Physics-based and Statistical Features for Image Forensics

Physikalische und statistische Merkmale in der Bildforensik

Der Technischen Fakultät der
Universität Erlangen–Nürnberg
zur Erlangung des Grades

DOKTOR–INGENIEUR

vorgelegt von

Christian Riess

Erlangen — 2013

Als Dissertation genehmigt von der
Technischen Fakultät der
Universität Erlangen-Nürnberg

Tag der Einreichung:	05. 11. 2012
Tag der Promotion:	21. 12. 2012
Dekan:	Prof. Dr.-Ing. habil. M. Merklein
Berichterstatter:	Prof. Dr.-Ing. J. Hornegger
	Prof. Dr.-Ing. M. Stamminger

Abstract

The objective of blind image forensics is to determine whether an image is authentic or captured with a particular device. In contrast to other security-related fields, like watermarking, it is assumed that no supporting pattern has been embedded into the image. Thus, the only available cues for blind image forensics are either a) based on inconsistencies in expected (general) scene and camera properties or b) artifacts from particular image processing operations that were performed as part of the manipulation.

In this work, we focus on the detection of image manipulations. The contributions can be grouped in two categories: techniques that exploit the statistics of forgery artifacts and methods that identify inconsistencies in high-level scene information. The two categories complement each other. The statistical approaches can be applied to the majority of digital images in batch processing. If a particular, single image should be investigated, high-level features can be used for a detailed manual investigation. Besides providing an additional, complementary testing step for an image, high-level features are also more resilient to intentional disguise of the manipulation operation.

Hence, the first part of this thesis focuses on methods for the detection of statistical artifacts introduced by the manipulation process. We propose improvements to the detection of so-called copy-move forgeries. We also develop a unified, extensively evaluated pipeline for copy-move forgery detection. To benchmark different detection features within this pipeline, we create a novel framework for the controlled creation of semi-realistic forgeries. Furthermore, if the image under investigation is stored in the JPEG format, we develop an effective scheme to expose inconsistencies in the JPEG coefficients.

The second part of this work aims at the verification of scene properties. Within this class of methods, we propose a preprocessing approach to assess the consistency of the illumination conditions in the scene. This algorithm makes existing work applicable to a broader range of images. The main contribution in this part is a demonstration of how illuminant color estimation can be exploited as a forensic cue. In the course of developing this method, we extensively study color constancy algorithms, which is the classical research field for estimating the color of the illumination. In this context, we investigate extensions of classical color constancy algorithms to the new field of non-uniform illumination. As part of this analysis, we create a new, highly accurate ground truth dataset and propose a new algorithm for multi-illuminant estimation based on conditional random fields.

Zusammenfassung

In der Forschungsrichtung „Blinde Bildforensik“ werden Methoden entwickelt, um die Authentizität und das Aufnahmegerät digitaler Bilder zu ermitteln. Hierfür kann — im Unterschied zu verwandten Gebieten, wie zum Beispiel der Wasserzeicheneinbettung — nicht auf eine speziell hinzugefügte Sicherheitssignatur zurückgegriffen werden. Dementsprechend können die verfügbaren Hinweise auf Ursprung und Authentizität eines Bildes lediglich aus zwei Quellen bezogen werden: a) aus Inkonsistenzen in erwarteten, allgemeinen Szenen- oder Kameraeigenschaften, oder b) aus Bildverarbeitungsartefakten, die durch eine Fälschungsoperation entstanden sind.

Diese Arbeit beschäftigt sich mit dem Erkennen von Bildmanipulationen. Die Beiträge teilen sich in zwei Kategorien auf: die Ausnutzung statistischer Fälschungsartefakte und das Finden von Inkonsistenzen in abstrakterer Szeneninformation. Beide Kategorien ergänzen sich gegenseitig. Die statistischen Ansätze können in einem automatisierten Ablaufplan auf die meisten digitalen Bilder einfach angewandt werden. Für den Fall, dass ein einzelnes Bild speziell untersucht werden soll, können Szeneneigenschaften in einer manuellen Analyse miteinbezogen werden. Neben dem Umstand, dass hiermit ein weiterer, zu den statistischen Merkmalen komplementärer Prüfschritt möglich wird, bieten Szeneneigenschaften im allgemeinen einen besseren Schutz gegen die gezielte Vertuschung der Fälschungsoperation.

Der erste Teil behandelt die Erkennung statistischer Manipulationsartefakte. Wir schlagen Verbesserung für die Erkennung sogenannter Copy-Paste-Fälschungen und eine einheitliche, gründlich evaluierte Verarbeitungskette zur Erkennung von Copy-Paste-Fälschungen vor. Zum Leistungsvergleich verschiedener Erkennungsmerkmale innerhalb der Verarbeitungskette wird ein neuartiges Konstruktionsgerüst für die gesteuerte Erzeugung semi-realistischer Fälschungen vorgestellt. Des weiteren, falls das zu untersuchende Bild im JPEG-Format vorliegt, wird ein effektives Verfahren entwickelt, um Inkonsistenten in den JPEG-Koeffizienten aufzudecken.

Der zweite Teil dieser Arbeit beschäftigt sich mit der Verifikation von Szeneneigenschaften. Für diese Algorithmenkategorie schlagen wir einen Vorverarbeitungsschritt für die Analyse der Lichtrichtung vor, der die Anwendbarkeit bestehender Methoden auf eine größere Vielfalt von Bildern ermöglicht. Der Hauptbeitrag in diesem Teil ist die Vorstellung eines Ansatzes zur Ausnutzung von Lichtfarbschätzern als Fälschungsindikatoren. Im Rahmen der Entwicklung dieser Methode untersuchen wir ausführlich Algorithmen zur Farbkonstanz, dem klassischen Forschungsfeld zur Schätzung der Lichtfarbe. Hierbei werden Erweiterungen klassischer Farbkonstanz-Methoden auf eine neue Problemstellung untersucht, der Schätzung inhomogener Lichtumgebungen. Im Rahmen dieser Analyse stellen wir einen neuen, sehr präzisen Datensatz zur Evaluation von Szenen mit inhomogener Beleuchtung vor. Des weiteren wirde eine neue Methode zur Schätzung inhomogener Lichtquellen entwickelt, die auf Conditional Random Fields beruht.

Acknowledgment

I would like to acknowledge the contribution of a number of people to this thesis. Achim and Elli, as my supervisors, had the strongest influence on this work. Among our collaborators, the work with Joost van de Weijer and Shida Beigpour from the Computer Vision Center in Barcelona, Spain, and with Tiago Carvalho and Anderson Rocha from the Universidade Estadual de Campinas, Brazil, was highly inspiring. Some of the undergraduate students under my supervision greatly contributed to the research work in this thesis. These are Vincent Christlein, Fabian Zach, Sven Pfaller and Michael Bleier. I would also like to thank my room mates Andre Linarth, Eva Eibenberger and Johannes Jordan for many fruitful discussions, and the nice working environment at the lab.

I would also like to thank the subjects for the dataset in Appendix E for their support and permission to use their pictures, in alphabetical order Andre Aichert, Daniel Danner, Bruno Kleinert, Michael Gernoth, Julian Hammer, Thomas Kemmer, Eva Kollorz, Rainer Müller, Daniela Novac, Jens Schedel.

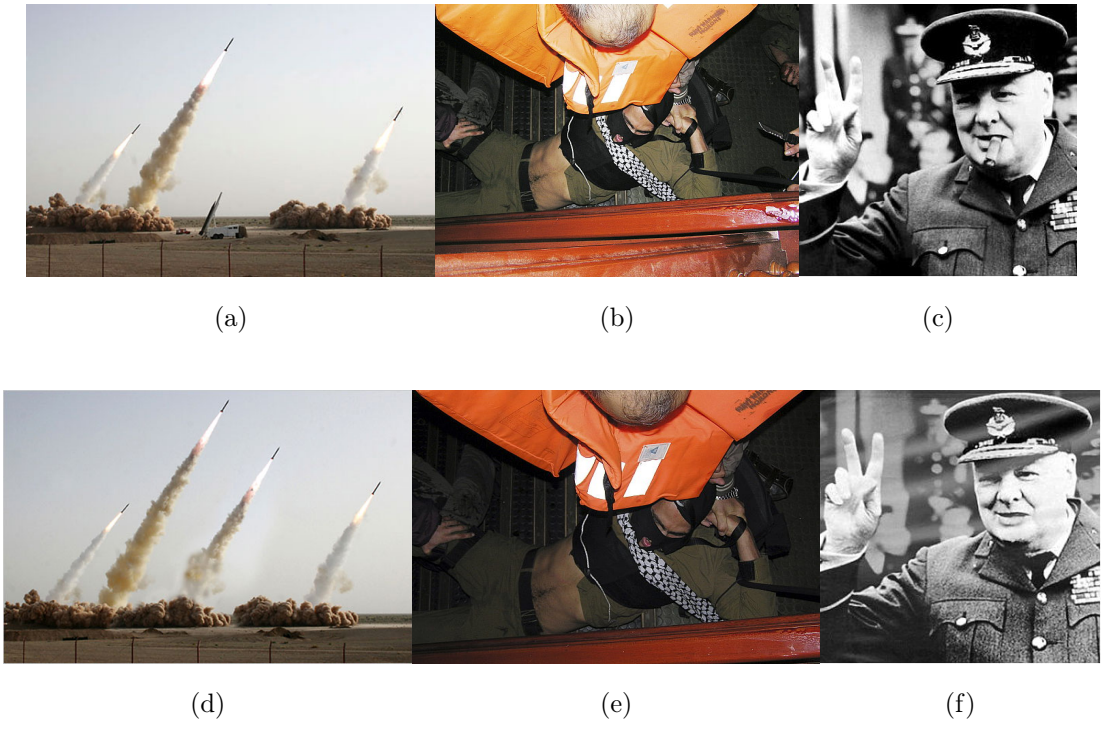
Christian Riess

Contents

Chapter 1

Introduction

Blind Multimedia Forensics is a relatively new research direction in multimedia security. It aims at the detection of altered media content, but does not assume any embedded security scheme. Video footage, scanned images, as well as digital and analog photographs can be the target for manipulations. In this thesis, we limit ourselves to digital photographs. From a forensics perspective, several changes in a photograph are widely acceptable. For instance, it is well accepted to improve the image quality, e. g. to enhance the contrast, denoise an image, or highlight important regions. Forensics investigators search for changes in an image that create a different statement of the image. Thus, an "image forgery" is semantically defined, by considering the information communicated by the original image and the tampered image. The creation of forgeries can be motivated politically, economically, commercially, socially, or individualistically. Real-world examples for two of these motivations are shown in Fig. 1.1. On the left side, an allegedly propaganda-motivated retouch of the failed missile launch is shown. Note that the smoke billows in the inserted missile are the same as for the right rocket. In the middle, another politically motivated forgery is shown. Interestingly, the technique is completely different. The manipulation was created by just removing unwanted information (in this case, the knife) from the source image. Finally, the column on the right shows an example for a socially motivated forgery. For an exhibition in London 2010, Churchill's cigar was removed from a poster allegedly due to the anti-smoking movement. Thus, an image forgery is not defined independently of the applied technique. As a side note, not even every motive is considered a forgery. For instance, photo collages are typically acceptable, because it is not expected that the image shows a real event. One particular example is an advertisement in the German news magazine Der Spiegel [Spiegel 10], shown in Fig. 1.2. Technically, the same form of manipulation has been used as in Fig. 1.1d. Thus, the crowd of soccer fans was enlarged by simply copying groups of people within the same image. The middle and the right images highlight the copied regions. The lower and upper part of the image contain identical image regions, which can be best seen in the two large identical blocks in the right part of the image. These aspects illustrate that it does not suffice to look at an image from a solely technical viewpoint. Nevertheless, algorithmic methods can greatly support a human expert.



(a) (b) (c)

(d) (e) (f)

Figure 1.1: Examples of real-world image manipulations. Top original, bottom forgery. Left: Allegedly, Sepah news has been hiding the failed missile launch [Nizz 08]. Middle: The knife of the defending guard has been hidden in the press images [Mozg 10]. Right: Churchill's trademark, the cigar, has been removed from the image [Hale 10].

1.1 Categories of Forensic Methods

Figure 1.3 subsumes the potential stages in the image formation process that can be exploited by forensic algorithms. From left to right, the world reflects light to the camera. Thus, the shown scene must adhere to the laws of physics, in particular with respect to perspective and lighting. The light is refracted by the lens. Again, by physical laws, the lens introduces chromatic aberrations which act as an intrinsic signature in the image. Then, the sensor converts the incident light into an electric signal. Variations in the sensor sensitivity allow the extraction of a sensor-specific watermark, in the sense of digital watermarking. Subsequent processing in the digital signal processor (DSP) can also be used for the extraction of camera-specific signatures. Finally, the output image can be explicitly examined for tamper-specific artifacts. Additionally, if the output image has undergone JPEG compression, JPEG artifacts can be used as a general purpose image signature.

Looking at the pipeline in Fig. 1.3 from a different viewpoint, there are two general approaches for detecting tampered images. We subsume these approaches as *Verification of Imaging Artifacts* and *Detection of Tampering Artifacts*, or abbreviated *Verification* and *Detection*.Verification-oriented techniques operate mainly on features from the in-camera processing, i. e. between the lens and the digital signal

Figure 1.2: Example of an image manipulation that is not considered a forgery. Soccer fans are copied and pasted within the same image. Middle: colored regions were highlighted using ZERNIKE features for copy-move forgery detection. Right: binary map of the highlighted regions. Note that the applied manipulation technique is the same as in Fig. 1.1d.

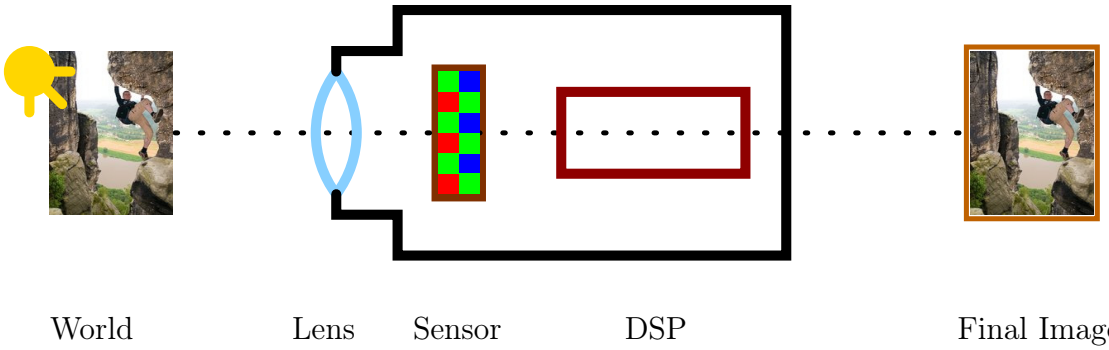


Figure 1.3: Image formation pipeline. The highlighted stages are popular anchor points for forensic algorithms (see text for details).

processor. For example, the chromatic aberration of a lens must follow a consistent pattern over the whole image. If a subregion of the image exhibits a contradicting behavior, this subregion is assumed to be manipulated.

Conversely, detection-oriented techniques aim at discovering a particular mistake in the process of creating a manipulated image. In the schematic of Fig. 1.3, these techniques can be attributed to the last step, i. e. the "final image". Irregularities that are introduced after capturing the image, for instance by resampling or copying image content, can often directly be detected from statistical artifacts.

Often, forensic algorithms can not strictly dichotomically be divided into Verification and Detection methods. However, we consider this distinction to be helpful in characterizing the application domain and particularities of forensic algorithms. As a particular example for methods that fall in both categories, we consider detection techniques that exploit the first step of Fig. 1.3, the depicted "World". On one hand, the laws of physics can be verified on a natural image. Thus, such techniques can be

seen as verification techniques. On the other hand, these algorithms are specialized to detect image splicing. As such, it can also be seen as a detection technique for spliced objects.

In this thesis, we present improvements and novel features in Detection approaches. First, two methods are presented that examine the final, possibly tampered, image. First, we thoroughly investigate features for copy-move forgery detection. We present our work on these algorithms in Chap. 2. The second technique we employ investigates JPEG artifacts as a cue for possible image manipulation history, in Chap. 3. As part of this thesis we also examine illumination-based physical constraints on natural images. We present our work on these algorithms in Chap. 5. As a theoretical foundation to this part, several novel insights in color constancy under non-uniform illumination are presented in Chap. 4.

The diversity of our methods offers various, complementary tools to the forensic analyst. Algorithms that operate on tampering artifacts of the output images can run fully automatically. Thus, these algorithms can be incorporated in a standard document processing pipeline, for instance at an insurance company. Whenever a digital image is submitted, these algorithms can be executed as a baseline test before forwarding the document to the person in charge. Algorithms that verify the physical consistency of the scene typically require semantic scene understanding. Thus, a human expert is required to support the scene analysis. However, this drawback is outweighed by two factors. Firstly, methods for scene analysis are not limited to digital imagery. They can be directly translated to analog photographs. Secondly, automated counter-detection methods are typically difficult to implement for these methods.

1.2 Contributions

This thesis investigates cues for the detection of manipulated images in blind image forensics. We examined two types of indicators, low-level statistical anomalies of tampered images, and high-level illumination properties. In the latter case, particular emphasis has been put on the estimation of the color of the illumination. As a consequence, a number of results contribute to color constancy, i. e. the foundations of research on illumination color. Part of this work was done in collaborations with other researchers. The detailed contributions by the author are stated at the beginning of the respective sections.

Statistical Cues in Image Forensics We developed a semi-realistic dataset for the controlled evaluation of statistical algorithms for detecting image forgeries. This benchmark database was introduced in [Chri 12]. We also thoroughly investigated prior work for the detection of copy-move forgeries. We cast these algorithms in a uniform pipeline and compared 15 of the proposed feature sets [Chri 12]. Upon closer investigation of the individual steps in the pipeline, we argue for using approximate nearest neighbors for matching these features [Chri 10a] and proposed a novel rotation- and theoretically also scale-invariant matching strategy [Chri 10b]. As a second low-level cue, we investigated JPEG-compressed images, and proposed a

pattern recognition-based automated detector for the so-called JPEG ghost observation [Zach 12]. This method removes the requirement for the user to browse dozens or even hundreds of intermediate images in order to find JPEG ghosts that discriminate single- and double-compressed regions in JPEG-images.

Color Constancy As a theoretical foundation for illumination cues in image forensics, we thoroughly investigated existing color constancy algorithms. Upon reviewing the challenges with specularity- and shadow processing [Ries 09a, Ries 09c], we developed a physics-based single- or multi-illuminant estimator that avoids specularity segmentation [Ries 11, Ries 09b]. Aiming ultimately at a robust estimator for non-uniform illumination, we investigated the potential of extending off-the-shelf single-illuminant estimators [Blei 11]. However, although the overall benchmark results were acceptable, we concluded that non-uniform illumination is so severely underconstrained, that it is worth investigating more sophisticated approaches. Ultimately, we propose to use a Conditional Random Field to incorporate spatial information in the local illuminant estimates. The resulting algorithm is currently under review [Beig 12]. Quantitative results for this algorithm are obtained from a newly created ground truth dataset, which exploits a novel idea for computing accurate ground truth from multiple input images.

Scene Understanding in Image Forensics Exploiting the insights from our research on color constancy, we proposed the use of the estimated illumination color as a cue for image manipulations [Ries 10]. We extended this work towards automated forgery assessment by interpreting local estimates of the illuminant color as texture maps. As a second high-level cue, we investigated the direction of the incident light as an indicator for image manipulations. The original method [John 07a] suffers from relatively strict assumptions. Thus, we develop a preprocessing step, inspired from intrinsic image decomposition, to make the exploitation of the illumination direction applicable to a broader range of images.

1.3 Thesis Outline

The thesis is structured in three main parts: statistical cues for detecting image manipulations, fundamental work on illuminant color estimation and illumination cues for detecting image manipulations. The statistical cues are split in two chapters.

In Chap. 2, an in-depth examination of copy-move forgery detection is presented. For quantitative analysis, we present a novel ground truth dataset and framework in Sec. 2.1. This dataset is used in Sec. 2.2, where a unified pipeline for copy-move forgery detection is proposed. The most important steps in this pipeline are thoroughly examined in this section, namely the choice of features, matching of the features and subsequent filtering on the found matches. The feature comparison is the largest part of this section. A comparison is conducted on copied regions under added noise, JPEG-compression, scaling and rotation. Additionally, a performance analysis is conducted under global downscaling of the images, a categorization into the type of the copied texture and a categorization on a semantic level.

In Chap. 3, we investigate the so-called JPEG-ghost observation for distinguishing single- and double-compressed regions in JPEG images. As the original method is a relatively tedious manual approach, we propose a fully automated classification scheme that performs this task with high sensitivity and specificity at also high spatial resolution.

In Chap. 4, we turn towards illuminant color estimation. After preliminary remarks and related work, we propose in Sec. 4.3 two approaches to create a ground truth dataset for scenes under non-uniform illumination. This opens the opportunity to quantitatively evaluate the effectiveness of estimation algorithms for scenes under multiple illuminants. In Sec. 4.4, we propose three approaches to multi-illuminant estimation. First, we experiment with off-the-shelf single-illuminant estimators on superpixels. Then, we propose a physics-based method that performs coarse clustering on similar estimates. Finally, we propose an energy-minimization approach that integrates color estimation and segmentation of differently illuminated regions in a single step.

In Chap. 5, we explore scene properties for image forensics. In Sec. 5.2, insights from the previous chapter are transferred to forgery detection. We propose to use physics-based features in a manually guided pipeline for image manipulation, and extend it to preliminary experiments on automated tampering detection using a machine learning approach. Besides the color of the illuminant, we also investigated the geometry factor in different lighting environments in Sec. 5.3. The exploitation of different directions of the illumination on manipulated images has been proposed before. However, existing work is only applicable to a small set of images, due to relatively strict constraints. In this section, we propose a preprocessing step that makes prior work applicable to a broader range of images, using insights from intrinsic image decomposition.

We conclude this work with a brief discussion and outlook in Chap. 6. Finally, in Chap. 7, we present a summary on the results in this thesis.

Chapter 2

Optimizing Copy-Move Forgery Detection

Copy-Move Forgery Detection (CMFD) is a classical approach in tampering artifact detection, in the sense that copy-move manipulations leave characteristic traces which can be directly discovered. The underlying assumption is that a region is copied and pasted within the same image. The pasted part may have been subject to additional transformations, for instance be slightly rescaled, rotated or parts of it can be repainted for artistic reasons. In general, if a duplicated region within the same image is found, the manipulation is directly proven. Thus, in terms of the image formation pipeline in Fig. 1.3 on page 3, CMFD methods are located in the last, rightmost step.

A considerable number of CMFD methods have been proposed. Despite this large body of work, there was a lack of a framework for consolidating the various insights, and additionally unifying existing approaches when applicable. In the first part of this thesis, such an overarching schema is proposed. It can encompass most of the existing CMFD methods. It is flexible and general and thus allows for the inclusion of future algorithms. It includes a benchmarking database.

In this chapter, we first introduce a novel, challenging benchmark database of close-to-real-world image forgeries (see Sec. 2.1). Second, we formulate proposed solutions to the CMFD problem in a unified processing pipeline in Sec. 2.2.1. Finally, we examine the three most influential steps in the pipeline separately and give concrete guidelines for practical implementations of the algorithms. In detail, we found that approximate nearest are in general a more solid choice for feature matching than lexicographic sorting (see Sec. 2.2.2). Then, we propose a novel method for postprocessing raw feature matches that have undergone affine transformations, called *Same Affine Transform Selection* (SATS) in Sec. 2.2.3. Finally, we conducted a large-scale study on the best performing features in Sec. 2.2.4. This comparison suggests that keypoint-based methods are excellent choices for quick screening of large databases, while several other feature sets provide higher detail in the detection performance, but at a much higher computational cost. The code for copy-move forgery detection was mostly written by Vincent Christlein, in the course of his study thesis under my supervision and optimized by him after submitting his thesis. The dataset, and the software framework were created by me. Most algorithmic ideas that are presented

in this chapter are also by me. The evaluation for the large comparison at the end of the chapter was done jointly by Vincent and me.

2.1 Ground Truth Database

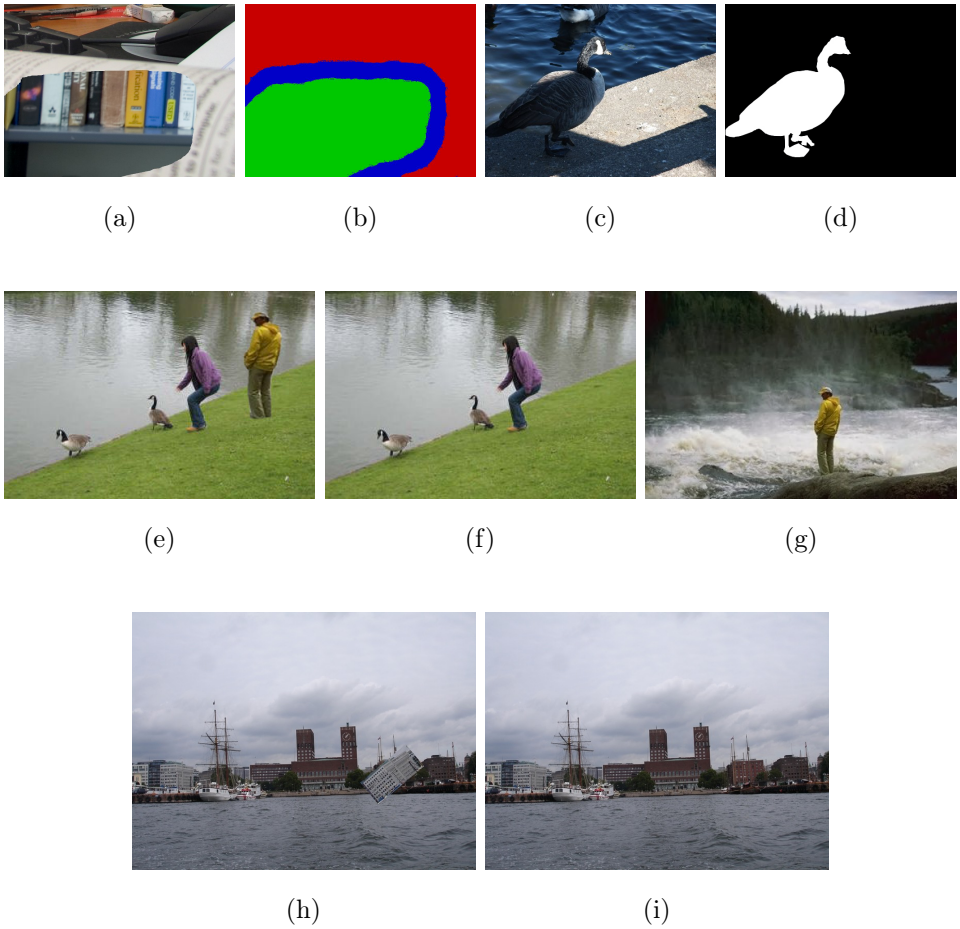
When a new algorithm is introduced in image forensics, it is typically evaluated on a number of specifically tailored images. Only a small number of publicly available datasets exists for a standardized comparison of similar methods. Some of these datasets aim at the identification of source cameras. Others are specifically developed for evaluating one particular type of manipulation. In this section, we address the lack of data by introducing a *framework* for evaluating copy-move forgery detection algorithms. The benchmark consists of images, image components for performing manipulations and a software to replay the manipulation. While replaying the manipulation, postprocessing, image artifacts and noise can be inserted on demand. Thus, it is a framework for creating semi-synthetic forgeries in a controlled way.

2.1.1 Related Work

Samples from publicly available datasets are shown in Fig. 2.1. Ng *et al.* [Ng 04] developed a dataset of automatically spliced images. It consists of 183 authentic and 180 tampered images. The size of the images ranges from 757×568 pixels to 1152×768 pixels. The dataset aims mainly at distinguishing different cameras types. Four cameras from different manufacturers have been used to create the dataset. For the tampered part, portions of an image from one camera are randomly copied and inserted in an image from a different camera, without additional post-processing (see Fig. 2.1a). The seams of the spliced regions often exhibit sharp edges, and the content of the spliced region is semantically not meaningful. However, meaningful image content was not the goal of this work. Algorithms for camera identification often use methods from signal processing approaches. Thus, it suffices to have different camera signatures within the same image. Additionally, the user can mask the boundary regions with the associated ground truth map (see Fig. 2.1b). Here, red and green denote data from different cameras, while blue masks the boundary between the areas.

Battiato *et al.* [Batt 09b] presented a tampered image database that is focused on the exploitation of JPEG artifacts. It consists of 59 images, taken mostly from the Uncompressed Colour Image Database (UCID) [Scha 04]. The authors provide masks and photoshop scripts to create artifacts for distinguishing single and double JPEG compression. As JPEG-based manipulation detectors often operate on independent image blocks of 8×8 or 16×16 pixels, image size is a secondary criterion. Consequently, these images have rather low dimensions: almost all images are 384×512 pixels. Fig. 2.1c and Fig. 2.1d show an example image and the associated mask for introducing artifacts from double JPEG compression.

The CASIA [CASIA] forensic dataset provides a large number of tampered images with realistic content. Version 2 of this dataset consists of 7491 authentic and 5123 manipulated (mostly JPEG-compressed) images. The manipulations are done by splicing two images from the authentic data (for an example, see Fig. 2.1e, Fig. 2.1f



(a) (b) (c) (d)

(e) (f) (g)

(h) (i)

Figure 2.1: Example images from other forensic databases. Top row, from left to right: A sample from the dataset of Ng *et al.* [Ng 04] (Fig. 2.1a and Fig. 2.1b), and an image with associated object mask from the dataset by Battiato *et al.* [Batt 09b] (Fig. 2.1c and Fig. 2.1d). Middle row: a tampered image from the CASIA dataset [CASIA], and the corresponding source images. Bottom row: tampered image from the copy-move detection dataset by Amerini *et al.* [Amer 11] and its associated source image.

and Fig. 2.1g). If desired, the inserted region is rotated, scaled, and perspectively distorted. Most of the image sizes are 384×256 pixels, which is unrealistically small. Additionally, tampered pixels are not indicated, the only available information is whether the whole image has been tampered or not. Although it should in principle be possible to compute such a map from the input images, we refrained from this approach, as it is an error-prone process to reverse-engineer pixelwise ground truth from output images, in particular after lossy JPEG compression.

Amerini *et al.* [Amer 11] published two ground truth databases for CMFD algorithms, called MICC F220 and MICC F2000. They consist of 220 and 2000 images, respectively. In each of these datasets, half of the images are tampered. The image size is 2048×1536 pixels. However, the type of processing on the copy-move forgeries is limited to rotation and scaling (for an example, see Fig. 2.1h and Fig. 2.1i). Ad-

ditionally, the source files are not available. Hence, it is not straightforward to add artifacts like noise to the copied region.

Several other databases target camera identification. For instance, the Dresden Image Database by Gloe and Böhme [Gloe 10] focuses on methods for camera identification. Dirik *et al.* [Diri 08] also developed a dataset on camera identification based on sensor dust. Similarly, Goljan *et al.* [Golj 09] created a large-scale database for the identification of sensor fingerprints.

There are a number of shortcomings in the presented work. First, image size is an important property for algorithms aiming at the detection of copy-move forgeries or resampling. Furthermore, for an evaluation that is focused on practical applications, it is always a better choice to operate on realistic forgeries rather than on randomly tampered images. Equally importantly, every single one of the existing databases has been created with a particular evaluation scenario in mind. Except of the work by Battiato *et al.*, the benchmark scenario is static, in the sense that it can not be extended or varied.

2.1.2 A Framework for Image Forensics Benchmarking

To address the limitations of prior datasets, we propose a novel *framework* for evaluation, consisting of large real-world images. An outline of the framework is shown in Fig. 2.2. We start by manually preparing semantically meaningful regions and a corresponding alpha channel (for partial transparency) which will be used for tampering (see Fig. 2.2, top row). These regions (called *snippets*) are taken from within the source image, so that the benchmark can also be used for copy-move forgery detection. Three persons of varying artistic skills manually created the snippets. When creating the snippets, we asked the artists to vary the size of the selected regions. Additionally, the snippet content should be either *smooth* (e. g., sky), *rough* (e. g., rocks) or *structured* (typically man-made buildings). These groups can be used as categories for CMFD images.

To create artifacts from tampering operations in a controlled setup (i. e. as the result of a parameterized algorithm), we developed a software to create forgeries using these snippets. The creation of a forgery involves three computational steps, denoted as green and blue boxes in Fig. 2.2. First, each snippet can be individually postprocessed by adding for instance noise, or by applying an affine transformation to it. Then, the postprocessed snippets and the source image are spliced by inserting the snippets on freely chosen positions in the source image (see Fig. 2.2, blue box). For spliced images that are semantically meaningful, we provide predefined snippet coordinates. The output of this combination step is the combined image and an associated ground truth map. The combined image can be exposed to global postprocessing, e. g. JPEG compression. The outcome of this processing chain becomes the benchmark image.

The combination of the snippets and the postprocessing steps are done in the C++ core of the framework. For a systematic evaluation, a variation of the parameters in the postprocessing steps is required. Example variations are shown in Fig. 2.2 within the green boxes. For instance, if different variants of resampled splicing shall be evaluated, the snippets can be resampled in equidistant steps, e. g. by rotation or by

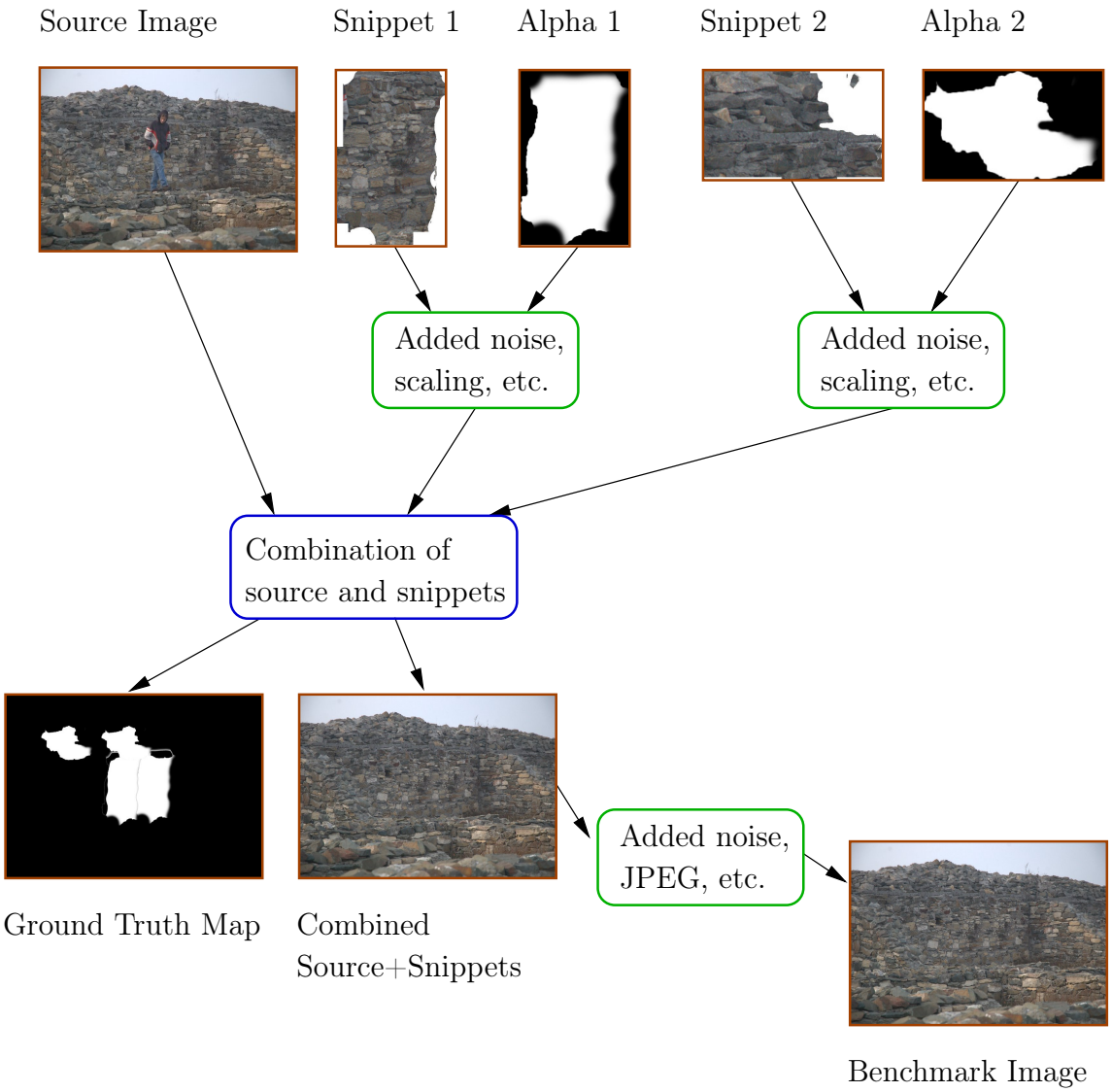


Figure 2.2: Overview of our proposed framework for benchmark creation.

scaling. This parametrization is highly problem-dependent. Thus, these variations are guided by perl-scripts that call the C++ core with the required parameters.

In total, the dataset consists of 48 source images. A group of 12 images was collected from the same source, namely a Panasonic DMC-TZ4, a Canon EOS, a Nikon D70 and flickr [Yaho 12]. The images from the Panasonic and the Canon EOS cameras contain weak artifacts from JPEG compression. The pictures from Nikon D70 were converted from raw images. The flickr images are downloaded from various authors and serve as images from uncontrolled origin. We prepared a total of 87 snippets. The average size of an image is about 3000×2300 pixels. Little less than 5% of the pixels are used for creating the snippets. For a copy-move scenario, every snippet is copied, thus about 10% of the pixels belong to the ground truth. Tab. 2.1 summarizes these numbers on the dataset. In Appendix C.4, we show the reference manipulations together with the computed ground truth maps.

# of images	48
# of snippets	87
Total # of pixels	348972116
Total # of pixels in snippets	17395849
#tampered pixels / # total pixels	0.0498
Total # of copy-move tampered pixels	34791698
#copy-moved pixels / # total pixels	0.0997

Table 2.1: Dataset statistics

Figure 2.3: From left to right: The image *beachwood* (first image) is forged with a green patch to conceal a building (second and third image). A ground truth map (fourth image) is generated where copy-moved pixels are white, unaltered pixels are black and boundary pixels are gray.

Figure 2.3 illustrates the various components and the final output of an example copy-move forgery in our database. The source image is shown in the left. The snippet (in its position of insertion) in the middle left. To the middle right, the combined image is shown. The associated ground truth map is shown in the right image. White pixels denote one-to-one copies, black pixels denote background. Gray pixels state that the pixels have been copied, but are no direct copies, for instance due to partial transparency of these pixels.

2.1.3 Generation of Spliced Copies

The snippets are inserted in the source image in the order of their numbering, i. e. snippet 1 is inserted before snippet 2, and so on. This may play a role in the case that multiple inserted snippets overlap. The alpha channel is used to linearly interpolate the intensity of a snippet pixel with the corresponding source pixel. When setting the value of a pixel, the ground truth is updated with the opacity of the pixel. When a snippet pixel has to be placed with full opacity over another snippet pixel, we simply overwrite it. When partially transparent pixels from two snippets overwrite each other, we set the ground truth to the opacity of the overwriting pixel. This can lead to inaccuracies, but as partially transparent pixels should be excluded from evaluation, these inaccuracies are not critical.

A particular challenge is the definition of ground truth for copy-move detection algorithms. In these methods, it is assumed that a region is copied and pasted within the same image. Thus, the ground truth computation must also consider the inserted region, as well as the source region where it has been copied from. The location of the source region is provided with the software framework. However, a 1-to-1

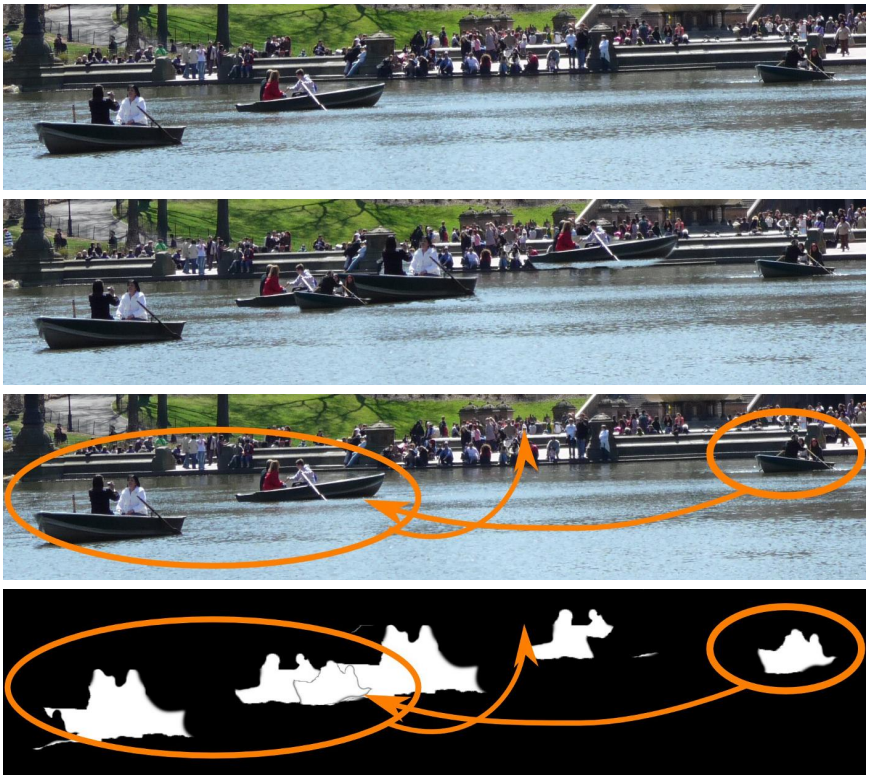


Figure 2.4: Artificial example of several types of occlusion in the ground truth generation. From top to bottom: original image; final tampered image, where the two boats from the left and the single boat from the right are copied one over another; visualization of which image parts were copied and to which location; when computing the ground truth, occlusions are characterized by the insertion order.

pixel relationship between the source and the target regions of the snippet must be maintained. When a second snippet masks a pixel in either of these regions (source or target), it must also be excluded from the ground truth in the corresponding other region. This property is illustrated with an (exaggerated) example in Fig. 2.4 , using the "central park" motif. In the top row, the original image is shown. In the second row, the two boats from the left are copied and moved to the right, while the single small boat on the right is copied and moved to the left. The third row illustrates these moves. Note that the source and target regions of the boats considerably overlap each other in the central part of the image. The fourth row contains the move annotations for the ground truth map. The impact of the overlap can be seen from the shape of the boats in the last row. The source region from the boats on the left is partially occluded, which leads to excluded pixels in the copy of this region (as can be seen in the ground truth annotation of the second boat from the right). The bow of the boat that was copied from left is also occluded from the rightmost boat. Thus, the bow is also excluded from the source region (as can be seen in the ground truth annotation of the leftmost boat).

Although such mutual occlusion rarely happens in practice, it can lead to very complicated situations. In our implementation, we used a recursive formulation that resolves overlaps when a new snippet is inserted. We omit the algorithmic description

here, as we consider it of mostly theoretical interest. From a practitioners viewpoint, we do not necessarily see a reason for producing such particularly difficult overlapping cases.

2.1.4 Performance Measures

The database is suitable for evaluations at two levels of detail. At a broad level, one can examine the images in their entirety. The evaluation focuses on the number of images that were correctly detected as original or tampered. We call this a performance evaluation at *image level*. The second possibility is to evaluate the detection performance *within* an individual image. In this case, we count the number pixels that were correctly detected. This can be done using the ground truth map together with the pixelwise output of the respective benchmark image. We call this a performance evaluation at *pixel level*.

At both levels, it is possible to count the number of correct detection (true positives, n_{TP}), false detections (false positives, n_{FP}) and correctly omitted images or pixels (true negatives, n_{TN}). This metric has been used in several papers, for instance by Bravo *et al.* [Brav 09] and Luo *et al.* [Luo 06]. From these measures, related performance metrics can be computed. We recommend to use *Precision* prec and *Recall* rec. They are defined as:

$$\mathrm{prec} = \frac{n_{\mathrm{TP}}}{n_{\mathrm{TP}} + n_{\mathrm{FP}}} \tag{2.1}$$

and

$$\mathrm{rec} = \frac{n_{\mathrm{TP}}}{n_{\mathrm{TP}} + n_{\mathrm{FN}}} \ . \tag{2.2}$$

Precision denotes the probability that a reported detection is indeed correct, while *Recall* shows the probability that a manipulation is detected. *Recall* is often also called true positive rate.

Alternatively, one can use *Specificity* and *Sensitivity*. *Specificity* spec is defined as

$$\mathrm{spec} = \frac{n_{\mathrm{TN}}}{n_{\mathrm{TN}} + n_{\mathrm{FN}}} \ , \tag{2.3}$$

while *Sensitivity* is equivalent to *Recall*. The intuition between spec is to reward non-marked (i. e. omitted) areas, if they are truly not part of the sought region. Note that two other popular measures, the false positive rate and the false negative rate, additively complement specificity and sensitivity to 1.

If a single performance metric is required, we recommend the use of the F_1 score. It combines *Precision* and *Recall* in a single measure:

$$F_1 = 2 \cdot \frac{\mathrm{prec} \cdot \mathrm{rec}}{\mathrm{prec} + \mathrm{rec}} \ . \tag{2.4}$$

Figure 2.5: Common processing pipeline for the detection of copy-move forgeries. The feature extraction differs for keypoint-based features (top) and block-based features (bottom). Except of method-specific threshold values, all remaining steps are the same.

2.2 Copy-Move Forgery Detection

Copy-move forgery detection (CMFD) is the most popular topic among all approaches for exposing image manipulations: more than 30 different CMFD algorithms have been proposed. The undelying assumption is that an image region has been copied and pasted within the same image. Although this assumption is very restrictive, copy-move forgeries are commonly seen in practice. Two real-world examples have already been shown in Fig. 1.1d and Fig. 1.2 (on page 2 and page 3), respectively.

Our main contributions are: a) a large-scale comparison of different CMFD feautures, and b) a novel algorithm, SATS, for robust post-processing of copied regions. In Sec. 2.2.1, we formulate the CMFD algorithms within a unified pipeline. The joint pipeline allows us to relate the various algorithms and to compare the various design decisions. The remainder of this section covers our findings for several steps of the pipeline. In Sec. 2.2.2, we show that feature matching is better done using approximate nearest neighbors, rather than the earlier proposed lexicographic sorting. In Sec. 2.2.2, we propose the Same Affine Transformation Selection (SATS) to perform rotation- and scale-invariant postprocessing of matched features. Finally, in Sec. 2.2.4, we use the derived insights for a large-scale evaluation of feature types that have been proposed in earlier work.

2.2.1 Related Work and the CMFD Pipeline

Although a large number of CMFD methods has been proposed, the general workflow is typically very similar. Most of the CMFD-methods follow a common pipeline, as shown in Fig. 2.5. Given an original image, there exist two processing alternatives. CMFD-methods are either keypoint-based methods or block-based methods. In both cases, preprocessing of the images is possible. For instance, most methods operate on grayscale images, and as such require that the color channels be first merged. For feature extraction, block-based methods subdivide the image in rectangular regions. For every such region, a feature vector is computed. Similar feature vectors are matched. By contrast, keypoint-based methods do not perform explicit image subdivision. Instead, their feature vector is computed at image regions containing high entropy. Subsequently, similar features within an image are matched. A forgery shall be reported if regions of such matches cluster to larger areas. Both, keypoint- and block-based methods include further filtering for removing spurious matches. An optional postprocessing step of the detected regions may also be performed, to group matches that jointly follow a transformation pattern.

Due to differences in the computational cost, as well as the detected detail, we consider the difference between block- and keypoint-based methods very important. We describe these two variants for feature vector computation in detail in the last two subsections of this section. Additional relevant details to the remaining steps in the pipeline are presented below.

Matching High similarity between two feature descriptors is interpreted as a cue for a duplicated region. Most authors propose the use of *lexicographic sorting* in identifying similar feature vectors (see e. g. [Bash 10, Bayr 05, Bayr 09, Brav 11, Dyba 07, Frid 03, Ju 07, Kang 08, Ke 04, Li 07, Lang 06, Lin 01, Lin 09a, Luo 06, Myrn 07, Pope 04, Ryu 10, Shie 06, Wang 09b, Wang 09a, Zhan 08]). In lexicographic sorting, a matrix of feature vectors is built so that every feature vector becomes a row in the matrix. This matrix is then row-wise sorted. Thus, the most similar features are in consecutive rows to each other.

Other authors [Huan 08, Pan 10, Mahd 07] propose to use the Best-Bin-First search method [Beis 97] derived from the kd-tree algorithm to get approximate nearest neighbors with lower computational cost than the original kd-tree [Frie 77].

Filtering Filtering schemes have been proposed in order to reduce the probability of false matches. For instance, a common noise suppression measure is the removal of matches between spatially close regions. Neighboring pixels often have similar intensities, which can lead to false forgery detection. Additionally, different distance criteria have been proposed to filter out weak matches. For example, several authors proposed the Euclidean distance between matched feature vectors [Mahd 07, Wang 09a, Ryu 10]. In contrast, Bravo-Solorio *et al.* [Brav 11] proposed the correlation coefficient between two feature vectors as a similarity criterion.

Postprocessing The goal of this last step is to only keep those matches that exhibit a common behavior. Consider a set of matches that belongs to the same copied region. These matches are expected to be spatially close to each other in both the source, and the target blocks (or keypoints, resp.). Furthermore, multiple matches must follow a joint pattern with respect to translation, rotation and scaling.

The most widely used postprocessing variant handles outliers by imposing a minimum number of same-shift-vectors between matches (see e. g. [Bash 07, Bayr 09, Frid 03, Luo 06, Pope 04]). A shift vector contains the translation (in image coordinates) between two matched feature vectors. Consider a number of blocks which are simple copies, without rotation or scaling. Then, the histogram of shift vectors exhibits a peak at the translation parameters of the copy operation.

Mahdian and Saic [Mahd 07] consider a pair of matched feature vectors as forged if: a) they are sufficiently similar, i. e. their Euclidean distance is below a threshold, and b) the neighborhood around their spatial locations contains similar features. Other authors use morphological operations to connect matched pairs and remove outliers [Zhan 08, Lang 06, Pan 10, Wang 09b]. An area threshold can also be applied, so that the detected region has at least a minimum number of points [Luo 06, Pan 10, Wang 09b, Wang 09a]. To handle rotation and scaling, Pan and Lyu [Pan 10] proposed to use RANSAC. For a certain number of iterations, a random subset of

the matches is selected and the transformations of the matches are computed. The transformation which is satisfied by most matches (i. e. which yields most inliers) is chosen. Recently, Amerini *et al.* [Amer 11] proposed a scheme which first builds clusters from the locations of detected features and then uses RANSAC to estimate the geometric transformation between the original area and its copy-moved version. Alternatively, SATS (see Sec. 2.2.3 on page 20) can be used for both, block-based and keypoint-based methods. Although not explicitly reported, we evaluated the impact of each of these methods. Ultimately, we adopted two strategies. For block-based approaches, we used a threshold τ_2 based on the SATS-connected area to filter out spurious detections, as SATS provided the most reliable results in early experiments. To obtain pixel-wise results for keypoint-based methods, we combined the methods of Amerini *et al.* [Amer 11] and Pan and Lyu [Pan 10]. We built the clusters described by Amerini *et al.*, but avoided the search for the reportedly hard to calibrate *inconsistency threshold* [Amer 11]. Instead, clusters stop merging when the distance to their nearest neighbors are too high, then the affine transformation between clusters is computed using RANSAC and afterwards refined by applying the gold standard algorithm for affine homography matrices [Hart 03, pp. 130]. For each such estimated transform, we computed the correlation map according to Pan and Lyu [Pan 10]. For further details on our implementation, please refer to the appendix C.1.

2.2.1.1 Block-based Algorithms

The image is subdivided into equally sized blocks. A feature vector is extracted for every block. A number of different features has been recently proposed. We investigated 13 block-based features, which we considered representative of the entire field. They can be grouped in four categories: moment-based, dimensionality reduction-based, intensity-based, and frequency domain-based features (see Tab. 2.2).

Moment-based: We evaluated 3 distinct approaches within this class. Mahdian and Saic [Mahd 07] proposed the use of 24 blur-invariant moments as features. We refer to this method as Blur. Wang *et al.* [Wang 09b] used the first four Hu moments as features. We refer to this method as Hu. Finally, Ryu *et al.* [Ryu 10] recently proposed Zernike moments for features, which we denote as Zernike.

Dimensionality reduction-based: In [Pope 04], the feature matching space was reduced via principal component analysis (PCA). Bashar *et al.* [Bash 10] proposed another variant of PCA, called Kernel-PCA (KPCA). Kang *et al.* [Kang 08] computed the singular values of a reduced-rank approximation (SVD). A fourth approach using a combination of discrete wavelet transform and Singular Value Decomposition [Li 07] did not yield reliable results in our setup and is, thus, excluded from the evaluation.

Intensity-based: The first three features used in [Luo 06] and [Brav 11] are the average red, green and blue components. Additionally, Luo *et al.* [Luo 06] use directional information of blocks (Luo) while Bravo-Solorio *et al.* [Brav 11] consider the entropy of a block as a discriminating feature (Bravo). Lin *et al.* [Lin 09a] (Lin) compute the average grayscale intensities of a block and its sub-blocks. Wang *et al.* [Wang 09a]

[1]Some feature-sizes depend on the block size, which we fixed to 16×16. Also note that the feature-sizes of PCA and SVD depend on the image or block content, respectively.

Group	Methods	Feature-length [1]
Moments	Blur [Mahd 07]	24
	Hu [Wang 09b]	5
	Zernike [Ryu 10]	12
Dimensionality reduction	PCA [Pope 04]	–
	SVD [Kang 08]	–
	KPCA [Bash 10]	192
Intensity	Luo [Luo 06]	7
	Bravo [Brav 11]	4
	Lin [Lin 09a]	9
	Circle [Wang 09a]	8
Frequency	DCT [Frid 03]	256
	DWT [Bash 10]	256
	FMT [Bayr 09]	45
Keypoint	Sift [Huan 08],[Pan 10],[Amer 11]	128
	Surf [Shiv 11],[Bo 10]	64

Table 2.2: Categorization and size of the CMFD feature sets that have been examined for this work.

(Circle) use the mean intensities of circles with different radii around the block center.

Frequency-based: Fridrich *et al.* [Frid 03] proposed the use of the 256 coefficients of the discrete cosine transform as features (DCT). The coefficients of a discrete wavelet transform (DWT) using Haar-Wavelets are proposed as features in work by Bashar *et al.* [Bash 10]. Bayram *et al.* [Bayr 09] recommended the use of the Fourier-Mellin Transform (FMT) for generating feature vectors.

2.2.1.2 Keypoint-based Algorithms

Unlike block-based algorithms, keypoint-based methods rely on the identification and selection of high-entropy image regions (i. e. the "keypoints"). Whereas in block-based techniques a feature vector was computed per block, keypoint approaches extract a feature vector per keypoint. Consequently, fewer feature vectors are estimated, resulting in reduced (by an order of magnitude) computational complexity of feature matching and post-processing. As a side-effect of the lower number of feature vectors, the postprocessing thresholds have to also be an order of magnitude lower than that of block-based methods. A drawback of keypoint methods is that copied regions are often only sparsely covered by matched keypoints. If the copied regions exhibit little structure, it may happen that the region is completely missed. We examined two different versions of keypoint-based feature vectors. One uses the SIFT features while the other uses the SURF features (see e. g. [Huan 08, Shiv 11]). They are denoted as Sift and Surf, respectively.

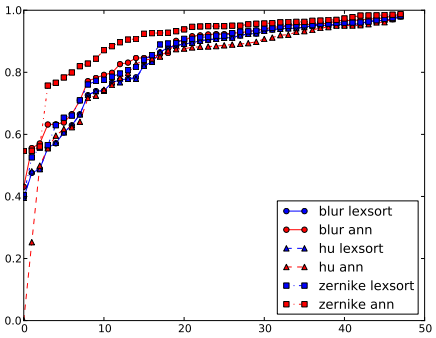
2.2.2 Feature Matching using Approximate Nearest Neighbors

We first focused on the matching step in the CMFD pipeline shown in Fig. 2.5. In prior work, lexicographic sorting and an approximate nearest neighbor search have been proposed. Although lexicographic sorting is the method that has been predominantly used in the related work, early experiments suggested that computing approximate nearest neighbors leads to better results. Consequently, we investigated this issue more thoroughly. The foundation of this section is [Chri 10a]. However, in contrast to [Chri 10a], we investigate in this work the performance difference on a much larger dataset. We confirm the results in our prior work, but upon closer investigation, the gained advantage is less pronounced as reported in [Chri 10a].

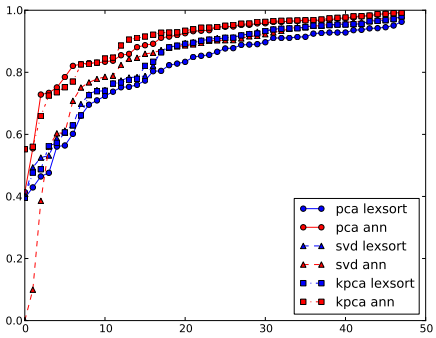
We set up an experiment consisting of the 13 block-based feature sets from Tab. 2.2. Keypoint-based features have been omitted, as SIFT and SURF keypoints are matched using the Euclidean distance, and have not been subject to this discussion. As benchmark data, we used the 48 reference manipulations from our proposed dataset according to Sec. 2.1. Note that we did not apply any postprocessing like noise, JPEG artifacts, rotation or scaling to the benchmark images. We used the same-shift-vector approach for determining the final matches, and set the minimum number of same shifts to 800. Per image, we computed the F_1 score, as proposed in Eqn. 2.4 on page 14. We sorted the obtained F_1 scores for one test run, i. e. one feature type combined with either lexicographic or nearest neighbor sorting, and plotted the F_1 scores feature group, as shown in Fig. 2.6. In these plots, matching by approximate nearest neighbors is shown in red, while lexicographic sorting is shown in blue. For the large majority of the feature sets, sorting by approximate nearest neighbors outperforms lexicographic sorting.

To analyze this more closely, we counted the number of times that kd-tree outperforms lexicographic sorting with respect to the F_1 score, but also with respect to precision and recall. The results are shown in Tab. 2.3. In the left part of the table, we counted the number of times that approximate nearest neighbors outperformed lexicographic sorting. For the F_1 score (see column 1), CIRCLE, HU, DWT, LIN and SVD exhibit more often worse performance when using nearest neighbors. The second and third column highlight an interesting aspect, namely that the improvement of using approximate nearest neighbors is almost exclusively due to the better recall rate, i. e. improved detection of actually copied areas. Conversely, precision is in almost all cases worse compared to lexicographic sorting.

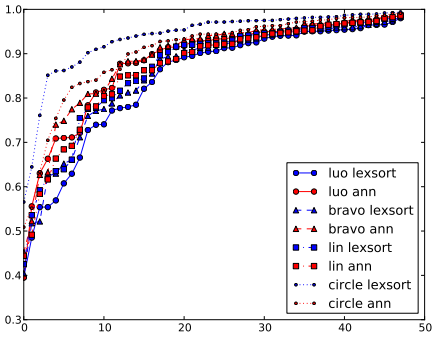
The right part of Tab. 2.3 provides further details of our analysis. We sorted the results by performance and computed the mean difference of the F_1 score, Precision and Recall obtained from approximate nearest neighbors versus lexicographic sorting, and its standard deviation. A positive sign denotes better performance by approximate nearest neighbors, and vice versa. Thus, for instance the F_1 score of CIRCLE is on average three points worse when approximate nearest neighbor sorting is used. On the other hand, overall the benefits of approximate nearest neighbors outweigh the smaller losses in some feature sets. Considering the two rightmost columns on precision and recall, it can also be seen that in general the gain in the recall is typ-



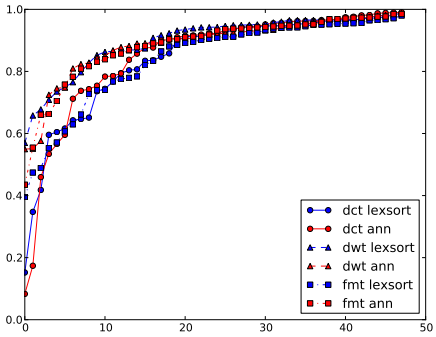
(a) Moment-based features



(b) Features from dimensionality reduction



(c) Intensity-based features



(d) Frequency-based features

Figure 2.6: Comparison of the F_1 score for lexicographic sorting (green) to approximate nearest neighbors sorting (red) over all 48 images, for all feature sets.

ically larger than the loss in precision. Thus, we believe that in general, sorting by approximate nearest neighbors is a better choice than using lexicographic sorting.

2.2.3 Detection of copies after affine transforms

One of the advantages of keypoint-based methods over block-based methods is their intrinsic rotation- and scale-invariance (see e. g. [Huan 08],[Pan 10]). For instance, SIFT features consist of an orientation- and scale-invariant part that can be used for matching. The orientation- and scale-dependent part can then be used for postprocessing the detected matches.

In some works on block-based methods, the feature vectors were also designed to be rotation- and scale-invariant. Typical examples of such rotation-invariant features are CIRCLE proposed by Wang *et al.* [Wang 09a] and ZERNIKE proposed by Ryu *et al.* [Ryu 10]. Features that are pseudo-invariant to rotation and scaling have also been proposed. Bayram *et al.* [Bayr 09] proposed the FMT features, where a ro-

	# cases ann > lex.-sort			median of rel. difference		
Feature	F_1	Precision	Recall	F_1	Precision	Recall
Blur	46	0	47	0.02 ± 0.02	-0.01 ± 0.04	0.04 ± 0.03
Hu	11	0	17	-0.02 ± 0.06	-0.04 ± 0.18	-0.01 ± 0.04
Zernike	48	0	48	0.05 ± 0.05	-0.05 ± 0.08	0.11 ± 0.07
PCA	48	0	48	0.09 ± 0.06	-0.04 ± 0.07	0.17 ± 0.07
SVD	26	0	36	-0.01 ± 0.09	-0.03 ± 0.14	0.01 ± 0.07
KPCA	48	0	48	0.06 ± 0.05	-0.06 ± 0.08	0.14 ± 0.08
Luo	47	0	47	0.04 ± 0.03	-0.02 ± 0.05	0.07 ± 0.05
Bravo	48	0	48	0.03 ± 0.03	-0.01 ± 0.03	0.05 ± 0.04
Lin	22	36	24	-0.00 ± 0.02	-0.00 ± 0.01	-0.00 ± 0.02
Circle	0	1	0	-0.03 ± 0.03	-0.01 ± 0.01	-0.04 ± 0.04
DCT	39	0	46	0.01 ± 0.04	-0.06 ± 0.11	0.07 ± 0.07
DWT	15	2	46	-0.01 ± 0.03	-0.05 ± 0.05	0.03 ± 0.04
FMT	48	0	48	0.05 ± 0.05	-0.01 ± 0.02	0.07 ± 0.06

Table 2.3: Numerical results for lexicographic sorting versus approximate nearest neighbor sorting on the 48 images of the dataset. No noise, rotation or similar has been added to the copied regions.

tation of the block corresponds to a rotation of the feature dimensions. In this work, the authors proposed to match every region against all 180 circular shifts of a feature vector to achieve rotation invariance. However, for images of realistic dimensions, this brute-force approach is computationally excessively expensive.

However, even when a rotation- and scale-invariant feature set is used, it is not straightforward to exploit its full potential. The reason is that the remainder of the pipeline (see Fig. 2.5) must also be extended for rotation- and scale-invariance. More specifically, filtering and postprocessing are important steps to remove false matches. Most authors, e. g. Ryu *et al.* [Ryu 10], limit themselves to morphological filtering of the matches. However, more advanced postprocessing of the matches is not invariant to affine transformations. The same-shift-vector approach can only be used for translations of the copies, i. e. a plain shifted copy in x- and y-direction.

As a consequence, we propose in this section an extension to the same-shift-vector approach. We call it *Same Affine Transform Selection* (SATS) [Chri 10b]. Instead of estimating a translation between the source region and the target region of a copied snippet, we propose to estimate the full affine transformation. Thus, SATS includes translation, rotation, scaling and reflection. At the same time, the runtime of SATS within the whole pipeline is low compared to the more expensive feature extraction and matching steps (confer also Tab. 2.10 on page 40).

We demonstrate the effectiveness of the proposed method by evaluating invariance to rotation. First, we present the performance of 12 block-based feature sets with respect to their rotation invariance. With the 3 best performing features, we show the effectiveness of SATS.

	cattle, max 0°: 2588			*tree*, max 0°: 4755		
Feat.	60°	120°	180°	60°	120°	180°
BRAVO	**2108**	**2154**	**1875**	**2628**	**2625**	**2512**
CIRCLE	**774**	**738**	**541**	**1650**	**1663**	**1762**
ZERNIKE	**609**	**544**	**363**	**1725**	**1686**	**1698**
LUO	736	310	184	853	539	506
HU	294	296	389	1172	1210	1308
FMT	54	60	766	191	199	1633
SVD	198	221	232	1084	982	913
BLUR	91	148	112	691	667	715
LIN	130	71	64	853	803	662
DWT	127	73	64	44	49	76
DCT	9	0	1	20	25	16
PCA	0	0	0	7	1	2

Table 2.4: Number of correct block pair matches, for 60°, 120° and 180° rotations. For comparison, under no rotation, the best performing feature sets found 2588 and 4755, respectively, true closest matches[2].

2.2.3.1 Rotation-invariant features

We evaluated the ability of existing feature sets to match similar blocks when they have undergone rotation. For this purpose, we considered 12 different features, from all four categories (according to Tab. 2.2 on page 18). In detail, we used BLUR, HU and ZERNIKE as moment-based features, PCA and SVD as features resulting from dimensionality reduction, LUO, BRAVO, LIN and CIRCLE as intensity-based features, and DCT, DWT and FMT as frequency-based features.

For testing the feature performance for rotational CMFD, we picked two images from our benchmark dataset, namely *tree* and *cattle*. We inserted the copied parts with rotations of 0°, 60°, 120° and 180° (see Fig. 2.7 for an example of the *tree* image). Then, we subdivided the resulting images into blocks, computed the respective feature vectors per block and matched every feature vector to its nearest neighbor in feature space. Each such match constitutes a *block pair*. Note that, for this particular experiment, no noise has been added to the copies, since we are only interested in the performance of the features under pure rotation.

As a first straightforward measure of the suitability of the features, we counted the block pairs, where one block stems from the source and one from the target region of the copied part. Features with good discriminating power which are also rotational invariant will exhibit a low number of false nearest neighbors in feature space. This will result in a high number of correctly matched block pairs. Without rotation, the best-performing feature set found 2588 correct matches in *tree*, and 4755 correct matches in *cattle*. The columns of Table 2.4 show the correctly matched pairs under rotations of 60°, 120° and 180°.

Based on the results of these experiments, we chose BRAVO, CIRCLE and ZERNIKE for the demonstration of our proposed method. Our findings support the claim of the authors that these three methods are rotation invariant. BRAVO [Brav 09] uses

the average color information of a circular block as the first three features, and the area's entropy as its fourth component. CIRCLE [Wang 09a] uses the mean intensities of circles with different radii around the block center. Finally, the feature vector of ZERNIKE [Ryu 10] is based on the Zernike moments of circular blocks.

Fig. 2.7 shows a visualization of this test. White pixels belong to block pairs, where both blocks truly belong to a copied region. Gray pixels denote matches where at least one block is outside the copied area (and thus a false match). Finally, as a copied region has a minimum size, two blocks are not allowed to lie too close to each other. Thus, black pixels belong to matches where two blocks are located within a certain distance.

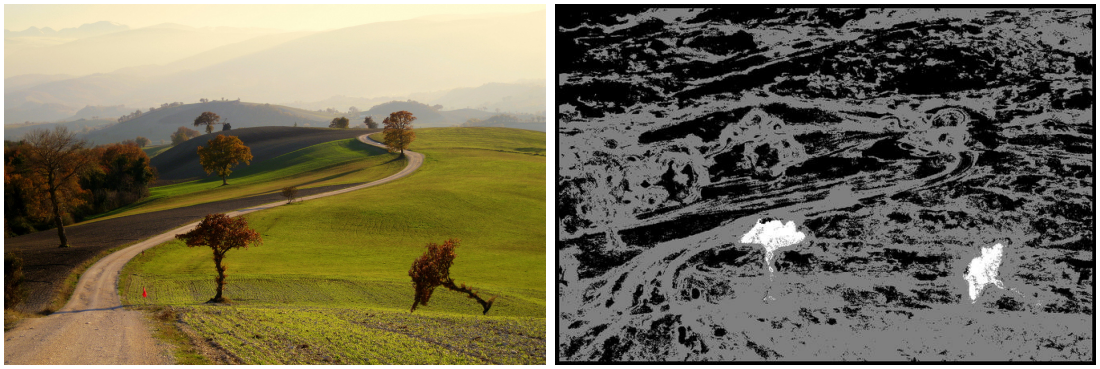


Figure 2.7: Visualization of the performance check of the CMFD features under rotation. White denotes matched feature pairs where both blocks came from copy-moved regions. Gray denotes matched pairs where at least one block is in a non-copied region.

2.2.3.2 Same Affine Transform Selection

We propose a straightforward yet effective replacement for the shift vectors, that can expressly handle affine transformations. The core idea is to explicitly estimate the rotation and scaling parameters from a few blocks, expressed as an affine transformation matrix. Starting from an initial estimate, we apply region growing on block pairs with similar transformation parameters.

Consider the i-th matched pair ${}^{i}_{\mathrm{M}}f$ of feature vectors ${}^{i}_{\mathrm{M}}f_1$ and ${}^{i}_{\mathrm{M}}f_2$, ${}^{i}_{\mathrm{M}}f = ({}^{i}_{\mathrm{M}}f_1, {}^{i}_{\mathrm{M}}f_2)$. In order to determine the rotation and translation between block pairs, we need to examine the coordinates of the block centers. Let $\mathrm{coord}({}^{i}_{\mathrm{M}}f_j)$ denote the coordinates (in row vector form) of the block center from where ${}^{i}_{\mathrm{M}}f_j$ was extracted. Further, let

$$\boldsymbol{a}_i = \mathrm{coord}({}^{i}_{\mathrm{M}}f_1), \qquad \boldsymbol{a'}_i = \mathrm{coord}({}^{i}_{\mathrm{M}}f_2). \tag{2.5}$$

If ${}^{i}_{\mathrm{M}}f$ stems from a copy-move operation with rotation and scaling, then $\boldsymbol{a'}_i$ is related to $\boldsymbol{a}_i$ via an affine transformation:

$$\boldsymbol{a'}_i = \boldsymbol{a}_i \cdot \boldsymbol{A} + \boldsymbol{b} \ , \tag{2.6}$$

where $\boldsymbol{A}$ is a 2×2 matrix containing rotational and scaling parameters, and $\boldsymbol{b}$ is a translation vector. The six unknowns in $\boldsymbol{A}$ and $\boldsymbol{b}$ can be found if at least three matched pairs ${}^{1}_{\mathrm{M}}f$, ${}^{2}_{\mathrm{M}}f$, ${}^{3}_{\mathrm{M}}f$ are available.

Equation 2.6 can be satisfied by searching for matched block pairs that are spatially close to each other, i.e. within a distance t_1. We recover the transformation and treat it as an initial solution to a rotated and/or scaled copied (RS-CMFD) region. Then, we search for further matched block pairs that fit this hypothesis, which is iteratively refined. If the number of block pairs that satisfy the hypothesis exceeds a certain limit t_2, we consider the transformation a candidate for a copy-moved region. We report the involved blocks as well as the transformation parameters as an RS-CMFD result. SATS follows the same principles as shift vectors for robustness to outliers: clustering of similar results, and required minimum number of similar transformations. Thus, it is expected that SATS be equally robust to this type of noise. The details of the proposed verification method is shown in Algorithm 1.

Algorithm 1 SATS: Rotation and scale invariant match filtering.

for every matched pair ${}^{1}_{\mathrm{M}}f = ({}^{1}_{\mathrm{M}}f_1, {}^{1}_{\mathrm{M}}f_2)$ **do**
  Let the hypothesis-set $\mathcal{H} = \{{}^{1}_{\mathrm{M}}f\}$;
  for matches ${}^{i}_{\mathrm{M}}f$ **do**
    if $\mathrm{d}(\mathrm{coord}({}^{1}_{\mathrm{M}}f_1), \mathrm{coord}({}^{i}_{\mathrm{M}}f_1)) < t_1$ and $\mathrm{d}(\mathrm{coord}({}^{1}_{\mathrm{M}}f_2), \mathrm{coord}({}^{i}_{\mathrm{M}}f_2)) < t_1$ **then**
      $\mathcal{H} = \mathcal{H} \cup {}^{i}_{\mathrm{M}}f$;
    end if
  end for
  if $|\mathcal{H}| < 3$ **then**
    continue; // At least three spatially close block pairs
  end if
  From $\mathcal{H}$, compute $\boldsymbol{A}$ and $\boldsymbol{b}$ as described in the text
  for every ${}^{i}_{\mathrm{M}}f$ where $\mathrm{coord}({}^{i}_{\mathrm{M}}f_1)$ is close to matched blocks in $\mathcal{H}$ **do**
    compute $\boldsymbol{a'}_i = \boldsymbol{a}_i \cdot \boldsymbol{A} + \boldsymbol{b}$ as in Eqn. 2.6
    if $\mathrm{d}(\mathrm{coord}({}^{i}_{\mathrm{M}}f_2)), \boldsymbol{a'}_i) < t_1$ **then**
      $\mathcal{H} = \mathcal{H} \cup {}^{i}_{\mathrm{M}}f$
      if $|\mathcal{H}| \mod 10 \equiv 0$ **then**
        recompute $\boldsymbol{A}$ and $\boldsymbol{b}$ to increase stability of the estimate
      end if
    end if
  end for
  if $|\mathcal{H}| > t_2$ **then**
    store $\boldsymbol{A}$, $\boldsymbol{b}$ and mark the blocks in $\mathcal{H}$ as copy-moved.
  end if
end for

Thus, SATS naturally extends the known shift vector selection. Given a rotation-invariant feature set, it can handle arbitrary rotations. The incorporation of different rotation-invariant features is smoothly integrated in the RS-CMFD pipeline. One could equally seamlessly use rotation-and-scale-invariant features. Additionally, we also provided a variant of SATS to serve as a post-processing step for keypoint-based methods.

The runtime complexity is affordable in practice, despite of the two nested loops in Algorithm 1. This comes from the fact that we select suitable neighbors for the initial

Figure 2.8: Four images from [Mahd 07]. Top row: original images, bottom row: manipulated images. From left to right: *kamen*, *beton*, *soldiers*, *helikopter*.

hypothesis greedily. If the candidates fail the neighborhood test, we immediately examine the next region. Theoretically, greedy selection can lead to suboptimal results, but we found the performance to be sufficiently good in practice. Without such assumptions, the complexity mainly consists of: a) an iteration over all blocks and b) a per-block neighborhood search for suitable pairs. More precisely, let n_{B} be the total number of blocks in the image, n_{CB} the number of copied blocks and N the neighborhood size. Then the worst-case runtime is $O(n_{\mathrm{B}} n_{\mathrm{CB}} N)$. It is reasonable to assume that $n_{\mathrm{CB}} \ll n_{\mathrm{B}}$. Thus, the complexity is mainly influenced by the number of blocks in the image, with a (potentially large) coefficient for the neighborhood. With the aforementioned greedy approach, we dampen the effect of this coefficient as well.

An implementation of SATS, called fastSATS, can be downloaded from the web page of our lab[3]. When timing our code, we noticed that the running time of fastSATS is negligible in comparison with the other steps in the CMFD pipeline (see column "postprocessing" in Tab. 2.10 on page 40).

2.2.3.3 Performance Evaluation for SATS

We selected ten test images to evaluate SATS. In each of the images one or more regions were selected for copying. The size of the regions varies among the images. Five of these images stem from our benchmark dataset. The five remaining images are created by Mahdian and Saic [Mahd 07] and are shown in Fig. 2.8. When we conducted this study, we considered these images as relatively typical benchmark images for the related work in copy-move forgery detection. As we did not directly have the copy-moved snippets for these images, we copied rectangular regions from these images, copied and rotated them. Small copied regions are more difficult to detect, while larger copied regions are computationally more demanding for SATS. The copied regions were rotated by 0° to 180° angles, in steps of 15°. Thus, our dataset consisted of $10 \cdot 13 = 130$ images. Ground truth labels were created for every image using our framework for benchmarking CMFD algorithms (see Sec. 2.1).

[3] `http://www5.informatik.uni-erlangen.de`

Detection Error Measures We employed as error metrics the percentage of erroneously matched blocks n_{FP} (false positives) and erroneously missed blocks n_{FN} (false negatives). Note that, as long as a copied region is detected, a high n_{FP} rate is considered to be worse than a high n_{FN} rate. High n_{FP} rates can lead to a confusing overdetection result, which: a) requires a man-in-the-loop to examine every result and b) might even conceal the truly tampered regions.

Comparison to the Originally Proposed Methods We evaluated SATS using the rotation invariant features Bravo, Circle and Zernike. In our implementation of SATS, we set the neighborhood size N to 16 (i. e. we search a 4×4 grid). The distance of matched block pairs t_1 was set to 7 and the minimum number of connected matches t_2 was set to 30. For computational efficiency, the underlying feature extraction was performed on every second block. When the spatial offset between two true positive blocks is odd, a pixelwise exact match is not possible anymore. Thus, there is a trade-off between computational efficiency and feature performance. Furthermore feature extraction was only computed on those blocks with a minimum entropy of 4.0, following the idea of [Bayr 09]. This drastically decreases the runtime and prevents false matches due to too uniform blocks. For the matching step we used a kd-tree as it gives fewer false positives than lexicographic sorting [Chri 10a] (these steps were of course also included in the evaluation of the original methods).

Table 2.5 summarizes the performance of SATS in comparison to the original methods. The results show the average performance over the entire dataset. For all three features, SATS drastically reduces the false positive rate n_{FP}, making false alarms very unlikely. This is mainly due to the clustering of transformation hypotheses controlled by t_2.

A further drawback of the original methods of Circle and Zernike is the proper adjustment of the Euclidean distance threshold (used as similarity criterion). This threshold depends on the image size while, when using SATS, we have a threshold, which is independent of the image size but dependent of the patch size we want to detect.

	Prior art: "same-shift-vectors"		Proposed: SATS	
Feat.	n_{FP}	n_{FN}	n_{FP}	n_{FN}
Bravo	4 ± 3	96 ± 9	0 ± 0.4	22 ± 2
Circle	24 ± 19	66 ± 30	0 ± 0.0	41 ± 32
Zernike	0.4 ± 1	88 ± 24	0 ± 0.0	23 ± 1

Table 2.5: Comparison of the original CMFD method and the proposed SATS approach. The average n_{FP} and n_{FN} rates and the standard deviation are computed over the entire dataset and are given as percentages.

Table 2.6 shows the detailed results for one of the most successful features, Zernike. The average and standard deviation over all rotation angles is depicted. Note that, consistently over all image sizes, about 75% of the copied block pairs are found. A common convention of most copy-move authors is to *mark a copy-moved region as detected, if at least one block pair is correctly matched.* Under this definition, our proposed method exhibits a 100% detection rate of copy-move forgeries.

However, the use of a stricter evaluation measure, like n_{FP} and n_{FN} rates, provides a better insight on the performance of a method.

SATS with Zernike				
Image	x-dim.	y-dim.	n_{FP}	n_{FN}
soldiers	420	300	$0.00\% \pm 0.0\%$	$23.0\% \pm 0.9\%$
concrete	640	480	$0.00\% \pm 0.0\%$	$23.6\% \pm 0.4\%$
kamen	640	480	$0.00\% \pm 0.0\%$	$23.3\% \pm 0.3\%$
helicopter	640	480	$0.00\% \pm 0.0\%$	$21.8\% \pm 0.9\%$
giraffe	800	533	$0.00\% \pm 0.0\%$	$22.1\% \pm 0.4\%$
tree	1024	683	$0.00\% \pm 0.0\%$	$21.8\% \pm 0.6\%$
cattle	1280	854	$0.00\% \pm 0.0\%$	$22.7\% \pm 0.8\%$
beach wood	3264	2448	$0.00\% \pm 0.0\%$	$23.1\% \pm 1.2\%$
kore	3872	2592	$0.00\% \pm 0.0\%$	$22.8\% \pm 0.9\%$
swan	3888	2592	$0.00\% \pm 0.0\%$	$22.0\% \pm 1.2\%$

Table 2.6: SATS performance of the Zernike features, and the sizes of the respective test images. The columns show the average and the standard deviation over all rotation angles.

We also tested our approach with different degrees of JPEG compression, ranging from JPEG quality 50% to 100% in steps of 10%. Since the performance of SATS-Zernike and Bravo did not significantly vary with the rotation angle, we only tested a 90° rotation. The results over the different images were highly stable, with a n_{FP} rate of 0% and n_{FN} rates that were comparable to those of the uncompressed images.

2.2.4 Comparison of existing methods

We conclude the examination of CMFD algorithms with a comprehensive evaluation of the most important feature sets that have been proposed at the time of writing. We incorporate the insights from the previous sections, and use approximate nearest neighbors for matching (see Sec. 2.2.2) and SATS for postprocessing of the matches (see Sec. 2.2.3). The evaluation is conducted on all images of the proposed dataset (see Sec. 2.1). We analyse the performance of 15 block-based and keypoint-based feature sets. The experimental setup is challenging, we examine various types of postprocessing on the manipulated images, as well as on the manipulated regions. Our experiments show that Sift features are ideally suited for a real-time online screening of large databases, while the best performing block-based methods, in alphabetical order DCT, DWT, KPCA, PCA and Zernike, are best for an offline, in-depth examination of an image.

First, we describe the setup of the CMFD pipeline and the employed error metrics. We then present a series of experiments. We start with an evaluation on image level, then on pixel level (using both as presented in Sec. 2.1). Then, we downsampled the input images to 50% of their original size and report the detection rates. Afterwards, we comment on the computational and memory requirements of the algorithms. Fi-

nally, we discuss the obtained insights and give recommendations for implementing a real-world CMFD system.

2.2.4.1 Algorithm setup

Matching Matching with a kd-tree yields a relatively efficient nearest neighbor search. Typically, the Euclidean distance is used as a similarity measure. In prior work, it has been shown that the use of kd-tree matching leads, in general, to better results than lexicographic sorting, but the memory requirements are significantly higher [Chri 10a]. In our setup we matched feature vectors using the approximate nearest neighbor method of Muja *et al.* [Muja 09]. It uses multiple randomized kd-trees for a fast neighbor search.

Postprocessing We use our improved postprocessing method SATS, as presented in Sec. 2.2.3 [Chri 10b]. It groups locations of feature vectors to clusters which have undergone the same affine transformation. Although not explicitly reported in this paper, we experimented with all of these methods. Ultimately, we chose a threshold based on the SATS-connected area to filter out spurious detections, as SATS provided very reliable results and can be applied on both, block-based and keypoint-based features.

Our pipeline for this evaluation is a specific instance of the general pipeline, as it is stated in Sec. 2.2.1. From our investigations of the matching step and the postprocessing step, we used the approximate nearest neighbors for matching, and the Same Affine Transform Selection for postprocessing.

Given an image, the detected regions are computed as follows.

1. Convert the image to grayscale when applicable (exceptions: the features of Bravo-Solorio *et al.* [Brav 11] and Luo *et al.* [Luo 06], which require all color channels for the feature calculation)

2. For block-based methods:

   (a) Tile the image in n_B overlapping blocks of size $b \times b$, with a step size of one pixel.

   (b) For every block, compute a feature vector ${}^{i}_{\mathrm{M}}f$, where $1 \leq i \leq n_\mathrm{B}$.

   For keypoint-based methods:

   (a) Scan the image for keypoints (i. e. high entropy landmarks).

   (b) Compute for every keypoint a feature vector ${}^{i}_{\mathrm{M}}f$. These two steps are typically integrated in a keypoint extraction algorithm like SIFT or SURF.

3. Match every feature vector by searching its approximate nearest neighbor. Let F_{ij} be a matched pair consisting of features ${}^{i}_{\mathrm{M}}f$ and ${}^{j}_{\mathrm{M}}f$, where i, j denote feature indices, and $i \neq j$. Let $\mathrm{coord}({}^{i}_{\mathrm{M}}f)$ denote the image coordinates of the block or keypoint from which ${}^{i}_{\mathrm{M}}f$ was extracted. Then, $\boldsymbol{v}_{ij}$ denotes the translational difference ("shift vector") between positions $\mathrm{coord}({}^{i}_{\mathrm{M}}f)$ and $\mathrm{coord}({}^{j}_{\mathrm{M}}f)$.

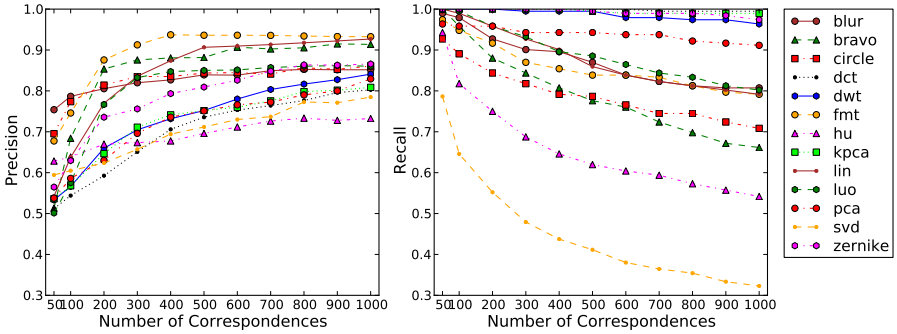


Figure 2.9: Results at image level for different minimum number of correspondences.

4. Remove pairs F_{ij} where $\|\text{coord}({}^{i}_{\text{M}}f), \text{coord}({}^{j}_{\text{M}}f)\|_2 < \tau_1$, where $\|\boldsymbol{x}, \boldsymbol{y}\|_2$ denotes the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$.

5. Let $|\mathcal{H}(A)|$ be the number of pairs satisfying the same affine transformation A. Remove all matched pairs where $|\mathcal{H}(A)| < \tau_2$.

6. If an image contains connected regions of more than τ_3 connected pixels, it is denoted as tampered.

2.2.4.2 Preface to the Experiments

In the first series of experiments, we evaluated the detection rate of tampered images. In the second series, we evaluated pixelwise the detection of copied regions, in order to obtain a more detailed assessment of the discriminative properties of the features.

In total, we conducted experiments with about 60000 images in order to better understand the behavior of the different feature sets. The complete result tables, as well as the source code to generate these results, can be downloaded from the web-site of the Pattern Recognition Lab of the University of Erlangen-Nuremberg[4].

2.2.4.3 Threshold Determination

Thresholds that are specific to a particular feature set were manually adjusted to faithfully fit the benchmark dataset. Most threshold values for the processing pipeline were fixed across the different methods when possible to allow for a fairer comparison of the feature performance.

- Block size b: We chose to use a block size of 16 pixels. We found this to be a good trade-off between detected image details and feature robustness. Note that the majority of the original methods also proposed a block size of 16 pixels.

- Minimum Euclidean distance τ_1: Spatially close pixels are closely correlated. Thus, matches between spatially close blocks should be avoided. In our experiments, we set the minimum Euclidean distance between two matched blocks to 50 pixels.

[4] `http://www5.cs.fau.de/`

- Minimum number of correspondences τ_2: This threshold reflects the minimum number of pairs which have to fulfill the same affine transformation between the copied and the pasted region. Thus, it compromises the improved noise suppression and the false rejection of small copied regions. τ_2 strongly depends on the features, as some features generate denser result-maps than others. Consequently, τ_2 has to be chosen for each feature individually. We empirically determined appropriate values τ_2 as follows. From our dataset, we created CMFD benchmark images with JPEG quality levels between 100 and 70 in steps of 10. Thus, we evaluated on the 48 tampered images for $48 \times 4 = 192$ images. The JPEG artifacts should simulate a training set with slight pixel distortions. Per block-based feature, we estimated τ_2 by optimizing the F_1-measure at image level. The results of the experiment are shown in Fig. 2.9. Please note that throughout the experiments, we were sometimes forced to crop the y-axis of the plots, in order to increase the visibility of the obtained results. The feature set specific values for τ_2 are listed in the rightmost column of Tab. 2.7. For the sparser keypoint-based methods, we require only $\tau_2 = 4$ correspondences.

- Area threshold τ_3: The area threshold corresponds to the minimum number of connected tampered pixels. In our experiments, we set $\tau_3 = \tau_2$ for the block-based methods and $\tau_3 = 1000$ for the keypoint-based methods to remove spurious matches[5].

- Individual feature parameters: We omitted the Gaussian pyramid decomposition for the Hu-Moments (in contrast to the original method by [Wang 09b]). This variant gave better results on our benchmark data. For Circle, we had to use a different block size $b = 17$, as this feature set requires odd sized blocks for the radius computation. For KPCA, two parameters had to be determined, namely the number of samples M and the variance of the Gaussian kernel σ. We set up a small experiment with two images (with similar proportions as images from our database) in which for both images a block of size 128×128 was copied and pasted. Then we varied the parameters and chose the best result in terms of the F_1-measure. We observed that with increasing σ and M the results became slightly better. We empirically determined that values of $M = 192$ and $\sigma = 60$ offer an overall good performance. Note that, these values are larger than what Bashar *et al.* [Bash 10] used. For the remaining features, we used the parameters as suggested in the respective papers.

2.2.4.4 Detection at Image Level

We split these experiments in a series of separate evaluations. As error metrics, we use precision and recall per image, as recommended in Sec. 2.1.4. In the tables, we additionally give the F_1 score. We start with the baseline results, i. e. direct 1-to-1 copies (no postprocessing) of the pixels. Subsequent experiments examine the cases

[5] Alternatively, it would be possible to set the threshold for keypoint matching stricter, and then to omit τ_3 completely. However, we preferred this variant (i. e. a more lenient matching threshold) in order to gain better robustness to noise.

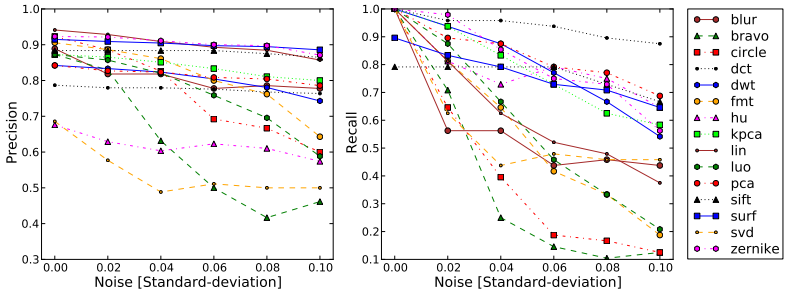
Method	Precision	Recall	F_1	τ_2
Blur	88.89	100.00	94.12	100
Bravo	87.27	100.00	93.20	200
Circle	92.31	100.00	96.00	200
DCT	78.69	100.00	88.07	1000
DWT	84.21	100.00	91.43	1000
FMT	90.57	100.00	95.05	200
Hu	67.61	100.00	80.67	50
KPCA	87.27	100.00	93.20	1000
Lin	94.12	100.00	96.97	400
Luo	87.27	100.00	93.20	300
PCA	84.21	100.00	91.43	1000
Sift	88.37	79.17	83.52	4
Surf	91.49	89.58	90.53	4
SVD	68.57	100.00	81.36	50
Zernike	92.31	100.00	96.00	800
Average	85.54	97.92	90.98	–

Table 2.7: Results for plain copy-move at image level in percent.

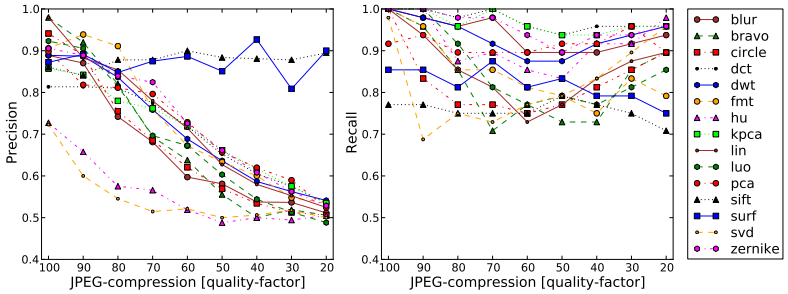
of noise on the copied region, JPEG compression on the entire image, rotation and scaling of the copied region.

Plain Copy-Move As a baseline, we evaluated how the methods perform under ideal conditions. We used the 48 original images, and spliced 48 images without any additional modification. We chose per-method optimal thresholds for classifying these 96 images. Interestingly, although the sizes of the images and the manipulation regions vary greatly on this test set, 13 out of the 15 tested features perfectly solved this CMFD problem with a recall-rate of 100% (see Tab. 2.7). However, only four methods have a precision of more than 90%. This means that most of the algorithms, even under these ideal conditions, generate some false alarms. This comes mainly from the fact that the images in the database impose diverse challenges, and the large image sizes increase the probability of false positive matches.

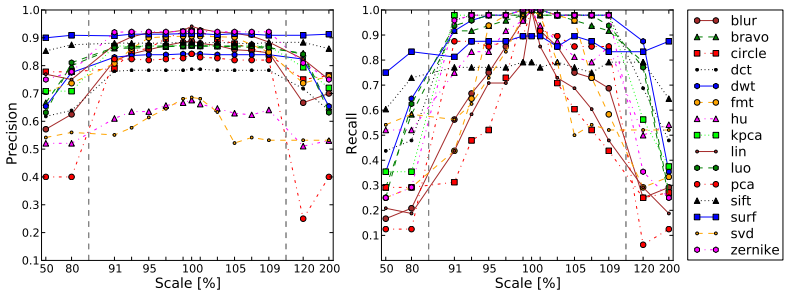
Robustness to Gaussian Noise We normalized the image intensities between 0 and 1 and added zero-mean Gaussian noise with standard deviations of 0.02, 0.04, 0.06, 0.08 and 0.10 to the inserted snippets before splicing. Besides the fact that a standard deviation of 0.10 leads to clearly visible artifacts, 7 out of 15 features drop to under 50% recall rate, while the precision decreases only slightly, see Fig. 2.10a. DCT exhibits a remarkably high recall, even when large amounts of noise are added. PCA, Sift, Surf and Hu also maintain their good recall, even after the addition of large amounts of noise. At the same time, several methods exhibit good precision. Among these, Surf provides a good balance between precision and recall, followed by PCA.



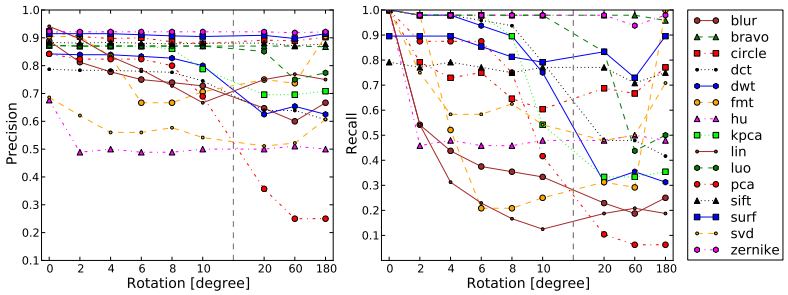
(a) White Gaussian noise



(b) JPEG compression



(c) Scaled copy



(d) Rotated copy

Figure 2.10: Results at image level for different postprocessing operations.

Robustness to JPEG compression artifacts We introduced a common global disturbance, JPEG compression artifacts. The quality factors varied between 100 and 20 in steps of 10, as provided by `libjpeg`[6]. Per evaluated quality level, we applied the same JPEG compression to 48 forgeries and 48 complementary original images. For very low quality factors, the visual quality of the image is strongly affected. However, we consider at least quality levels down to 70 as reasonable assumptions for real-world forgeries. Fig. 2.10b shows the results for this experiment. The precision of SURF and SIFT remains surprisingly stable, while block-based methods slowly degenerate to a precision of 0.5. On the other hand, many block-based methods exhibit a relatively high recall rate, i. e. miss very few manipulations. Among these, KPCA, DCT, ZERNIKE, BLUR and PCA constantly reach a recall of 90% or higher.

Scale-invariance One question that recently gained attention was the resilience of CMFD algorithms to affine transformations, like scaling and rotation. We conducted an experiment where the inserted snippet was slightly rescaled, as is often the case in real-world image manipulations. Specifically, we rescaled the snippet between 91% and 109% of its original size, in steps of 2%. We also evaluated rescaling by 50%, 80%, 120% and 200% to test the degradation of algorithms under larger amounts of snippet resizing. Note that we only scaled the copied region, not the source region. Fig. 2.10c shows the results for this experiment. Most features degenerate very fast even at small amounts of up- or down-sampling. Some methods, namely KPCA, ZERNIKE, LUO, DWT, DCT and PCA are able to handle a moderate amount of scaling. For more extreme scaling parameters, keypoint-based methods are the best choice: their performance remains relatively stable across the entire range of scaling parameters.

Rotation-invariance Similar to the previous experiment, we rotated the snippet between 2° to 10°, in steps of 2°, and also tested three larger rotation angles of 20°, 60° and 180°. In prior work [Chri 10b, Chri 10a], we already showed that ZERNIKE, BRAVO and CIRCLE are particularly well-suited as rotation-invariant features. Our new results, computed on a much more extensive data basis, partially confirm this. Fig. 2.10d shows the results. ZERNIKE features provide the best precision, followed by SURF, CIRCLE, LUO and BRAVO. In the recall-rate, BRAVO and ZERNIKE yield consistently good results and thus seem to be very resilient to rotation. For small amounts of rotation, KPCA and LUO perform also strongly, for higher amounts of rotation, the SURF features perform best. FMT, LIN, HU and BLUR seem not to be well suited for handling rotation.

Robustness to Combined Transformation In this experiment, we examined the performance under several joint effects. We rotated the snippet by 2°, scaled it up by 1% and compressed the image with a JPEG-compression level of 80. In three subsequent setups, we increased per step the rotation by 2°, increased scaling by 2%, and decreased the JPEG quality by 5 points. In setup 5 and 6, slightly stronger parameters were chosen: rotation was set to 20° and 60°, scaling was set to 120% and

[6] `http://libjpeg.sourceforge.net/`

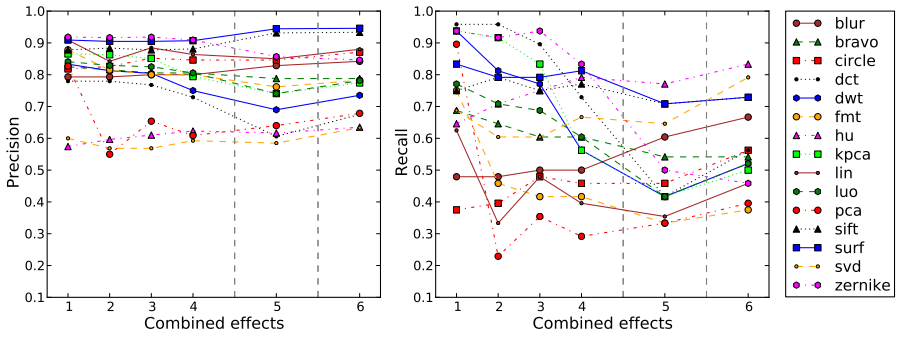


Figure 2.11: Results for different combined transformations at image level.

140%, and JPEG quality was set to 60 and 50, respectively. Fig. 2.11 shows that high precision can be achieved for several feature sets. The best recall for small variations is achieved by DCT and Zernike. For the fourth step, Surf and Sift are almost on a par with Zernike. Note that also in the fourth step, a number of features exhibit a recall below 50%, and can thus not be recommended for this scenario. For large rotations and scaling in the combined effects (see the scenarios 5 and 6), Sift and Surf have the best precision and very good recall.

2.2.4.5 Detection at Pixel Level

A second series of experiments considers the accuracy of the features at pixel level. The goal of this experiment is to evaluate how precisely a copy-moved region can be marked. However, this testing has a broader scope, as it is directly related with the discriminating abilities of a particular feature set. Under increasingly challenging evaluation data, experiments on per-match level allow one to observe the deterioration of a method in greater detail. We repeated the experiments from the previous subsections, with the same test setups. The only difference is that instead of classifying the image as original or manipulated, we focused on the number of detected (or missed, respectively) copied-moved matches.

For each detected match, we check the centers of two matched blocks against the corresponding (pixelwise) ground truth image. All boundary pixels are excluded from the evaluation (see also Fig. 2.3). Please note that all the measures, e. g. false positives and false negatives, are computed using all the pixels in the tampered images only. Note also, that it is challenging to make the pixelwise comparison of keypoint- and block-based methods completely fair: as keypoint-based matches are by nature very sparse, we are not able to directly relate their pixel-wise performance to block-based methods. Thus, we report the *postprocessed* keypoint matches (as described in Sec. 2.2.1).

Plain Copy-Move Tab. 2.8 shows the baseline results for the dataset at pixel level. Similarly to the experiment at image level, all regions have been copied and pasted without additional disturbances. Note that we calibrated the thresholds for all methods in a way that yields very competitive (comparable) detection performances.

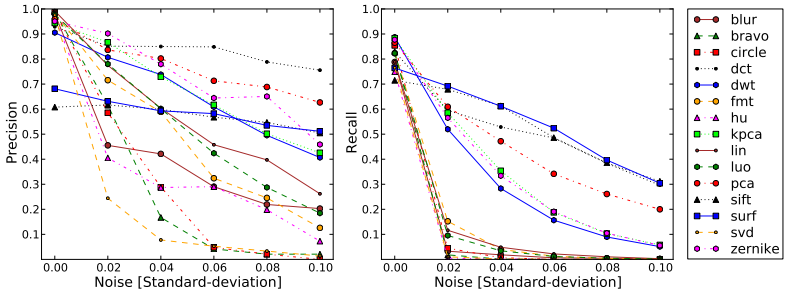
Method	Precision	Recall	F_1
Blur	98.07	78.81	86.19
Bravo	98.81	82.98	89.34
Circle	98.69	85.44	90.92
DCT	92.90	82.85	84.95
DWT	90.55	88.78	88.86
FMT	98.29	82.33	88.79
Hu	97.08	74.89	82.92
KPCA	94.38	88.36	90.24
Lin	99.21	78.87	86.69
Luo	97.75	82.31	88.41
PCA	95.88	86.51	89.82
Sift	60.80	71.48	63.10
Surf	68.13	76.43	69.54
SVD	97.53	76.53	83.71
Zernike	95.07	87.72	90.29
Average	92.21	81.62	84.92

Table 2.8: Results for plain copy-move at pixel level in percent.

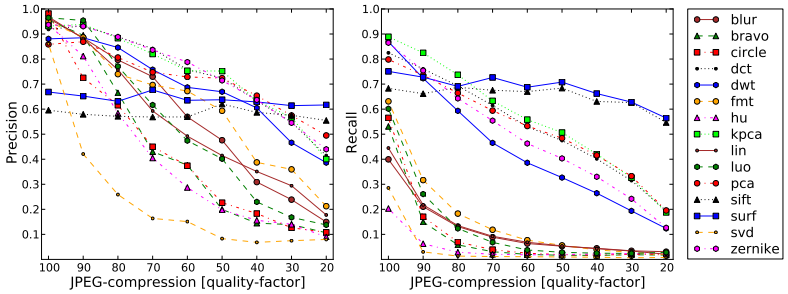
Robustness to Gaussian Noise We used the same experimental setup as in the per-image evaluation, i. e. zero-mean Gaussian noise with standard deviations between 0.02 and 0.1 has been added to the copied region. The goal is to simulate further postprocessing of the copy. At pixel level, this experiment shows a clearer picture of the performance of the various algorithms (see Fig. 2.12a). DCT, Sift and Surf provide the best recall. DCT also outperforms all other methods with respect to precision. The performance of the remaining features splits in two groups: Circle, Blur, Bravo, SVD and Hu are very sensitive to noise, while PCA, Zernike, KPCA and DWT deteriorate slightly more gracefully.

Robustness to JPEG compression artifacts We again used the same experimental setup as in the per-image evaluation, i. e. added JPEG compression between quality levels 100 and 20. Fig. 2.12b shows the resilience at pixel level against these compression artifacts. The feature sets form two clusters: one that is strongly affected by JPEG compression, and one that is relatively resilient to it. The resilient methods are Sift, Surf, KPCA, DCT, PCA, Zernike, and slightly worse, DWT. The robustness of DCT was foreseeable, as DCT features are computed from the discrete cosine transform, which is also the basis of the JPEG algorithm.

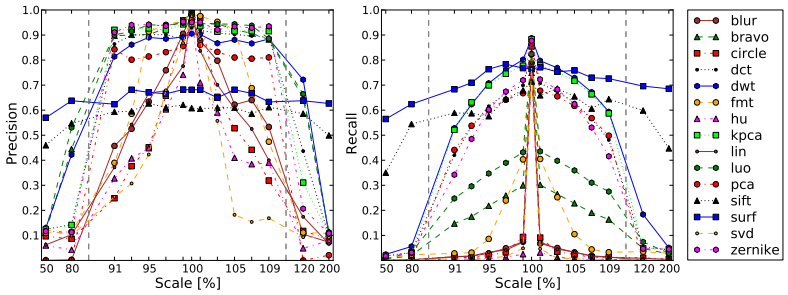
Scale-invariance The experimental setup is the same as on the per-image level analysis. The copy is scaled between 91% and 109% of its original size in increments of 2%. Additionally, we evaluated more extreme scaling parameters, namely 50%, 80%, 120% and 200%. As Fig. 2.12c shows, 7 feature sets exhibit scaling invariance for small amounts of scaling. However, the quality strongly varies. The best performers within these 7 feature sets are DWT, KPCA, Zernike, PCA and DCT. However,



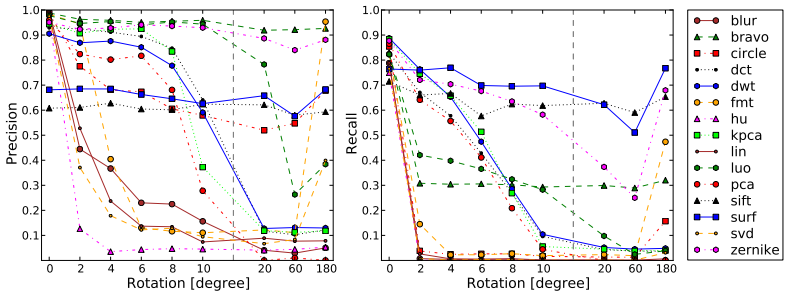
(a) White Gaussian noise



(b) JPEG compression



(c) Scaled copy



(d) Rotated copy

Figure 2.12: Results at pixel level for different postprocessing operations.

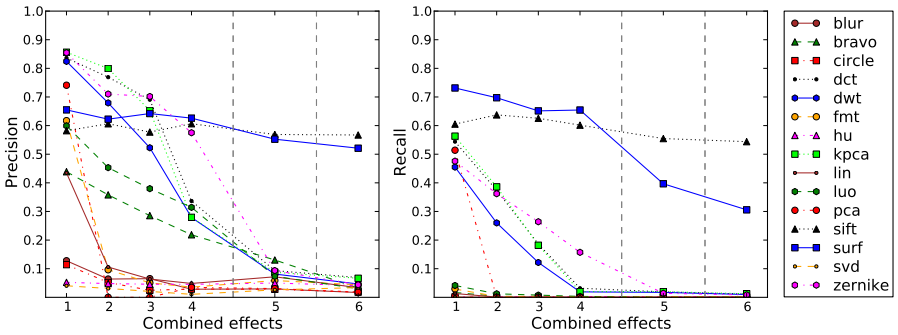


Figure 2.13: Results for different combined transformations at pixel level.

for scaling differences of more than 9%, the keypoint-based features SIFT and SURF are the only features sets that preserve acceptable precision and recall.

Rotation-invariance We evaluated cases where the copied region has been rotated between 2° and 10° (in steps of 2°), as well as for 20°, 60° and 180°. We assumed this to be a reasonable range for practical tampering scenarios. Fig. 2.12d shows the results. Most feature sets show only weak invariance to rotation. Similar to the scaling scenario, SIFT and SURF exhibit the most stable recall. From the block-based methods, ZERNIKE, and also BRAVO and LUO are the best features for larger amounts of rotation. Note that for the special case of 180°, also FMT and CIRCLE perform very well.

Robustness to Combined Transformation Besides the targeted study of single variations in the copied snippet, we conducted an experiment for evaluating the joint influence of multiple effects. Thus, we analyzed images where the copied part was increasingly scaled, rotated and JPEG-compressed. The setup was the same as on image level. Thus, scaling varied between 101% and 107% in steps of 2%, rotation between 2° and 8° in steps of 2°, and the JPEG quality ranges from 80 to 65 in steps of 5. Setup 5 and 6 have different parameters, namely a rotation of 20° and 60°, a scaling of 120% and 140%, and the quality of JPEG compression was set to 60 and 50, respectively. The performance results are shown in Fig. 2.13. In these difficult scenarios, SURF and SIFT perform considerably well, followed by ZERNIKE, DCT, KPCA and DWT. Note that it is infeasible to cover the whole joint parameter space experimentally. However, we take this experiment as an indicator, that the results of the prior experiments approximately transfer to cases where these transformations jointly occur.

2.2.4.6 Detection of Multiple Copies

In recent work, e. g. [Amer 11], the detection of multiple copies of the same region has been addressed. While at image level it typically suffices to recognize whether something has been copied, multiple-copies detection requires that all copied regions be identified. For such an evaluation, we modified the feature matching as follows. Instead of considering the nearest neighbor, we implemented the g2NN strategy

Method	Precision	Recall	F_1	Precision	Recall	F_1
Blur	95.24	52.50	67.31	89.91	54.11	65.20
Bravo	97.54	52.58	68.16	88.75	58.27	67.58
Circle	95.12	60.90	73.75	89.60	62.48	71.43
DCT	19.15	5.37	8.02	66.11	55.76	55.06
DWT	52.15	14.55	21.21	81.88	69.15	71.84
FMT	94.42	54.07	68.14	88.85	60.50	69.91
Hu	94.98	54.08	68.64	89.98	54.61	65.99
KPCA	37.01	7.50	12.05	87.79	62.27	70.06
Lin	96.84	51.04	66.61	90.86	59.96	70.63
Luo	95.53	51.70	66.72	89.32	58.95	68.47
PCA	37.79	9.05	13.95	88.20	61.95	71.77
Sift	11.37	4.95	6.74	17.00	7.34	10.07
Surf	37.49	21.86	26.15	38.31	22.93	26.79
SVD	91.91	59.06	71.51	71.98	58.91	59.33
Zernike	83.15	22.00	33.52	87.55	61.87	69.64
Average	69.31	34.75	44.83	77.31	53.71	60.65

Table 2.9: Results for multiple copies at pixel level. On the left side considering the single best match (as conducted in the remainder of the paper). On the right, we used the g2NN strategy by Amerini *et al.* [Amer 11]. All results are in percent.

byAmerini *et al.* [Amer 11]. This method considers not only the single best-matching feature, but the n best-matching features for detecting multiple copies. Although our dataset contains a few cases of single-source multiple-copies, we created additional synthetic examples. To achieve this, we randomly chose for each of the 48 images a block of 64×64 pixels and copied it 5 times.

Tab. 2.9 shows the results for this scenario at pixel level. On the left side, we used the same postprocessing method as in the remainder of the paper, i. e. we matched the single nearest neighbor. On the right side, we present the results using the g2NN strategy. For many feature sets, precision slightly decreases using g2NN. This is not surprising, as a multiple combinations of matched regions are possible, which also increases the chance for false matches. Still, some methods like Blur, Bravo, etc. are relatively unaffected by this change in postprocessing, while others experience a remarkable performance boost. In particular, DCT, DWT, KPCA, PCA, Zernike, i. e. the strong feature sets in the prior experiments, can significantly benefit from the improved matching opportunities of g2NN. As we discuss later (see Sec. 2.2.4.11), we see this as yet another indicator that these features exhibit very good discriminating properties. The performance of Sift and Surf drops considerably, mainly due to the fact that the random selection of small blocks often yields regions with very few matched keypoints. Although not explicitly evaluated, we expect that selecting copy regions with high entropy (instead of a random selection), would considerably improve the detection rates of Sift and Surf.

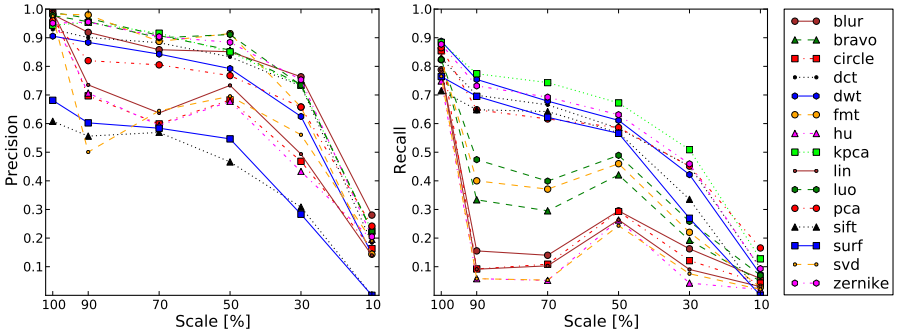


Figure 2.14: Results for interpolated downsampling of the final image prior to the detection process.

2.2.4.7 Downsampling: Computational Complexity versus Detection Performance

The evaluated methods vary greatly in their demand for resources. One widely-used workaround is to rescale images to a size that is computationally efficient. However, this raises the issue of performance degradation. In order to analyze the effects of downsampling, we scaled down all 48 noise-free, one-to-one (i. e. without further postprocessing) forgeries from our database in steps of 10% of the original image dimensions. Note that the detection parameters, as in the whole section, were globally fixed to avoid overfitting. In this sense, the results in this section can be seen as a conservative bound on the theoretically best performance. We observe that the performance of all features considerably drops. When downsampling by a factor of exactly 0.5, results are still better than for other scaling amounts (see Fig. 2.14 for more details). This shows that global resampling has considerable impact on the results. Some feature sets are almost rendered unusable. KPCA, Zernike, DWT, PCA, DCT, Luo, FMT and Bravo perform relatively well. Sift and Surf exhibit slightly worse precision, which might also be due to a suboptimal choice of τ_3 with respect to the reduced number of keypoints in the downscaled images. However, the recall rates are competitive with the block-based methods. For completeness, we repeated the analysis of subsections 2.2.4.5, 2.2.4.5 and 2.2.4.5 on a downsampled (50%) version of the tampered images. The results are presented in the appendix in Fig. C.1 on page 170. The general shape of the performance curves is the same as in the previous sections. However, the performance of recall is more strongly affected by downscaling than precision.

2.2.4.8 Resource Requirements

For block-based methods, the feature-size (see Tab. 2.2) can lead to very high memory use. For large images, this can reach several gigabytes. Tab. 2.10 (right column) shows the per-method minimum amount of memory in MB on our largest images, obtained from multiplying the length of the feature vector with the number of extracted blocks. In our implementation, the effective memory requirements were more than a factor of 2 higher, leading to peak memory usage for DCT and DWT of more than 20GB. Note however, that the feature size of DCT and DWT depends on the block size. For

Method	Feature	Matching	Postpr.	Total	Mem.
Blur	12059.66	4635.98	12.81	16712.19	924.06
Bravo	488.23	5531.52	156.27	6180.42	154.01
Circle	92.29	4987.96	19.45	5103.43	308.02
DCT	28007.86	7365.53	213.06	35590.03	9856.67
DWT	764.49	7718.25	119.66	8606.50	9856.67
FMT	766.60	6168.73	8.07	6948.03	1732.62
Hu	7.04	4436.63	5.36	4452.77	192.51
KPCA	6451.34	7048.83	88.58	13592.08	7392.50
Lin	12.41	4732.88	36.73	4785.71	346.52
Luo	42.90	4772.67	119.04	4937.81	269.52
PCA	1526.92	4322.84	7.42	5861.01	1232.08
Sift	15.61	126.15	469.14	610.96	17.18
Surf	31.07	725.68	295.34	1052.12	19.92
SVD	843.52	4961.11	7.65	5816.15	1232.08
Zernike	2131.27	4903.59	27.23	7065.18	462.03
Average	3549.41	4829.22	105.72	8487.63	2266.43

Table 2.10: Average computation times per image in seconds, and the theoretical minimum peak memory requirements in megabytes. Note that this is a lower bound, for instance our implementation actually requires more than twice of the memory.

better comparability, we kept the block size fixed for all methods. In a practical setup, the block size of these feature sets can be reduced. Alternatively, the feature sets can be cropped to the most significant entries. Some groups explicitly proposed this (e. g. [Pope 04], [Bash 10]). In our experiments, as a rule of thumb, 8GB of memory sufficed for most feature sets using `OpenCV`'s[7] implementation for fast approximate nearest neighbor search.

The computation time depends on the complexity of the feature set, and on the size of the feature vector. Tab. 2.10 shows the average running times in seconds over the dataset, split into feature extraction, matching and postprocessing. Among the block-based methods, the intensity-based features are very fast to compute. Conversely, Blur, DCT and KPCA features are computationally the most costly in our unoptimized implementation. The generally good-performing feature sets PCA, FMT and Zernike are also relatively computationally demanding.

Keypoint-based methods excel in computation time and memory consumption. Their feature size is relatively large. However, the number of extracted keypoints is typically an order of magnitude smaller than the number of image blocks. This makes the whole subsequent processing very lightweight. On average, a result can be obtained within 10 minutes, with a remarkably small memory footprint.

2.2.4.9 Qualitative Results

A more intuitive presentation of the numerical results is provided for four selected examples, shown in Fig. 2.15. On the left, the extracted contours of the keypoint-

[7] `http://opencvlibrary.sourceforge.net/`

Figure 2.15: Indicative performance of SURF (left) versus ZERNIKE (right) features. Top left: Plain copy-move example, SURF and ZERNIKE detect all copies. Top right: JPEG compression quality of 70 hides the copied people from the SURF keypoints. ZERNIKE generates many false positives in homogeneous regions. Bottom left: 20° rotation of the snippet is easily handled by both, SURF and ZERNIKE. Bottom right: Combined transformations, in addition to highly symmetric image content, result in SURF producing a large number of false positives. The block-based ZERNIKE features correctly detect the copied statues.

based method SURF are shown. On the right the matches detected by the block-based ZERNIKE features are depicted. Matched regions are highlighted as brightly colored areas. In the top left image, the people in the middle were covered by a region copied from the right side of the image. Additionally the circle was closed by copying another person. SURF and ZERNIKE correctly detected all copied regions. In the top right image, three passengers were copied onto the sea. The image was afterwards compressed with JPEG quality 70. SURF yielded one correct match but missed the two other persons. ZERNIKE marked all passengers correctly. However, it also generated many false positives in the sky region. In the bottom left image, a 20° rotation was applied to the copy of the tower. Both methods accurately detected the copied regions. This observation is easily repeatable, as long as: a) rotation-invariant descriptors are used, and b) the regions are sufficiently structured. Similarly to the JPEG-compression example, ZERNIKE produced some false positives above the left tower. In the bottom right picture, the two stone heads at the edge of the building were copied in the central part. Each snippet was rotated by 2° and scaled by 1%. The entire image was then JPEG compressed at a quality level of 80. This image is particularly challenging for keypoint-based methods, as it contains a number of high-contrast self-similarities of non-copied regions. ZERNIKE clearly detected the two copies of the stone heads. SURF also detected these areas, but marked as copied a large number of the background due to the structural symmetries.

2.2.4.10 Results by Image Categories

To investigate performance differences due to different texture of the copied regions, we computed the performances according to categories. We subdivided the dataset into the object class categories *living*, *man-made*, *nature* and *mixed*. Although *man-*

made exhibited overall the best performance, the differences between the categories were relatively small. This finding is in agreement with the intuition that the descriptors operate on a lower level, such that object types do not lead to significant performance differences. In a second series of experiments, we used the artists' categorization of the snippets into *smooth*, *rough* and *structure* (see Sec. 2.1.2). Overall, these results confirm the intuition that keypoint-based methods require sufficient entropy in the copied region to develop their full strength. In the category *rough*, SIFT and SURF are consistently either the best performing features or at least among the best performers. Conversely, for copied regions from the category *smooth*, the best block-based methods often outperform SURF and SIFT at image or pixel level. The category *structure* ranges between these two extremes. The full result tables for both categorization approaches can be found in the appendix.

2.2.4.11 Discussion

We believe that the obtained insights validate the creation of a new benchmark dataset. The selection of the evaluation data for the CMFD algorithms is a non-trivial task. While early attempts were mostly centered around small test sets of merely a dozen images, more recent work has used a more extensive picture basis. However, to our knowledge, all existing test sets are somewhat limited in one aspect or another. For instance, preliminary experiments suggested that image size strongly influences the detection result of CMFD algorithms. One workaround is to scale every input image to a fixed size. However, as we show in Fig. 2.14, that interpolation itself influences the detection performance. Furthermore, in the case of downsampling, the size of the tampered region is also reduced, further inhibiting detection. Thus, we conducted all experiments, unless otherwise stated, in the full image resolution (note, however, that the images themselves had different sizes, ranging from 800×533 pixels to 3900×2613 pixels). This greatly increased the risk of undesired matches in feature space, especially when a feature set exhibits weak discriminative power. Consequently, differences in the feature performance became more prominent.

Which CMFD method should be used? During the experiments, we divided the proposed methods in two groups. SIFT and SURF, as keypoint-based methods, excel in computational time and memory footprint. The advantage in speed is so significant, that we consider it worth applying these methods always, independent of the detection goal. Tab. 2.7 and subsequent experiments indicate slightly better result for SIFT than for SURF. The computation of SURF is faster, but we consider this advantage negligible. Thus, we recommend to use SIFT instead of SURF. One should, however, be aware that keypoint-based methods lack the detail that might be desirable for highly accurate detection results. When regions with little structure are copied, e. g. the cats image in Fig. C.10 top right (page 183), keypoint-based methods are prone to miss them. In contrast, highly self-similar image content, as the building in Fig. 2.15 can provoke false positive matches.

The best-performing block-based features can relatively reliably address these shortcomings. Experiments on per-image detection indicate that several block-based features can match the performance of keypoint-based approaches. We conducted additional experiments to obtain stronger empirical evidence for the superiority of one block-based method over another. These experiments measured the pixelwise preci-

sion and recall of the block-based approaches. Experiments on the robustness towards noise and JPEG artifacts showed similar results. DCT, PCA, KPCA, Zernike and DWT outperformed the other methods by a large margin with respect to recall. Their precision also outperformed the other methods for large amounts of noise and heavy JPEG compression. Note, however, that as shown by example in Fig. 2.15 (top), a bad performance in the precision leads to a considerable number of false positive matches. When the copied region is scaled, the aforementioned five block-based features also perform well. Under scaling, the precision of Luo and Bravo might also be sufficient, at least if the copied region is large enough to compensate for weaker recall. For rotated copies, these seven feature sets Zernike, Luo Bravo, DWT, KPCA, DCT and PCA again constitute the best performing group. In general, for detecting scaled and rotated copies, Zernike performed remarkably well. Note also that the computation of Zernike features is very cheap, in relation to their performance.

In a more practical scenario, running a CMFD algorithm on full-sized images can easily exceed the available resources. Thus, we examined, how block-based algorithms perform when the examined image is downsampled by the investigator to save computational time. Not surprisingly, the overall performance drops. However, the best performing feature sets remain relatively stable, and confirm the previous results at a lower performance level.

In all the previous discussion, we tailored our pipeline for the detection of a single one-to-one correspondence between source region and copied region. However, we also evaluated, at a smaller scale, the detection of multiple copies of the same region. We adapted the matching and filtering steps to use g2NN, as proposed by Amerini *et al.* [Amer 11], so that not the single best-matching feature, but the n best-matching features were considered. Interestingly, the already good features DCT, DWT, KPCA, PCA and Zernike profited the most from the improved postprocessing. This re-emphasizes the observation that these feature sets are best at capturing the relevant information for CMFD. With the improved postprocessing by Amerini *et al.*, the advantages of these features can be fully exploited.

In a practical setup, one should consider a two-component CMFD system. One component involves a keypoint-based system, due to its remarkable computational efficiency and small memory footprint. This allows the screening of large image databases, or online, nearly real-time screening within a document processing pipeline. With high precision and a typically over-the-average recall, we consider Sift features a good candidate for such a system.

The second component should be a block-based method, for close and highly reliable examination of an image. We consider Zernike features as a good choice for this component. A single Zernike feature vector consists of only 12 dimensions (see Tab. 2.2), thus the memory requirements are relatively low. Additionally, the computational time for Zernike features is relatively low, compared to the matching performance. Note, however, that a final recommendation has of course to be made based upon the precise detection scenario. We assume that the provided performance measurements, together with the publicly available dataset, can greatly support practitioners and researchers to hand-tailor a CMFD pipeline to the task at hand.

Chapter 3

Exploitation of JPEG artifacts

JPEG compression is currently the most commonly used image format for digital photographs. Most consumer cameras store the picture already in the JPEG format. The main advantages are the simplicity of the format, spatially local compression operations, and the fact that it is an open standard. JPEG compression is lossy, thus every time an image is stored in this format, the content is slightly changed. This property has been the starting point for developing forensic algorithms. The information loss enables analysts to distinguish whether an image has been compressed once or multiple times with the JPEG algorithm. Depending on the scenario, an answer to this question can be very useful in practice. For instance, assume that a photographer claims that an image is directly copied from his camera. Thus, the image should be single-compressed. Evidence that the image, or a part of it, is double-compressed can deliver an initial suspicion to a forensic investigator. A second example is image cropping. For instance, between Fig. 1.1b and Fig. 1.1e on page 2, relevant information has been removed. Cropping a JPEG image and resaving it in the JPEG format is considered as a classical shifted double-compression scenario. Besides that, methods that are tailored for JPEG compression fill an important gap for the forensic investigator. The performance of most blind forensic algorithms quickly deteriorates under increasingly strong JPEG compression.

In this section, we present a fully automated detection scheme for JPEG images that exhibit partially single and double JPEG compression. It exploits Farid's so-called JPEG ghost observation [Fari 09]. First, we present an overview of related JPEG-based algorithms in Sec. 3.2. Then, we restate the JPEG ghost observation in Sec. 3.2, and present an overview on our algorithm in Sec. 3.3. The extracted features are explained in Sec. 3.4, details on the classification are presented in Sec. 3.5. Our experimental result are listed and discussed in Sec. 3.6. The code for this chapter was written by Fabian Zach during his diploma thesis under my supervision. He provided also the evaluation results for this section. Most ideas in this section are from me.

3.1 Related Work

Lukáš and Fridrich [Luka 03] developed one of the first methods of double JPEG-compression detection. The authors exploit the fact that often during recompression, different quantization matrices are used, which leads to a significant high frequency

component in the spectrum of the coefficients. Their method yields a global "yes/no" statement on whether the image has undergone double compression.

A different approach has been presented by Fu *et al.* [Fu 07], and later extended by Li *et al.* [Li 08]. The key observation is that double-compressed images violate Benford's law. The authors extract features from the first 20 AC coefficients[1] of the quantization matrix), and classify the features with an support vector machine.

In some forensic setups, a global solution may suffice. In many cases, however, a local cue for double-compression is sought. Lin *et al.* extended the idea from Lukáš and Fridrich, so that individual blocks can also be detected as tampered [Lin 09b]. During JPEG compression, the image is subdivided in a 8×8 pixel grid. The method assumes that the JPEG block grids of the first and second compression are exactly aligned. This holds for background image regions in particular. The background remains often untouched during manipulation of the image. Thus, as long as the background is not cropped or rescaled, the grids of the first and second compression are aligned.

In more general scenarios, so-called "shifted double-JPEG (SD-JPEG) compression" can be detected. For instance, Qu *et al.* [Qu 08] developed a method for handling arbitrary block grid alignments using independent component analysis. Ye *et al.* [Ye 07] proposed the detection of SD-JPEG cases via the power spectrum of the JPEG DCT coefficients. Barni *et al.* proposed a method that purely relies on non-matching grids [Barn 10]. Recently, Bianchi and Piva developed an integrated approach for jointly detecting aligned double-JPEG compression and SD-JPEG compression [Bian 12a].

All these methods assume different quantization matrices for the first and second compression. Huang *et al.* [Huan 10], on the other hand, presented a method that can detect double compression, even if the same quantization matrix is used. The authors exploited the fact that due to numerical imprecisions, every recompression step alters the statistics of an image.

3.2 JPEG Ghost Observation

We briefly restate Farid's ghost observation [Fari 09]. Let $\boldsymbol{I}_{q_1}$ be an input image that has been compressed with JPEG quality q_1. Assume that a region of the image has previously been compressed with JPEG quality q_0, where $q_0 < q_1$. To detect this double compressed region, define a set of quality factors

$$\mathcal{Q} = \{q_2 | 0 < q_2 < q_1\} \ . \tag{3.1}$$

Recompress image $\boldsymbol{I}_{q_1}$ with the factors in $\mathcal{Q}$, yielding a set of test images $\boldsymbol{I}_{q_1,q_2}$. Now, the pixel-wise squared difference of $\boldsymbol{I}_{q_1}$ and $\boldsymbol{I}_{q_1,q_2}$ is defined as the difference image $\boldsymbol{D}_{q_2}$,

$$\boldsymbol{D}_{q_2}(x,y) = \frac{1}{3} \sum_{i \in \{R,G,B\}} \left(\boldsymbol{I}_{q_1}(x,y,i) - \boldsymbol{I}_{q_2}(x,y,i)\right)^2 \ , \tag{3.2}$$

[1] The first JPEG quantization coefficient is often called DC component, the second to 64th entries are called AC components.

Figure 3.1: Example JPEG ghost. Left: a rectangular region has been double-compressed with primary compression rate $q_0 = 35$. Middle and right: in the difference images $\boldsymbol{\Delta}_{60}$ and $\boldsymbol{\Delta}_{35}$ a "ghost" gradually appears as a darker region. Note also the noise in $\boldsymbol{\Delta}_{60}$ and $\boldsymbol{\Delta}_{35}$ due to the image texture.

where x and y denote the pixel coordinates, and $i \in \{R, G, B\}$ the red, green and blue color channels.

If a region of the image has previously been compressed with a compression factor q_0, $q_0 < q_1$, the squared differences become smaller for this part of the image as q_2 approaches q_0. This local region, termed "ghost", appears darker than the remaining imaage. The smaller difference values are due to the fact, as the coefficients induced by q_2 become increasingly similar to q_0, similar artifacts are introduced in the image. For robustness to image noise and texture, these computed differences are averaged across small windows of size w. Thus, the differences are computed as

$$\begin{aligned}\boldsymbol{\Delta}_{q_2}(x,y) = &\frac{1}{3b^2}\sum_i \sum_{w_x=0}^{w-1}\sum_{w_y=0}^{w-1} (\boldsymbol{I}_{q_1}(x+w_x, y+w_y, i) - \\ &\boldsymbol{I}_{q_2}(x+w_x, y+w_y, i))^2 \quad ,\end{aligned} \tag{3.3}$$

and normalized in the range between 0 and 1.

Fig. 3.1 shows an example of such a ghost. A rectangular double compressed region has been embedded with $q_0 = 35$ (left). In the difference images $\boldsymbol{\Delta}_{60}$ and $\boldsymbol{\Delta}_{35}$, this region appears visually. The exploitation in image forensics is quite direct: examine a number of difference images $\boldsymbol{\Delta}_{q_2}$ for varying q_2 values. If a dark region appears, consider it as doubly compressed. The method is particularly appealing due to its straightforward idea and simplicity of implementation.

However, in practice, this approach is often not applicable. The amount of human interaction can become disproportionally time-consuming for human experts for two main reasons. First, a mixture of differently textured regions leads to interfering noise patterns. In order to recognize a ghost, it is often necessary to closely examine the images (see e. g. [Batt 09a]). Second, the number of difference images can become very large. A ghost is only detected, if the JPEG grid of $\boldsymbol{\Delta}_{q_2}$ is exactly aligned with the JPEG grid of the compression using q_0. For all possible 64 JPEG grid alignments, difference images must be created. Thus, a human expert has to browse $64 \cdot |\mathcal{Q}|$ images. Depending on the scenario, this can easily amount to 300 difference images for examination.

We propose a pattern recognition method that performs this selection automatically for the user. For images under examination, the user can either resort to the

fully automated process, or use the automation as a preprocessing step and then visually inspect the ghost.

3.3 Algorithm Overview

The proposed algorithm consists of 5 steps.

1. Read out the JPEG quality level q_1 of the image under examination. If it has been resaved in a different image format, such as PNG, estimate the former JPEG quality level.

2. For every of the 64 possible JPEG block alignments, recode the image with JPEG quality levels between 30 and q_1.

3. Slide a window over the image under examination. For every JPEG block alignment, compute features ${}_{\mathrm{J}}f_1$ through ${}_{\mathrm{J}}f_6$ on the image window.

4. Classify the feature vector as double- or single-compressed.

5. Apply morphological opening and closing on the blocks that are marked as double-compressed. We continue with the JPEG block alignment with the maximum number of windows classified as double-compressed. If this number exceeds a chosen threshold, the image is considered double-compressed.

The next sections add details to these steps, in particular to feature extraction and classification. The output of this algorithm is either an automated decision at image level ("partially double-compressed: yes/no"), or a single, aggregated output image as shown in Fig. 3.4. In particular, the latter representation is a substantially easier representation than the hundreds of difference images, as proposed in the original method.

3.4 Feature Extraction

The information about a JPEG ghost is contained in the differences of one single window over different quality levels. Consider ROIs which have been single- or double compressed. Example difference curves for this case are shown in Fig. 3.2. Here, the differences are computed over four windows of an image with compression quality $q_1 = 80$, over 50 quality levels $30 \leq q_2 \leq 80$ on a window size $w = 16$. All differences are normalized between 0 and 1. The green curve denotes the differences for single compressed windows. Note that the choice of the window size can be varied, and is a tradeoff between detection detail and robustness. The red curves show the same ROIs, but this time double compressed with $q_0 = 75$. Note that, although differences on single-compressed images are consistently higher than on double-compressed images, these curves are not straightforward to distinguish across different windows: for most datapoints, the single-compressed curve in the left plot lies in below the double-compressed curve in the right plot. These differences are due to different image texture.

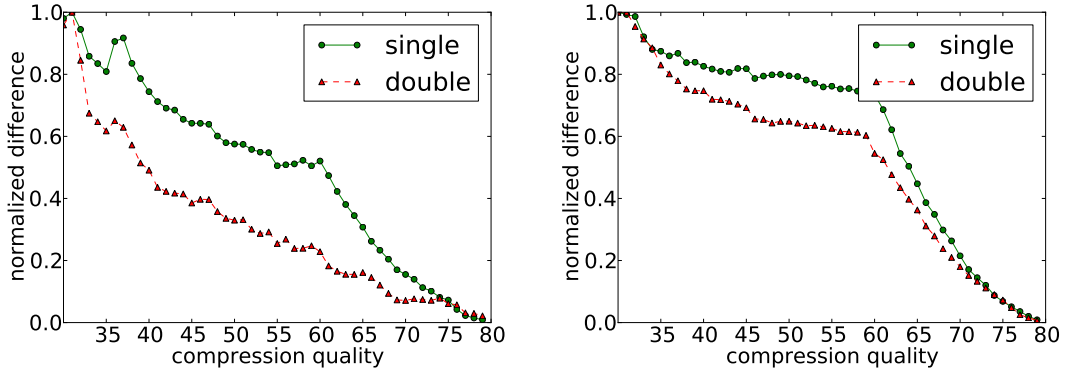


Figure 3.2: Difference curves from example JPEG ROIs. The same ROI has been single- and double-compressed and is plotted in a joint diagram. In green, the difference curves for single compression are shown, in red for double compression.

Our method is based on the analysis of these difference curves. The core idea of this feature set is to target the steeper decay of the double-compression difference curve (see Fig. 3.2). We estimate the quality level q_1 as the global minimum over the curves derived from all windows in the image. We then proceed as follows. Let $c(x)$ be the value of the difference curve for quality level x. We extracted six features that are defined on $c(x)$ for $30 \leq x \leq q_1$. Note that this range implies that we can not detect ghosts with $q_0 < 30$. However, this has been mainly an engineering decision, as we considered cases of $q_0 < 30$ as very unlikely. Let furthermore

$$w_1(x) = \frac{x - 30}{q_1 - 30} \tag{3.4}$$

denote a linear weighting function that puts more emphasis on high JPEG qualities, and

$$w_2(x) = 1 - w_1(x) \tag{3.5}$$

a linear weighting function that emphasizes low JPEG qualities. We employ the following features:

1. The weighted mean value of the curve,

$$_{\mathrm{J}}f_1 = \frac{1}{\sum_{x=30}^{q_1} w_1(x)} \sum_{x=30}^{q_1} w_1(x) \cdot c(x) \ . \tag{3.6}$$

2. The median of all function values on $c(x)$ for $30 \leq x \leq q_1$, i. e. $_{\mathrm{J}}f_2 = \mu_{1/2}$ where

$$\left(P(c(x) \leq \mu_{1/2}) \geq \frac{1}{2}\right) \wedge \left(P(c(x) \geq \mu_{1/2}) \geq \frac{1}{2}\right) \tag{3.7}$$

and $P(c(x) \leq x)$ denotes the cumulative distribution function of the difference values.

3. The slope m, ${}_{\mathrm{J}}f_3 = m$ of the regression line $y = mx + h$ through $c(x)$ for $30 \leq x \leq q_1$.

4. The y-axis intercept, $h = {}_{\mathrm{J}}f_4$, of the regression line $y = mx + h$ through $c(x)$ for $30 \leq x \leq q_1$.

5. The weighted number of points of $c(x)$ with $c(x) < 0.5$,

$$ {}_{\mathrm{J}}f_5 = \frac{1}{\sum_{x=30}^{q_1} w_2(x)} \sum_{x=30}^{q_1} w_2(x) \cdot g_5(x) \ , \tag{3.8} $$

where

$$ g_5(x) = \begin{cases} 1 & \text{if } c(x) < 0.5 \\ 0 & \text{else} \end{cases} \tag{3.9} $$

6. The average squared distance of the actual curve and the linear function

$$ l(x) = 1 - \frac{x - 30}{q_1 - 30} \ , \tag{3.10} $$

i. e. the line with endpoints (30, 1) and (q_1, 0). More formally,

$$ {}_{\mathrm{J}}f_6 = \sum_{x=30}^{q_1} g_6(x) \ , \tag{3.11} $$

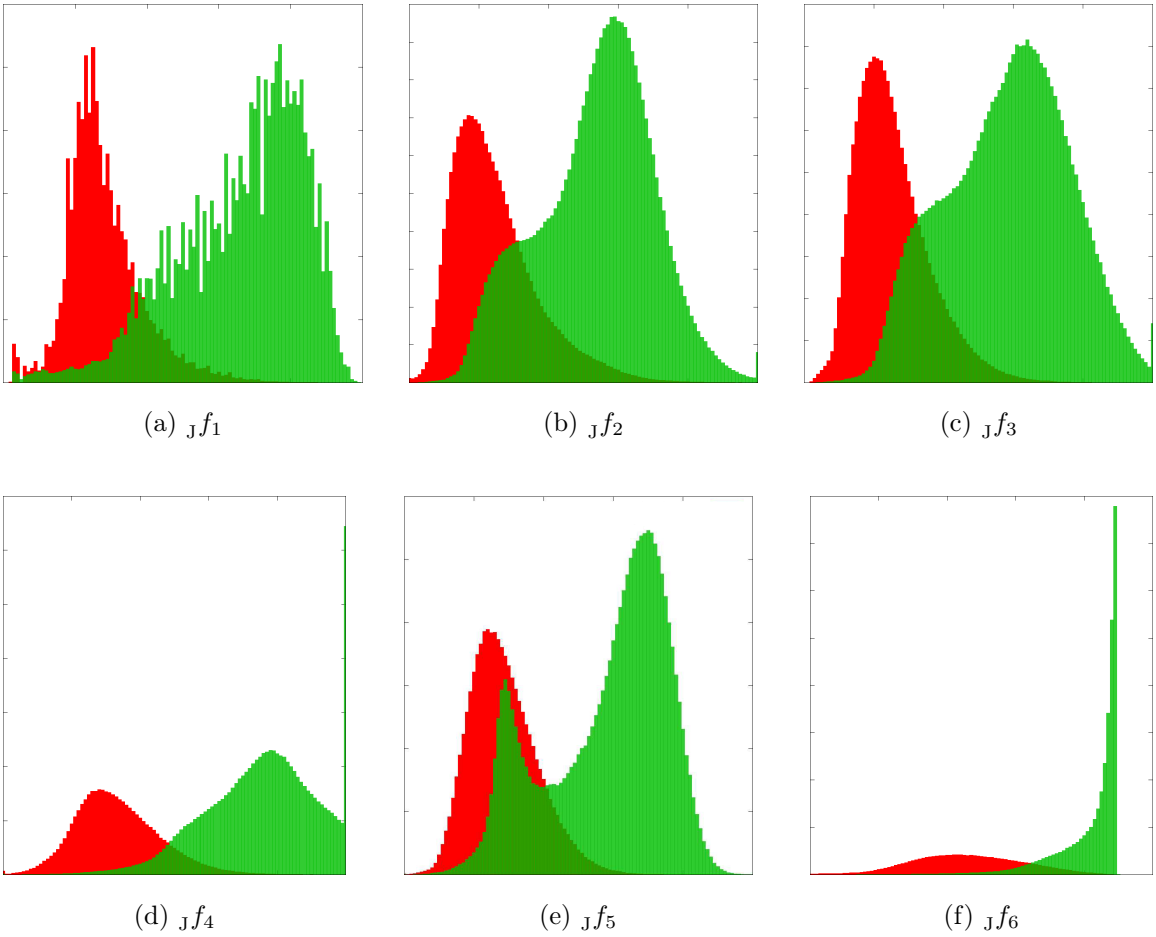
where

$$ g_6(x) = \begin{cases} (l(x) - c(x))^2 & \text{if } l(x) > c(x) \\ 0 & \text{else} \end{cases} \tag{3.12} $$

Note that these features can be computed on an isolated window, i. e. no spatial assumptions or dependencies have been added to the detection of JPEG ghosts. As a consequence, the feature computation can be directly parallelized.

Feature ${}_{\mathrm{J}}f_1$ exploits the fact that for high quality levels, the double-compressed curves are generally lower. Double-compressed areas also exhibit a tendency to more shallow regression lines, exploited in particular in the features ${}_{\mathrm{J}}f_2$, ${}_{\mathrm{J}}f_3$ and ${}_{\mathrm{J}}f_4$. Finally, the squared distance of the error curve to the diagonal line in the error diagram is generally larger for double-compressed images. This relation is captured in feature ${}_{\mathrm{J}}f_6$.

For each feature, we computed a histogram based on more than 2.5 million image windows, see Fig. 3.3. The compression quality levels of these windows were randomly chosen between 50 and 95, with a fixed distance $q_0 - q_1 = 20$. Feature values for single-compressed windows are plotted in green, while features for double-compressed windows are shown in red. From top left to top right, features ${}_{\mathrm{J}}f_1$ to ${}_{\mathrm{J}}f_3$ are shown. From bottom left to bottom right features ${}_{\mathrm{J}}f_4$ to ${}_{\mathrm{J}}f_6$ are plotted. As seen in Fig. 3.3, the proposed features exhibit good separability between single- and double compressed areas. None of the features is perfect, in the sense that there is in every histogram an area where both features co-occur. However, all histograms exhibit distinct peaks for the classes “single-compressed” and “double-compressed”.



(a) ${}_{\mathrm{J}}f_1$ (b) ${}_{\mathrm{J}}f_2$ (c) ${}_{\mathrm{J}}f_3$

(d) ${}_{\mathrm{J}}f_4$ (e) ${}_{\mathrm{J}}f_5$ (f) ${}_{\mathrm{J}}f_6$

Figure 3.3: Histograms for each of the six features, ${}_{\mathrm{J}}f_1$, ${}_{\mathrm{J}}f_2$, ..., ${}_{\mathrm{J}}f_6$, shown in a top to bottom, left to right order. Red histograms are from double compressed windows. Green ones correspond to single-compressed windows.

3.5 Classification

Every block was classified separately, solely based on the features from this block. We experimented with different classification algorithms, namely thresholding, Neural Networks, Random Forests, AdaBoost and Bayesian classification. The `openCV` implementations of the algorithms were used.

The values for thresholding were determined by computing the mean values of the feature distributions for single- and double compressed areas. The actual threshold is then determined as the mean of means for each feature. A block was considered double compressed, if at least three quarters of the feature values exceeded their respective threshold. The Neural Network was a Multilayer Perceptron with a 10-node hidden layer. The activation function is a sigmoid function with $\alpha = \beta = 1$. For the Random Forests, we used 50 trees for classification, while for Discrete AdaBoost 100 trees. The cost functions of all classifiers were set to a balanced state of false positive and false negative rates.

Note, that this classification is conducted for a single window on one out of the 64 possible JPEG block alignments. If applied on an unknown image, this does not suffice: an investigator would still be forced to browse through 64 images in order to get an overview. To avoid this situation, we chose a straightforward post-processing scheme, in order to obtain an automated decision on top of the classified windows. For every of the 64 possible JPEG block alignments, we created a binary map where blocks classified as single-compressed are set to 0. Blocks classified as double-compressed are set to 1. On each of these maps, we applied morphological opening with cross-topology in order to remove outliers that were wrongly classified as double-compressed. Counting the remaining ones in the maps, we keep only the map with the maximum number of entries. If the number of ones in this map exceeds a particular threshold, the image is classified as double-compressed. The JPEG block offset that has been used to create this map is the estimated JPEG block alignment of the ghost.

Additionally, this map can be nicely colored and presented as a visual representation to the forensic investigator, as shown in Fig. 3.4.

3.6 Experiments

We evaluated our method on the Uncompressed Colour Image Database (UCID) by Schaefer and Stich [Scha 04], which was also used in the original presentation of the JPEG ghost approach [Fari 09]. It consists of 1338 images of size 512×384. For each image, we created three variants: A single-compressed version, i. e. an "authentic" image. A version containing a small double-compressed region, as used in [Fari 09] for evaluation. In the third version, only a small region is single-compressed, and the background region is double-compressed, as recommended by [Lin 09b]. When considering all 4014 test images, we have in total twice as many single- as double-compressed pixels. The differently compressed regions were of size 192×192 pixels. In our experiments, for the training of the classifiers, we used 10 percent of the images from all three variants. For each image from the UCID database, we randomly drew the JPEG compression quality q_1 in a range of 50 to 95, and set $q_0 = q_1 - \delta$, where $5 \leq \delta \leq 20$. The window sizes have been varied as well, between 8×8 and 64×64 pixels. Very smooth image regions barely contain compression artifacts. Thus, following [Fari 09], windows with an intensity variance below 5 points have been excluded from the evaluation.

All results are presented as specificity/sensitivity pairs, as described in Sec. 2.1.4. To do so, we define n_{TP}, n_{TN}, n_{FP} and n_{FN} as

$$n_{\mathrm{TP}} = P(\text{double-compressed}|\text{double-compressed}) \tag{3.13}$$

$$n_{\mathrm{TN}} = P(\text{single-compressed}|\text{single-compressed}) \tag{3.14}$$

$$n_{\mathrm{FP}} = P(\text{double-compressed}|\text{single-compressed}) \tag{3.15}$$

$$n_{\mathrm{FN}} = P(\text{single-compressed}|\text{double-compressed}) \ . \tag{3.16}$$

Experiments on Individual Image Windows Tab. 3.1 shows the raw results for the evaluation on individual windows, without any post-processing like the morphological operators. Here, "Thresholding", "MLP", "RF", "Boosting" and "Bayes" denote

classification by thresholding, multi-layer perceptron, random forests, discrete AdaBoost and the Bayesian classifier, respectively. $\delta = q_1 - q_0$ denotes the difference in the primary and secondary compression quality levels q_0 and q_1. In order to have a relatively balanced number of single and double compressed pixels, we only computed on the correct shift of the doubly-compressed region. However, when we evaluated the whole pipeline, we tested all 64 shifts of the JPEG grid.

	δ	8×8	16×16	32×32	64×64
Thresholding	5	0.798/0.702	0.805/0.704	0.816/0.717	0.840/0.596
	10	0.815/0.728	0.821/0.730	0.833/0.745	0.837/0.755
	20	0.844/0.778	0.849/0.783	0.857/0.796	**0.861/0.804**
MLP	5	0.866/0.826	0.855/0.880	0.865/0.891	0.897/0.833
	10	0.865/0.864	0.869/0.873	0.835/0.935	0.917/0.847
	20	0.888/0.910	0.895/0.906	0.865/0.935	**0.925/0.892**
RF	5	0.804/0.889	0.811/0.901	0.838/0.925	0.889/0.890
	10	0.831/0.901	0.834/0.914	0.850/0.941	0.908/0.893
	20	0.865/0.938	0.864/0.937	0.879/0.935	**0.931/0.913**
Boosting	5	0.834/0.886	0.841/0.893	0.847/0.919	0.907/0.870
	10	0.852/0.896	0.858/0.910	0.865/0.934	0.924/0.869
	20	0.895/0.934	0.902/0.938	**0.912/0.957**	0.938/0.916
Bayes	5	0.469/0.984	0.483/0.983	0.505/0.981	0.562/0.976
	10	0.479/0.986	0.497/0.984	0.520/0.983	0.579/0.979
	20	0.506/0.985	0.520/0.984	0.553/0.983	**0.622/0.977**

Table 3.1: Experiments on UCID database for shifted ghost detection on misaligned DCT grids at a per-window level.

Results are presented as pairs of specificity and sensitivity. The best performance per classifer on all combinations of δ and the region size is printed in bold face. As expected, typically the highest tested compression distance of $\delta = 20$, together with the largest examined window size 64×64 yields best results. Note that AdaBoost performed best below the maximum windows size. Note also that for $\delta = 10$ and $\delta = 5$, boosting, followed by neural networks (MLP) and random forests (RF) all provide very strong results. Furthermore, the performance of these three methods degrades gracefully for smaller windows. Thus, as a preprocessing step for guiding a human expert towards a JPEG ghost location, we consider these three classifiers highly suitable. At the same time, the good discrimination for small values of δ clearly improves over the results reported in [Fari 09], which reports $\delta \geq 20$ as a good quality distance for detection. In [Lin 09b], detection rates vary between 50% and 70% for $\delta \leq 10$.

To judge the method's performance in a real-world scenario, we can not assume to know δ. However, note that at least at a region size of 32×32 pixels, the results MLP, Random Forests and Boosting are also for $\delta = 5$ highly competitive.

Automated Tampered Image Detection For better comparison to other methods, we evaluated the proposed algorithm also on full images. For comparison to the

	δ	8×8	16×16	32×32	64×64
Lin	5	0.583/0.640	-	-	-
	10	0.658/0.597	-	-	-
	20	**0.705/0.605**	-	-	-
Thresholding	5	0.806/0.766	0.816/0.760	0.812/0.755	0.830/0.481
	10	0.820/0.759	0.827/0.758	0.830/0.737	0.852/0.439
	20	0.832/0.880	**0.858/0.883**	0.867/0.868	0.933/0.484
MLP	5	0.783/0.870	0.783/0.871	0.749/0.889	0.963/0.355
	10	0.774/0.874	0.777/0.870	0.776/0.857	0.946/0.377
	20	0.864/0.941	**0.892/0.957**	0.865/0.947	0.866/0.517
RF	5	0.756/0.929	0.762/0.934	0.726/0.949	0.978/0.375
	10	0.913/0.857	0.897/0.873	0.882/0.883	0.968/0.416
	20	0.905/0.960	0.904/0.973	**0.908/0.972**	0.959/0.519
Boosting	5	0.823/0.867	0.832/0.867	0.811/0.883	0.982/0.378
	10	0.918/0.880	0.916/0.899	0.916/0.895	0.989/0.412
	20	**0.997/0.939**	0.996/0.956	0.993/0.960	0.997/0.507
Bayes	5	0.576/0.846	0.646/0.841	0.742/0.822	0.882/0.606
	10	0.594/0.892	0.772/0.815	0.840/0.845	0.938/0.684
	20	0.533/0.997	0.647/0.997	0.839/0.988	**0.960/0.955**

Table 3.2: Experiments on UCID database for ghost detection on aligned DCT grids at image level.

method by Lin *et al.*, we split this section in two experimental parts: In the first part, we assumed perfect JPEG block alignment. In the second part, we evaluated the more general (and more realistic) scenario of shifted-double JPEG compression.

Recognition of partially double-compressed JPEG images was conducted as stated in Sec. 3.5. On the marked windows from the previous section, we applied a 3×3 morphological opening with cross-topology on these markings to remove outliers. Then, we considered an image tampered, if 10% of the windows are marked. Note that an an embedded foreground-ghost contains about 20% double-compressed pixels, a background-ghost about $100\% - 20\% = 80\%$. As before, we created three images from every UCID image. Once completely single-compressed, once with an embedded foreground-ghost, and once with an embedded background-ghost.

Tab. 3.2 shows the result for JPEG ghosts that were exactly aligned with the JPEG grid. We used the same notation as in the previous Subsection.

For comparison, we computed the results of Lin *et al.* [Lin 09b] on our test set. As this method operates on 8×8 windows, only these results are presented. Note that the method by Lin *et al.* is more general, in the sense that it can also detect double-compression where $q_0 > q_1$, which is not possible with the JPEG ghost approach. However, this comes at the expense of the accuracy of the method in the presence of very small differences in the compression parameters. Thus, if the initial assumption $q_0 < q_1$ for JPEG ghosts is fulfilled, the proposed method provides much higher specificity and sensitivity rates.

	δ	8×8	16×16	32×32	64×64
Thresholding	5	0.742/0.772	0.755/0.708	0.750/0.698	0.763/0.468
	10	0.762/0.794	0.779/0.733	0.791/0.728	0.795/0.646
	20	0.784/0.836	**0.802/0.795**	0.806/0.786	0.810/0.659
MLP	5	0.982/0.904	0.993/0.934	0.973/0.915	0.967/0.765
	10	0.978/0.932	0.975/0.913	**0.981/0.968**	0.984/0.755
	20	0.977/0.955	0.963/0.939	0.862/0.953	0.988/0.833
RF	5	0.923/0.960	0.940/0.952	0.969/0.952	0.992/0.843
	10	0.938/0.957	0.955/0.954	0.961/0.957	0.993/0.803
	20	0.969/0.972	**0.978/0.972**	0.948/0.963	0.994/0.806
Boosting	5	0.990/0.955	0.987/0.944	0.984/0.947	1.000/0.809
	10	0.978/0.951	0.984/0.948	0.986/0.950	0.998/0.775
	20	0.993/0.971	0.995/0.969	**0.995/0.971**	0.999/0.818
Bayes	5	0.907/0.981	0.919/0.982	0.923/0.983	0.951/0.983
	10	0.923/0.983	0.945/0.983	0.951/0.983	0.966/0.986
	20	0.987/0.985	0.993/0.987	0.995/0.988	**0.997/0.988**

Table 3.3: Experiments on UCID database for shifted ghost detection on misaligned DCT grids at image level.

The best success rates do not occur at the larger window sizes. This is due to the fact that our inserted foreground ghosts of 192×192 pixels are comparably small. When applying morphological opening on the windows that have been marked as double-compressed, more accurate detectors loose too many windows on the boundary of the marked region. This renders very large window sizes less successful. One notable exception is the Bayesian classifier. As can be seen from Tab. 3.1, Bayesian classification exhibits very low specifity, i. e. creates many erroneously marked regions. The morphological operator removes a large number of these false positive markings, and makes also detection with larger window sizes possible. On the converse, the remaining classifiers exhibit their peak performance at window sizes around 16×16 pixels. Again, Discrete Adaboost clearly outperforms the other methods.

In shifted double-compression, the grid of the inserted region is not required to properly align with the JPEG grid of the background. To detect such tampered images, we computed all 64 shifts and selected the one with the highest response of double-compressed blocks (see Tab. 3.3 for the results). A surprising result is that shifted double JPEG compression can be discriminated than the non-shifted version: while e. g. the overall best result in Tab. 3.2 is 0.997/0.939, several results in Tab. 3.3 perform better, e. g. the Bayesian classifier on a 64×64 grid for $\delta = 20$ with 0.997/0.988. We assume that this comes from the shifted recompression with quality factors q_2 during the ghost detection. During feature extraction (see Sec. 3.4), q_1 must be estimated. This estimation, however, has a strong tendency to default to $q_1 = 100$, as the shifted recompression is not aligned with the JPEG grid from q_1. As a consequence, q_1 is estimated as 100, which raises the gap between q_0 and the estimated q_1. Apparently, this adds discriminating power to the features. In future

work, we plan to further investigate this effect. For the moment, we just note that all classifiers except of the simple thresholding provide strong results on picture-level.

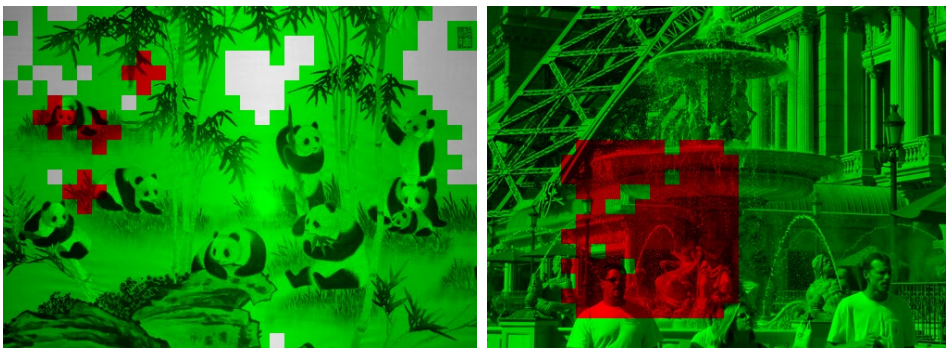


Figure 3.4: Two example markings on individual windows. Green and red are single- and double compressed, respectively. Gray denotes low contrast regions. Left: the rectangular double compression region could only in the high-contrast windows be recovered. Right: the double compression region is clearly visible. In a classification at image level, the left example is a false negative case, the right example true positive.

Based on our experimental evaluation, we recommend to use the boosted classifier for JPEG ghost detection. It yields very good detection results, even on block sizes of 8×8 pixels. Thus, when classifying individual windows, this method preserves in many cases a high level of detail in the marked blocks (for example, see Fig. 3.4. At image level, it performs still very reliably. The relation between specificity and sensitivity on image level can be adjusted with the detection threshold.

Results on larger quality differences like $\delta = 20$ were very reliable. However, this is not a hard limit, as our experiments suggest that δ can be decreased to an empirical minimum of 5 points. In such cases, other methods often exhibit difficulties in correctly pointing out the location of double-compression, e. g. [Fari 09] and [Lin 09b] reported a highly increased error rate for $\delta < 20$ and $\delta < 10$, respectively. In practice, we assume that the proposed method yields the biggest advantages in scenarios where $\delta < 20$. Here, automated per-block classification can create relatively fine-grained markings. In adversial situation, like low-contrast regions, or low values of δ, a human observer can directly examine these markings by looking at a single image which are similar to Fig. 3.4. We believe that this is much more feasible than browsing dozens of low-contrast black-and-white difference images in the style of Fig. 3.1. Alternatively, the experiments suggest that in a fully automated pipeline, the fully automated assessment of the images can also well be used.

Chapter 4

Illumination Color Estimation

Color is widely used in computer vision, but in a very basic, primitive way. One reason for employing very basic color primities is that the color information of a pixel is always a mixture of illumination, geometry and object material. Consider, for example, changes in illumination, which are not unlikely: the spectrum of sunlight changes over the daytime, shadows can fall on the object, or artificial light is switched on. Fig. 4.1 shows two examples for different color appearances. The pictures are part of the dataset by Barnard *et al.* [Barn 02b]. The scene is once exposed to relatively neutral (white) light, and once to illuminants that approximate the environment light at night. Thus, for robustness, methodologies that employ color should explicitly address such appearance variations.

Figure 4.1: Example of the influence of illumination on the perceived object color. The same scene is shown, once exposed to white illumination, once exposed to illuminants that approximate illumination at night. The pictures are part of the dataset by Barnard *et al.* [Barn 02b]).

Human vision can adapt to changing illumination in many real-world situations (see, e. g., [Land 71, Funt 04]). This observation motivated the investigation of computational methods for the neutralization of illumination in machine vision. Thus, in computer vision, the term *color constancy* subsumes techniques that either aim at determining the color of the scene illumination, or at producing an illumination-invariant, colored representation of the scene. Both problems are commonly considered equivalent[1].

[1]Under the simplifying, but highly common von Kries hypothesis, both problems are indeed equivalent. See Sec. 4.1 for details.

In 1998, Funt *et al.* [Funt 98] asked "Is Colour Constancy Good Enough?". Their evaluation showed failure cases of existing color constancy algorithms. In principle, the same question can be asked also today. The reason for this long-term struggle with color constancy lies in the fact that an analytical solution is severely underdetermined: as the perceived color of every pixel is a mixture of illumination and material colors, a general solution has to assign values to two unknowns per known value. Thus, current research focuses on the search for reasonable constraints or assumptions to make color constancy algorithms practical for the application on real-world images. In Sec. 4.1, we introduce some underlying theory for color and reflectance that is required for better comprehension of the color constancy methods.

Related work is briefly described in Sec. 4.2. One commonly used assumption is that a scene is typically exposed to a single dominant illuminant. However, in many scenes, one can observe more than one illuminant. Thus, it is not clear whether the current model of uniform illumination is intrinsically too limited. In Sec. 4.4, we show some examples for typical scenes under non-uniform illumination. In such cases, perfect color balancing can only be achieved if the assumption of uniform illumination is relaxed. We present a number of contributions towards multi-illuminant color constancy.

First, in order to be able to benchmark color constancy methods on multi-illuminant scenes, we investigate two approaches to create ground truth data for such setups. For both approaches, we created multi-illuminant datasets and a new computational method to obtain pixelwise ground truth. The details are presented in Sec. 4.3.

We also investigated a number of approaches for multi-illuminant color constancy in Sec. 4.4. We first examined the possibility of narrowing the spatial support of single-illuminant estimators. Although this approach is in principle feasible, its simplicitly makes it difficult to incorporate further cues on the scene composition, if they are available. As a consequence, we finally propose a novel energy minimization-based color constancy algorithm that performs highly competitively on scenes containing multiple, differently colored illuminants. The first dataset in Sec. 4.3 and the implementation and evaluation in Sec. 4.4.1 were done by Michael Bleier during his project work under my supervision. The pictures in the second dataset were captured by Shida Beigpour and Joost van de Weijer. The ground truth computation has been done by me. The algorithm in Sec. 4.4.2 is joint work with Elli Angelopoulou. Finally, the energy-minimization method in Sec. 4.4.3 was joint work with Shida Beigpour, Joost van de Weijer and Elli Angelopoulou. The code was mostly written by Shida and me, important ideas were contributed by Joost and Elli.

4.1 Basics of Color Analysis

Color is typically denoted as a a vector of three components, consisting of the red, green and blue intensities that can typically be observed with an off-the-shelf digital color camera. Very often, it is preferred to use a brightness-normalized representation of color, called *chromaticity*, which, for a given color $\boldsymbol{p}$, can be defined as

$$\boldsymbol{\chi}(\boldsymbol{p}) = \frac{1}{p_R + p_G + p_B} \cdot \begin{pmatrix} p_R \\ p_G \\ p_B \end{pmatrix} \quad , \tag{4.1}$$

where p_R, p_G and p_B denote the red, green and blue components of $\boldsymbol{p}$. To access the red, green or blue component of $\boldsymbol{\chi}(\boldsymbol{p})$, we define

$$\boldsymbol{\chi}(\boldsymbol{p}) = \begin{pmatrix} \chi_R(\boldsymbol{p}) \\ \chi_G(\boldsymbol{p}) \\ \chi_B(\boldsymbol{p}) \end{pmatrix} \quad . \tag{4.2}$$

Note that this brightness normalization removes one dimension from the data, because all channels now sum up to 1. Some authors used an alternative definition of chromaticity where they replace the sum in the denominator of Eqn. 4.1 by the Euclidean norm. In this work, we use the definition of chromaticity as stated in Eqn. 4.1 unless stated otherwise.

The most widely used assumption is that the light reaching the camera is due to light that has been reflected from a surface. Thus, several reflectance models have been proposed to describe the formation of color images. We present the two most popular choices, Lambertian reflectance and dichromatic reflectance, in Sec. 4.1.1. The von Kries model, i. e. the most common choice for the transformation from an input pixel color and an input illumination color to a neutral representation is presented in Sec. 4.1.2. Lastly, in Sec. 4.1.3, we present criteria for evaluating the accuracy of automatically extracted illuminant color estimates.

4.1.1 Lambertian and Dichromatic Reflectance

The color response in a pixel can be modeled as the joint influence of the illumination color, direction and intensity, the color and geometry of a reflecting surface and the camera function. We discuss these points in greater detail below, and state for the moment that the commonly used reflection models are mainly influenced by the choice of the surface reflectance. While a physically accurate description of surface reflectance is difficult, reflectance can be roughly subdivided into a diffuse and a specular component. Photons under diffuse (body) reflectance penetrate the object surface (interface) and adopt the color of the chromophores in the object. Photons under specular (surface) reflectance are reflected from the surface without entering the material at all. Note that for specular reflectance, the angle of reflection is almost exactly equal to the angle of incidence, mirrored about the surface normal. Body reflectance distributes the reflected light in all directions. Fig. 4.2 illustrates this difference.

In the *Lambertian reflectance model*, all pixels are assumed to result from an ideal diffuse surface [Wysz 82, pp. 273-274], i. e. the reflected light is uniformly distributed in all directions. Under this assumption, the intensity of a pixel p_c in color channel $c \in \{R, G, B\}$ is

$$p_c = \cos(\theta) \int_0^\infty \rho(\lambda) e(\lambda) \check{c}_c(\lambda) d\lambda \ , \tag{4.3}$$

where λ denotes the wavelength of the light, θ the angle between the surface normal and the lighting direction, $\rho(\lambda)$ the surface albedo (i. e. the wavelength-dependent reflectivity of the material), $e(\lambda)$ the wavelength-dependent intensity of the light source, and $\check{c}_c(\lambda)$ the camera color response function for channel c. Note that $\check{c}_c(\lambda)$ is a sensor-dependent function. It describes the transmittance behavior of the red,

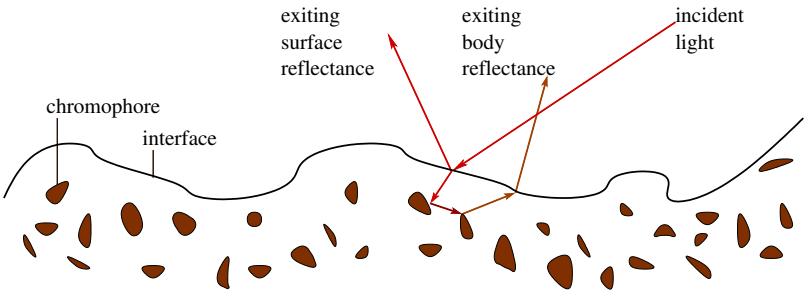


Figure 4.2: Illustration of diffuse and specular reflectance, according to [Lee 86, Klin 90] (see text for details).

green and blue filters of the camera. Particular post-processing like image gamma or Bayer pattern interpolation are not modelled. θ denotes the angle between the light source and the surface normal. The so-called "geometry factor" $\cos\theta$ is independent of wavelength, and can thus be written outside the integral[2]. This model, or variants of it, is most widely used in color constancy research. For reference, see e. g. [Finl 01a, Weij 07a, Rose 03, Finl 96]. Although it does not cover common observations in real-world images like interreflections, specularities or even image gamma, its simplicity makes mathematical reasoning particularly straightforward. This can be seen if the model is further simplified, such that sharpened sensors are assumed. Sensor sharpening has been investigated in a number of works, e. g. by Drew and Finlayson [Drew 00], Barnard *et al.* [Barn 98, Barn 01] and Funt and Jiang [Funt 03]. Here, a transform of the raw sensor data is sought, such that the correlation between the color channels is minimized. Ultimately, the red, green and blue sensor color responses should approximate disjoint impulse functions. Using this assumption, the integral in Eqn. 4.3 can be removed. Then, the Lambertian model for each color channel $c \in \{R, G, B\}$ becomes a multiplication of the components,

$$p_c = \cos(\theta)\rho_c e_c \ . \tag{4.4}$$

Here, p_c, e_c and ρ_c denote the per-channel responses of $\boldsymbol{p}$, $e(\lambda)$ and $\rho(\lambda)$, respectively. We use this model in particular in Sec. 4.3.2 on page 71 of this thesis.

Very few surfaces exhibit pure Lambertian reflectance. Thus, Shafer proposed the *dichromatic reflectance model*[3] [Shaf 85], which describes the reflectance of a surface as a sum of specular and diffuse reflectance. In this model, p_c is expressed as

$$p_c = \int_0^\infty m_d(\boldsymbol{x}) S_d(\lambda) e(\lambda) + m_s(\boldsymbol{x}) S_s(\lambda) e(\lambda) \check{c}_c(\lambda) d\lambda \ , \tag{4.5}$$

where the geometric influences are more generally captured by $m_d(\boldsymbol{x})$ and $m_s(\boldsymbol{x})$ for the diffuse and specular intensity, depending on the position of the light source, the surface normal and the position of the camera. Different models can be used for the functions $m_d(\boldsymbol{x})$ and $m_s(\boldsymbol{x})$. For example, $m_s(\boldsymbol{x})$ could be modelled as Fresnel

[2]Note that one of the assumptions of Lambertian reflectance is the invariance of the perceived intensity to the viewer position. Thus, the geometry factor only includes θ.

[3]Note that Shafer actually used the term "reflection" instead of "reflectance". However, in later work, we got the impression that the term "reflectance" was more widely used.

reflectance. This model has been validated for so-called dielectric materials[4] by Cook and Torrance [Cook 81], but is also commonly used in less constrained environments (like, for instance, on human skin [Blan 03]).

One very common additional assumption is the *Neutral Interface Assumption*, which states that for specular reflectance, $m_s(\boldsymbol{x})$ can be ignored. Then, Eqn. 4.5 simplifies to

$$p_c = \int_0^\infty \left(m_d(\boldsymbol{x}) S_d(\lambda) e(\lambda) + m_s(\boldsymbol{x}) e(\lambda)\right) \check{c}_c(\lambda) d\lambda \ . \tag{4.6}$$

In this case, the color of the illuminant is the same as the color of the specular portion. Finally, one can assume Lambertian reflectance for the diffuse part of Eqn. 4.6, which leads eventually to

$$p_c = \int_0^\infty \left(m_d(\boldsymbol{x}) \rho(\lambda) e(\lambda) + m_s(\boldsymbol{x}) e(\lambda)\right) \check{c}_c(\lambda) d\lambda \ . \tag{4.7}$$

We use this model as a basis for our work in Sec. 4.4.2 and Sec. 5.2.

The dichromatic reflectance model assumes a single illuminant. Extensions of this definition to two illuminants have been proposed by Maxwell *et al.* [Maxw 08] and Riess *et al.* [Ries 09c]. However, experimental evidence of the benefits of such an extension is lacking so far, and it is currently not clear how such an extension can be algorithmically exploited.

4.1.2 The von Kries Model for Color Correction

One of the most popular ways to correct for the color of the illuminant is the von Kries model [Krie 70]. The von Kries model was originally developed to explain human vision. Subsequently, it was also applied to color correction in machine vision. According to the von Kries hypothesis, illumination effects can be per color channel independently adapted to produce a white balanced image. Under certain conditions, this can be approximated by a diagonal transform between a pixel $\boldsymbol{p}$ and an illumination-neutralized vector $\breve{\boldsymbol{p}}$. This is commonly expressed[5] in the form

$$\breve{\boldsymbol{p}} = \begin{pmatrix} e_R^{-1} & 0 & 0 \\ 0 & e_G^{-1} & 0 \\ 0 & 0 & e_B^{-1} \end{pmatrix} \boldsymbol{p} \ , \tag{4.8}$$

i. e., an illumination-neutralized pixel is obtained from dividing the input pixel channel-wise by the color of the illuminant. Note that the diagonal transform is typically only approximately correct [Wysz 82, pp. 431-432]. One prominent reason for the introduced error is that in practice, camera color response functions overlap with respect to the wavelength λ. To overcome this, either a non-diagonal matrix has to be used for correction, or the images have to be preprocessed with sensor sharpening [Drew 00, Barn 01, Drew 09] in order to better fulfill the diagonal model. However, sensor sharpening relies on an often cumbersome camera calibration step for computing a transformation from overlapping camera-sensitivity functions to ones that are

[4] Tominaga lists as some common dielectric materials for instance plastics, paints, ceramics, vinyls, tiles, fruits, leaves and different types of woods [Tomi 91].

[5] For reference, see for instance e. g. [Gijs 11, Barn 02a].

maximally decorrelated [Barn 98, Alva 08]. Thus, most of the color constancy algorithms ignore the non-diagonality of a correct color transform and use the von Kries model as a sufficiently good approximation to the underlying physical model. In this work, we also adopt the simplified von Kries model (see Sec. 4.4.1). Under this assumption, a three-component vector suffices to transform a pixel exposed to colored illumination to a pixel under neutral illumination, as stated in Eqn. 4.8. In summary, *the problem of estimating the color of the illuminant becomes equivalent to correcting the color under the von Kries hypothesis.* Thus, unless expressly stated otherwise in this thesis, we do not distinguish these two concepts.

4.1.3 Evaluation of Color Constancy Methods

To assess the correctness of an RGB-estimate of the illuminant color, different error metrics have been proposed. We briefly restate the most influential variants.

Several authors evaluated the algorithm performance indirectly, in three steps. First, the illumination is estimated on scenes with known objects. Then, the illuminant estimate is used to create an illuminant-invariant representation. Finally, the performance of a color-based object recognition system is evaluated to quantify the effectiveness of the color correction [Swai 91, Funt 98, Schi 00, Schi 96, Ebne 09]. This indirect way of measuring the performance with respect to the usefulness of the result is very interesting from a machine vision viewpoint. In this case, the desired result is not the illuminant estimate itself, but instead a white-balanced image, i. e. the successful application of the illuminant estimate. However, care has to be taken that the obtained performance metric is not biased by parameters of the recognition system.

As a consequence, several other authors proposed distance metrics between the estimated illumination color and a separately obtained ground truth illuminant. The ground truth is typically measured by placing a Macbeth color chart [XRite In 12] within the scene. If the scene is carefully exposed to only one illuminant, the color of the illumination can be obtained from the gray and white areas of the color checker.

As a distance measure between the estimated illuminant color $\tilde{\boldsymbol{e}}$ and the ground truth illuminant color $\boldsymbol{e}$, the Euclidean distance and the angular distance have been used, as it is also recommended by Hordley and Finlayson [Hord 06] in their thorough study on the evaluation of color constancy algorithms. The Euclidean distance is computed from 2D-chromaticities of the estimated and the ground truth illuminant, i. e.

$$\epsilon_{\text{Euclidean}} = \sqrt{\left(\chi_R(\tilde{\boldsymbol{e}}) - \chi_R(\boldsymbol{e})\right)^2 + \left(\chi_G(\tilde{\boldsymbol{e}}) - \chi_G(\boldsymbol{e})\right)^2} \ , \tag{4.9}$$

where $\chi_R(\tilde{\boldsymbol{e}})$ denotes the red chromaticity component of the estimated illuminant $\tilde{\boldsymbol{e}}$, $\chi_R(\boldsymbol{e})$ denotes the red chromaticity component of the ground truth illuminant $\boldsymbol{e}$, and so on.

The angular distance is defined on the 3D-chromaticities as

$$\epsilon_{\text{ang}} = \operatorname{acos}\left(\frac{\boldsymbol{\chi}(\tilde{\boldsymbol{e}})^{\mathrm{T}}\boldsymbol{\chi}(\boldsymbol{e})}{\|\boldsymbol{\chi}(\tilde{\boldsymbol{e}})\|\|\boldsymbol{\chi}(\boldsymbol{e})\|}\right) \ , \tag{4.10}$$

where the chromaticities are defined as in the previous equation, and the norm denotes the length of a vector.

Figure 4.3: Example scenes from the dataset by Barnard *et al.* [Barn 02c].

To summarize the performance of an illuminant estimator over a number of images, Barnard *et al.* [Barn 02a, Barn 02b] used the root mean square error (RMSE) on $d_{\text{Euclidean}}$ in their in-depth evaluation of color constancy methods. Hordley and Finlayson examined the distribution of $d_{\text{Euclidean}}$ and d_{Angular} [Hord 06]. It turned out that both distributions are neither Gaussian nor symmetric, which makes RMSE a bad choice to summarize the algorithm performance. Instead, the authors propose a number of metrics to characterize the distribution of the errors. One suggestion is the currently most commonly used median of d_{Angular}. The authors note that the angular error should be accompanied by additional metrics. In many publications, the maximum error and the mean of d_{Angular} are typically reported. Note that the mean is according to [Hord 06] not an optimal choice. Nevertheless, due to its popularity, we also use it in our analysis. Thus, in this work, we typically report the median and mean angular error.

4.2 Related Work

We briefly review the main directions of research on single-illuminant color constancy. We start with a section on overview papers, and present then a number of related methods in detail. Note, however, that this presentation is necessarily a selection from the large body of existing methodologies.

Surveys and datasets In 2002, Barnard *et al.* [Barn 02a, Barn 02b] conducted an extensive survey on color constancy algorithms. The accompanying dataset is still be considered an important benchmark for single-illuminant methods. Example images from the dataset are shown in Fig. 4.3. In 2011, Gijsenij *et al.* [Gijs 11] complemented this evaluation by a broader evaluation on more recent methods. According to this survey, the (at that time) strongest algorithms operate on a classifier selection based on semantic image information. As a basis for the evaluation, Gijsenij *et al.* use the grayball dataset by Ciurea and Funt [Ciur 03] and the color-checker dataset by Gehler *et al.* [Gehl 08]. Example images are shown in Fig. 4.4 and Fig. 4.5, respectively.

Virtually all publications in the last ten years benchmark their respective approach using at least one of these three datasets. The dataset by Barnard *et al.* is a very carefully captured set of 51 scenes under laboratory conditions using 11 illuminants with a resolution of 637×468 pixels. Typically, for evaluation, 321 images are selected, stemming from 31 scenes that exhibit diffuse and dichromatic reflectance. Ground truth was obtained by measuring the color of the illuminants separately with

Figure 4.4: Example scenes from the grayball dataset by Ciurea and Funt [Ciur 03].

Figure 4.5: Example scenes from the colorchecker dataset by Gehler *et al.* [Gehl 08].

a Macbeth color chart. The dataset by Ciurea and Funt pursuits a different goal. The authors aimed at creating real-world ground truth data. A gray ball was mounted in front of a camera to obtain the color of the illuminant. The dataset consists of 11346 white-balanced, JPEG-compressed images from 15 video clips with a resolution of 360×240 pixels, captured with a Sony VS-2000 camera. For an unbiased comparison, the image regions containing the gray sphere have to be masked out when benchmarking illuminant estimation methods. Although the authors were criticized for the poor quality of the images, the dataset was widely adopted for two reasons. First, capturing real-world ground truth data is a challenge on its own, and second this was for several years the only publicly available, large real-world dataset. Gehler *et al.* proposed a high-resolution dataset in 2008, consisting of 568 images with a resolution between 1359×2041 and 2193×1460 pixels. To obtain ground truth, a color checker is placed within the scene. This color checker must be masked out when evaluating on the dataset. Shi and Funt [Shi 11] noted that the original ground truth was computed under image gamma, which is why they offer a recomputed ground truth from the linear versions of these pictures to download from their web page. Since its publication, it is often used as a replacement for the grayball dataset.

Few datasets have been proposed that can be used for cases where scenes contain non-uniform illumination. Foster *et al.* [Fost 04] captured static outdoor scenes using a multispectral camera. The illuminant spectrum has been measured for part of the scene with a gray reflectance target. RGB images can be obtained from these scenes using a known camera response function, as described for, e. g., the work by Barnard *et al.* [Barn 02c] (see Fig. 4.6 (left) for two example images). To use this dataset for the evaluation of multiple illuminants, Gijsenij *et al.* [Gijs 12b] embedded two illuminants per scene, following a three-step approach: first, the measured illuminant spectrum is removed from the scene, using the diagonal model within the

Figure 4.6: Example images from the multi-illuminant datasets by Foster *et al.* [Fost 04] (left), Gijsenij *et al.* [Gijs 12b] (middle) and Ebner [Ebne 09] (right).

von Kries hypothesis[6]. Then, the scene is split in two halfs, and both sides are multiplied with different ground truth illuminant spectra. Finally, the spectral image is converted to an RGB image. Note that Foster *et al.* [Fost 04] already point out, that the single illuminant color that is measured by the gray reflectance target is in some cases a poor approximation to the actual, inhomogeneous scene illumination. However, lacking better alternatives, this approach can provide semi-natural multi-illuminant ground truth.

Gijsenij *et al.* [Gijs 12b] created a laboratory dataset, where one illuminant was located on the left and one illuminant was located at the right of a centrally placed object in front of gray background (see Fig. 4.6 (middle)). To determine the ground truth, the gray background was taken as-is[7], while the illuminant colors on the foreground object have been manually annotated. Upon manual investigation, however, it turned out that this process is very error-prone. Thus, we consider this approach for creating multi-illuminant ground truth as too unreliable to be used in practice. Other works, like e. g. by Ebner [Ebne 09] (see Fig. 4.6 (right)), either evaluate on very few images or completely lack quantitative evaluation. Multi-illuminant algorithms will be discussed in Sec. 4.4 in greater detail.

To overcome these limitations, we investigated two alternate approaches to create ground truth information. In Sec. 4.3.1, we assume Lambertian reflectance and negligible influence of interreflections. The ground truth is measured from a repainted version of the scene. In Sec. 4.3.2, we use the same assumptions, but in an improved, more robust and non-intrusive setup using multiple images that are exposed to different illuminants.

[6]Assuming that the measured spectrum is a product of illuminant times albedo, removing the illuminant consists of a per-channel division of the measured spectrum by the measured illuminant.

[7]Assuming that the background albedo is neutral, the reflected color of the background corresponds to the mixture of the illuminants.

Generalized Gray World Buchsbaum [Buch 80] proposed a hypothesis for color constancy stating that the average reflectance of a scene is a known quantity. Any deviation from this quantity is due to colored illumination. This assumption is commonly called *Gray World hypothesis*, as in the absence of particular prior knowledge on the scene, one can still assume that the average of all observed reflectances is neutral (gray).

Van de Weijer *et al.* [Weij 07a] coined the generalized Gray World hypothesis, which puts the Gray World assumption in a larger statistical framework. Let $\boldsymbol{p}(\boldsymbol{x}) = (p_R(\boldsymbol{x}), p_G(\boldsymbol{x}), p_B(\boldsymbol{x}))^{\mathrm{T}}$ denote the color of a pixel at position $\boldsymbol{x}$. Furthermore, let ${}_\sigma\boldsymbol{p}(\boldsymbol{x})$ denote $\boldsymbol{p}(\boldsymbol{x})$ after a Gaussian blur filter with standard deviation σ is applied to the images. van de Weijer *et al.* define the generalized Gray World hypothesis for obtaining an illuminant color estimate $\tilde{\boldsymbol{e}}$ as

$$k\boldsymbol{\chi}(\tilde{\boldsymbol{e}}(\boldsymbol{x}))) = \left(\int \left| \frac{\partial^{n_{\mathrm{GW}}} {}_\sigma\boldsymbol{p}(\boldsymbol{x})}{\partial \boldsymbol{x}^{n_{\mathrm{GW}}}} \right|^{\tau_{\mathrm{GW}}} d\boldsymbol{x} \right)^{\frac{1}{\tau_{\mathrm{GW}}}} . \tag{4.11}$$

Here, τ_{GW}, n_{GW} and σ are parameters to the estimator. The illumination is estimated as a function of the n_{GW}-th derivative of the image, computed per color channel independently. Due to the use of the derivatives, this algorithm is also called *Gray Edge algorithm*. However, several published algorithms are integrated within this framework: for instance, one obtains the well-known white patch method by Land and McCann [Land 71] (also referred to as max-RGB) if $n_{\mathrm{GW}} = 0$ and τ_{GW} is set to infinity. The choice of the parameters is a difficult problem in practice. Thus, in this and the follow-up work, often multiple performance numbers are reported, which include the best parameter settings for a particular dataset.

For properly chosen parameters, the Gray Edge algorithm is still one of the best performing algorithms. As the intuition for the good performance is unclear, Gijsenij *et al.* [Gijs 09, Gijs 12a] conducted a studies to examine the suitability of different edge types for color constancy. The authors concluded that edges that occur due to specularities are better suited for Gray Edge than edges due to surface geometry or shadows, and these again are better suited than edges due to texture changes. Thus, one can speculate that a proper choice of σ and τ_{GW} particularly emphasizes the contribution of relatively bright specularities to the illuminant estimates.

Several follow-up methods have been proposed on the basis of the generalized Gray World method. For instance, van de Weijer *et al.* [Weij 07b] proposed to segment the image in semantic categories. Then, a number of generalized Gray Edge estimates is created as hypotheses. The final illuminant color is determined by a hierarchical probabilistic inference on the labels and the estimates. The performance gain is due to the fact that the semantic labels allow to compensate structural weaknesses of the algorithm. For instance, one can use particularly trained estimators for homogeneous, strongly colored categories like grass. In another semantically motivated work, Lu *et al.* [Lu 09] examined ways to roughly estimate the scene geometry for selecting good Gray Edge parameters. Bianco *et al.* [Bian 08, Bian 10] proposed to improve the parameter selection by classifying images in indoor and outdoor using features for color, texture and edge distribution.

Gamut-Constrained Methods A classic method to estimate the color of the illuminant is the so-called gamut mapping, originally proposed by Forsyth [Fors 90]. The assumption is that the set of observed sensor responses (i. e., pixel colors) forms a convex 3D-shape whose position and scale depends on the color of the illuminant. Thus, the task is to map the gamut of an unknown image to a reference gamut, constructed from images under a known illuminant. The mapping parameters can then be used to color correct the unknown image, or to explicitly output the difference in the unknown illuminant and the illuminant of the reference gamut. To compute the reference gamut, all Gamut Mapping methods depend on training data.

However, given an input image under unknown illumination, the mapping to the canonical gamut is not unique. Forsyth originally proposed to select the mapping that maximizes the volume of the mapped gamut inside of the canonical gamut [Fors 90]. Most of the follow-up literature centers around alternate heuristics to select the best mapping. For instance, Barnard [Barn 00] proposed to relax the diagonal model to select the feasible solution. Finlayson [Finl 96] proposed to conduct Gamut Mapping in chromaticity space to increase the robustness. Finlayson *et al.* [Finl 06] also showed that if the lights that might occur in a scene are known, the accuracy of Gamut Mapping can be greatly improved. Recently, Gijsenij *et al.* [Gijs 10a] successfully demonstrated how to combine Gamut mapping with the generalized Gray World algorithm.

Perceptual Land and McCann [Land 71] proposed the well-known Retinex method to capture the *sensation of lightness*, in contrast to physical reflectance or perceived reflectance. To illustrate this, Funt *et al.* [Funt 04] provided in follow-up work an example of the difference to other methods: consider a cube with white reflectance where one side is lit by the sun, the other is shadowed. A physical viewpoint decomposes the reflectance in albedo and illumination. The perceived reflectance is white, as the human cognition assumes the darker side to be in shadow, and compensates this impression. The color sensation, however, differs. There, the sunny side is (for instance) more yellowish, and the blue side more bluish.

Besides this unusual claim (from a computer vision viewpoint), Retinex gained enormous popularity in different research directions, such as high dynamic range imaging [Meyl 06, Kim 11], intrinsic image decomposition[8] [Gehl 11, Shen 11] and, of course, color constancy itself [Brai 86, Jobs 97, McCa 99, Gijs 11].

Several variants of the algorithm have been proposed, until Funt *et al.* [Funt 04] proposed a standardized implementation, which is currently most commonly used.

Statistical Methods Brainard and Freeman [Brai 97] proposed to learn linearly transformed surface color distributions within a Bayesian framework. A similar, approach called "color by correlation" was proposed by Finlayson *et al.* [Finl 01a]. Also based on the Bayesian theorem, Rosenberg *et al.* [Rose 03] and Gehler *et al.* [Gehl 08] proposed to learn histograms of scene colors for color constancy. Alternatively, Cardei *et al.* [Card 02] proposed to use a neural network classifier to solve the color constancy problem. Recently, Chakrabarti *et al.* [Chak 12] achieved very good results

[8] For more details, see also Sec. 5.3

on several standard datasets using a Maximum Likelihood estimator on image patch statistics.

Similar to the work in gamut mapping, all these methods are strongly dependent on the training data. Thus, within one set of scenes, the performance is often relatively strong. But it is still an open problem to achieve competitive results when training and testing set differ, which is often the case in scenarios with few constraints.

Physics-Based Methods The dichromatic reflectance model by Shafer [Shaf 85] built the foundation for a series of physics-based illumination-related algorithms (see also Eqn. 4.5 on page 60). Most authors use additionally the Neutral Interface Assumption, i. e. the assumption that the color of the specularity corresponds to the color of the light source. Notable examples for algorithms that are based on the dichromatic reflectance model are the segmentation of specularities [Klin 88, Bajc 96, Tan 05], albedo segmentation [Klin 90, Geus 01]. In the field of color constancy, Lee [Lee 86] exploited the fact that pixels containing different mixtures of specular and diffuse reflectance form lines in 2D-chromaticity space that intersect in the color of the illuminant. Finlayson and Schaefer [Finl 01c] proposed to use the Planckian locus as an additional constraint to be able to estimate the color of an illuminant from a single surface patch. Tan *et al.* [Tan 04] considerably extended Lee's idea to the definition of the *inverse-intensity chromaticity space* (IIC). We built parts of our work on the IIC space. Thus, the details of the method by Tan *et al.* are presented in greater detail in Sec. 4.4.2.

The dichromatic model states that for all pixels from a single surface color align on the so-called dichromatic plane in RGB-space. Schaefer *et al.* [Scha 05] presented a hybrid method that used the physics-based dichromatic planes to constrain the statistical color by correlation method. Recently, Toro and Funt [Toro 07] proposed a direct exploitation of the dichromatic planes. In contrast to prior work, their approach does not require a pre-segmentation of surface colors in an image.

Few physics-based methods operate on purely diffuse pixels. One example is the work by Geusebroek *et al.* [Geus 03]. Here, the authors propose to exploit spatial and spectral derivatives in the scene to achieve illumination invariance.

Multiple Illuminants Some illuminant estimation methods are explicitly designed to handle illumination that varies across the scene. The first method was presented by Barnard *et al.* [Barn 97]. Within a relatively complicated algorithm, the method contains additional constraints to Gamut mapping, such that the localization problem is solved by a gamut-constrained segmentation step, and the colors of the illuminants are obtained from a predefined set of common real-world illuminants. Besides the rather hands-on algorithm, the preselected illuminants limit the applicability of the method to well-defined cases. Additionally, the method requires smooth transitions of the illumination. Ebner [Ebne 09] followed a different approach by applying a diffusion operator on the pixel intensities. It is based on the assumption that the content of the image consist of large grayish areas, and the illumination varies very smoothly. Then, the diffusion removes the image content and only the low-pass illumination information remains. Unfortunately, the assumptions of this algorithm are very limiting in practice, and often lead to inaccuracies, especially in colorful

scenes [Hsu 08]. Kawakami *et al.* [Kawa 05] proposed a physics-based method specifically designed to handle illumination variations between shadowed and non-shadowed regions in outdoor scenes. Due to its explicit assumption of hard shadows and skylight/sunlight combination (or even more general Planckian illuminants), this method does not generalize well on arbitrary images. Gijsenij *et al.* [Gijs 12b] recently proposed an algorithm to compensate two light sources in a scene. It applies generalized Gray Edge estimators for single illuminants on a grid segmentation of the image. With the additional assumption of two light sources and sub-grid postprocessing, the method achieves good results on a small benchmark dataset that has been created by the authors. For cases where additionally the chromaticity of the two illuminant is known, Hsu *et al.* [Hsu 08] proposed an algorithm for high quality white-balanced images. Thus, this method only solves the localization problem, but with high accuracy. Unfortunately, the color of the input illuminants can often only be obtained under laboratory conditions, which makes it difficult to apply this algorithm in practice. Thus, none of the existing multi-illuminant estimation methods can handle arbitrary images and as such, none of them has been extensively tested on a large variety of real images.

4.3 Datasets for Multi-Illuminant Recovery

Until two or three years ago, almost all work on color constancy assumed single-illuminant scenes. However, in real-world images, the assumption of a single homogeneous illuminant is typically violated. Thus, an extension to multiple illuminants is a natural question in order to make illuminant color estimation more broadly applicable.

The presence of multiple illuminants makes the color constancy problem considerably more challenging: for single-illuminant methods, it suffices to estimate a three-component vector, i. e. the color of the illuminant. If multiple illuminants are present, it is required to estimate a) the colors of the illuminants, and b) their relative contribution per pixel. The number of unknown illuminant colors increases linearly in the number of illuminants. However, the problem of localizing the region of influence per illuminant is completely new, and adds the illuminant mixture parameters per pixel as another (large) set of unknowns. Additionally, we require a new protocol for quantitative evaluation. As the mixture of the illuminants may change between every pixel, high-resolution (ideally pixelwise) ground truth is required. More precisely, it does not suffice anymore to place a Macbeth color chart in the scene, as it is often done to determine the color of a single illuminant. Instead, more advanced methods for creating ground truth have to be found.

Prior multi-illuminant work is sparse, and mainly proposed to operate on synthetic data. To obtain "true" multi-illuminant data, we investigated two ways of creating such datasets. First, in Sec. 4.3.1, we discuss ground truth that is obtained after recoloring the scene under investigation with diffuse gray paint. In Sec. 4.3.2, we investigate a novel approach to generate ground truth from a set of input images under partially known illuminants.

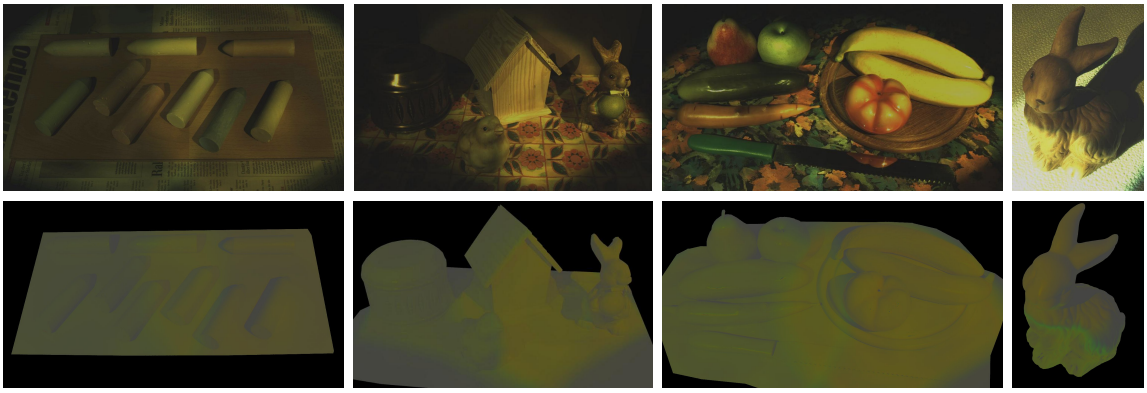


Figure 4.7: Example images from the proposed multi-illuminant dataset using gray paint (non-linear representation for visualization).

4.3.1 Ground Truth from Fixed, Static Scenes

We drew inspiration by a dataset on intrinsic image decomposition, which was recently proposed by Grosse *et al.* [Gros 09]. The goal of this work is to benchmark algorithms that aim to separate albedo from shading. To do so, the authors firmly attached the objects to a base plate. Then, pictures from the object were taken from defined positions. In a second step, the objects were painted white. A second series of pictures was then taken from the same positions. Thus, for every object there exists an image pair, where one image contains shading *and* albedo, and one contains only shading.

We transferred this idea to color constancy. Four scenes (see Fig. 4.7) were taken under 17 different illumination conditions, 9 of which were truly multi-illuminant, for a total of 36 multi-illuminant images. Our scene is mostly composed of diffuse materials. Exceptions are the tin object in the second scene in Fig. 4.7, the knife and some of the fruits in the third scene. The different lighting setups were created by two Reuter lamps [Reuter 12] with LEE color filters [LEE Filt 12]. One Reuter lamp was positioned on the left side of the scene and was combined with the LEE filters 201, 202 and 281. The other Reuter lamp was positioned on the right side and was used with the LEE filters 204, 205 and 285. We also took images with only one filtered light on at a time, both Reuter lamps on without any filters and one under ambient illumination. As ground truth we spray-painted each scene gray and took a series of images under the exact same 17 illumination conditions. We used RAL 7035 and RAL 7047 spray paints[9] which were verified with a Macbeth color checker. The color of the illuminant can now be directly obtained from the gray painted objects. One limitation of this approach is that interreflections are not preserved in the gray painted scenes.

The data was captured with a Canon EOS 550D camera and a Sigma 17-70 lens. The aperture and ISO settings were the same for all the images. The RAW data was converted using `dcraw` [Coff 12] with gamma set to 1.0 and without any white balancing. Different fixed shutter speeds were used across the 17 different illumination

[9]The RAL colors are standardized colors, nowadays maintained by the German organization "RAL Deutsches Institut für Gütesicherung und Kennzeichnung e.V." [RAL gGmb 12].

Figure 4.8: Example scenes containing two illuminants. Pictures courtesy of Trine Juel [Juel 08], Risa Ikeda [Iked 10] and David Domingo [Domi 05].

conditions in order to avoid under- and over-exposure. Note that the collected data, as well as the code for this work can be downloaded from the web[10].

We use this dataset to evaluate parts of our work as presented in Sec. 4.4.1. One limitation of this approach to obtain ground truth is that it does not faithfully preserve the interreflections between the scene objects. Thus, we can recommend this approach only when for capturing clean scenes that containisolated objects.

4.3.2 Ground Truth from Multiple Light Situations

Another limitation of the previously presented approach is the tedious and destructive capturing process. Thus, this method does not scale above well-defined, small laboratory settings (consider, e. g., an outdoor scene that has to be completely spray-painted for ground-truth recovery).

To address this issue, we investigated a second, non-intrusive method that recovers multi-illuminant ground truth from multiple images under different illuminations. It assumes exactly two illuminants, Lambertian reflectance, a linear sensor response, sharpened sensors and no interreflections. Although the restriction to two illuminants might appear very limiting, we note that the largest part of the images from many scenes in the wild can be well approximated with two illuminants. As shown by example in Fig. 4.8, scenes that are mainly influenced by two illuminants are for instance outdoor scenes containing shadow and sunlight, indoor scenes that are partially illuminated through windows, or scenes at night captured with camera flash light. Thus, we assume that the restriction to two illuminants is compensated by the largely increased applicability of the method.

4.3.2.1 Real-world Two-illuminant Datasets

We created two datasets containing two dominant illuminants, one under laboratory conditions, and one containing indoor and outdoor real-world images. The images were captured with a Sigma Foveon X3 camera. This model is capable of capturing per pixel red, green and blue intensities, instead of interpolating image information, e. g. from a Bayer pattern. Image gamma has been deactivated, and camera output has been set to raw 12 bit mode. The exposure time has been fixed, such that specularities are not clipped. One known limitation of this model is that the separation of

[10] `http://www5.cs.fau.de`

the color channels is relatively poor for raw images [Coff 12]. Thus, for the real-world images, we decided to apply the perceptual sharpening by Vázquez i Corral [Vazq 11].

Under laboratory conditions, we selected three illuminants, referred to as "red", "white" and "blue", due to their relative color differences. The illuminants were firmly mounted on a structure, such that one illuminant was located on the left, the other on the right of the scene. We created 11 scenes containing a selection of specular and diffuse objects and varying number of objects. One scene was empty, i. e. only the gray ground plane is shown (see Fig. 4.11 on page 77). 5 scenes contain one or two mugs in different colors. One scene contained a diffusely reflecting stuffed animal. The 4 remaining scenes contain multiple objects, with varying amount of specularities (see the first three images in Fig. 4.10 on page 76 for an example). Note that the high intensity of the red illuminant strongly influences the overall appearance of the scene. For a complete list of the scenes, see Appendix D).

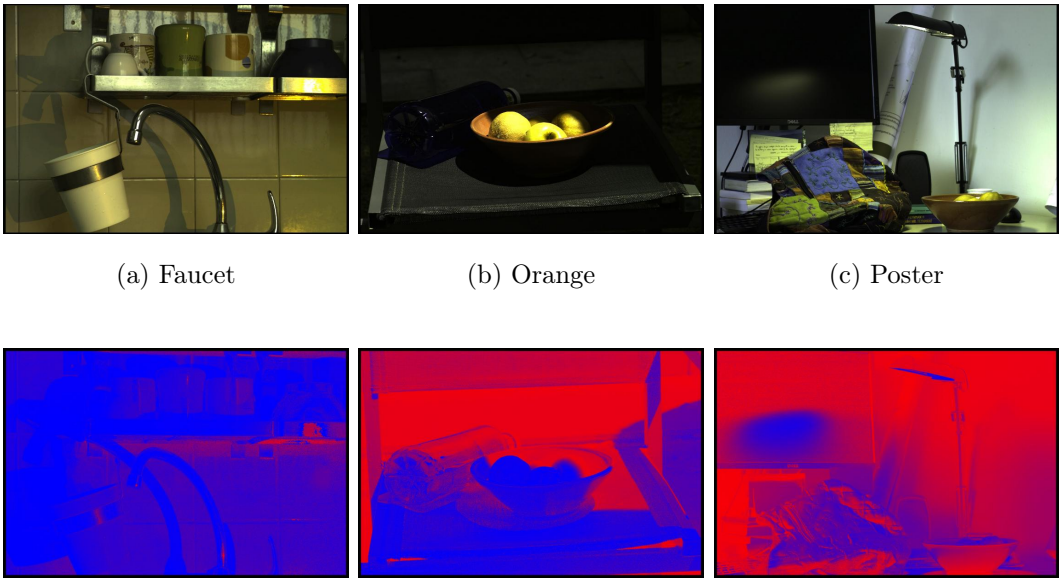
For the creation of the real-world scenes, we selected 20 scenes. 17 of these scenes contain one environmental light, like sun light or incandescent/indoor illumination, and one colored light from a colored projector image. Two images, "dark tools" and "orange", contain sunlight and ambient illumination, and "poster" contains a mixture of incandescent light and sunlight. Sample images are shown in Fig. 4.9. In the top row, three representative scenes are shown. On the left, "faucet" shows a mixture of incandescent light and projector light. In the middle and on the right, "orange" and "poster" are shown. In the bottom row, the relative influence of the two illuminants is shown color-coded in red and blue. The algorithm to compute this relative influence is subject of this section, and explained below.

4.3.2.2 Theoretical Model for Multi-Illuminant Ground Truth Computation

In order to determine the illuminant ground truth for two light sources per pixel, two subproblems have to be solved. First, the color of the illuminants is required. Second, one must determine the (potentially overlapping) spatial distribution how these illuminants influence the scene. We call the second task the *Localization Problem*.

We address the first problem, i. e. the recovery of the ground truth illuminant color, in the next section. In this section, we describe a solution for the localization problem to determine the spatial influence of the illuminants. Thus, we assume for the moment that the chromaticities of the illuminants are known.

General Procedure We determine the relative influence of both illuminants on an image pixel from multiple input images. The main result of this section is to show that, under certain assumptions, it is straightforward to compute the relative influence of two illuminants per pixel if multiple images are available. More precisely, assume that the scene and camera setup is static, and three images are available: one that is exposed to both illuminants, and two images where only one of these two illuminants is switched on. Assume also that the colors of both illuminants are known. We are going to show that for each pixel, the relative influence of the illuminants in the two-illuminant image is equal the relative brightness of the illumination-corrected single-illuminant images.



(a) Faucet (b) Orange (c) Poster

(d) Illumination influence in faucet (e) Illumination influence in orange (f) Poster illumination in poster

Figure 4.9: Example images from the proposed real-world dataset, and the influence areas of the two illuminants in red and blue. Left: mixture of incandescent light and projector light. Middle: mixture of sunlight and shadow. Right: mixture of incandescent light and sunlight.

Proof Without loss of generality, let the scene illumination consist of a bluish illuminant from the left and a more reddish illuminant from the right. For simplicity, we refer to these illuminants as blue and red lights, respectively. We denote this image as $\mathbf{I}^{(\mathrm{B};\mathrm{R})}$. To compute the ground truth, we require two additional images, one that is only exposed to the blue illuminant, and one that is only exposed to the red illuminant. We denote these images as $\mathbf{I}^{(\mathrm{B};\emptyset)}$ and $\mathbf{I}^{(\emptyset;\mathrm{R})}$, respectively. Note that all three images must be taken from the same position, and the scene must be static between capturing these images. If the camera color response function is linear, then

$$\mathbf{I}^{(\mathrm{B};\mathrm{R})} = \mathbf{I}^{(\mathrm{B};\emptyset)} + \mathbf{I}^{(\emptyset;\mathrm{R})} \quad , \tag{4.12}$$

holds, i. e. a direct addition of the single-illuminant images results in the two-illuminant image. Excluding third illuminants and effects from interreflections, the illuminant in a pixel at position $\boldsymbol{x}$ in $\mathbf{I}^{(\mathrm{B};\mathrm{R})}$ must be a linear combination of the illuminants in $\mathbf{I}^{(\mathrm{B};\emptyset)}$ and $\mathbf{I}^{(\emptyset;\mathrm{R})}$ at position $\boldsymbol{x}$. Thus, given the chromaticities $\boldsymbol{\chi}(\boldsymbol{e}^{(\mathrm{B})})$ and $\boldsymbol{\chi}(\boldsymbol{e}^{(\mathrm{R})})$ of the blue and red illuminants, we seek a pixelwise weighting factor w_{GT}, such that

$$\boldsymbol{\chi}(\boldsymbol{e}^{(\mathrm{B};\mathrm{R})}(\boldsymbol{x})) = w_{\mathrm{GT}}\boldsymbol{\chi}(\boldsymbol{e}^{(\mathrm{B})}) + (1 - w_{\mathrm{GT}})\boldsymbol{\chi}(\boldsymbol{e}^{(\mathrm{R})}) \tag{4.13}$$

is the illuminant chromaticity of the pixel at position $\boldsymbol{x}$ in image $\mathbf{I}^{(\mathrm{B};\mathrm{R})}$. We assume the illuminant chromaticity to be constant over the image[11], and thus $\boldsymbol{x}$ is omitted. A function to compute w_{GT} is given in Eqn. 4.19 at the end of the proof.

To properly model non-uniform illumination, we need to define the Lambertian reflectance model with sharpened sensors more precisely. In contrast to Eqn. 4.4 (see Sec. 4.1.1) we incorporate the (potentially) spatially varying intensity of the light source explicitly. Let $p_c^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ be the intensity of $\mathbf{I}^{(\mathrm{B};\emptyset)}$ at pixel $\boldsymbol{x}$ in channel c. $p_c^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ can then be written as

$$p_c^{(\mathrm{B};\emptyset)}(\boldsymbol{x}) = \cos(\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x}))\rho_c(\boldsymbol{x})k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})\chi_c(\boldsymbol{e}^{(\mathrm{B})}) \ . \tag{4.14}$$

Here, $\rho_c(\boldsymbol{x})$ and $\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ denote the albedo in channel c and the angle between light source and surface normal for the blue light in pixel $\boldsymbol{x}$. The intensity of the blue light $\boldsymbol{e}^{(\mathrm{B})}$ may be spatially variant, which is why it depends on $\boldsymbol{x}$. $\chi_c(\boldsymbol{e}^{(\mathrm{B})})$ denotes the chromaticity of $\boldsymbol{e}^{(\mathrm{B})}$ in channel c. Because $\chi_c(\boldsymbol{e}^{(\mathrm{B})})$ is intensity-invariant, an intensity factor $k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ is added. It is defined as

$$k^{(\mathrm{B};\emptyset)}(\boldsymbol{x}) = \frac{e_c^{(\mathrm{B})} e_{b,c}(\boldsymbol{x})}{\chi_c(\boldsymbol{e}^{(\mathrm{B})})} = \frac{e_c^{(\mathrm{B})}}{\frac{e_c^{(\mathrm{B})}}{\sum_{i\in(R,G,B)} e_i^{(\mathrm{B})}}} = \sum_{i\in(R,G,B)} e_i^{(\mathrm{B})} \ . \tag{4.15}$$

Thus, $k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ denotes the spatially varying sum of intensities of the light source over all color channels. As such, $k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ does not depend on the choice of the color channel c. $\mathbf{I}^{(\emptyset;\mathrm{R})}$ is analogously defined to $\mathbf{I}^{(\mathrm{B};\emptyset)}$.

We now outline the procedure of obtaining pixelwise ground truth:

1. As the chromaticities of the blue and red illuminants are assumed to be known, one can transform $\mathbf{I}^{(\mathrm{B};\emptyset)}$ and $\mathbf{I}^{(\emptyset;\mathrm{B})}$ into images under neutral (white) illumination. Using the diagonal version of the von Kries model of illuminant change (see Eqn. 4.8 on page 61), this can be accomplished by dividing every color channel by the respective illuminant intensity. We denote by $\breve{\mathbf{I}}^{(\mathrm{B};\emptyset)}$ and $\breve{\mathbf{I}}^{(\emptyset;\mathrm{R})}$ the illuminant-normalized version of $\mathbf{I}^{(\mathrm{B};\emptyset)}$ and $\mathbf{I}^{(\emptyset;\mathrm{R})}$.

   Note that $\breve{\mathbf{I}}^{(\mathrm{B};\emptyset)}$ and $\breve{\mathbf{I}}^{(\emptyset;\mathrm{R})}$ are not identical, although the scene is assumed to be static when capturing $\mathbf{I}^{(\mathrm{B};\emptyset)}$ and $\mathbf{I}^{(\emptyset;\mathrm{B})}$. This is mainly due to

   (a) the spatially varying light intensities,

   (b) different angles between the light sources and the surface normals, and

   (c) effects due to object and scene geometry, e. g. different shadow geometries.

2. These brightness differences in $\breve{\mathbf{I}}^{(\mathrm{B};\emptyset)}$ and $\breve{\mathbf{I}}^{(\emptyset;\mathrm{R})}$ are used to compute the mixture of the illuminants.

From the outline above, point 1 is a direct application of the von Kries model and the given assumptions. We show it for $\breve{\mathbf{I}}^{(\mathrm{B};\emptyset)}$, $\breve{\mathbf{I}}^{(\emptyset;\mathrm{R})}$ can be treated analogously. $\breve{\mathbf{I}}^{(\mathrm{B};\emptyset)}$ is obtained for each color channel c separately by computing

$$\breve{\mathbf{I}}_c^{(\mathrm{B};\emptyset)} = \frac{\mathbf{I}_c^{(\mathrm{B};\emptyset)}}{\chi_c(\boldsymbol{e}^{(\mathrm{B})})} \quad \text{for } c \in \{R, G, B\} \ . \tag{4.16}$$

[11] Note that this assumption would be invalid if for instance differences in large-scale atmospheric effects could be observed within the image. However, we neglect this case.

Note that this color correction is wrong in occlusion regions, i. e. shadow regions of $\breve{\mathbf{I}}^{(\mathrm{B};\emptyset)}$ However, we found in practice (see below) this error to be negligible.

Let $\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ denote the intensity of $\breve{\mathbf{I}}_c^{(\mathrm{B};\emptyset)}$ in pixel $\boldsymbol{x}$. Then, Eqn. 4.16 and Eqn. 4.14 can be written per pixel as

$$\begin{aligned}\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x}) &= \frac{p_c^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{\chi_c(\boldsymbol{e}^{(\mathrm{B})})} \\ &= \frac{\cos(\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x}))\rho_c(\boldsymbol{x})k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})\chi_c(\boldsymbol{e}^{(\mathrm{B})})}{\chi_c(\boldsymbol{e}^{(\mathrm{B})})} \\ &= \cos(\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x}))\rho_c(\boldsymbol{x})k^{(\mathrm{B};\emptyset)}(\boldsymbol{x}) \ .\end{aligned} \tag{4.17}$$

We proceed to show point 2 from the outline above. In analogy to $\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$, let $\breve{p}^{(\emptyset;\mathrm{R})}(\boldsymbol{x})$ be the intensity in channel c of $\breve{\mathbf{I}}^{(\emptyset;\mathrm{R})}$ at $\boldsymbol{x}$. As both images are aligned, consider the ratio

$$\frac{\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{\breve{p}^{(\emptyset;\mathrm{R})}(\boldsymbol{x})} = \frac{\cos(\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x}))\rho_c(\boldsymbol{x})k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{\cos(\theta^{(\emptyset;\mathrm{R})}(\boldsymbol{x}))\rho_c(\boldsymbol{x})k^{(\emptyset;\mathrm{R})}(\boldsymbol{x})} = \frac{\cos(\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x}))k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{\cos(\theta^{(\emptyset;\mathrm{R})}(\boldsymbol{x}))k^{(\emptyset;\mathrm{R})}(\boldsymbol{x})} \ . \tag{4.18}$$

Equation 4.18 shows that the difference in the ratio of the color channels of $\breve{\mathbf{I}}^{(\mathrm{B};\emptyset)}$ and $\breve{\mathbf{I}}^{(\emptyset;\mathrm{R})}$ comes exactly from the different geometry factors multiplied by the potentially different light source intensities. The weight w_{GT} from Eqn. 4.13 can be directly computed from this relationship. In our implementation, we clipped the weighting if one intensity was more than 40 times larger than the other, which lead to the function

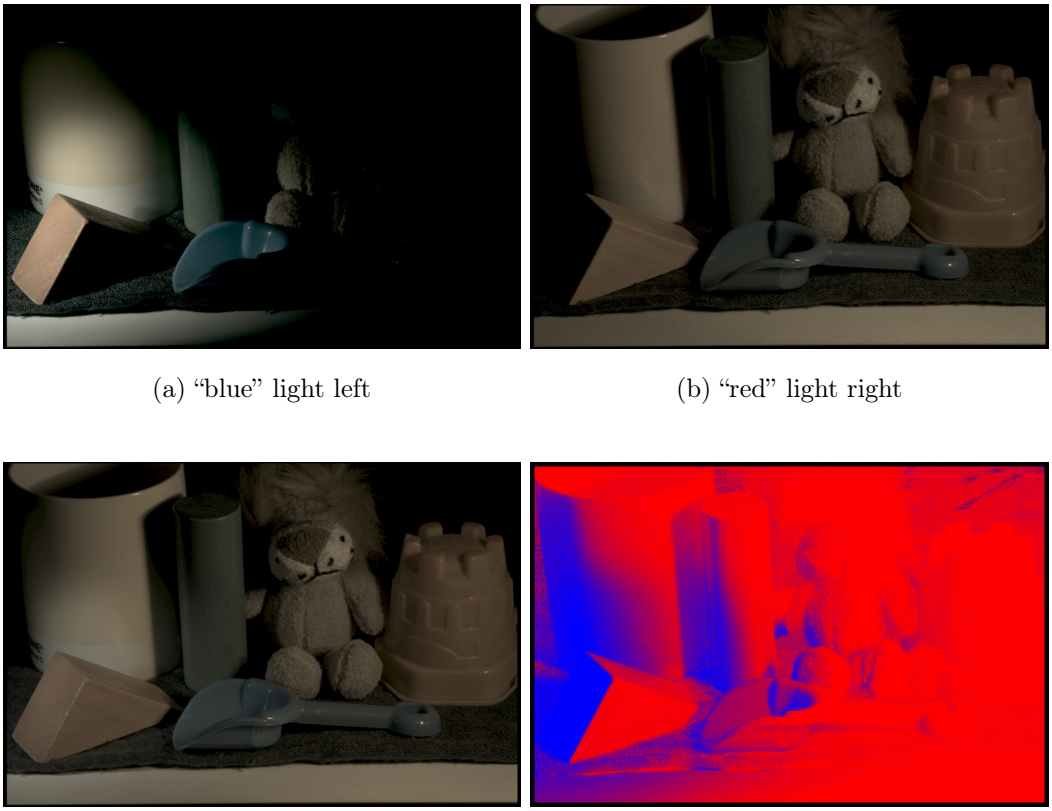
$$w_{\mathrm{GT}} = \begin{cases} 1 & \text{if } \frac{\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{\breve{p}^{(\emptyset;\mathrm{R})}(\boldsymbol{x})} > 40 \\ 0 & \text{if } \frac{\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{\breve{p}^{(\emptyset;\mathrm{R})}(\boldsymbol{x})} < \frac{1}{40} \\ \frac{\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})+\breve{p}^{(\emptyset;\mathrm{R})}(\boldsymbol{x})} & \text{otherwise} \end{cases} \ . \tag{4.19}$$

Here, the fact that w_{GT} is a linear function of $\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ and $\breve{p}^{(\emptyset;\mathrm{R})}(\boldsymbol{x})$ is a direct consequence of the linearity assumption of the camera responses in Eqn. 4.12.

Figure 4.10 illustrates this computation. In the top row, the input images under the blue and red illuminant are shown. In the bottom row, the image under blue and red illumination is shown on the left, and the computed weights are shown on the right. The weighting function has been linearly scaled between saturated red for $w_{\mathrm{GT}} = 0$ to saturated blue for $w_{\mathrm{GT}} = 1$. Several images in our dataset contain specular reflections. Although we use the Lambertian reflectance model for the computation of the influence of the illuminants, we found upon manual inspection that the introduced error can be tolerated. As specularities are typically very bright in relation to the remaining pixels, highly specular intensities are always more than 40 times brighter than the reflected intensity from the other image. Thus, in the ground truth, these pixels are completely assigned to the illuminant where the specularity stems from.

4.3.2.3 Obtaining the Illuminant Color Chromaticities

When deriving the solution to the localization problem in the previous section, the recovery of the illuminant chromaticities has been post-poned. The classical method



(a) "blue" light left

(b) "red" light right

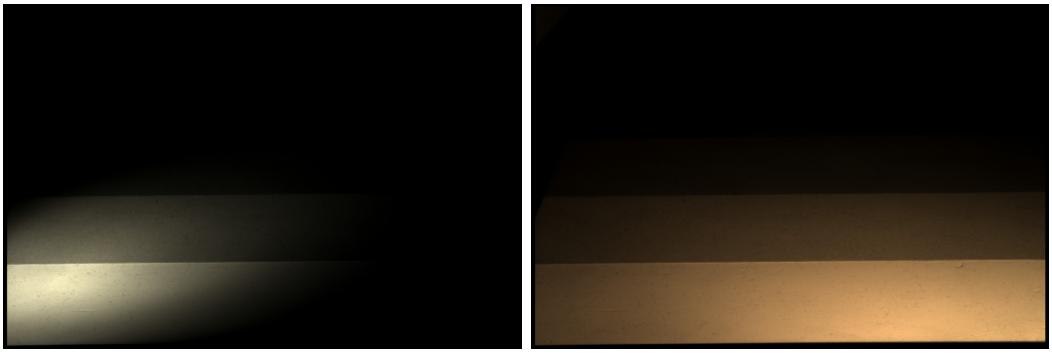
(c) "blue" left, "red" right

(d) Visualization of $w(\boldsymbol{x})$

Figure 4.10: Example images from the proposed multi-illuminant dataset (non-linear representation for visualization).

is to measure the illuminant chromaticities with a Macbeth color chart. Ideally, the chart is positioned in an empty scene perpendicular to the light source, such that the maximum intensity is reflected from this light source. The chromaticity is then computed as an average of the neutral reflectance patches on the Macbeth chart. To minimize noise from the capturing process, typically the brightest, non-clipped patch is selected. This is typically either the white patch (if the camera exposure was carefully set) or a light gray patch.

For some of the images that we have captured for benchmarking, such as the scene shown in Fig. 4.10, information from a Macbeth color chart was not available. Instead, only an empty scene containing a gray floor plate could be used for color picking (see Fig. 4.11), which was not oriented towards the light source. This fact, together with inhomogeneities in the gray surface, lead to considerable differences in the estimation of the ground truth illuminants, depending on which pixels were selected. Table 4.1 shows the chromaticities that were obtained by selecting different groups of pixels from the ground truth patch. In the row "manual selection", we aimed at selecting particularly clean pixels. The last row shows the maximum angular difference d_{Angular} (see Eqn. 4.10 on page 62) between these estimates. While for instance the first and



(a) empty scene, "blue" light left (b) empty scene, "red" light right

Figure 4.11: Example empty scene as only source of ground truth information.

	Illuminant								
Selection strategy	Red			White			Blue		
all pixels	.440	.340	.221	.395	.348	.257	.353	.354	.293
bright pixels	.458	.334	.207	.404	.344	.251	.365	.348	.287
darker pixels	.441	.339	.220	.394	.348	.258	.351	.358	.291
manual selection	.429	.344	.227	.389	.348	.263	.341	.358	.301
Max. angular error	3.41			1.90			2.92		

Table 4.1: Color picking results for the ground truth illuminants.

the third row are in high agreement, the second and fourth row exhibit in the red channel more than 3 degrees difference.

To address the ambiguity in the region selection, we assumed that it is worth to investigate a novel, alternative approach to estimate the illuminant chromaticities. We again exploit the fact that several images are taken from the same scene with fixed camera and light positions. As shown in the previous section, images from the same scene differ per pixel only in the color and intensity of the illuminant. We exploit this fact by setting up two constraints for a least squares solution for estimating the chromaticity of the illuminant:

1. For pixels that are exposed to only one illuminant, the ratio inbetween color channels is fixed. We use the same notation as above. Without loss of generality, we reuse the scenario that there is an image $\mathbf{I}^{(\mathrm{B};\mathrm{R})}$ under blue and red illumination, and separate, aligned images $\mathbf{I}^{(\mathrm{B};\emptyset)}$ and $\mathbf{I}^{(\emptyset;\mathrm{R})}$ that are exposed to only the blue or only the red illuminant. Let $\boldsymbol{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ the intensity of $\mathbf{I}^{(\mathrm{B};\emptyset)}$ at pixel $\boldsymbol{x}$ in channel c, for $c \in \{R, G, B\}$. Using the same assumptions and

notation as in the previous section, the ratio e. g. between the red and green channel for image $\mathbf{I}^{(\mathrm{B};\emptyset)}$ is

$$\begin{aligned}\frac{p_R^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{p_G^{(\mathrm{B};\emptyset)}(\boldsymbol{x})} &= \frac{\cos(\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x}))\rho_R k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})\chi_R(\boldsymbol{e}^{(\mathrm{B})})}{\cos(\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x}))\rho_G k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})\chi_G(\boldsymbol{e}^{(\mathrm{B})})} \\ &= \frac{\rho_R\chi_R(\boldsymbol{e}^{(\mathrm{B})})}{\rho_G\chi_G(\boldsymbol{e}_b(\boldsymbol{x}))\chi_G(\boldsymbol{e}^{(\mathrm{B})})} \ ,\end{aligned} \tag{4.20}$$

where the notation is the same as in the chapter above, i. e. $\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ is the angle between the light source and the surface normal, ρ_R the red channel of the albedo, $k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$ denotes the intensity of the light source and $\chi_R(\boldsymbol{e}^{(\mathrm{B})})$ denotes the red component of the blue illuminant chromaticity. Equation4.20 shows that the ratio between the color channels is the ratio of the chromaticity components of the illuminant, multiplied with the ratio of the albedo components. For a gray scene (such as shown in Fig. 4.11), the ratio of the albedo cancels. Additionally, this formulation does not depend on the geometry factor $\omega_b(\boldsymbol{x})$, i. e. the ratio does not vary with the location $\boldsymbol{x}$ of the pixel. Note that this argument is independent of the choice of channels, i. e. it can be applied to all combinations of color channels.

2. The ratio between two pixels under different illuminants *at the same position* depends on illuminant chromaticity and brightness. This step requires to use an additional input images. To relate e. g. the blue and the red illuminants, both must be located at the same side. Thus, let $p_c^{(\emptyset;\mathrm{B})}(\boldsymbol{x})$ and $p_c^{(\emptyset;\mathrm{R})}(\boldsymbol{x})$ the intensity at pixel $\boldsymbol{x}$ in channel c for another input image resulting from the blue and right illuminant, respectively, mounted on the right side. Then,

$$\begin{aligned}\frac{p_c^{(\emptyset;\mathrm{B})}(\boldsymbol{x})}{p_c^{(\emptyset;\mathrm{R})}(\boldsymbol{x})} &= \frac{\cos(\theta^{(\emptyset;\mathrm{B})}(\boldsymbol{x}))\rho_c k^{(\emptyset;\mathrm{B})}(\boldsymbol{x})\chi_c(\boldsymbol{e}^{(\mathrm{B})})}{\cos(\theta^{(\emptyset;\mathrm{R})}(\boldsymbol{x}))\rho_c k^{(\emptyset;\mathrm{R})}(\boldsymbol{x})\chi_c(\boldsymbol{e}^{(\mathrm{R})})} \\ &= \frac{\cos(\theta^{(\emptyset;\mathrm{B})}(\boldsymbol{x}))k^{(\emptyset;\mathrm{B})}(\boldsymbol{x})\chi_c(\boldsymbol{e}^{(\mathrm{B})})}{\cos(\theta^{(\emptyset;\mathrm{R})}(\boldsymbol{x}))k^{(\emptyset;\mathrm{R})}(\boldsymbol{x})\chi_c(\boldsymbol{e}^{(\mathrm{R})})} \ ,\end{aligned} \tag{4.21}$$

where due to the collocation of the red and blue illuminant, the geometry factors are equal, i. e.

$$\cos(\theta^{(\emptyset;\mathrm{B})}(\boldsymbol{x})) = \cos(\theta^{(\emptyset;\mathrm{R})}(\boldsymbol{x})) \ . \tag{4.22}$$

Thus, the geometry term cancels. Computing the chromaticities over the color channels, the intensity terms $k_b(\boldsymbol{x})$ and $k_r(\boldsymbol{x})$ can be cancelled as well. Thus, the ratio of the chromaticities of the pixels corresponds to the ratio of the illuminant chromaticities:

$$\frac{\chi_c(\boldsymbol{p}^{(\emptyset;\mathrm{B})}(\boldsymbol{x}))}{\chi_c(\boldsymbol{p}^{(\emptyset;\mathrm{R})}(\boldsymbol{x}))} = \frac{\chi_c(\boldsymbol{e}^{(\mathrm{B})})}{\chi_c(\boldsymbol{e}^{(\mathrm{R})})} \ . \tag{4.23}$$

The ratios from Eqn. 4.20 and Eqn. 4.23 can be computed for all combinations of illuminants (i. e., red, white and blue) on the left and on the right side of the scene.

	Illuminant								
Strategy	Red			White			Blue		
Least squares solution from the empty scenes	0.446	0.338	0.217	0.403	0.345	0.251	0.364	0.351	0.286

Table 4.2: Least squares results for the ground truth illuminants.

Entering these ratios in a linear system of equations yields the illuminant colors. For instance, Equation 4.20 translates to the condition

$$\chi_R(\boldsymbol{e}^{(\mathrm{B})}) - \chi_G(\boldsymbol{e}^{(\mathrm{B})}) \frac{p_R^{(\mathrm{B};\emptyset)}(\boldsymbol{x})}{p_G^{(\mathrm{B};\emptyset)}(\boldsymbol{x})} = 0 \ , \tag{4.24}$$

and analogously Equation 4.23

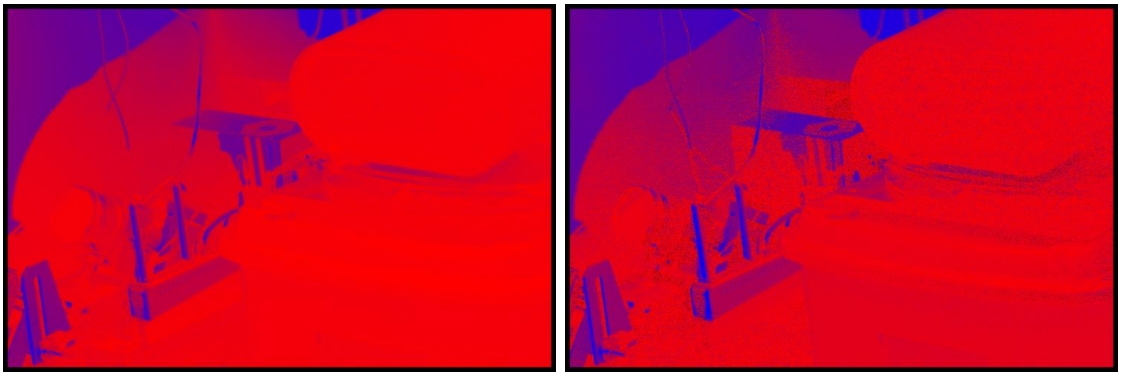
$$\chi_c(\boldsymbol{e}^{(\mathrm{B})}) - \chi_c(\boldsymbol{e}^{(\mathrm{R})}) \frac{\chi_c(\boldsymbol{p}^{(\emptyset;\mathrm{B})}(\boldsymbol{x}))}{\chi_c(\boldsymbol{p}^{(\emptyset;\mathrm{R})}(\boldsymbol{x}))} = 0 \ . \tag{4.25}$$

Setting these equations up for each color channel, each of the sought illuminants and for both, the left and right side, the unknown ground truth illuminants are fully specified up to a multiplicative factor. To avoid the trivial solution where all variables are set to 0, one variable can be forced to 1. This does not lead to any problems, as the final result is anyways rescaled to an overall sum of 1. In our implementation, we solved these equations with a least squares optimizer. Table 4.2 shows the obtained results for our laboratory data. One advantage of the method over naive color picking are the suppression of geometric effects and brightness differences. On the downside, this approach requires a complete set of images showing all combinations of illuminant positions. Thus, we believe that this approach should only be applied if for some reason direct measurements from a Macbeth color chart are not available or can not be used.

4.3.2.4 Full Algorithm for Ground Truth Computation

In a separate experiment, we verified that the linearity assumption of the images (see Eqn. 4.12) is approximately correct on the raw output of the sensor. Thus, the raw images have been used to create the laboratory dataset. For the ground truth computation on the laboratory data, we first estimate the colors of the illuminants using the least squares solution. To do so, we require a full set of single-illuminant input images, i. e., images where either the left or the right illuminant is activated on changing positions. Then, we estimate the distribution of the illuminants for every two-illuminant image. As input, we use the two single-illuminant images that add up to the two-illuminant image (as described by Eqn. 4.12). The overlapping color channels of the Sigma Foveon X3 sensor grossly violate the assumption of sharp sensors. Thus, to determine the distribution of illuminants, we only use the green channel[12]. The presented algorithm does not cover regions that are not directly

[12]Using the green channel instead of a combination of color channels sometimes improves the robustness, if the camera does not fully satisfy the theoretic assumptions (for another application, see e. g. [John 07a])



(a) Weights on linear input (b) Weights on visually enhanced input

Figure 4.12: Example for the error in the ground truth computation, if the input images underwent a non-trivial non-linear transformation before ground truth computation.

illuminated by any of the two illuminants. We visually verified that such regions occur rarely in our scenes. Additionally, such regions are typically very dark, such that the majority of these regions are excluded from processing, due to high image noise. Specularities are also not modelled within the algorithm. However, in our data, all specular regions are correctly assigned to their respective illuminant. Thus, we assume that the presented method is — although physically only approximately correct — sufficiently accurate for benchmarking color constancy algorithms.

The raw output of the Sigma Foveon X3 sensor looks relatively grayish, due to the overlap of the sensor response functions. To create images with more realistic colors, such as from consumer cameras, the images taken from the real-world scenes were visually enhanced using the method by Vázquez i Corral [Vazq 11]. In total, four images were captured per scene: one containing both illuminants, one containing only one illuminant, and both scenes again containing a Macbeth color chart for estimating the chromaticity of the illuminants. The missing single-illuminant image was obtained by subtracting the single-illuminant image from the mixed-illuminant image. With the hand-picked ground truth from the Macbeth color chart and the multiple input images under different illuminants, we computed the illuminant distribution analogously as for the laboratory data.

4.3.2.5 Inaccuracies in the Outdoor Dataset

Creating the missing single-illuminant image via subtraction (see previous section) is valid as long as the input images are linear. Unfortunately, the color enhancement [Vazq 11] is a non-linear transform. Thus, using this preprocessing step introduces noticeable errors in the ground truth computation. One potential workaround to lower this error might be to first perform the subtraction, and then convert the images, which has for technical reasons not been further pursued. Figure 4.12 illustrates an example of the introduced inaccuracies. The base image is “cameras”, as

(a) Sun and ambient light (b) Part of the image only ambient (c) Difference image

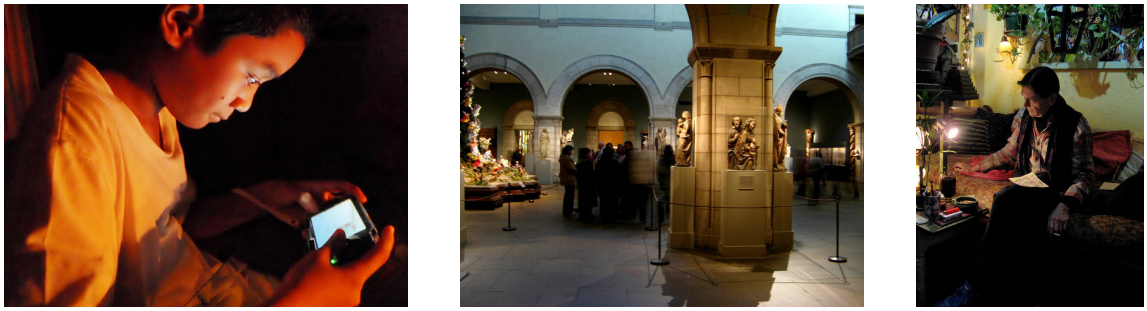
Figure 4.13: Linear version of the "orange" image. Left: sunlight and ambient light on the whole region. Middle: the shadowed area is ambient only, while the remaining image area contains again sunlight and ambient light. Right: subtraction of the left image minus the middle image. This should yield pure sunlight — however, the joint sunlit and ambiently lit area is canceled.

shown in Fig. D.3c on page 188. On the left, the ground truth weights are shown as obtained from linear input, on the right from visually enhanced input. Note the largely increased noise on the right, in areas that are smooth when linear data is used. As a consequence, we originally captured more than 20 images and removed the cases where the noise level was considered inacceptable.

Another inaccurate case are the two sun light/shadow images, "dark tools" and "orange". In these cases, one input image was fully exposed to the sun (see Fig. 4.13a), one was partially shadowed (see Fig. 4.13b). However, for our algorithm, we assume that we can observe two images, in each of which only one of the two illuminants is present. Thus, in the case of outdoor images, we would require one image that is fully shadowed (i. e., contains only ambient light), and one image that is exposed to pure sun light. In this case, correct ground truth could be computed. However, in our case, the illumination that is observed in the sun light image is in reality a mixture of sun light and ambient light. Additionally, no image under full shadow was available.

To obtain a pure sun light image, we would need to subtract the image containing sun light plus ambient light from a pure shadow image, i. e. pure ambient light. For our actual data, this fails in two areas: a) in regions where combined sun/ambient light image is shadowed (i. e. no pure sun light is observed), and b) in regions where the shadow image is still exposed to the sun.

Noting this issue, we aimed to quantify the introduced error by measuring the Macbeth color checker responses on pure shadows, pure sun light, and combined sun/ambient light. For sunny areas, we estimated the error of the illuminant distribution to be about 1°. For areas that are shadowed in the sunny image, the error is larger. Ultimately, we decided to leave these two images in the dataset, for three reasons. First, we thought that realistic outdoor images are a valuable addition to the dataset for research purposes. Second, we considered an error of 1° to be tolerable for the sunny regions, and third, the shadowed areas in the sun/ambient images are relatively small, compared to the total number of pixels. However, for future work,



(a) (b) (c)

Figure 4.14: Examples for multi-illuminant situations. (a) Two dominant light sources, spatially separated (b) Two dominant light sources mix smoothly on the floor (c) Complex illumination situation.

we strongly recommend to capture true shadow images to minimize the error in the ground truth.

4.4 Multi-Illuminant Estimation

Most state-of-the-art color constancy methods were designed for recovering a single, dominant illuminant. In real-world images, this assumption is often violated. Fig. 4.14 shows some examples of multi-illuminant situations. On the left image, the game console acts as a local second dominant light source on the face of the boy. In the middle, two dominant light sources mix smoothly on the floor of the church. On the right, several local light sources create a complex multi-illuminant scene.

It is not straightforward how to incorporate these different effects in a single multi-illuminant CC algorithm. When the influence of an illuminant is spatially limited to a distinct object (like the face in the left image), object segmentation and subsequent single-illuminant estimation may lead to a satisfactory recovery of the illumination colors (see, e. g., the face-specific illuminant estimation by Bianco and Schettini [Bian 12b]). In a different scenario, the church floor in the middle image is also a single "object" from a segmentation viewpoint. However, its surface is illuminated by distinct light sources at different locations. Thus, an object-based illumination extraction is inappropriate in this case. Pixel diffusion-based approaches like the one by Ebner [Ebne 09] can be expected to solve the church floor example, but are expected to fail on object boundaries as in the left image.

How does estimating multiple illuminants differ from estimating one single illuminant? Color constancy for scenes under non-uniform illumination adds a new level of complexity to the algorithms: in addition to the estimation of the chromaticities of the illuminants, one is also required to solve the *localization problem*. Thus, the distribution of the influence of each illuminant in the image must be estimated. We consider this as a segmentation task, although it could also be considered as a blind deconvolution or source separation problem.

In this section, we investigate several directions for multi-illuminant estimation. In Sec. 4.4.1, we report experiments with methods that are direct extensions of single-illuminant estimators. In Sec. 4.4.2, a novel physics-based, but spatially coarse multi-illuminant estimator is presented. Finally, in Sec. 4.4.3, we propose a novel method that operates on a Conditional Random Field (CRF). Experimental results are promising, also in respect to prior work by Gijsenij *et al.* [Gijs 12b].

4.4.1 Color Constancy with Small Spatial Support

Color constancy for multiple illuminants can be considered equivalent to color constancy for uniform illumination with (infinitely) small spatial support. Thus, we investigated whether existing color constancy methods, originally developed assuming uniform illumination, can be applied on smaller image regions. In order to obtain such localized estimates, we examine how the uniform-illuminant assumption of state-of-the-art color constancy methods can be relaxed. We use image sub-regions to compute the illuminant color locally. We then compensate for the loss of accuracy, by combining multiple independently obtained local estimates. Most of the existing fusion strategies try to solve the combination problem by extracting additional features from the image, e. g. color and texture statistics. Based on these features the best algorithm is selected or a weighted average of the estimates is computed. If the best algorithm could always be selected, we could gain in theory significant performance. Recent evaluations show that algorithm combination based on image features does not perform substantially better than the best performing algorithm [Gehl 08]. Computing regression on the estimates has been shown to be more robust than selecting a single estimate [Bian 10]. Thus, we expand this idea to make it applicable to local estimates. An important part of such a comparative analysis is quantitative evaluation. To our knowledge, currently available databases do not provide sufficient information for evaluating multiple illuminant algorithms. Thus, we use our own multi-illuminant ground truth data, as presented in Sec. 4.3.1 on page 70. We then evaluate on this database different color constancy as well as fusion algorithms. We conclude that machine-learning based regression consistently outperformed all other combination strategies, as well as individual estimates.

4.4.1.1 Gamut Mapping and Bayesian Color Constancy

For this study, we used (among other algorithms) also gamut-constrained methods and Bayesian color constancy. Thus, we first add technical details on these methods.

Gamut-Constrained Methods A key component of the gamut constrained methods is the definition of a canonical gamut $\mathcal{G}(\mathcal{O})$, which denotes the convex set of sensor responses $\mathcal{O} = \{{}^1\boldsymbol{o}, \ldots, {}^n\boldsymbol{o}\}$ to n surface reflectances under a canonical illuminant:

$$\mathcal{G}(\mathcal{O}) = \left\{ \sum_i \alpha_i \, {}^i\boldsymbol{o} \;\middle|\; {}^i\boldsymbol{o} \in \mathcal{O},\ \alpha_i \geq 0,\ \sum_i \alpha_i = 1 \right\}, \tag{4.26}$$

where α_i denote scaling parameters. A mapping between the gamut of an unknown illuminant and the canonical gamut is reveals then the illuminant color. Based

on Forsyth's original algorithm [Fors 90], a number of variations have been developed [Finl 96, Finl 00, Finl 06, Gijs 10a].

The most well-known of these methods is *Gamut Mapping*, originally proposed by Finlayson. When transforming an image gamut $\mathcal{G}(\mathbf{I})$ to the canonical gamut $\mathcal{G}(\mathcal{O})$, gamut mapping uses a diagonal matrix transform $\mathbf{T}$. Because only a limited number of surfaces is observed within a single image, the unknown gamut can only be approximated by the observed image gamut. The set of all possible mappings from $\mathcal{G}(\mathbf{I})$ to $\mathcal{G}(\mathcal{O})$ is calculated and the best mapping (with respect to a selection criterion) is selected.

In more detail, let $\mathbf{T}^{\boldsymbol{p},\boldsymbol{o}}$ be a diagonal matrix that maps a point $\boldsymbol{p} = (p_R, p_G, p_B)^{\mathrm{T}}$ in the image gamut to a point $\boldsymbol{o} = (o_R, o_G, o_B)^{\mathrm{T}}$ in the canonical gamut,

$$\forall \boldsymbol{p} \in \mathcal{G}(\mathbf{I}), \;\; \boldsymbol{p}\mathbf{T} \in \mathcal{G}(\boldsymbol{\mathcal{O}}) \;\; , \tag{4.27}$$

where $\mathbf{T}^{\boldsymbol{p},\boldsymbol{o}}\boldsymbol{p} = \boldsymbol{o}$. The set of possible mappings for one point $\boldsymbol{p}$ in the image gamut can be calculated as

$$\mathcal{M}(\boldsymbol{p}) = \left\{ \boldsymbol{a}^{\boldsymbol{p},\boldsymbol{o}} \;\middle|\; \boldsymbol{a}^{\boldsymbol{p},\boldsymbol{o}} = \left(\frac{o_R}{p_R}, \frac{o_G}{p_G}, \frac{o_B}{p_B}\right)^{\mathrm{T}}, \;\; \boldsymbol{o} \in \mathcal{G}(\mathcal{O}) \right\} \;\; . \tag{4.28}$$

These sets are likewise convex. The feasible set $\check{\mathcal{M}}$ can be calculated by intersecting all elements of $\mathcal{M}(\boldsymbol{p})$ for each point $\boldsymbol{p} \in \mathcal{G}(\mathbf{I})$ in the image gamut, i. e.,

$$\check{\mathcal{M}} = \bigcap_{\boldsymbol{p} \in \boldsymbol{\Gamma}(\mathbf{I})} \mathcal{M}(\boldsymbol{p}) \;\; . \tag{4.29}$$

Their intersection is also a convex set. Each map in $\mathcal{M}$ corresponds to a possible illuminant. For the final decision, Forsyth proposed to choose the diagonal matrix transform with the maximum trace from the feasible set, which results in the most colorful gamut.

Finlayson and Hordley [Finl 96, Finl 00] showed that Gamut Mapping can be also performed in a 2D chromaticity space, as only intensity information is lost. We refer to this method as **2D-Gamut Mapping**.

The three-dimensional vector of sensor responses $\boldsymbol{p} = (p_R, p_G, p_B)^{\mathrm{T}}$ is projected onto the plane at $p_B = 1$, yielding 2-D chromaticities in the red and green channel. For 2D-Gamut Mapping, the diagonal transform has only two parameters. Additionally, the feasible set is constrained by the set of possible illuminants. Using Monte Carlo estimation, the explicit computation of the intersection between illuminants and the feasible set can be avoided. Rather, random points in 3D-sensor space are generated. The illuminant is the mean or median of a set of randomly chosen points lying within the feasible set.

Bayesian Color Constancy Bayesian color constancy generates a probabilistic model for surface reflectances and illuminants. Assuming statistical independence of illuminants and surfaces, Bayes' rule is used to decide for the illuminant $\boldsymbol{e}$ according to a loss function $g_{\mathrm{B}}(\hat{\boldsymbol{e}}, \boldsymbol{e}')$ [Brai 97, Rose 03].

Let the probability of coocurrence of illuminants and surface reflectances be known. The illuminant $\boldsymbol{e}$ that minimizes the average loss is

$$\boldsymbol{e} = \underset{\boldsymbol{e}'}{\operatorname{argmin}} \sum_{\boldsymbol{e}'} g_{\mathrm{B}}(\hat{\boldsymbol{e}}, \boldsymbol{e}') P(\boldsymbol{e}'|\boldsymbol{I}) \ , \tag{4.30}$$

where $\boldsymbol{I}$ is the observed image, and the loss function g_{B} is the Euclidean distance between $\hat{\boldsymbol{e}}$ and $\boldsymbol{e}'$. Using Bayes' rule, set

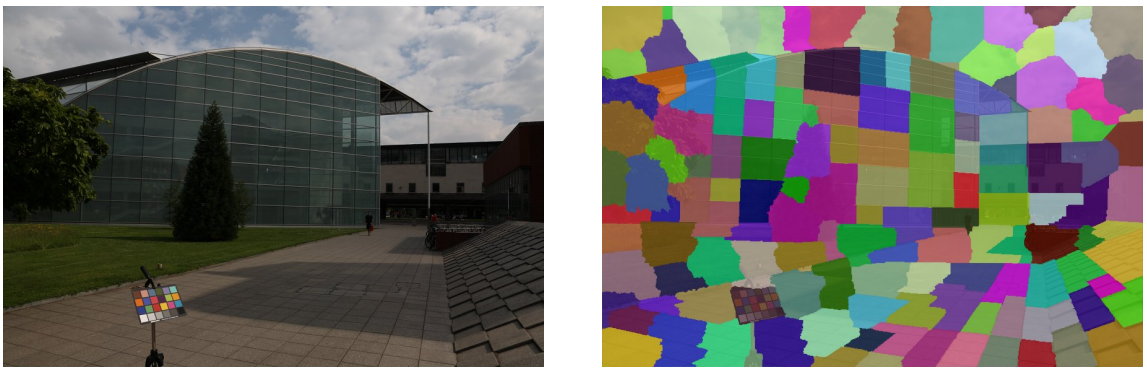
$$P(\boldsymbol{e}'|\boldsymbol{I}) = \frac{P(\boldsymbol{I}|\boldsymbol{e}')P(\boldsymbol{e}')}{P(\boldsymbol{I})} = k_{\mathrm{B}} \ P(\boldsymbol{I}|\boldsymbol{e}')P(\boldsymbol{e}') \ , \tag{4.31}$$

where $P(\boldsymbol{I})$ has a uniform prior density and k_{B} is a constant over the variables of interest.

Rosenberg *et al.* [Rose 03] proposed to model the likelihood $P(\boldsymbol{I}|\boldsymbol{e}')$ using reflectances. The illuminant prior $P(\boldsymbol{e}')$ can be estimated from training data, or assumed to be equally distributed.

4.4.1.2 From Uniform to Non-Uniform Illumination

We segment the image in a set of superpixels, i. e., small image sub-regions such that all pixels in a single superpixel satisfy the same property, in our case color value. A collection of color constancy algorithms is then applied on each superpixel independently. The per-superpixel output of the algorithms is fused afterwards. For the superpixel segmentation, we used the algorithm by Veksler *et al.* [Veks 10, Boyk 01, Kolm 04, Boyk 04], though other segmentation methods can also be employed. It segments the image into non-overlapping, compact superpixels based on their RGB values (see, for example, Fig. 4.15).



(a) (b)

Figure 4.15: Example superpixel segmentation on an image from the Gehler database with the method by Veksler *et al.*. Left: original image, right: the segmentation typically preserves object boundaries.

The underlying assumption is, that the illumination is approximately locally constant on a single superpixel. We then apply state-of-the-art color constancy algorithms on a per-superpixel basis. The superpixel segmentation can address, without

any fine-tuning, a large range of multiple scenarios. Since the superpixels follow object boundaries, our method can handle object-specific illuminants as in Fig. 4.14a. Because superpixels are small, a large object that is illuminated differently at distinct locations (like the church floor in Fig. 4.14b) is subdivided and its subregions are separately processed.

However, when applying the algorithms locally, a trade-off between spatial resolution and color constancy performance has to be made. For instance, gamut mapping and Bayesian color constancy draw their accuracy from extensive statistics over the range of colors in the image. A superpixel offers only a limited selection of the observable colors. Hence, a performance drop for statistics-based methods is expected. Equivalently, zero-order gray world is clearly affected from the fact that superpixels typically contain pixels of similar colors. In order to partially alleviate these problems, we constrained the possible illuminants to the convex set of illuminants in our training data.

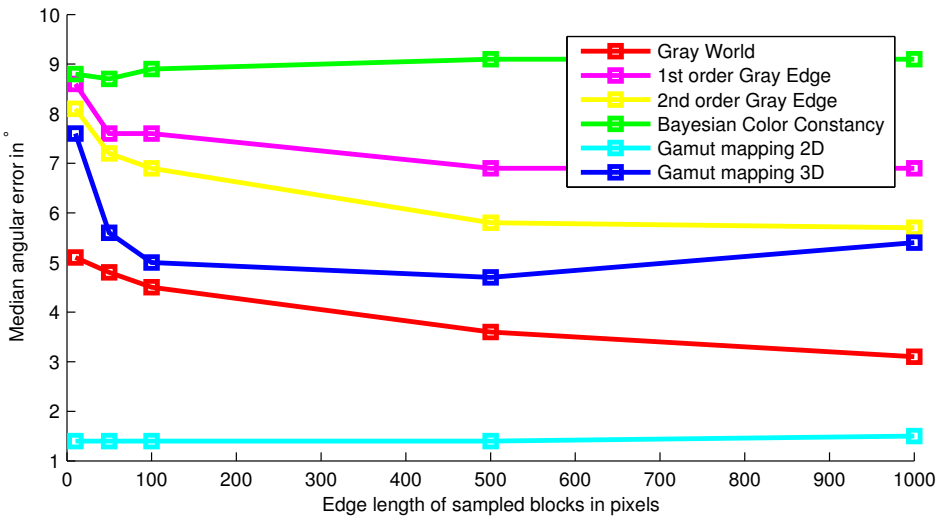


Figure 4.16: Error rates on rectangular subregions of different sizes.

As part of our analysis, we evaluated how the size of local subregions can affect the performance of color constancy algorithms. We selected 80 images from the reprocessed Gehler database [Gehl 08, Shi 11] with approximately uniform illumination. We excluded the parts of the image containing the Macbeth chart. Rectangular regions of different sizes were selected, and Bayesian color constancy and several instances of the generalized Gray World and Gamut Mapping were applied on it. We use the median angular error on these regions (see Eqn. 4.10 on page 62) as a performance measure. As shown in Fig. 4.16, blocks of size less than 100×100 pixels typically exhibit increased error rates. One notable exception is the two-dimensional Gamut Mapping. Here, the fine-tuned set of possible illuminants dominated the overall estimation result, leading to a very low error rate.

4.4.1.3 Fusion of Multiple Illuminants

In order to improve the illumination estimation on small regions, we fused the outputs of different algorithms based on their error statistics. We evaluated different fusion approaches. As a straightforward baseline method, we computed the average of all combined estimates. Besides this, we used two approaches based on machine learning regression. First, we used Gradient tree boosting as a classical machine learning algorithm to combine multiple weak predictors to a single strong one. As an alternative, we used Random forest regression. It consists of a set of tree predictors, which are trained on randomly chosen, different training sets. The output is computed as the average response over all trees in the forest.

The implementations of both machine learning algorithms were taken from the `openCV` library [Open 12]. For Gradient tree boosting, we used a squared loss function, 200 learning iterations and a maximum depth of 20. The Random forest regression was trained with a maximum depth of 50 with at most 100 trees. Note that, in difference to prior work, we did not use additional features to guide the fusion process. Instead, only the estimates (plus for the training set the ground truth, of course) were available for computing the regression. Thus, this approach can be seen as a "brute force" approach to localized illuminant estimation. Its outcome should serve as a cue whether we can use variants of these established algorithms also for illumination estimation on non-uniformly illuminated scenes.

4.4.1.4 Evaluation

In our evaluation, we used as a performance measure the angular error ϵ_{ang} as stated in Eqn. 4.10 on page 62. The results over multiple estimates are most often aggregated by computing the median of the errors. Additionally, we computed the mean, root mean square error and the maximum error for every estimator.

For the evaluation we used leave-one-out cross validation. The base estimators for the fusion schemes have been trained on the database by Gehler *et al.*, using the reprocessed ground truth by Shi and Funt [Shi 11]. Accordingly, the fusion algorithms have also been trained on this database. Thus, our own captured testing data was for the algorithms completely unknown.

Evaluation on Uniform Illumination First, we validated our implementations on the reprocessed database by Gehler *et al.* and Shi and Funt [Gehl 08, Shi 11]. To our knowledge, this is the largest real-world dataset which is available as raw data. The reflection target has been masked out. The results show the overall performance of our implementations of individual algorithms — as there are always implementational ambiguities. They also serve as a basis for comparison to the subsequent evaluation on images containing non-uniform illumination. Even in the Gehler database, there are images taken under multiple illuminants, even though a single illuminant is provided as ground truth. This is a source of error which is more prominent in the indoor scenes. Thus, we subdivided the images in outdoor and indoor images (see Tab. 4.3 and Tab. 4.4).

"Do nothing" assumes a white illuminant, and "Average illuminant" estimates always the average of all the training illuminants. For the generalized Gray World

Algorithm	Angular error in °			
	RMS	Mean	Median	Max
Do nothing	13.6	13.5	13.6	21.2
Average illuminant	3.0	2.3	1.9	18.7
White patch Retinex	10.5	8.4	6.6	37.3
Gray World	7.6	6.4	5.3	25.2
1st order Gray Edge	5.6	4.7	3.8	24.3
2nd order Gray Edge	5.3	4.2	3.2	24.3
Best Gray World / Edge	5.3	4.2	3.2	24.3
Gamut Mapping (max)	8.4	6.6	5.2	31.6
Gamut Mapping (mean)	5.0	4.6	4.5	17.4
Bayesian Color Constancy	4.1	3.2	2.5	19.5
Average estimate	6.1	4.9	3.6	21.1
Gradient tree boosting	3.5	2.6	1.9	18.6
Random forest regression	3.6	2.7	2.2	18.7

Table 4.3: Root mean square, mean, median, and maximum errors for outdoor images from the reprocessed Gehler *et al.* database.

we used the following settings: We varied the Minkowski norm with $1 \leq p \leq 10$, $0 \leq \sigma \leq 4$ in steps of 1. We explicitly report only the results for "White patch Retinex" ($n = 0$, $p \to \infty$, $\sigma = 0$), "Gray World" ($n = 0$, $p = 1$, $\sigma = 0$), "1st order Gray Edge" ($n = 1$, $p = 1$, $\sigma = 1$), "2nd order Gray Edge" ($n = 2$, $p = 1$, $\sigma = 1$), and the best performing generalized Gray World algorithm based on the median angular error. For Gamut Mapping, "Gamut Mapping (max)" denotes 3D Gamut Mapping that chooses the illuminant based on the maximum trace. "Gamut Mapping (mean)" denotes 2D Gamut Mapping with the mean selection strategy. Only the 2D Gamut Mapping uses illuminant constraints. "Bayesian color constancy" denotes Bayesian illumination estimation using the Euclidean distance as loss function. For the fusion of the estimates, "Average estimate" denotes the mean of the outputs of the fused estimators, "Gradient tree boosting" and "Random forest regression" denote the two machine learning-based regression approaches.

From Tab. 4.3, we observe that the error of choosing the mean illuminant is very small for outdoor images. Thus, the variability of illuminants is small for the outdoor images. The 2D Gamut Mapping variants benefit the most from that fact. The best performing generalized Gray World algorithm for outdoor images was $n = 2$, $p = 1$, $\sigma = 1$. Using regression and a collection of estimators, we were able to obtain results that are better than the best performing single algorithm. For the training and testing itself, we used k-fold cross-validation. Interestingly, the average illuminant from the ground truth performs still slightly better with respect to the mean and RMS error, which suggests that the variability of illuminants is not very high in this dataset. For indoor images (see Tab. 4.4), the best performing Gray World method was Gray Edge with $n = 1$, $p = 1$, $\sigma = 1$. Here, the variation of illuminants is significantly higher, as can be seen from the higher error of the "Average illuminant". Note that

among the fusion schemes, Random forest regression performs best and improves the final error for approximately 0.9°, compared to the single estimators.

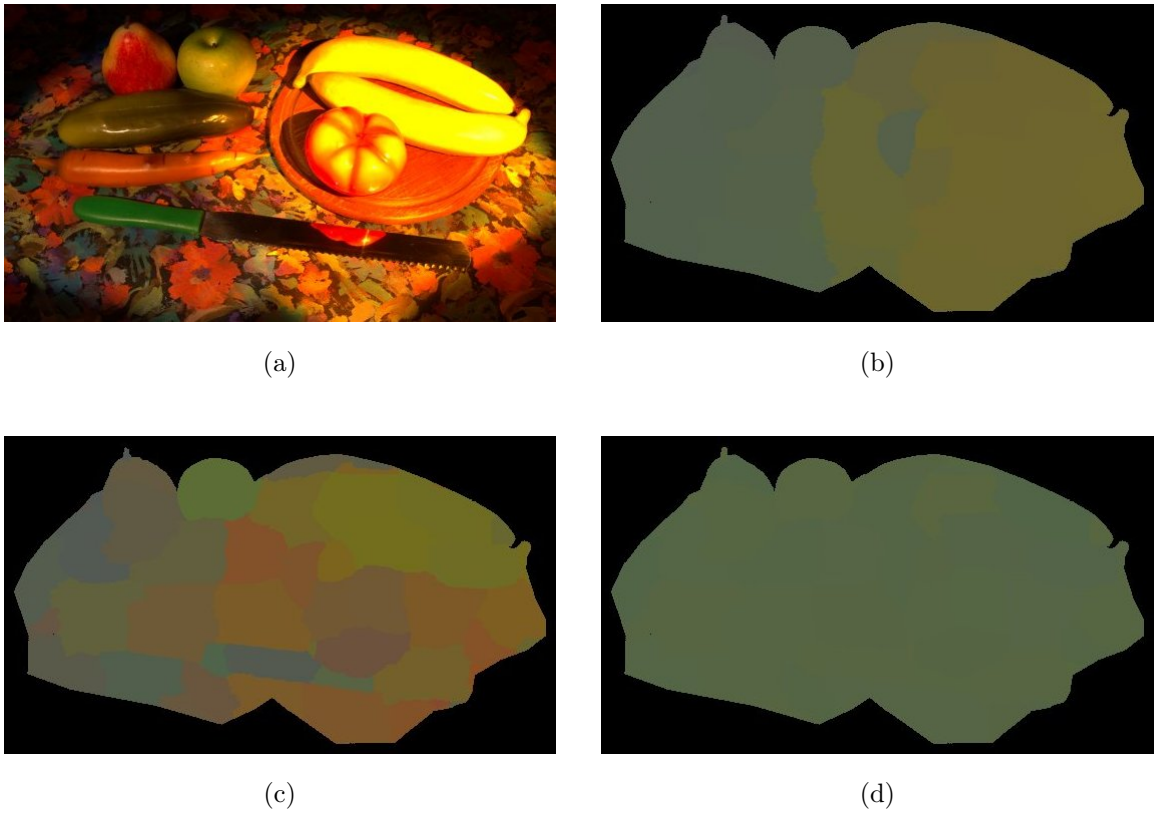
Algorithm	Angular error in °			
	RMS	Mean	Median	Max
Do nothing	14.4	13.8	13.4	27.4
Average illuminant	8.1	7.0	6.5	22.8
White patch Retinex	12.6	10.9	10.7	48.1
Gray World	7.7	6.6	6.3	24.8
1st order Gray Edge	5.3	4.7	4.3	15.0
2nd order Gray Edge	6.4	5.6	4.9	17.3
Best Gray World / Edge	5.3	4.7	4.3	15.0
Gamut Mapping (max)	10.9	9.5	8.9	27.4
Gamut Mapping (mean)	8.2	7.1	6.7	20.6
Bayesian Color Constancy	7.0	5.9	5.4	20.0
Average estimate	8.2	7.0	6.5	29.4
Gradient tree boosting	5.3	4.5	3.9	15.7
Random forest regression	4.7	3.9	3.4	16.2

Table 4.4: Root mean square, mean, median, and maximum errors for indoor images from the reprocessed Gehler *et al.* database.

Evaluation on Non-Uniform Illumination The 36 multi-illuminant images have been segmented in superpixels. The segmentation was transferred to the ground truth images, and the per-segment ground truth was determined by averaging the ground truth over the superpixels. This averaging introduces inaccuracies on an illumination boundary. However, we considered it reasonable, because the algorithms produce illuminant estimates per-superpixel, and as such this is the level of detail in which we require ground truth.

An example segmentation and estimation is shown in Fig. 4.17. The segmentation parameters were chosen, so that an image was subdivided into approximately 30 superpixels, with the individual superpixel size varying between approximately $10,000$ and $50,000$ pixels.

We trained the illuminant estimators on the indoor images from the reprocessed Gehler database. The results are shown in Tab. 4.5. The best gray world configuration was $n = 0$, $p \to \infty$, $\sigma = 1$. Note that the individual errors on each superpixel are largely increased, and range from a median of about 4.6 degrees to 13.7 degrees. This has been expected. The superpixel segmentation aims to provide areas of approximately the same color, which is theoretically poor input for almost all applied estimators. For instance, Gamut Mapping expects colorful images, while gray world expects balanced colors. Picking a superpixel with mainly only one color undermines such algorithms. However, this effect is apparently limited. For instance, in the present case, the 1st and 2nd order gray world algorithms perform considerably worse than the statistical methods.



(a) (b)

(c) (d)

Figure 4.17: Example segmentation, ground truth and illuminant estimation (colors are scaled for the purpose of printing) (a) Original image (b) ground truth (c) Result of gray world (d) Result of 2D Gamut Mapping

In this scenario, we performed all necessary training steps on the Gehler indoor database. Thus, for Gamut Mapping and Bayesian Color Constancy, as well as the fusion algorithms, the tested database was unknown beforehand. Similarly, also the "Average Illuminant" refers to the average of the ground truth of the database by Gehler *et al.*. In this experiment, regression by boosting did not improve the results. However, regression based on Random forests could lower the median error to 4.1 degrees. The improved estimates yield an error level that again clearly improves over the single illuminant estimates. This is a surprising result, as the underlying estimates by themselves are comparably weak, and no additional image features are used for guidance of the fusion process. As a consequence, we conclude that to some extent, color constancy under non-uniform illumination can be addressed by ensembles of locally applied single-illuminant estimators.

4.4.1.5 Discussion

We evaluated single illuminant estimation algorithms and fused versions of these algorithms in three scenarios. The two cases on the database by Gehler *et al.* can be used as a baseline reference for the performance of our implementation. Among these, the indoor pictures exhibit a greater variability. Thus, when evaluating on the newly

Algorithm	Angular error in °			
	RMS	Mean	Median	Max
Do nothing	11.7	10.4	10.0	21.6
Average illuminant	8.3	7.7	7.8	15.4
White patch Retinex	7.0	5.8	5.1	21.6
Gray World	5.7	5.1	5.0	14.4
1st order Gray Edge	14.9	14.2	13.7	29.0
2nd order Gray Edge	13.4	12.8	12.6	27.3
Best Gray World / Edge	6.3	5.4	4.6	21.6
Gamut Mapping (max)	7.2	5.9	5.2	21.3
Gamut Mapping (mean)	7.1	6.4	6.3	14.6
Bayesian Color Constancy	8.1	6.5	4.8	20.7
Average estimate	8.4	7.8	7.7	16.2
Gradient tree boosting	6.8	6.2	6.0	17.5
Random forest regression	5.0	4.4	4.1	12.9

Table 4.5: Root mean square, mean, median, and maximum errors non-uniform illumination estimation on superpixels.

captured data, we trained our algorithms on the Gehler indoor images. As we only captured 4 scenes for this experiment, we did not attempt to perform cross-validation on the limited data. In general, machine learning algorithms have difficulties when training on one database and evaluating on another. During our experiments, we also noted this behavior. Nevertheless, results show that we managed to avoid overfitting to some extend. The methods generalize sufficiently, such that their good performance can be confirmed on the unknown data.

We consider several aspects of the evaluation section worth to discuss in greater detail. At first, it is counter-intuitive that the statistical methods, i. e. Gamut Mapping and Bayesian Color Constancy, yield at all useful results on the superpixel estimation. Indeed, we observed that a drop color variation yields to several problems with the statistical algorithms. However, note that in the present case, the superpixels are still relatively large with respect to the level of detail in the image. As a consequence, at least two surfaces and some shading artifacts are typically contained within one superpixel.

Another particularity of this approach is the fact that no additional image features are used to support the fusion of the estimates. Thus, the fusion methods can be seen as a way of integrating brute force results on the multi-illuminant problem. Thus, the results suggest – although this should be confirmed in more extensive experiments – that the error behavior of existing algorithms contains useful patterns to estimate the true illuminant.

As a consequence, we see in these results two main points. First, evaluating a large number of existing algorithms for every superpixel in the image dramatically increases the computational cost. Additionally, methods like 2D-Gamut Mapping can become slow if the observed Gamut is very small. More efficient algorithms are necessary for a more practical solution to the multi-illuminant problem. At the same time, we

are surprised how well the lack of color information can be compensated by adding more estimators. Looking at the error rates, we eventually reached a median error of 4.1° on non-uniform illumination, which is even better than the best performing estimator on the single-illuminant indoor dataset by Gehler *et al.*.

4.4.2 Physics-based Multi-Illuminant Estimation and Localization

Although the results on off-the-shelf single-illuminant estimators are encouraging, one can assume that tailored algorithms might be better suited for multi-illuminant estimation. One particular drawback of most methods in the previous section is their dependence on the training data. One can expect that the performance of these methods considerably drops when applied on arbitrary images in the web. Additionally, the localization problem of multiple illuminants (see page 82) is not directly addressed, rather avoided by directly classifying each superpixel on its own.

Thus, we investigated and extended a physics-based method to perform rough illuminant segmentation on real-world images without prior training. More precisely, we used the method by Tan *et al.* [Tan 04], and propose an extension to it that relaxes the requirement of clean, segmented specularities. We call the core of this method "Illuminant Estimation by Voting". A segmentation on local illuminant estimates provides a first step towards a solution of the localization problem.

4.4.2.1 Inverse-intensity chromaticity (IIC) space

Most surfaces exhibit a mixture of diffuse and specular reflectance. Thus, we adopt the dichromatic reflection model and the Neutral Interface Assumption, as presented in Eqn. 4.6 on page 61. Assuming sharpened sensors and a linear camera, one can rewrite Eqn. 4.6 at pixel $\boldsymbol{x}$ in a simplified form as a sum of diffuse and specular reflectance,

$$\boldsymbol{p}(\boldsymbol{x}) = m_d(\boldsymbol{x})\boldsymbol{s}^d(\boldsymbol{x}) + m_s(\boldsymbol{x})\boldsymbol{s}^s \ , \tag{4.32}$$

where $\boldsymbol{s}^d(\boldsymbol{x}) = (s^d_R(\boldsymbol{x}), s^d_G(\boldsymbol{x}), s^d_B(\boldsymbol{x}))^{\mathrm{T}}$ and $\boldsymbol{s}^s = (s^s_R, s^s_G, s^s_B)^{\mathrm{T}}$ denote the per-channel responses of specular and diffuse reflectance. Here, analogously to Eqn. 4.6, the specular component $\boldsymbol{s}^s$ does not depend on $\boldsymbol{x}$. Thus, we are currently assuming one single illuminant, which will be relaxed later. We operate on chromaticities $\chi_c(\boldsymbol{p})$ of a pixel color $\boldsymbol{p}$ where $c \in \{R, G, B\}$ denotes the color channel of the chromaticity. For summations over the color channels, we define the index $i \in \{R, G, B\}$. In a similar manner, one can define the diffuse chromaticity $\zeta_c(\boldsymbol{x})$ at pixel $\boldsymbol{x}$ and the specular chromaticity γ_c in channel c as

$$\zeta_c(\boldsymbol{x}) = \frac{s^d_c(\boldsymbol{x})}{\sum_i s^d_i(\boldsymbol{x})} \ , \tag{4.33}$$

$$\gamma_c = \frac{s^s_c}{\sum_i s^s_i} \ . \tag{4.34}$$

In terms of chromaticities, the dichromatic reflectance model in Eqn. 4.6 on page 61 can then be rewritten as

$$p_c(\boldsymbol{x}) = w_d(\boldsymbol{x})\zeta_c(\boldsymbol{x}) + w_s(\boldsymbol{x})\gamma_c \ , \tag{4.35}$$

where

$$w_d(\boldsymbol{x}) = m_d(\boldsymbol{x}) \sum_i s_i^d(\boldsymbol{x}) \ , \tag{4.36}$$

$$w_s(\boldsymbol{x}) = m_s(\boldsymbol{x}) \sum_i s_i^s \ . \tag{4.37}$$

In this formulation, w_d and w_s act as geometry- and brightness-dependent weighting factors.

A convenient color space for analyzing the chromaticity relationship between purely specular, purely diffuse and mixtures of specular and diffuse reflection is the *inverse-intensity chromaticity space* introduced by Tan *et al.* [Tan 04]. According to that work, by dividing Eqn. 4.35 by $\sum_i p_i(\boldsymbol{x})$, the chromaticity at $\boldsymbol{x}$, $\chi_c(\boldsymbol{p}(\boldsymbol{x}))$ becomes:

$$\chi_c(\boldsymbol{p}(\boldsymbol{x})) = \frac{w_d(\boldsymbol{x})\zeta_c(\boldsymbol{x}) + w_s(\boldsymbol{x})\gamma_c}{w_d(\boldsymbol{x}) \sum_i \zeta_i(\boldsymbol{x}) + w_s(\boldsymbol{x}) \sum_i \gamma_i} \ . \tag{4.38}$$

Solving Eqn. 4.38 for $w_s(\boldsymbol{x})$ and inserting this back in Eqn. 4.35 one gets:

$$p_c(\boldsymbol{x}) = w_d(\boldsymbol{x})(\zeta_c(\boldsymbol{x}) - \gamma_c) \left(\frac{\chi_c(\boldsymbol{p}(\boldsymbol{x}))}{\chi_c(\boldsymbol{p}(\boldsymbol{x})) - \gamma_c} \right) \tag{4.39}$$

which leads to the definition of $s_c(\boldsymbol{x})$ as:

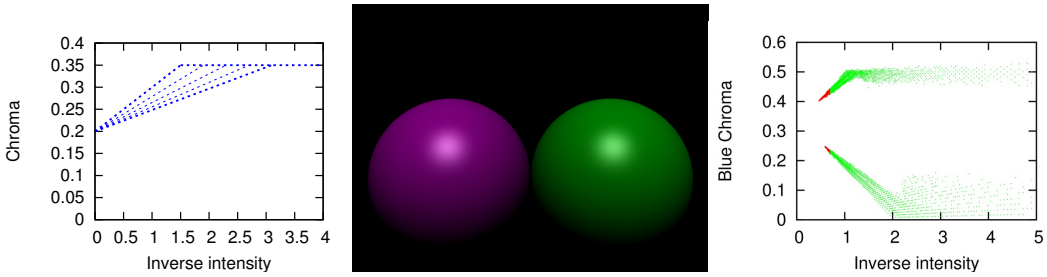
$$s_c(\boldsymbol{x}) = w_d(\boldsymbol{x})(\zeta_c(\boldsymbol{x}) - \gamma_c) \ . \tag{4.40}$$

Thus, a linear relationship between the image chromaticity $\chi_c(\boldsymbol{p}(\boldsymbol{x}))$ and the inverse-intensity $1/\sum_i p_i(\boldsymbol{x})$ can be established,

$$\chi_c(\boldsymbol{p}(\boldsymbol{x})) = s_c(\boldsymbol{x}) \frac{1}{\sum_i p_i(\boldsymbol{x})} + \gamma_c \ . \tag{4.41}$$

In this representation, $s_c(\boldsymbol{x})$ can be seen as the slope of a line with intercept γ_c. The domain of the line is determined by $1/\sum_i p_c(\boldsymbol{x})$, the range is given by $0 \leq \chi_c(\boldsymbol{p}(\boldsymbol{x})) \leq 1$. The space spanned by $1/\sum_i p_i(\boldsymbol{x})$ and $\chi_c(\boldsymbol{p}(\boldsymbol{x}))$ is called *inverse-intensity chromaticity space* [Tan 04]. The chromaticity value where this line intersects the vertical axis gives the illuminant chroma estimate for channel c.

The inverse-intensity diagram shown in Fig. 4.18a is an idealized graphical representation of the distribution of pixels in inverse-intensity chromaticity space. The horizontal axis corresponds to the inverse-intensity $1/\sum_i p_c(\boldsymbol{x})$, and the vertical axis $\chi_c(\boldsymbol{p}(\boldsymbol{x}))$, the illuminant chromaticity. This figure visualizes the ideal distribution of pixels of a monochrome object in IIC space. The diffuse pixels lie on a single horizontal line, while pixels that exhibit specular reflection align according to their specific $s_c(\boldsymbol{x})$-values in lines between the illuminant color on the vertical axis and the diffuse line. In the context of the more widely used RGB color histograms, the horizontal line of the IIC space corresponds to the diffuse line which in RGB space emanates from the origin. Similarly, in RGB space the specular pixels (which in IIC space emanate from the illuminant chroma) form lines whose direction is the color of the incident light.



(a) (b) (c)

Figure 4.18: Left: Idealized pixel distribution of a monochrome object in inverse-intensity chromaticity space. Middle: An image with few distinct albedoes. Right: the distribution of pixels in IIC space for the blue chromaticity. Highly specular pixels are shown in red.

Note that, though the formulation is mathematically elegant, it is, in general, not possible to obtain $s_c(\boldsymbol{x})$ directly in order to estimate the illuminant color. Typically, different values of $s_c(\boldsymbol{x})$ can occur within a single specular region. According to Eqn. 4.40, distinct $s_c(\boldsymbol{x})$-values can result from variations in the underlying geometry, w_d. However, specular pixels with the same underlying albedo and the same geometric factors $w_d(\boldsymbol{x})$ and $w_s(\boldsymbol{x})$ have the same slope in the IIC space. For a complete discussion, see [Tan 04].

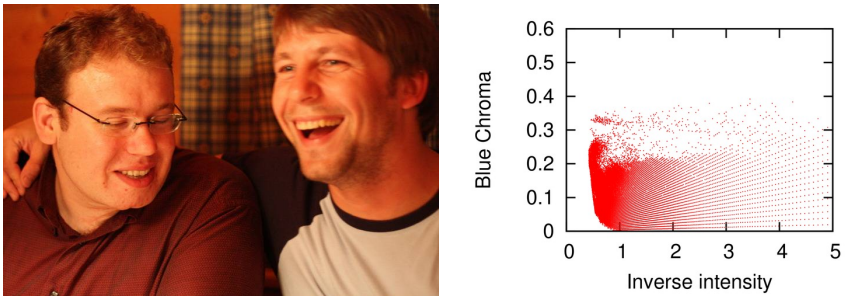
4.4.2.2 IIC distributions of real-world images

Though an explicit calculation of the illuminant chromaticity γ_c is in most cases not feasible, one can exploit the distribution of pixels in inverse-intensity chromaticity space in order to detect the $\chi_c(\boldsymbol{p}(\boldsymbol{x}))$ intercept.

Tan *et al.* [Tan 04] developed a methodology which analyzes the location of very highly specular pixels in IIC space. They estimated the illuminant color by using a Hough transform of the specular pixels with parameters $s_c(\boldsymbol{x})$ and γ_c. They demonstrated that for images with few distinct albedo values it is feasible to exploit the spatial distribution of specular pixels and obtain a reliable estimate of the illuminant color. A sample distribution for an image with few distinct albedos is shown in Fig. 4.18b and Fig. 4.18c.

However as scenes become more complex, the distribution of pixels in IIC space does not form clearly separable clusters (see Fig. 4.19). Though one could try to identify specularities and focus the analysis on the specular pixels, the results are not reliable enough for precise illuminant color estimation [Finl 01b, Ries 09a].

Rather, we propose using pixels that exhibit a mixture of specular and diffuse reflectance. In both IIC space and RGB space, each specular cluster includes *all the pixels that are not purely diffuse*. The larger the specular component, the farther the pixels will lie from the diffuse line. In particular in IIC space the closer the pixels will lie on the $\chi_c(\boldsymbol{p}(\boldsymbol{x}))$-intercept. However, as can be seen in Fig. 4.18c, the highly specular pixels comprise only a small part of a specular cluster. Under the assumption of a single underlying albedo, the intercept γ_c can still be extracted from the bisector



(a) (b)

Figure 4.19: An arbitrary image and its distribution of pixels in IIC space (blue chromaticity).

(along the axis of elongation) of the entire specular cluster. Thus, in obtaining an estimate of the illuminant, one can exclude all highly specular pixels and use all the remaining pixels with a mixture of diffuse and specular reflection. This offers certain distinct advantages:

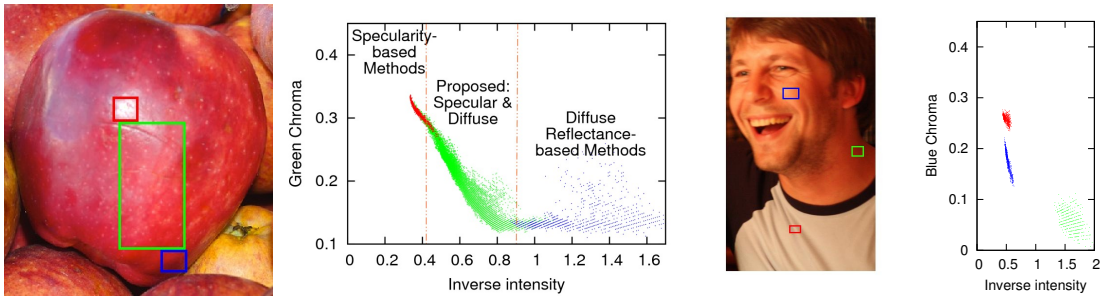
1. It uses a larger number of pixels making the estimate less sensitive to outliers.

2. It excludes pixels whose values are at the upper limit of the sensor's dynamic range and are thus unreliable (e. g., due to color clipping, blooming) [Gijs 09, Klin 88, Finl 01b].

3. It typically examines pixels whose values are neither to large nor too small. Hence, it uses that part of information in an image, where the typical commercial camera is designed to give the best fidelity.

Fig. 4.20a shows an image downloaded from the web and the domains of the different families of physics-based color constancy algorithms. Three regions representative of highly specular (in red), purely diffuse (in blue) and a mixture of specular and diffuse (green) are hand-selected. The corresponding clusters in IIC space are shown in Fig. 4.20b.

Furthermore, to increase the robustness of the illuminant color estimate we propose the collection of multiple independent local estimates which can be combined in deriving a more reliable global estimate. Local estimates can be obtained by performing the IIC distribution analysis in *small image regions* as shown in Fig. 4.20c and Fig. 4.20d. The use of small image regions has the additional advantage that it increases the probability that the pixels within a patch have the same underlying albedo. The subsequent section describes in greater detail the process of collecting appropriate local samples.



(a) (b) (c) (d)

Figure 4.20: Left: Illustration of the domain of the proposed method in comparison with other existing physics-based color constancy techniques. Right: Hand selected image regions (using a close-up of Fig. 4.19a) and their corresponding distribution in IIC space (blue chromaticity).

4.4.2.3 Sample selection

One of the key ideas of the proposed methodology is the derivation of a robust global illuminant estimate $\boldsymbol{e}$ through the use of multiple local estimates ${}^{i}\tilde{\boldsymbol{e}}$. This global estimate is obtained from n independent and identically distributed (iid) samples.

The overall goal is to minimize the angular estimation error ϵ_{ang} (see Eqn. 4.10 on page 62) between the true illuminant $\boldsymbol{e}$ and the final estimate $\tilde{\boldsymbol{e}}$.

Our approach is to sample over the entire image. This leads to a set $\mathcal{S} = \{{}^{i}\tilde{\boldsymbol{e}} | i = 1 \ldots n\}$ of independent and identically distributed (iid) estimates. This set consists of a subset of "positive" samples $\mathcal{P}$, whose angular distance to the true illuminant $\boldsymbol{e}$ is small, and a subset of "negative" samples $\mathcal{N}$ with a large distance of the true illuminant. Thus,

$$\mathcal{S} = \mathcal{P} \cup \mathcal{N} \ . \tag{4.42}$$

Then, the elements of $\mathcal{P}$ form a unimodal distribution around the true illuminant $\boldsymbol{e}$, such that

$$\lim_{|\mathcal{P}| \to \infty} \text{argmax} \ \ \text{hist}(\mathcal{P}) = \boldsymbol{e} \ , \tag{4.43}$$

where $\text{hist}(\mathcal{P})$ denotes the histogram of the illuminant estimates in $\mathcal{P}$.

The elements of $\mathcal{N}$ can be arbitrarily distributed. Our goal is to reduce the influence of $\mathcal{N}$ while preserving $\mathcal{P}$, so that finally

$$\lim_{|\mathcal{S}| \to \infty} \text{argmax} \ \ \text{Hist}(\mathcal{S}) = \boldsymbol{e} \ . \tag{4.44}$$

In order to increase the probability that an estimate ${}^{i}\tilde{\boldsymbol{e}}$ obtained from a local region will be a good estimate (i.e. ${}^{i}\tilde{\boldsymbol{e}} \in \mathcal{P}$), the image region where this estimate is computed should satisfy the following properties:

- *uniform albedo.* Both the mathematical analysis in Section 4.4.2.1 and the study of specular clusters in Section 4.4.2.2 assume uniform albedo. Thus, any local region used in estimating the illuminant color should satisfy this assumption.

- *elongated, non-horizontal clusters in IIC space.* Our goal is to identify non-diffuse clusters. Since in IIC space diffuse clusters are horizontal, we wish to exclude image regions that generate such IIC distributions. Furthermore, in order to more accurately estimate the non-diffuse cluster bisector (see Section 4.4.2.2), we require that the corresponding cluster is clearly elongated.

The iid sampling and screening of samples in our implementation is performed as follows. We first segment the image in superpixels of approximately uniform chromaticity values. A superpixel is a locally connected region of pixels that share low-level properties, like in our case similar chromaticity values. We use the graph-based segmentation by Felzenszwalb and Huttenlocher [Felz 04], but any segmentation method that decomposes an image into regions with approximately the same albedo could also be employed.

We sample with replacement small regions within superpixels with probability proportional to the size of the superpixel. Any iid sampling which results in small regions of approximately uniform albedo could be employed.

In practice, we sample small rectangles over the entire image and single pixel positions $\boldsymbol{x}$ within these rectangles. The region used in computing a local estimate ${}^{i}\tilde{\boldsymbol{e}}$ is composed of all the pixels which lie in the intersection of the rectangle and the superpixel $\mathcal{F}_i$ such that $\boldsymbol{x} \in \mathcal{F}_i$.

The next step is to examine the shape of the distribution of pixels in the candidate region. One way of doing this is via PCA. Let $\mathcal{R}_{\text{IIC}}$ be the set of pixels under investigation in IIC space, λ_1 its largest eigenvalue, λ_2 its second largest eigenvalue. Then the eccentricity $\text{ecc}(\mathcal{R}_{\text{IIC}})$ is

$$\text{ecc}(\mathcal{R}_{\text{IIC}}) = \sqrt{1 - \frac{\lambda_2}{\lambda_1}} \ . \tag{4.45}$$

We consider only sets $\mathcal{R}_{\text{IIC}}$ that exhibit a minimum eccentricity (in our experiments typically 0.2). In order to avoid purely diffuse pixels we compute also the slope of the eigenvector belonging to λ_1. A set $\mathcal{R}_{\text{IIC}}$ must also satisfy a minimum slope (0.003, in our experiments). See Section 4.4.2.5 for further discussion on the region size. The actual illuminant estimate is computed from the $\chi_c(\boldsymbol{p}(\boldsymbol{x}))$-intercept of this eigenvector, as an approximation for γ_c in Eqn. 4.41.

Please note that like [Tan 04] we exclude pixels with duplicate values in our sample validation analysis, since our focus is on the spatial distribution of pixels in IIC space. We also exclude any pixels that are very close to the limits of the dynamic range of the camera (i.e. saturated and very dim pixels).

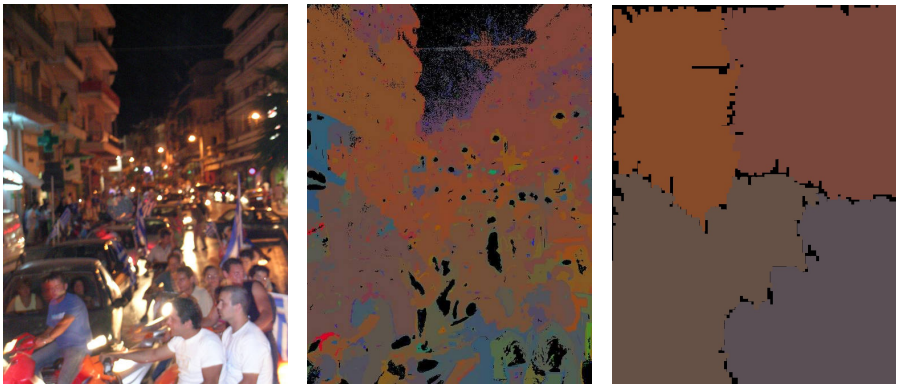
4.4.2.4 Multiple Illuminants

The algorithm that has been presented so far can directly be used to obtain multiple illuminant estimates. Once local illuminant estimates are obtained per superpixel, the local information can be combined as follows for the final computation of the number and color of the dominant illuminants in the scene.

1. Group local estimates into regions with consistent/similar illuminant color.

2. Obtain a new estimate per illuminant region.

An example of this process is shown in Fig. 4.21. The details of each of the aforementioned steps are provided in the following subsections.



(a) (b) (c)

Figure 4.21: (a) Original image. (b) Local illuminant estimation. (c) Segmented regions, colored according to the illuminant estimate.

We extended our algorithm as described above to handle multiple illuminants by examining the estimates per superpixel more closely. Note two assumptions. First, multiple illuminants are often clearly visible in the superpixel map, see Fig. 4.21b for an example. Second, outlier estimates occur typically isolated, both spatially and in the distribution of estimated colors. In order to extract the regions of the dominant regions, we do the following steps.

1. Create an illuminant map by recoloring every superpixel by its local illuminant estimate.

2. Downscale the map, such that the larger dimension of this image is only 140 pixels.

3. Group regions of similar estimates with the Quick Shift algorithm [Veda 08].

The downscaling suppresses a large amount of relatively small noisy regions. Its purpose is to speed up the Quick Shift algorithm. Quick Shift is a method for seeking modes in densities, which is why we preferred it over [Felz 04] for grouping similar estimates. In our case, we obtained the best results by applying it on the joint spatial and chromaticity domain, using red and blue chromaticities. Quick Shift creates trees of data points and distances between these nodes, such that similar regions can be segmented by separating subtrees from this graph. By discarding smaller segments, we typically obtain three to six major regions in the downscaled image.

For refining the estimation, we use the estimated illuminant regions for iid sampling instead of the whole image. The resulting per-region illuminant estimates can further be merged. In this work, we merged regions that were smaller than a predefined threshold of 10% of the image region.

Scene	Median ϵ_{ang}
Gamut Mapping	**3.1**°
Gray World	8.8°
White Patch Retinex	5.0°
Color-by-Correlation	8.6°
Physics-based diff+spec	4.4°

Table 4.6: Algorithm performance (comparison numbers by [Gijs 10b]) on benchmark laboratory images by Barnard *et al.* [Barn 02b].

Scene	Median ϵ_{ang}
Regular gamut+offset-model	5.7°
Gray World	7.0°
White Patch Retinex	6.7°
Color-by-Correlation	6.5°
1^{st}-order Gray Edge	5.2° (∗)
2^{nd}-order Gray Edge	5.4° (∗)
Physics-based diff+spec	**4.4**°

Table 4.7: Algorithm performance on benchmark real-world images, using the images by Ciurea and Funt [Ciur 03]. The numbers marked with an (∗) are computed over a subset of the dataset.

4.4.2.5 Experiments

For validating the performance of the method, we conducted quantitative evaluation on the single-illuminant datasets by Barnard *et al.* [Barn 02b] and Ciurea and Funt [Ciur 03] (see Sec. 4.2 on page 63). Instead of estimating the illuminant per superpixel, a global consensus is formed over the whole image. The error metric used in the evaluation of the two benchmark datasets is the angular error d_{Angular}, as defined in Eqn. 4.10 on page 62. As we randomly draw subregions per superpixel, we obtained 10 estimates and computed the mean error per image.

We also present qualitative results for single- and multi-illuminant estimation on images downloaded from websites like flickr [Yaho 12]. In the next section, this methodology is embedded in a more complex algorithm, and quantitatively evaluated on the proposed multi-illuminant dataset from Sec. 4.3.2.

Parameter selection For the segmentation of the chromaticity images by Felzenszwalb and Huttenlocher [Felz 04], the parameters were fixed by visual inspection to $\sigma = 0.3$, $k = 200$, and minimum segment size $m = 15$. The sampling rectangle size was set to 7×31 pixels. Our tests, however, indicated that the lab database was more challenging for our methodology. Hence, for the lab images we tried different rectangles and concluded that a larger size of 30×55 pixels gave the best performance.

Benchmark laboratory images Table 4.6 summarizes the performance of the presented methodology in comparison to state-of-the-art algorithms on the dataset

(a) Town (b) Woman (c) Castle (d) Sculpture

(e) Pool (f) People (g) Cows (h) Chapel

Figure 4.22: Subset of the selected real-world images, including two challenging cases in Fig. 4.22h and Fig. 4.22g.

by Barnard *et al.* [Barn 02b]. Our physics-based technique is only outperformed by the Gamut Mapping, which, however, is dependent on a training stage.

Benchmark real-world images The database of Ciurea and Funt [Ciur 03] includes images that are more representative of the pictures taken by arbitrary users. As can be seen in Table 4.7, the proposed physics-based method achieves a considerable improvement over (at the time of development) state-of-the-art methods. The referenced angular errors marked with an asterisk ($*$) are taken from [Lu 09] and are evaluated only on a subset of 711 images. The remaining measurements are extracted from [Gijs 10a] and are, like our evaluation, computed on the entire set of 11,000 images.

Out of the 15 provided scenes, the best result was obtained for "FalseCreek1" ($\epsilon_{\text{ang}} = 1.57°$), while "CIC2002_3" resulted in the worst performance ($\epsilon_{\text{ang}} = 11.46°$). The "CIC2002_3" is a sequence of indoor images, where there is high probability that the single illuminant assumption is violated. This observation is consistent with other indoor sequences of this dataset, as well as with arbitrary images we tested from the web (see Section 4.4.2.5).

Approximately uniformly illuminated real-world images Since our algorithm was designed for illuminant estimation of images typically found on the web, we also performed a qualitative evaluation on a set of almost 250 images we downloaded from various websites. The database contains images both of indoor and outdoor scenes (see Fig. 4.22). It includes a variety of different subjects, such as nature, people, animals and architecture. The outdoor images were acquired at different daytimes and under various weather conditions.

Scene	γ_r	γ_g	γ_b
Town	0.37 ± 0.02	0.34 ± 0.01	0.29 ± 0.02
Woman	0.33 ± 0.01	0.34 ± 0.01	0.33 ± 0.00
Castle	0.31 ± 0.02	0.33 ± 0.01	0.36 ± 0.02
Sculpture	0.31 ± 0.07	0.36 ± 0.05	0.33 ± 0.08
Pool	0.42 ± 0.04	0.37 ± 0.02	0.21 ± 0.02
People	0.48 ± 0.06	0.38 ± 0.06	0.14 ± 0.04
Cows	0.34 ± 0.01	0.32 ± 0.00	0.34 ± 0.01
Chapel	0.28 ± 0.02	0.30 ± 0.01	0.42 ± 0.03

Table 4.8: Stability of the algorithm performance on arbitrary real-world images. For the images of Fig. 4.22, the mean results and standard deviations of ten estimation runs are listed.

To illustrate the performance of the physics-based estimation method on different lighting and scene contents, we present estimates we obtained on a subset of representative images. The example images are shown in Fig. 4.22 and contain three outdoor scenes (upper row) and three indoor scenes (lower row), all captured under different illumination conditions. Fig. 4.22h and Fig. 4.22g show two challenging cases. Table 4.8 lists the corresponding estimates. For each scene, the mean estimate of ten randomized runs is given in combination with the standard deviation.

One can observe that the red component of the illuminant color in the outdoor scenes decreases from left ("Town", Fig. 4.22a) to right ("Castle", Fig. 4.22c). At the same time, the blue component is increasing. This tendency is captured quite well in the estimation results (Table 4.8). Furthermore, a comparison with the CIE standard illuminants leads to a reasonable interpretation of the estimates. The result of "Town" (Fig. 4.22a) corresponds to CIE D50 ($\boldsymbol{e}^{D50} = (0.37, 0.34, 0.30)^{\mathrm{T}}$), which is the standard used for horizon light. The estimate of "Woman" (Fig. 4.22b) is almost identical to CIE D65 ($\boldsymbol{e}^{D65} = (0.33, 0.33, 0.33)^{\mathrm{T}}$), which describes noon daylight. For "Castle" (Fig. 4.22c) the result corresponds to CIE D75 ($\boldsymbol{e}^{D75} = (0.32, 0.33, 0.35)^{\mathrm{T}}$), which is used for no direct sunlight. The chromaticity values of the different CIE standard illuminants are computed using the CIE 1931 2° standard observer in sRGB color space. There is a further interesting remark regarding "Woman" (Fig. 4.22b): in the estimates, the chromaticities of the three channels of the illumination color are well balanced, although the surfaces in the scene are significantly dominated by blue and red. This aspect indicates that the illumination estimation is not significantly affected by the presence of pure diffuse patches.

The estimation results for the indoor scenes fit nicely to the chromaticities of CIE A ($\boldsymbol{e}^{A} = (0.48, 0.33, 0.19)^{\mathrm{T}}$), which describes a tungsten light bulb. Furthermore, the visual impression of increasing red chromaticity and decreasing blue chromaticity from left ("Sculpture", Fig. 4.22d) to right ("People", Fig. 4.22f) is well captured in the estimation results.

An important aspect of the evaluation on natural images is the stability of the global estimate. The standard deviations of the results listed in Table 4.8 are very small. A key component of the proposed methodology is the use of identically and

(a) Dancer (b) Guide (c) Doors

Figure 4.23: Subset of the selected real-world images. The images are annotated with the segment numbers.

independently distributed samples of non-diffuse image regions. This stability in the results indicates that our technique for sample collection and verification is effective. For indoor scenes, the standard deviation is slightly increased. This observation is consistent with the quantitative evaluation on the database of Ciurea and Funt [Ciur 03] (see Section 4.4.2.5).

The images in Fig. 4.22h and Fig. 4.22g, and their estimation results (Table 4.8) show the limitations of the proposed method. In "Cows" (Figure 4.22g), the illuminant estimation fails. The estimated illuminant is composed of almost the same proportion of red, green and blue though it is apparent that the value of the red chromaticity should slightly dominate. As the grass on the ground and the coat of the buffalo are highly textured, it is difficult to find a sufficient quantity of patches which pass the selection step. This drawback could be reduced by an appropriated preprocessing of the images. Another limitation of the method is its applicability to non-dielectric surfaces. As the estimation approach is based on the dichromatic reflectance model, the illuminant estimation results for "Chapel" (Fig. 4.22h) seem unreasonable.

Multiple Illuminants Quantitative results of the first part of the method are presented within the more advanced framework for multi-illuminant estimation in the next section. In this section, we present qualitative results on multi-illuminant segmentation on images downloaded from websites like flickr. We collected about 30 mixed-illuminant scenes, and examined also the images used by Hsu *et al.* [Hsu 08]. The code for our method can be downloaded from the web[13]. The error metric used in the evaluation of the two benchmark datasets is the angular error ϵ_{ang} as stated in Eqn. 4.10 on page 62. For the segmentation of the chromaticity images by Felzenszwalb and Huttenlocher [Felz 04], the parameters were fixed by visual inspection to $\sigma = 0.3$, $k = 200$, and minimum segment size $m = 15$. The sampling rectangle size was set to 7×31 pixels.

The spatial location of the segments is denoted by their overlaid respective numbers in the images. In Fig. 4.23a, flash light illuminates the heads of the spectators,

[13] `http://www5.cs.fau.de/`

Scene	Segment 1	Segment 2
Dancer	$(0.327, 0.336, 0.337)$	$(0.330, 0.334, 0.336)$
Guide	$(0.312, 0.343, 0.345)$	$(0.347, 0.336, 0.316)$
Doors	$(0.309, 0.339, 0.352)$	$(0.294, 0.337, 0.369)$
Scene	Segment 3	Segment 4
Dancer	$(0.415, 0.306, 0.279)$	$(0.354, 0.319, 0.327)$
Guide	$(0.343, 0.327, 0.330)$	$(0.331, 0.334, 0.335)$
Doors	$(0.379, 0.334, 0.287)$	-

Table 4.9: Per segment illuminant chromaticity estimates for the multi-illuminant images.

while the remaining scene is mainly reddish illuminated. In Fig. 4.23b, the tourist in the foreground is illuminated from behind by a blueish light source. The rest of the scene contains mainly light from the lamps. Finally, Fig. 4.23c is taken from the dataset by Hsu et al. [Hsu 08]. Table 4.9 shows the illuminant estimates per segment. The tendency of the illuminant colors is well captured by the localized estimates.

4.4.3 CRF-based Multi-Illuminant Estimation

The method that was presented in the previous section suffers from several drawbacks. First, only one dominant illuminant can be estimated per region, and spatial context is limited to the segmentation of the estimates. A more elegant formulation could estimate a set of scene illuminants, and a per-region contribution to each of these. Likewise, the localization problem can also be addressed with respect to these illuminant candidates. Finally, the lack of ground truth for arbitrary real-world scenes under non-uniform illumination makes it difficult to judge the real performance.

The algorithm that is proposed in this section addresses these concerns. We consider the estimation of multiple illuminants as an energy minimization problem on local estimates. The proposed algorithm to solve this task jointly estimates the colors of the illuminants and their spatial distribution. To quantify the performance of the method, we used our multi-illuminant dataset that was proposed in Sec. 4.3.2.

4.4.3.1 Methodology

We propose to solve the multiple illuminant estimation problem by using a Conditional Random Field (CRF) framework. The nodes in the graph represent patches, the labels correspond to illuminant colors, and the edges connect neighboring patches. In such a representation local illuminant estimation becomes equivalent to finding the maximum a posteriori (MAP) labelling of the CRF. Such a framework facilitates both the local computation of illuminant color, as well as the incorporation of spatial information about the distribution of illuminants.

More specifically, a conditional random field can be viewed as an undirected graphical model, globally conditioned on observations. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes representing the n patches and $\mathcal{E}$ is the set of

edges connecting neighboring patches. We define a discrete random field $\mathcal{X}$ over the graph $\mathcal{G}$. Each node $i \in \mathcal{V}$ is associated with a random variable in $\mathcal{X}$, which can take on a value $\boldsymbol{u}_i$ from the illuminant-color label set $\mathcal{L} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_k\}$. At each node $i \in \mathcal{V}$ we also have a local observation $\mathcal{F}_i$, which is the set of (R, G, B) values of all the pixels belonging to the corresponding patch together with their spatial locations. The probability $P(\mathcal{X} = \breve{\boldsymbol{u}}|\mathcal{F})$ of a particular labelling $\breve{\boldsymbol{u}} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$ conditioned on the observations $\mathcal{F}$ of the entire image will be denoted as $P(\breve{\boldsymbol{u}}|\mathcal{F})$. Then according to the Hammersley-Clifford theorem

$$P\left(\breve{\boldsymbol{u}}|\mathcal{F}\right) \propto \exp\left(-\sum_{\mathcal{C}\in\mathcal{C}_{\text{All}}} \xi^{\mathcal{C}}(\breve{\boldsymbol{u}}^{\mathcal{C}}|\mathcal{F})\right) , \tag{4.46}$$

where $\xi^{\mathcal{C}}(\breve{\boldsymbol{u}}^{\mathcal{C}}|\mathcal{F})$ are potential functions defined over the observations $\mathcal{F}$ and the variables $\breve{\boldsymbol{u}}^{\mathcal{C}} = \{\boldsymbol{u}_i, i \in \mathcal{C}\}$ belonging to clique $\mathcal{C}$. A clique $\mathcal{C}$ is a set of random variables which are conditionally dependent on each other and $\mathcal{C}_{\text{All}}$ is the set of all cliques in $\mathcal{G}$. Finding the labelling $\breve{\boldsymbol{u}}^*$ with the maximum a posteriori (MAP) probability is then equal to

$$\breve{\boldsymbol{u}}^* = \underset{\breve{\boldsymbol{u}}\in\mathcal{U}}{\operatorname{argmax}}\, P\left(\breve{\boldsymbol{u}}\,|\mathcal{F}\right) = \underset{\breve{\boldsymbol{u}}\in\mathcal{U}}{\operatorname{argmin}}\, E(\breve{\boldsymbol{u}}|\mathcal{F}) \tag{4.47}$$

where $\mathcal{U}$ is the set of all possible labellings on $\mathcal{X}$ and $E(\breve{\boldsymbol{u}}|\mathcal{F})$ is the corresponding Gibbs energy defined as

$$E(\breve{\boldsymbol{u}}|\mathcal{F}) = \sum_{\mathcal{C}\in\mathcal{C}_{\text{All}}} \xi^{\mathcal{C}}(\breve{\boldsymbol{u}}^{\mathcal{C}}|\mathcal{F}) \tag{4.48}$$

Hence, computing the MAP labelling is equal to finding the labelling which minimizes the energy $E(\mathbf{x}|\mathcal{F})$. In our case, this means that obtaining the MAP assignment of illuminants to patches can be accomplished by finding that assignment which minimizes the corresponding Gibbs energy. Considering only up to pairwise clique potentials, the energy function becomes

$$E(\breve{\boldsymbol{u}}|\mathcal{F}) = \sum_{i\in\mathcal{V}} \phi\left(\boldsymbol{u}_i|\mathcal{F}_i\right) + w_{\text{PW}} \sum_{(i,j)\in\mathcal{E}} \psi\left((\boldsymbol{u}_i, \boldsymbol{u}_j)|(\mathcal{F}_i, \mathcal{F}_j)\right) \tag{4.49}$$

where ϕ denotes the unary potential and ψ the pairwise potential. The unary potentials ϕ penalize the discrepancy between the observations, i.e. the colors of the pixels in a patch $\mathcal{F}_i$, and the solution, i.e. the illuminant-color label assigned to the patch. The pairwise potentials ψ provide a definition of smoothness by penalizing changes in the labels of neighboring patches. Then the constant $w_{\text{PW}} > 0$ controls the balance between smoothness and data fit. In the next section we propose several unary potentials which allow us to represent several well-known illumination estimation algorithms as CRFs.

4.4.3.2 Unary Potentials

We show that by picking one particular unary potential we can write several existing color constancy methods as an error minimization problem. When we use the

pairwise potential to enforce a single label for all patches, we obtain the same result as with traditional single illuminant estimation methods. Reducing the influence of the pairwise potential results in multi-illuminant estimates for the scene. To prevent overfitting to the local data we propose also several adaptations to the unary potentials.

Statistics-based Illuminant Color Estimates Illuminant estimation methods are generally evaluated based on the angular error, which for two normalized illuminants (typically the estimated illuminant $\tilde{\boldsymbol{e}}$ and true illuminant $\boldsymbol{e}$) is given by

$$\varphi(\tilde{\boldsymbol{e}}, \boldsymbol{e}) = \arccos((\tilde{\boldsymbol{e}})^T \boldsymbol{e}). \tag{4.50}$$

For obtaining illuminant estimates on local patches, we use the generalized Gray World estimates by van de Weijer *et al.* [Weij 07a], as introduced in Eqn. 4.11 on page 66. In this section, we adopt a slightly different, but equivalent notation. One Gray World estimate in patch i is denoted as

$${}^{i}\tilde{\boldsymbol{e}}^{\mathrm{GW}} \approx \sqrt[\tau_{\mathrm{GW}}]{\sum_{\boldsymbol{p}(\boldsymbol{x})\in\mathcal{F}_i} \left| \frac{\partial^{n_{\mathrm{GW}}} {}_{\sigma}\boldsymbol{p}(\boldsymbol{x})}{\partial \boldsymbol{x}^{n_{\mathrm{GW}}}} \right|^{\tau_{\mathrm{GW}}}} \quad . \tag{4.51}$$

In this formulation, the generalized Gray World algorithm is applied locally to a patch $\mathcal{F}_i$. As in Eqn. 4.11 on page 66, n_{GW} denotes the order of differentiation, τ_{GW} the Minkovski norm, and σ the standard deviation of a Gaussian smoothing kernel prior to the computation.

We now define the statistics-based unary potential ϕ^s, which defines the cost for patch i to take on illuminant $\boldsymbol{u}_i$, as

$$\phi^s(\boldsymbol{u}_i|\mathcal{F}_i) = w_{\mathcal{F}_i} t^{w_{\mathrm{r}}} \left(\varphi(\boldsymbol{u}_i, {}^{i}\tilde{\boldsymbol{e}}^{\mathrm{GW}})\right) \quad , \tag{4.52}$$

where $w_{\mathcal{F}_i}$is a scalar weight per patch, and $t^{w_{\mathrm{r}}}$ denotes an error norm to the power of w_{r}. For example, choosing $t(e) = e^2$ yields the least squares error.

Choosing as an error norm $t(e) = 1 - \cos(e)$ and for $w_{\mathcal{F}_i}$a weight that is proportional to the magnitude of the generalized Gray World estimate, we obtain

$$\phi^s(\boldsymbol{u}_i|\mathcal{F}_i) = \left\| \sqrt[\tau_{\mathrm{GW}}]{\sum_{\boldsymbol{p}(\boldsymbol{x})\in\mathcal{F}_i} \left| \frac{\partial^{n_{\mathrm{GW}}} {}_{\sigma}\boldsymbol{p}(\boldsymbol{x})}{\partial \boldsymbol{x}^{n_{\mathrm{GW}}}} \right|^{\tau_{\mathrm{GW}}}} \right\|_2 \left(1 - \cos\left(\varphi\left(\boldsymbol{u}_i, {}^{i}\tilde{\boldsymbol{e}}^{\mathrm{GW}}\right)\right)\right). \tag{4.53}$$

If we choose $n_{\mathrm{GW}} = 1$ and $\tau_{\mathrm{GW}} = 1$, then minimizing Eqn. 4.49 with this unary potential results in the standard gray-edge algorithm by van de Weijer *et al.* [Weij 07a].

We proceed by proposing several adaptations to the unary potential to optimize it for multi-illuminant estimation. If we increase the influence of the pairwise potential, by choosing a large w_{PW} in Eqn. 4.49, we can enforce the whole image to have the same label, and therefore the same estimate for the illuminant. If we look at the other extreme where we pick $w_{\mathrm{PW}} = 0$ every patch would take on the label of the illuminant which is closest (in a angular error sense) to its local estimate. However, the local estimates of the statistical color constancy algorithms are very noisy and in

general this will lead to unsatisfying results. This can be countered by choosing an intermediate w_{PW} (by means of cross validation), that enforces multiple neighboring patches to take on the same label, and thereby reducing the noise of the statistical estimate. We will look at two adaptations to the unary potential which improve robustness with respect to noisy statistical measurements.

Robust error norm: To reduce the influence of outliers on the energy, we found the usage of a robust error norm indispensable. Throughout this algorithm we use the following error norm

$$t\left(\epsilon_{\mathrm{ang}}\right) = \mathrm{robust}_{\sigma}\left(\epsilon_{\mathrm{ang}}\right) = 1 - \exp\left(-\frac{\epsilon_{\mathrm{ang}}^2}{2\sigma^2}\right) \tag{4.54}$$

Its main effect is that outliers have less influence on the overall energy.

Uneven color balance: Statistical methods are known to be biased towards large segments of the same color. To counter this we propose the following adaptation:

$$\phi^s(\boldsymbol{u}_i|\mathcal{F}_i) = \left(\left\| \sqrt[\tau_{\mathrm{GW}}]{\sum_{\boldsymbol{p}(\boldsymbol{x})\in\mathcal{F}_i} \left|\frac{\partial^{n_{\mathrm{GW}}}{}_{\sigma}\boldsymbol{p}(\boldsymbol{x})}{\partial \boldsymbol{x}^{n_{\mathrm{GW}}}}\right|^{\tau_{\mathrm{GW}}}} \right\|_2\right)^{w_{\mathrm{Damp}}} \mathrm{robust}_{\sigma}\left(\varphi\left(\mathbf{i}_i, \mathbf{x}_i\right)\right). \tag{4.55}$$

The parameter w_{Damp} allows to dampen the results of uneven color balance in the image. Consider the standard gray-world assumption ($\tau_{\mathrm{GW}} = 1$ and $n_{\mathrm{GW}} = 0$). If we then choose $q = 0$, the unary potential is equal to

$$\phi^s\left(\boldsymbol{u}_i|\mathcal{F}_i\right) = \left(1 - \cos\left(\varphi\left(\boldsymbol{u}_i, {}^i\tilde{\boldsymbol{e}}^{\mathrm{GW}}\right)\right), \right), \tag{4.56}$$

which is one of the more popular implementations of gray-world. Here, instead of considering one value per pixel, one value for each patch is chosen. This was proposed by Barnard *et al.* [Barn 02b] to counter the dominance of large uniformly colored regions in images. In the results we consider $q \in \left\{0, \frac{1}{2}, 1\right\}$.

Physics-based color constancy We also make use of physics-based estimates, as presented in Sec. 4.4.2. Per patch i, we obtain an illuminant estimate ${}^i\tilde{\boldsymbol{e}}^{\mathrm{IIC}}$ To find regions with partial specularities, we used the specularity segmentation by Lehmann and Palm [Lehm 01]. It selects bright, achromatic pixels in the image, guided by two thresholds τ_b and τ_s. In our implementation, we set $\tau_b = 0.2$ and $\tau_s = 0.8$.

A patch is considered specular if the sum of intensities of its specular pixels exceeds a threshold τ_{sp}. In this case, we set $w_{\mathrm{sp}} = 1$, otherwise $w_{\mathrm{sp}} = 0$. We use this weight in the physics-based unary potential:

$$\phi^p\left(\boldsymbol{u}_i|\mathcal{F}_i\right) = w_{\mathrm{sp}}\mathrm{robust}_{\sigma}(\varphi(\boldsymbol{u}_i, {}^i\tilde{\boldsymbol{e}}^{\mathrm{IIC}})) \tag{4.57}$$

We use the same robust error norm as for statistical methods as given by Eqn. 4.54.

Combining Statistical and Physics based Illuminant Estimation The both statistical and physics-based illuminant estimation can be incorporated in a CRF framework using different unary potentials. An advantage of defining each method

as an energy minimization problem is that there is a natural way for combining them into a single color constancy method by defining the local potential as

$$\phi^s\left(\boldsymbol{u}_i|\mathcal{F}_i\right) = (1 - w_{\mathrm{UW}})\phi^s\left(\boldsymbol{u}_i|\mathcal{F}_i\right) + w_{\mathrm{UW}}\phi^p\left(\boldsymbol{u}_i|\mathcal{F}_i\right). \tag{4.58}$$

where w_{UW} is weighting the importance of the physics-based unary potential versus the statistical-based unary potential. Minimizing this energy will combine information from statistical cues as well as specularities into the final local illuminant estimate.

Constrained Illuminant Estimation Constraint illuminant estimation methods have been popular because they allow to incorporate prior knowledge about the illuminants. Several methods have been proposed which constrain the illuminant set to be on the Planckian locus [Finl 06]. Incorporating such constraints is straightforward in our framework. The constraints can be enforced on the illuminant label set $\mathcal{L}$. Here, we use a simple constraint were we exclude illuminants which are too saturated, such that

$$\left\{\forall i|\varphi\left(\boldsymbol{l}_i, \tfrac{1}{\sqrt{3}}\begin{pmatrix}1 & 1 & 1\end{pmatrix}^{\mathrm{T}}\right) < \tau_{\mathrm{Sat}}\right\} \ . \tag{4.59}$$

As a second constraint on the illuminants, we use the fact that in the majority of the multi-illuminant scenes only two illuminants are present. Given a pair of labels $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$, the optimal labeling $\breve{\boldsymbol{u}}^*(i,j)$ for the observation $\mathcal{F}$ is determined with

$$\breve{\boldsymbol{u}}^*(i,j) = \underset{\breve{\boldsymbol{u}}\in\mathcal{L}^{i,j}}{\operatorname{argmin}} E\left(\breve{\boldsymbol{u}}|\mathcal{F}\right), \tag{4.60}$$

where $\mathcal{L}^{i,j}$ is the set of all possible labellings on $\mathcal{X}$ restricting the illuminants to $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$. The two illuminant constraint is enforced by finding those two illuminants which minimize the energy function. Thus, the selected illuminants are computed with

$$\breve{\boldsymbol{u}} = \underset{(\boldsymbol{l}_i,\boldsymbol{l}_j)\in\mathcal{L}^2}{\operatorname{argmin}} \left(E(\breve{\boldsymbol{u}}^*(i,j)|\mathcal{F})\right) \tag{4.61}$$

Note that this also allows for single illuminant estimation in the case that $i = j$.

4.4.3.3 Pairwise Potential

The purpose of the pairwise potential functions, $\psi\left((\boldsymbol{u}_i, \boldsymbol{u}_j)|(\mathcal{F}_i, \mathcal{F}_j)\right)$ is to ensure, when appropriate, the smooth transition of labels in neighboring vertices. Similar to Boykov *et al.* [Boyk 98], we consider pairwise potentials that resemble a well. In MRFs, especially as described in [Boyk 98], $\psi(\boldsymbol{u}_i, \boldsymbol{u}_j) = u(1-\delta_{ij})$, where u is the well "depth" and the function $(1-\delta_{ij})$ controls the shape of the well. Here, u is defined as a constant and is based on the unit impulse function, $\delta(i,j) = \delta(\boldsymbol{u}_i - \boldsymbol{u}_j)$, to define the well shape.

In a CRF (see also [Kohl 09]) the "depth" depends on the observations $h(\mathcal{F}_i, \mathcal{F}_j)(1-\delta(i,j))$. Thus, our pairwise potential function has the form

$$\psi\left((\boldsymbol{u}_i, \boldsymbol{u}_j)|(\mathcal{F}_i, \mathcal{F}_j)\right) = h(\mathcal{F}_i, \mathcal{F}_j)(1 - \delta(i,j)) \ . \tag{4.62}$$

We also propose the use of a smoother well function which permits small deviations in illuminant colors between neighboring patches. Thus, our well is defined as

$$(1 - \delta(i,j)) = (1 - \cos^{w_{PWsh}}(\varphi(\boldsymbol{u}_i, \boldsymbol{u}_j))) \ , \tag{4.63}$$

where w_{PWsh} controls the sharpness of the impulse-like function.

If two neighboring labels are distinct, then there are two possibilities. It can be that the two patches, though spatially close, are illuminated by distinct illuminants, in which case, we should allow for a transition in labels and not significantly penalize the difference in their values. It may, however, be the case that we assigned an erroneous label and the two patches are illuminated by the same illuminant. The depth function $h(\mathcal{F}_i, \mathcal{F}_j)$ attempts to distinguish between these two cases.

In this work, we use the insight of Logvinenko *et al.* [Logv 05] that the shape of an edge (curvature, fuzziness and closedness) conveys discriminatory information about illuminant versus material edges. Influenced by this idea, we use the length of the border between two adjacent patches as an indicator of whether the patches should be sharing incident illumination, i. e.,

$$h(\mathcal{F}_i, \mathcal{F}_j) = \text{length}(\text{boundary}(\mathcal{F}_i, \mathcal{F}_j)) \ . \tag{4.64}$$

Longer boundaries imply that the distinct color of the patches is due to differences in material and, hence, the illuminant labels of the adjacent patches should be similar.

However, the proposed framework is general and allows the incorporation and/or combination of multiple methods that can provide information on the discontinuity of illuminants in the scene. For example, one could employ the Retinex [Land 77] heuristic that illumination is expected to vary slowly, thus large changes in surface reflectance are due to differences in material. A Retinex-inspired depth function could then be $h(\mathcal{F}_i, \mathcal{F}_j) = \exp\left(-\beta_R \|\bar{\mathcal{F}}_i - \bar{\mathcal{F}}_j\|^2\right)$, where $\bar{\mathcal{F}}_i$ is the average $(R,G,B)^{\mathrm{T}}$ value in patch p_i. Yet another option is to employ photometric quasi-invariants [Weij 05] which help distinguish between shading edges and material edges. Note that if multiple cues for illuminant transitions are available, the different functions h can be directly combined via summation.

4.4.3.4 MIRF: Overall algorithm

In this section, we present the full algorithm for multi-illuminant estimation. We call it Multi-Illuminant Random Field (MIRF). In the first step we divide the image into subregions or patches. There are several ways used in the literature for obtaining adequate patches. In contrast to the work in Sec. 4.4.1 and Sec. 4.4.2, we decided against using superpixels because they are more likely to follow object boundaries rather than subtle illuminant changes. Hence, a grid provides more diverse patch content, and thus more information for the statistical estimators.

Next, we obtain a local illuminant estimate for each patch using the Eqn. 4.51 and the per-patch version of our physics-based estimator as presented in Sec. 4.4.2. To add more robustness, these illuminants are then clustered to k illuminants based on their chroma. Additionally, we add a single illuminant estimate $\tilde{\boldsymbol{e}}^{\mathrm{GW}}$ to the illuminant set by applying Eqn. 4.11 on the whole image. To reduce the computational cost, we reduce the number of labels by averaging the ones whose angular distance is less

than half a degree. We calculate the unary potentials using equation Eqn. 4.55 and Eqn. 4.57.

In the next step, for every pair of labels we solve the energy minimization problem of Eqn. 4.60. We use the matlab implementation by Bagon [Bago 06] of the algorithm by Boykov *et al.* [Boyk 04, Boyk 01, Kolm 04] as an efficient approximate algorithm for multi-label energy minimization tasks. The Gibbs energy in Eqn. 4.49 can directly be translated to a multiway graph cut problem. Such a formulation has a globally optimal solution for the case of two illuminants, and an approximately optimal solution for three or more illuminants. Within the graph cut framework, different solution strategies can be chosen. In our case, we used the so-called alpha-expansion method. As output, we obtain the proper labeling (the assignment of the labels to patches) along with the residual error, i. e. the estimation error for the whole image (see Eqn. 4.60). The pair of two labels which minimizes the error is then chosen, which is a solution to Eqn. 4.61.

Finally, the label colors are assigned to their respective patches, and the estimated illumination map M is generated. In the last step of the algorithm, a Gaussian smoothing filter with standard deviation σ_p is applied to M as a post processing step in order to reduce artifacts created by the patch boundaries. The methodology is compactly presented in Algorithm 11.

Algorithm 2 Method

1: Apply an $a \times a$ grid on the image to divide it to a set of patches (subregions) $\{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n\}$
2: Extract the local illuminant colors for each patch.
3: Cluster the illuminants using the K-means algorithm to obtain n_k cluster centers. Add the single estimate $\tilde{\boldsymbol{e}}^{\mathrm{GW}}$.
4: Reduce the number of labels by averaging two estimates whose angular distance is less than .5 degrees.
5: Calculate the unary potentials using Eqn. 4.55 and Eqn. 4.57.
6: **for all** pairs $\boldsymbol{l}_i$ and $\boldsymbol{l}_j \in \mathcal{L}$ **do**
7: Calculate $\breve{\boldsymbol{u}}^* \boldsymbol{u}^*(i,j)$ using Eqn. 4.60.
8: **end for**
9: Find the pair of illuminants $\tilde{\mathcal{L}}$ which yields the lowest error when assigned to the image patches.
10: Back project $\tilde{\mathcal{L}}$ and create an illumination map $\mathbf{I_M}$.
11: Post processing: Apply Gaussian smoothing on $\mathbf{I_M}$ to fade out the artificial edges of the grid (artifacts).

4.4.3.5 Evaluation

In this section we compare the performance of the proposed method MIRF to several other approaches. We evaluate our results on three datasets, our proposed multi-illuminant dataset of laboratory images, our proposed multi-illuminant on real-world images (for both, see Sec. 4.3.2), and the outdoor dataset that was part of the work by Gijsenij *et al.* [Gijs 12b]. As an error metric, we obtain an error per image by

computing the mean of the pixelwise angular error (see Eqn. 4.10 on page 62) between the estimated illuminant color and the ground truth maps. Pixels that were too dark (i. e., for our 12 bit images, pixel intensities below 50) have been excluded from evaluation due to their relatively large noise component. Over these per-image errors, we computed the median and mean errors per dataset.

As a baseline, we computed results for a number of established algorithms that address color constancy under uniform illumination. So far, little prior work exists for estimating non-uniform illumination. We implemented the recent method by Gijsenij *et al.* [Gijs 12b], as it showed very competitive performance in a number of experiments.

Both, the method by Gijsenij *et al.* [Gijs 12b] and MIRF use as input illuminant estimates with small spatial support. Such illuminant estimates can be obtained from different estimators. We chose to use gray world ("GW"), which can be obtained from Eqn. 4.11 on page Eqn. 4.11 by using the parameters $n = 0$, $m = 1, \sigma_{\mathrm{GW}} = 0$, white patch ("WP", with $n = 0$, $m = \infty$, $\sigma_{\mathrm{GW}} = 0$), first order gray-edge ("GE1", with $n = 1$, $m = 1$, $\sigma_{\mathrm{GW}} = 1$) and second-order gray edge ("GE2", with $n = 2$, $m = 1$, $\sigma_{\mathrm{GW}} = 1$). Additionally, we use the physics-based estimator, as presented in Sec. 4.4.2, denoted as "IEbV" (derived from "illuminant estimation by voting"). We used these base estimators for comparing the performance of the three families of methods as described above. Additionally, we provide the error if the illuminant color is assumed to be already perfectly balanced to white. The "do nothing" ("DN") estimator shows these results. For the evaluation on our proposed dataset, we resampled the images to 20% of their original size to reduce the computational load.

Parameters A number of parameters have been fixed for the evaluation of MIRF. As patches we used a rectangular grid with cells of 20×20 pixels for the downscaled version of our proposed dataset, and cells of 10×10 pixels for the outdoor images by Gijsenij *et al.* [Gijs 12b]. In both cases, this corresponds to a cell size of about 15×20 pixels. The number of cluster centers k for the k-means algorithm has been set to the square root of the number of grid cells. To obtain the physics-based estimates, we set the Lehmann and Palm parameters $t_b = 0.2$ and $t_s = 0.8$, and the overall specularity threshold $t_{\mathrm{sp}} = 10$ for pixel intensities between 0 and 1. The subgrid size for single physics-based estimates was 20×20 pixels with a step size of 10 pixels[14], as proposed in [Ries 11]. The settings for the CRF framework were as follows: the saturation threshold for illuminant labels τ_{Sat} (see Eqn. 4.59) is set to 15°. The parameter σ in Eqn. 4.54 for robust thresholding on the unary potentials has been set to 2.5°. Finally, the standard deviation for the Gaussian smoothing on the reprojected illuminant labels has been set to 10.

Besides these globally fixed parameters, we determined three parameters via two-fold cross validation on each dataset. These were the weighting between unary and pairwise potentials w_{PW} (see Eqn. 4.49), the power $wDamp$ (see Eqn. 4.55) for computing the unary potentials, and finally, if data costs from different estimators are

[14]Note that for the downscaled images from our dataset, this leads to only one estimate per patch, i. e. the voting part is effectively clamped off. However, if the method is applied on larger images (or patches, respectively), the histogram voting is used.

	Single-illuminant		Gijsenij *et al.*		MIRF	
	Mean	Median	Mean	Median	Mean	Median
Do nothing	10.6°	10.5°	-	-	-	-
Gray World	3.2°	2.9°	6.4°	5.9°	3.1° (-3%)	2.8° (-3%)
White patch Retinex	7.8°	7.6°	5.1°	4.2°	3.0°(-41%)	2.8°(-33%)
1st order Gray Edge	3.1°	2.8°	4.8°	4.2°	2.7°(-13%)	**2.6**° (-7%)
2nd order Gray Edge	3.2°	2.9°	5.9°	5.7°	**2.6**°(-19%)	**2.6**°(-10%)
IEbV	8.5°	8.3°	-	-	4.5°(-47%)	3.0°(-64%)

Table 4.10: Comparative results on the proposed laboratory dataset.

combined, w_{UW} (see Eqn. 4.58) for the relative influence of physics-based and statistical estimators.

Comparing Single- and Multi-illuminant Methods In Tab. 4.10, we present the mean and median errors on our proposed laboratory dataset. In the column "single-illuminant", these results are based on a single global illuminant estimate. The columns "Gijsenij *et al.*" and "MIRF" report results for the multi-illuminant methods by Gijsenij *et al.* [Gijs 12b] and our proposed algorithm "Multi-Illuminant Random Field". It turns out, that some single-illuminant estimators, namely gray world, first and second order gray edge, already perform relatively well on our dataset. This comes from the fact that in many cases, the ground truth illuminant colors are not very distant from each other. Thus, the overall error can be small, even if only one of the two illuminants (or a color in between both illuminants) is reported as global estimate. However, in all cases, MIRF improves over these estimates. The physics-based estimates for IEbV yield a considerably weaker performance in the mean error, which might be due to the fact that the individual patches are relatively small, such that the voting becomes ineffective. The method by Gijsenij *et al.* performed surprisingly weak, even worse than the single-illuminant estimators. We investigated this case more closely. It turned out that relatively often, weak candidate estimates are selected by the method, which penalizes the overall algorithm. MIRF avoids this particular problem, as the remaining energy from the energy minimization is used as a criterion for the quality of a solution. In the next paragraph, we excluded this source of error, to directly compare the performance for determining only the distribution of illuminants.

Table 4.11 shows a similar tendency in the results, but this time on our proposed real-world dataset. Note that the overall errors are higher, which is mainly due to the fact that the images have been perceptually enhanced, such that the overall spread of the colors in the image is larger. The largest gain is obtained using localized estimates of the physics-based estimates. This performance gain comes mostly from the robust error metric, which suppresses gross outliers in the physics-based estimates.

In Tab. 4.12, we report results on the outdoor dataset by Gijsenij *et al.* [Gijs 12b]. Note that the reported numbers for the method by Gijsenij *et al.* deviate from what the authors reported in their paper. When investigating their method, we noted that the evaluation in [Gijs 12b] was conducted on the non-gamma-corrected images[15].

[15]Without gamma correction, we obtain the same numbers as reported in [Gijs 12b].

	Single-illuminant		Gijsenij *et al.*		MIRF	
	Mean	Median	Mean	Median	Mean	Median
Do nothing	8.8°	8.9°	-	-	-	-
Gray World	5.2°	4.2°	4.4°	4.3°	3.7°(-16%)	3.4°(-19%)
White patch Retinex	6.8°	5.6°	4.2°	3.8°	**4.1**° (-2%)	**3.3**°(-13%)
1st order Gray Edge	5.3°	3.9°	9.1°	9.2°	4.0°(-25%)	3.4°(-13%)
2nd order Gray Edge	6.0°	4.7°	12.4°	12.4°	4.9°(-18%)	4.5° (-4%)
IEbV	6.0°	4.9°	-	-	5.6° (-7%)	4.3°(-12%)

Table 4.11: Comparative results on the perceptually enhanced real-world images.

	Single-illuminant		Gijsenij *et al.*		MIRF	
	Mean	Median	Mean	Median	Mean	Median
Do nothing	4.4°	3.6°	-	-	-	-
Gray World	15.0°	13.8°	12.2°	13.8°	10.0°(-18%)	10.1°(-27%)
White patch Retinex	10.3°	11.3°	10.0°	8.4°	7.7°(-23%)	6.4°(-24%)
1st order Gray Edge	10.1°	10.1°	8.5°	7.6°	**7.1**°(-16%)	**4.7**°(-38%)
2nd order Gray Edge	8.7°	8.5°	8.1°	7.4°	7.2°(-11%)	5.0°(-32%)
IEbV	10.0°	7.3°	-	-	9.3° (-7%)	7.3° (-0%)

Table 4.12: Evaluation results on the gamma corrected version of the outdoor dataset by Gijsenij *et al.* [Gijs 12b]

In our implementation, we performed gamma correction on the input images, as it was also originally intended by [Gijs 12b]. The overall errors are higher than in the previous two experiments. First, the images of this dataset are relatively small snippets, consisting mostly of two relatively homogeneous regions in sunlight and shadow. Thus, the underlying localized illuminant color estimators have to estimate on relatively uninformative input. Note that we did not evaluate on the laboratory data by Gijsenij *et al.*, as we found upon manual inspection that the ground truth for these images is not very reliable.

Benchmarking Separate Components of the Algorithm Estimating multiple illuminants can be considered as two interleaved tasks, namely estimating the illuminant colors and their spatial distribution. The recovery of the spatial distribution was not required for single-illuminant estimators. Hence, we empirically investigated the capability of finding the proper spatial distribution, by providing the methods in this experiment the ground truth illuminant colors. The results on our laboratory dataset are shown in Tab. 4.13. In the left two columns, it can be seen that the performance of the method by Gijsenij *et al.* greatly improved, compared to Tab. 4.10. Thus, we conclude that the selection of the correct illuminant color is one of the major challenges in the method of Gijsenij *et al.*. In the right columns, we show the performance of the proposed method. The best performing method is first order gray edge, with a median error of 1.7°. This shows that the spatial distribution of the illuminants is well approximated by our proposed framework.

	Gijsenij		MIRF	
	Mean	Median	Mean	Median
Gray World	2.4°	2.3°	2.3°	2.3°
White patch Retinex	2.2°	2.1°	2.0°	1.9°
1st order Gray Edge	2.1°	2.0°	**1.8°**	**1.7°**
2nd order Gray Edge	2.2°	2.1°	1.9°	1.8°

Table 4.13: Performance on our laboratory data for recovering the spatial distribution. The ground truth illuminant colors are provided to the methods.

In another experiment, we investigated the relative gain of the various improvements we have introduced (see Tab. 4.14). As an example illuminant estimation algorithm, we used the gray world ("GW") estimator. If we remove the constraint of two illuminants and allow an arbitrary number of illuminants, the error increases significantly on our two datasets. Similarly, the robust error norm (see Eqn. 4.54) yields an important performance gain on both our datasets. Removing the parameter w_{Damp} which counters uneven color balances only affects results on the Gijsenij dataset. Finally, removing the saturation constraint τ_{Sat} on the illuminants results in a performance drop on all datasets.

Combination of Statistical and Physics-based Estimates Table 4.15 demonstrates another benefit of the framework. By defining the unary potentials as a weighted sum of the physics-based and the statistical unary potentials, we are able to combine cues from multiple methods in a natural way. To determine the parameters, we performed a full cross-validation over w_{PW}, $wDamp$ and w_{UW}. It turns out, that a combination of physics-based and statistical estimates can indeed further improve the results (confer Tab. 4.15 (left) and Tab. 4.10), in particular for the white patch (WP) and first order Gray Edge (GE1) estimates. On the other hand, the performance of the combination of IEbV with GE2 (second order Gray World) slightly dropped, thus there is no guarantee that a combination of the unary potentials brings a performance gain.

The right columns of Table 4.15 show the performance on our proposed real-world dataset. It is interesting to note that the impact of combined unary potentials on the overall performance is quite different from the experiments on the laboratory data. Here, the majority of the results is slightly worse than the results reported in

	Laboratory data		Real-world data		*et al.* Gijsenij	
	Mean	Median	Mean	Median	Mean	Median
MIRF	3.1°	2.8°	3.7°	3.4°	10.0°	10.1°
all lights	4.6°	4.0°	4.2°	4.0°	10.0°	10.2°
without Eqn. 4.54	3.9°	3.7°	4.3°	4.0°	10.1°	10.1°
$w_{\text{Damp}} = 1$	3.0°	2.8°	3.6°	3.3°	10.7°	10.3°
without τ_{Sat}	3.6°	3.3°	4.6°	3.2°	11.2°	10.1°

Table 4.14: Gray world results for different configurations of MIRF for each dataset.

	Laboratory data		Real-world data	
Combination variant	Mean	Median	Mean	Median
IEbV-GW	3.0°	2.8°	4.2°	4.3°
IEbV-WP	2.6°	2.5°	4.0°	3.4°
IEbV-GE1	2.6°	2.4°	4.5°	4.2°
IEbV-GE2	2.8°	2.8°	4.7°	3.9°

Table 4.15: Combination of physics-based and statistical methods on our laboratory data.

Tab. 4.11. This behavior, however, is not consistent. For instance, the mean error of IEbV-WP lies slightly below the reported error in Tab. 4.11, similarly the median error for IEbV-GE2. From these results, we conclude that the framework is general enough to allow the straightforward integration of multiple cues. However, whether such a combination indeed brings the desired performance gain has to be investigated on a case-by-case basis.

Automatic White Balance Example results for automatic white balancing are shown in Fig. 4.24. All images are contrast enhanced for improved visualization. In the top row, from left to right, the input scenes "toys", "lion", "camera", and "detergents" are presented. The second row shows perfectly white balanced output using the computed ground truth. The third row shows white balancing results for a single global Gray World estimator. The resulting images suffer from a color cast, as both illuminant colors in the scene are corrected with only one estimate. Using the same estimator within the framework by Gijsenij *et al.* [Gijs 12b] (fourth row) clearly improves over the global estimator. However, the images look more grayish and with faded colors as the local estimations were not able to fully separate the effect of illumination from the object color. Also the "lion" is more reddish on the right side. Finally, in the last row, the output of the proposed MIRF is shown. In this case, the improved performance results from the improvement in the selection of the illuminant color, thus the global color cast is removed. Some inaccuracies in the estimation of the spatial distribution of the illuminants may lead to local color casts (e.g., several bluish "blobs" overlay considerable regions of the "camera" image). However, the overall performance of MIRF is in general quite solid, as demonstrated in the "toys" and "detergents" images.

In summary, we proposed an extensible framework for estimating and localizing the influence of multiple illuminants. The results are very encouraging, and outperform the current state-of-the-art. The framework offers a number of opportunities to integrate existing methodologies and future insights. For instance, the depth function h in the pairwise potentials is currently barely used. Thus, we expect that the framework can still be fine-tuned, to further improve the estimation of non-uniform illumination.

Figure 4.24: Examples for automated white balancing (WB). From top to bottom the rows present: original image from the camera, the WB images using the ground truth, global Gray World, Gijsenij *et al.* [Gijs 12b], and MIRF. Note that the images are enhanced to sRGB for visualization. The captions on the images denote their estimation error.

Chapter 5

Illumination Cues in Image Forensics

Color and direction of the scene illumination adhere to physical laws. With complete knowledge about the scene objects and the scene geometry, all brightness variations and color shifts can be directly explained. From a forensic viewpoint, this opens the door to physics-based cues for exposing image manipulations. When an image is spliced, i. e. a part of the image stems from another source, it is difficult to precisely adjust the snippet to the illumination situation of the host image. In this chapter, we propose methods to detect such inconsistencies.

We characterize algorithms that operate on illumination effects as "high-level" methods, as opposed to the statistics-driven approaches in the chapters 2 and 3. Both classes of methods complement each other: Statistical methods exploit properties of the digital representation of an image. High-level methods, in contrast, are only minimally dependent on the host medium. Thus, high-level methods could equally well be applied on analogue photographs or maybe even on high-quality printouts, where statistical methods lack the pixel information. This generality comes at the expense that high-level methods are more difficult to apply. Typically, user input is required, and often enough, a human expert must assess the output of the methods. Thus, given the current state-of-the-art, a full automation of these approaches is in many cases not possible.

Computational assessment of high-level forensic cues is also interesting as a support to human experts in their manual inspection. Farid and Bravo [Fari 10b] and Ostrovsky *et al.* [Ostr 05] point out that the human visual system performs relatively poorly in judging lighting and shadow inconsistencies in photographs.

In this chapter, we first review related work on high-level forensic cues in Sec. 5.1. Then, we propose a physics-based method for finding inconsistencies in the illumination color in Sec. 5.2. Finally, in Sec. 5.3, we propose a practical extension to the work by Johnson and Farid [John 07a] to extend the practical applicability of illumination direction as a forensic cue.

The work in Sec. 5.2.1 was entirely done by me. Tiago Carvalho did most of the work in Sec. 5.2.2, I contributed some ideas to the general direction of the work and on the evaluation (which is still work in progress). The implementation in Sec. 5.3 is joint work of Dominik Schuldhaus, Szabolcz Vita, Sven Pfaller and me. The ideas are contributed by me. Dominik wrote a first prototype, Szabolcz expanded the

code and transferred it to C++, and Sven eventually investigated intrinsic image decomposition, and implemented the intrinsic contour estimation algorithm.

5.1 Related Work

Illumination-based methods for forgery detection are either geometry-based or color-based. Color-based methods search for inconsistencies in the interactions between object color and light color [Ghol 08, Ries 10, Wu 11]. Geometry-based methods aim at detecting inconsistencies in light source positions between specific objects in the scene [Fari 10a, John 05, John 07a, John 07b, Kee 10, OBri 12].

Gholap and Bora [Ghol 08] introduced physics-based illumination cues to image forensics. The authors examined inconsistencies in specularities based on the dichromatic reflection model. Specularity segmentation on real-world images is challenging [Ries 09a]. Therefore, the authors require manual annotation of specular highlights. A second drawback of this approach is that it relies on the presence of specularities on all regions of interest making them difficult to deploy in many real-world scenarios. To avoid this problem, Wu and Fang [Wu 11] assume purely diffuse reflectance (i.e., scenes without specularities), and train a mixture of Gaussians to select a proper illuminant color estimator. The angular distance between illuminant estimates from selected regions can then be used as an indicator for tampering. Unfortunately, the authors require the manual selection of a "reference block", where the color of the illuminant is estimated with sufficient accuracy. Unfortunately, the selection criteria for such a reference block are not quite clear. This restricts the applicability of the method to scenes containing favorable background, and the selection itself requires a human expert.

Two methods have been proposed to use the direction of the incident light for exposing digital forgeries. Johnson and Farid [John 07a] proposed a method which computes a low-dimensional descriptor of the lighting environment in the image plane (i. e. in 2D). It estimates the illumination direction from the intensity distribution along manually annotated object boundaries of homogeneous color. Kee and Farid [Kee 10] extended this approach to additionally exploit known 3D surface geometry. The authors demonstrate, for the case of faces, that a dense grid of 3D normals can improve the estimate of the illumination direction. To achieve this, a 3D face model is registered with the 2D image using manually annotated landmarks.

Johnson and Farid [John 07b] also proposed solutions for special cases. For instance, to investigate spliced images where the image parts containing people stem from different sources, they proposed a method for detecting forgeries using specular highlights in the eyes. Saboia *et al.* [Sabo 11] automatically classified these images by extracting additional features, such as the viewer position. The applicability of both approaches, however, is somewhat limited in practice by the fact that people's eyes must be visible and available in high resolution.

Determining the direction of the incident illumination is also occasionally addressed in computer vision work. For instance, Li *et al.* [Li 03] propose a method to estimate the light direction based on shadow boundaries and highlights in the image. However, the applicability of the method is limited to convex objects on planar surfaces. Takai *et al.* proposed a method to estimate the position of near light sources by

placing two gray spheres within the scene [Taka 09]. In a forensic scenario, however, the conditions for image capturing can not be controlled. Thus, one can not assume to have such gray spheres placed in the scene under investigation.

In this chapter, we build upon these ideas and add own algorithms. For the exploitation of the illuminant color, we develop a novel algorithm in Sec. 5.2, based on insights from the previous chapter. In Sec. 5.3, we extend the method by Johnson and Farid for exploiting illumination direction in forensics for a preprocessing step that normalizes the albedo along the contours of objects.

5.2 Illumination Color

In cases when two objects are apparently exposed to the same illumination, a forensic cue can be constructed. The basic idea, independently developed by Gholap and Bora [Ghol 08] and us [Ries 10], is to set up constraints for consistent versus inconsistent illumination. This was also our motivation for the fundamental research presented in Chap. 4. The general approach is to estimate the illuminant color locally, e. g. per person in the scene, and to compare these estimates. To accomplish this, we require robust methods for illuminant estimation that require little spatial support.

One concern in forensics is the false positive rate, i. e. reporting a tampered region although the image is authentic. In the context of the illumination color estimation, physics-based methods provide by definition a fully explicable model. If the model constraints are approximately fulfilled, the result is well predictable and explicable. This property can greatly increase the trust in a decision. In contrast, statistical approaches require a thoroughly compiled training set to compensate the underlying heuristics.

We investigated both directions. In Sec. 5.2.1, we first present the idea, general considerations and a manual, physics-based approach for investigation forgeries. Then, in Sec. 5.2.2, we propose a machine learning-based approach for an automated original/tampered decision. We pick up the discussion of the previous paragraph in Sec. 5.2.3.

5.2.1 User-driven Assessment

We propose a new method for the assessment of illumination-color consistency over the scene by extracting local illumination estimates. At the time of the development of this method, we were not aware of a similar approach in image forensics. The method is based on an extension of an illumination estimation method that is based on the physical principles of image formation. In contrast, most state-of-the-art methods for illuminant color estimation are machine-learning based. However, it is our belief that deviations from the expected result can be easier explained using a physics foundation than by machine-learning results, as detailed in Sec. 5.2.1.5. We believe this is a highly desirable property in forensics applications. Depending on the number of light sources of the scene, we show that these local estimates can provide further insights on the scene construction. For instance, if a photographer took an image at night using flashlight (which is typically a relatively bluish light source), we

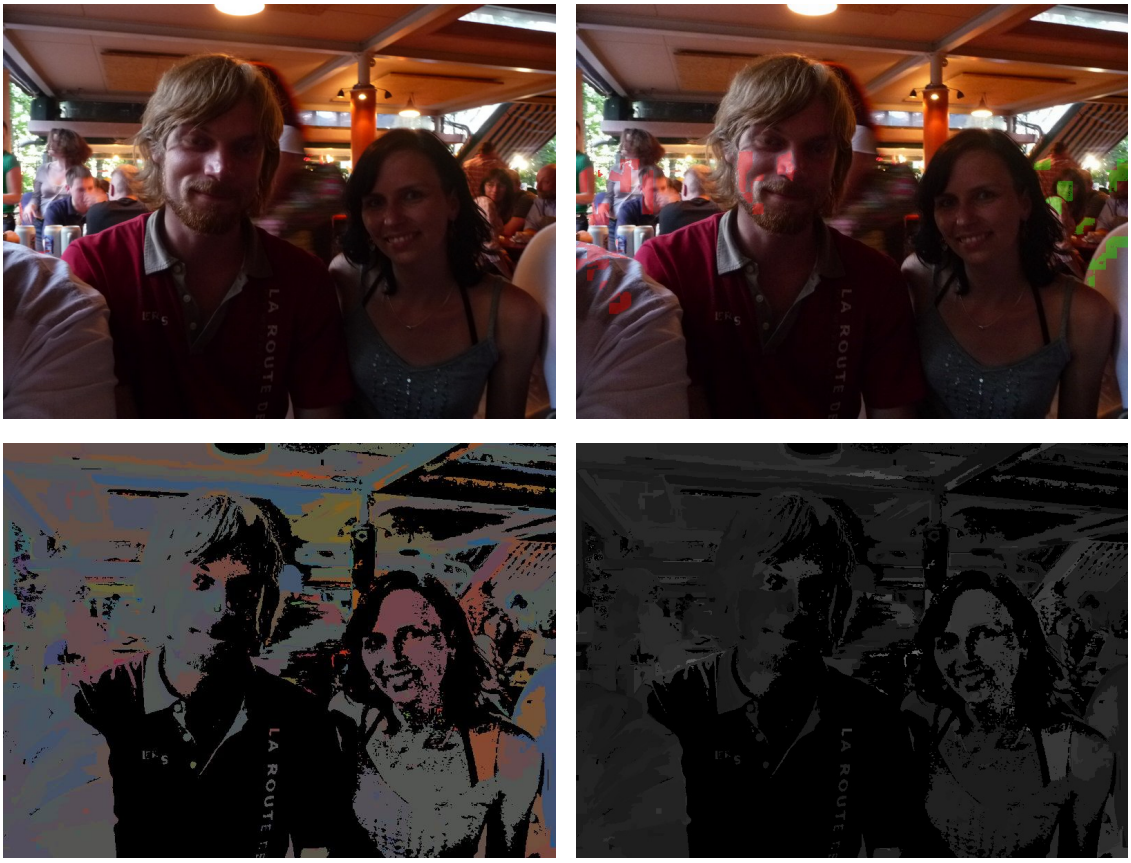


Figure 5.1: Illustration of the method. An image containing multiple illuminants (top left) is annotated in the regions of the illuminants of interest (top right, red and green markers). Then, the illuminant map is computed, local estimates for the illuminant color over the image (bottom left), as well as a distance map of every region to these illuminants (bottom right). Inconsistencies in these representations are interpreted as traces of tampering.

can obtain a rough relative depth estimate from the decay of the blue channel in the illuminant estimates. Inconsistencies in the illumination distribution can be used to distinguish original and spliced images. In detail, the contributions of this subsection are a) the development of a physics-based method for the recovery of the illuminant color for different objects in the scene, b) the introduction of an *illumination map* based on a distance measure on the estimated results, and c) The demonstration of the feasibility of employing this illuminant map in forensic analysis.

5.2.1.1 Overview of the Method

We propose a manual approach for assessing the color of the illumination. The method involves the following steps, as illustrated in Fig. 5.1.

1. The image is segmented in regions of approximately the same object color. These segments are called *superpixels*. A superpixel is required to a) be directly

illuminated by the light sources under examination and b) roughly adhere to the physical model presented in Sec. 4.4.2.

2. A user selects such superpixels whose incident illuminant he wants to further investigate. Each group of superpixels represents one illuminant color under investigation.

3. Estimation of the illuminant color is performed twice. First, the estimation is done on every superpixel separately. Second, the estimation is done on the user-selected superpixel groups for greater robustness.

4. The user-selected groups form the reference illuminants. A distance measure from these illuminants to every superpixel estimate is computed. We visualize these per-superpixel distances in what we call a *distance map* to support the analysis of the illumination color consistency.

In special cases, this method can be fully automated. On the other hand, since the estimation of the illuminant color is an underconstrained problem, there will always exist scenes that can not be correctly processed. We believe that a limited degree of human interaction is a valid tradeoff between the accuracy of the method and its usability.

5.2.1.2 Local Illuminant Estimation

In our forensic scenario, one must assume that the whole picture is composed from different sources. Thus, to estimate the illuminant color, only small, isolated regions of the image can be used. So far, limited research has been done in this direction. The work by Bleier *et al.* [Blei 11] indicates that many off-the-shelf single-illuminant algorithms do not scale well on smaller image regions. Additionally, the methods investigated by Bleier *et al.* are either statistical, or rely on assumptions that can barely be verified in a forensic scenario. As a consequence, we decided to use a localized variant of our single-illuminant estimator, as presented in Sec. 4.4.2 and Sec. 4.4.2. The algorithmic steps are listed below. The main variation of the fully automated version of the algorithm is that the reference illuminant colors for the comparison are computed from user-selected areas. Thus, in terms of Sec. 4.4, we avoid the localization problem in multi-illuminant estimation.

1. For every dominant illuminant in the scene, a user is required to select regions that a) follow the dichromatic reflectance model and b) are mostly lit by that light source.

2. Segment these regions in superpixels with roughly uniform chromaticity.

3. Further subdivide these superpixels in a rectangular grid. We call each such rectangular subregion a *patch*.

4. Transform every patch to inverse intensity space.

5. Apply tests on the shape of the patch's pixel distribution. If the distribution passes, obtain a local illuminant color estimate for this patch.

6. Obtain a color estimate for each dominant illuminant, based on a majority vote on local estimates of the user-selected regions.

For the superpixel segmentation, we used the publicly available code by Felzenszwalb and Huttenlocher [Felz 04] on the image chromaticities, though any segmentation method could be used. We choose the segmentation parameters $0.1 \leq \sigma \leq 0.3$ and $100 \leq k \leq 300$. Typically $\sigma = 0.3$ and $k = 300$ gave satisfying results, dividing the image in not too small regions of similar object color. The grid size is adaptive to the image size, typically between 16 and 32 pixels in the horizontal and vertical directions. The overall goal when determining superpixel size and grid size is a trade-off between spatial detail, which is provided by smaller superpixels, and estimation robustness, which comes from a sufficiently large number of grid cells (at least 10) within a superpixel. For the experiments in this section, we did not compensate image gamma. This could be done with additional preprocessing, using for instance the method by Lin *et al.* [Lin 04].

5.2.1.3 Illuminant Color for Image Forensics

Once the illuminant color estimates for the user-annotated regions are computed, the whole image can be examined for illumination color inconsistencies, as described in Sec. 5.2.1.4. Since the estimation of the illuminant color is a severely under-constrained problem, we briefly discuss failure cases and possible workarounds in Sec. 5.2.1.5. Please note that our illumination estimation method performs comparably to other state-of-the-art single illuminant estimation methods. This will be shown in Sec. 5.2.1.6.

5.2.1.4 Detecting Inconsistencies in Illumination

The same process (see Sec. 5.2.1.2 and Sec. 4.4.2) that was used in computing the illuminant color estimates at the user-specified regions is now extended to the entire image. The voting, however, is now performed within each superpixel. Thus, every superpixel contains an individual illuminant estimate. We store these illuminant estimates in a new image, where each superpixel is colored according to its estimated illuminant color. We call this new image *illumination map*. This map gives already quite meaningful results for the analysis.

For forensic analysis, we aim to quantify the relationship between the illuminant estimates. In a scene with truly one dominant illuminant, this can be done by comparing the angular errors of the individual illuminant estimates. However, most real-world scenes contain a mixture of illuminants. Their influence on the scene is closely connected to the positions of the objects relative to the positions of the light sources. Since the geometric composition of the scene is typically unknown, we resort to developing a tool for supporting the visual assessment of the scene, which we call *distance map*. Figure 5.2 shows an example. From left to right, the input image, illumination map and distance map is shown.

The distance map captures how well the illuminant estimation at each superpixel fits to the estimated dominant illuminants. For improved clarity, we assume two dominant illuminants $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ that were obtained from two user-selected regions.

Figure 5.2: Original image, illumination map and distance map for the image under examination. Foreground persons are estimated with a bluish color, probably due to flashlight, while persons in the background are increasingly red illuminated. The distance map between foreground and background illumination spots captures this relationship as a black-to-white transition.

We aim to create a grayscale-image that depicts the relative influence of both light sources. The distance map is created by assigning the value 0 (black) to the user-defined region corresponding to illuminant $\boldsymbol{e}_1$. Similarly, the second user-defined region, which gave rise to dominant illuminant $\boldsymbol{e}_2$, is assigned the value 1 (white). Then, for all the remaining pixels, the distance value $d_{e_l}(\boldsymbol{x})$ of a local estimate $\boldsymbol{e}_l(\boldsymbol{x})$ is computed as

$$d_{e_l}(\boldsymbol{x}) = (\boldsymbol{e}_l(\boldsymbol{x}) - \boldsymbol{e}_1)^{\mathrm{T}}(\boldsymbol{e}_2 - \boldsymbol{e}_1) \ . \tag{5.1}$$

The values in the distance map form a grayscale image with values in the range $[0, 1]$ if the estimate is located between $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Values outside of this range are cropped, but could also be otherwise marked as outliers. Such a map captures the relative influence of both light sources in each pixel.

The illumination map and the distance map are used together for the analysis of the image. In order to be consistent, a local illuminant estimate in an image must a) either exhibit a relative illuminant contribution that fits in the spatial layout of the scene or b) fail to fulfill the underlying physical model. In the latter case, it must be ignored for the analysis.

By adjusting the values of the criteria on the pixel distributions[1], it is possible to obtain fewer estimates that fit the physical model better (at the expense of larger regions with sparse or no estimates). On the other hand, less strict parameters lead to a more complete map, where also more outliers are expected. In general, we preferred lenient settings in our experiments. For the slope we set a lower bound of 0.003, and for the eccentricity 0.5. A stricter set of values, i.e. 0.01 for the slope and 0.95 for the eccentricity, typically results in fewer outliers.

5.2.1.5 Caveats and Workarounds

In some cases, the estimation of the illuminant color can not be successfully applied. Fortunately, for a physics-based method like the proposed one, the reasoning about

[1] Adjustable parameters are the superpixel size, grid size, and the thresholding parameters on the eigenvector slope and eccentricity in Eqn. 4.45, see page 97

failure cases is often easier than for machine-learning methods. While failures in the latter case often arise due to limitations of the training data or algorithm-dependent assumptions on the color distributions, physics-based methods mainly fail due to violations of the assumed reflectance model. This makes it possible to argue about possible problems and look for workarounds.

We present some cases where our method is problematic. First, the camera response is assumed to be linear. This is leveraged by the fact that we exploit only the relationship between illuminant estimates, and do not consider absolute estimates. Nevertheless, a gamma estimation method, e.g. [Lin 04], can be used to normalize the image. Some non-dielectric surfaces are especially difficult to handle, e.g. fluorescent materials (see Fig. 5.3) and metals. Other failure cases involve areas that are mostly

(a) (b) (c) (d)

Figure 5.3: Failure cases for the proposed illuminant color estimation method. Figures 5.3a and 5.3c are the original images, Figures 5.3b and 5.3d the respective illumination maps. In Fig. 5.3b, the illuminant estimate in the shadowed area under the head of the left actor is biased towards the object color. In Fig. 5.3d, the fluorescent suit of the actor overproportionally pushes the illuminant estimate towards extreme values.

diffuse, or highly textured, or in shadow (see Fig. 5.3). Finally, the method is inherently limited by the assumption that the color of the specularity closely approximates the color of the illuminant.

We found that by visual inspection it is often possible to distinguish failure cases from real inconsistencies. It is also possible to follow specific rules to minimize the risk of misjudging the scene under observation. The most robust approach is to use only identical or very similar materials for the analysis, e.g. faces in a crowded scene. We reflect this by demanding the user to select regions that a) are of interest for the examination and b) roughly adhere to the model.

5.2.1.6 Experiments

For qualitative results on multiple illuminants, we collected approximately 430 images containing scenes with multiple illuminants or unusual single-illuminant setups from

Figure 5.4: Tampered image. Illumination map as well as distance map show a clear difference between the first two and the third person. Since the three stand close together in the image, it can be assumed that this difference is due to tampering.

various sources, mostly flickr [Yaho 12]. Besides these images, which were assumed (or known, respectively) to be original, 10 forgeries have been examined using the proposed method. We present three cases where image geometry and illumination create discontinuities. Figure 5.4 shows a case where the change in the illumination color is barely explicable with the scene setup. Both the illumination map as well as the distance map exhibit a sharp transition between the two persons in the foreground and the third in the back, which could only be feasible if there was a greater distance between them.

The example in Fig. 5.5 shows outdoor illumination with one dominant illuminant. Again, we compare the skin regions of the people, in order to have roughly comparable object materials. The selected regions are the directly lit skin of the inserted person versus the directly lit skin of other guests. The illumination map shows blueish estimates for the inserted man. The distance map makes this difference even more visible. Note that the estimates of the coast line in the background should be ignored (although they fit well in this particular case). The underlying pixels must be assumed to be purely diffuse, and thus do not satisfy our assumptions.

Figure 5.6 contains a more complex case. The woman in the right is inserted in the image. Illumination map and distance map are plausible, compared to the people that stand similarly close to the restaurant. However, by adding the scene geometry we obtain a strong clue that this scene is not original. Since the woman is turned away from the restaurant, the illuminant color on the woman's chest should share greater similarity with the body parts of the other people that are turned away from the restaurant lights.

Figure 5.5: Original image (top left) and tampered image (top right). A comparison of the skin regions of the people exposes the inserted man in the distance map.

5.2.2 Automated assessment of the Illumination Consistency

One drawback of the presented approach is the dependence on a human expert to judge the authenticity of an image based on illuminant map and distance map. We found that it is non-trivial to clarify the details of the method to untrained subjects. Thus, we investigated further constraints, in order to transfer the authenticity decision from humans to an automated algorithm. In this section, we present a preliminary investigation towards such a system. We limit ourselves to the comparison of illuminant estimates that were obtained from similar materials in the scene. More precisely, we used illuminant estimates on face regions to detect spliced images, relying on the fact that pictures of persons are often subject to manipulations [Fari 11]. The authenticity is determined via machine learning on feature vectors that were extracted from illuminant maps. First, we present an overview of the algorithm. Then, we present the algorithmic details for every step. Throughout this section, illuminant maps are abbreviates as IM.

5.2.2.1 Interpretation of Illuminant Maps

We aim to classify the illumination for each pair of faces in the image as either consistent or inconsistent. The proposed method consists of five main components:

Figure 5.6: Original image (top left) and tampered image (top right). At first glance, illumination map and distance map show plausible results on the tampered image. However, the illuminant estimates are obtained from the front of the inserted woman, which is turned away from the restaurant lights. Therefore, the expected illumination should be more bluish, like e.g. at the back of the person in the middle.

1. *Local Illuminant Estimation:* we create illuminant maps from different local illuminant estimators that were applied on superpixels. As an extension of the method in Sec. 5.2.1, we do not only use the proposed physics-based estimator, but also a generalized gray world estimator.

2. *Face Extraction:* an operator sets a bounding box around each face that should be investigated. Alternatively, an automated face detector can be employed. We then crop each illuminant map to every bounding box, such that only the illuminant estimates of the face regions remain. This is the only step that may require human interaction.

3. *Computation of Illuminant Features:* for all face regions, texture-based and gradient-based features are computed on the IM values.

4. *Paired Face Features:* our goal is to assess whether two faces in an image are consistently illuminated. For each pair of faces in the image, we create a combined feature vector by concatenating the features from the two faces that constitute the pair.

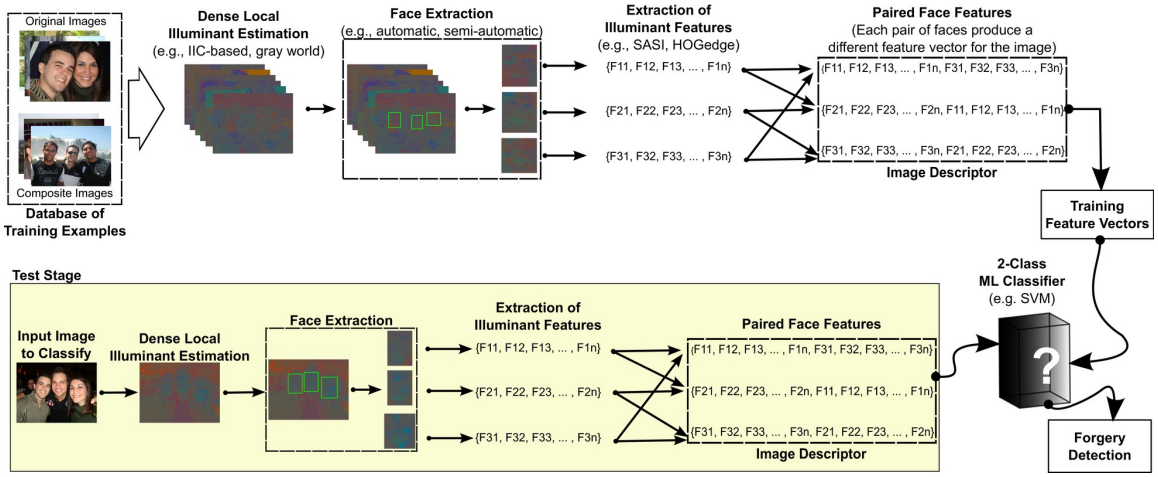


Figure 5.7: Overview of the proposed method.

5. *Classification:* we use machine learning to automatically classify the feature vectors. We consider an image as a forgery if at least one pair of faces in the image is classified as inconsistently illuminated.

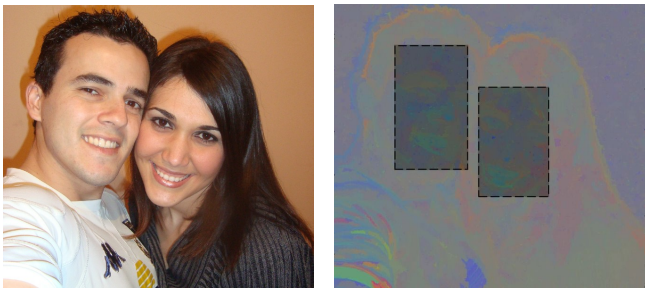
Figure 5.7 summarizes the main steps of the proposed method. The remainder of this section presents the details of these steps.

5.2.2.2 Dense Local Illuminant Estimation

To detect inconsistencies in the illumination color, we need a dense set of localized estimates. We segment the input image into regions of approximately constant chromaticity (so-called superpixels) with the algorithm proposed by Felzenszwalb and Huttenlocher [Felz 04]. Then we estimate the color of the illuminant per superpixel. By recoloring the superpixels with the estimated illuminant chromaticities, we obtain an illuminant map. We use two separate methods to obtain a version of this map: the statistical generalized gray world estimates, in particular the gray edge algorithm (see Eqn. 4.11 on page 66), and our variant of exploiting the inverse-intensity chromaticity space (see Sec. 4.4.2 and Sec. 5.2.1).

5.2.2.3 Face Extraction

Unconstrained estimation of the illuminant color can be error-prone and affected by the reflectance properties of the materials in the scene. However, it is possible to improve the accuracy of the relative error between two estimates by focusing only on objects of approximately the same material. For this work, we limit our examination of illumination consistency to human skin and, in particular, to faces. Pigmentation is the most obvious difference in skin characteristics between different ethnicities. This pigmentation difference depends on many factors as quantity of melanin, amount of UV exposure, genetics, melanosome content and type of pigments found in the skin [Igar 07]. However, this intra-material variation is typically smaller than that of all materials possibly occurring in a scene.



(a) Original

(b) Gray world with similar parts highlighted

Figure 5.8: An original image and its gray world map. Highlighted regions in the gray world map show a similar appearance.

All faces in the image that should be part of the investigation have to be annotated with a bounding box. In principle, this can be done automatically, through the use of a face detector [Schw 09]. However, we prefer a human operator for this task for two main reasons: a) this minimizes false detections or misses of faces; b) scene context is important when judging the lighting situation. For instance, consider an image where all persons of interest are illuminated by flashlight. The illuminants are expected to agree with one another. Conversely, assume that a person in the foreground is illuminated by flashlight, and a person in the background is illuminated by ambient light. Then, a difference in the color of the illuminants is expected. Such differences are hard to distinguish in a fully-automated manner. For this paper, we focused on faces that are exposed to supposedly similar illumination, which can be visually verified by the operator.

We illustrate this setup in Figure 5.8. The faces in Figure 5.8a can be assumed to be exposed to the same illuminant. As Figure 5.8b shows, the corresponding gray world illuminant map for these two faces also has similar values.

5.2.2.4 Interpreting Illuminant Maps as Texture Maps

Our main indicator for detecting inconsistencies in the illumination are the illumination maps. Thus, we consider an illuminant map as a texture, that exhibits a particular statistical structure, which may be disturbed when the image is tampered.

SASI Many different texture descriptors have been proposed in the literature thus far. One of the most effective [Pena 12] is the Statistical Analysis of Structural Information (SASI) [Cark 03] descriptor.

SASI [Cark 03] is a generic descriptor that measures the structural properties of texture. It computes the autocorrelation on multi-resolution sliding windows over the image. Given that each window is composed by a different orientation and resolution, a clique window $\mathbf{B_c}$ represents the window with index c under a specific orientation and resolution. For each of these clique windows, an autocorrelation coefficient is

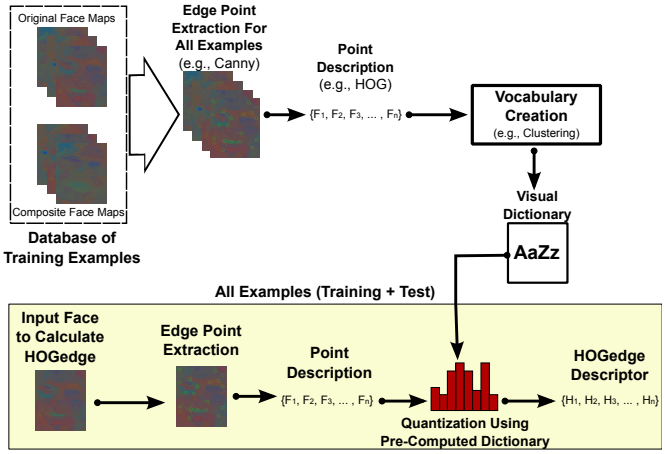


Figure 5.9: Overview of the proposed HOGedge algorithm.

computed. Ultimately, every clique window $\mathbf{B_c}$ provides a different autocorrelation image. We use the means and standard deviations extracted from these autocorrelation images to compose the image feature vector.

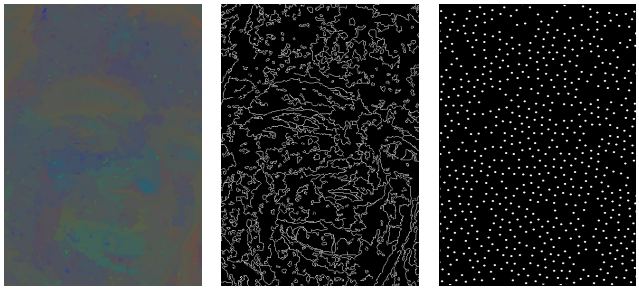
The most important advantage of SASI for our application is its good capability of capturing small granularities and discontinuities which are present in texture patterns. These patterns appear mainly in sharp corners and abrupt changes such as the ones present in illuminant maps, especially in the face region of composite images.

HOGedge: A New Algorithm for Interpreting Edges in Illuminant Maps
When a spliced forgery is created, the resulting local discontinuities affect mainly the edges of an illuminant map at the splicing boundary. To characterize this local information, we propose a new algorithm named *HOGedge*. It is based on the well-known HOG-descriptor, and computes visual dictionaries of gradient intensities in edge points. The full algorithm is described in the remainder of this section. Fig. 5.9 shows an algorithmic overview of the method.

The algorithm for characterizing a face using HOGedge descriptor is divided into two parts. First, we construct a visual dictionary using training examples. Then, we construct the final feature vector for every image in a dataset using a learned visual dictionary.

Given a face region from an illuminant map, we first extract edge points using the Canny edge detector [Cann 86]. This produces a large number of spatially close edge points. To reduce the number of points, we filter the Canny output using the following rule: starting from a seed point, we eliminate all other edge pixels in a region of interest (ROI) centered around the seed point. The edge points that are closest to the ROI (but outside of it) are chosen as seed points in the next iteration. Figure 5.10 depicts an example of the resulting points.

We compute Histograms of Oriented Gradients (HOG) [Dala 05] to describe the edge points. HOG is based on evaluating normalized local histograms of image gradient orientations in a dense grid. The HOG descriptor is constructed by dividing the region of interest into spatially small regions ("cells"). Each cell provides a local 1-D histogram of quantized gradient directions using all cell pixels. To construct the final



(a) IM derived from gray world (b) Canny Edges (c) Filtered Points

Figure 5.10: (a) The gray world IM for the left face in Figure 5.8a. (b) The result of the Canny edge detector when applied on this IM. (c) The final edge points after filtering using a square region.

feature vector, the histograms of all cells within a larger spatial region are combined and contrast-normalized using an accumulated measure of local histograms. We use the HOG output as a feature vector for the subsequent steps.

The number of edge points, and therefore the number of HOG vectors, varies depending on the face under examination. To obtain one feature vectors of equal length, we employ visual dictionaries [Csur 04]. Visual dictionaries constitute a robust representation of an object as a collection of regions. The only information of interest is the appearance of each region [Winn 05].

To construct the visual dictionary, feature vectors from original and tampered images are required. We compose a visual dictionary with $2n$ visual words by clustering each set with n centers using the k-means algorithm [Bish 06]. Every visual word is represented by a cluster center. Thus, the visual dictionary comprises the most representative feature vectors of the training set.

For evaluating the feature vectors, the HOG feature vectors are mapped to the visual dictionary. Each feature vector in an image is represented by the closest word in the dictionary with respect to the Euclidean distance. A histogram of word counts represents the distribution of feature vectors in a face.

Face Pair To compare two faces, we combine the same descriptors for each of the two faces. For instance, we can concatenate the SASI-descriptors that were computed on a gray edge illuminant map. The idea is that a feature concatenation from two faces is different when one of the faces is an original and one is spliced. Thus, for an image containing n faces with $n \geq 2$), the number of joint feature vectors is $(n(n-1))/2$.

Classification Assuming all selected faces are illuminated by the same light source, we tag an image as manipulated if one pair is classified as inconsistent. Individual feature vectors, i. e. SASI features or HOGedge features on either gray world or IIC-

(a) Original image. (b) Spliced image.

Figure 5.11: An original image (left) and a spliced image (right).

based illuminant maps, are classified using a support vector machine (SVM) classifier with a radial basis function (RBF) kernel.

The information provided by the SASI features is complementary to the information from the HOGedge features. Thus, we use a machine learning-based fusion technique for improving the detection performance. Inspired by the work of Ludwig *et al.* [Ludw 09], we use a late fusion technique named SVM-Meta Fusion. We classify each combination of illuminant map and feature type independently (i. e. SASI-Gray-World, SASI-IIC, HOGedge-Gray-World and HOGedge-IIC) using a two-class SVM classifier to obtain the distance between the image and the classifier decision boundary. SVM-Meta fusion consists of merging the marginal distances provided by all individual classifiers to build a new feature vector. Another SVM classifier then classifies the combined feature vector.

5.2.2.5 Experiments

To validate our approach, we performed two sets of experiments using a new database with 200 images involving people. The database we created is composed of 200 indoor and outdoor images, with an image resolution of 2048×1536 pixels. Each image contains two or more people. From this dataset, 100 images are original, the remaining 100 images are doctored. The forgeries have been composed by adding one or more people in a source image that already contained one or more people. When necessary, we performed color and image adjustments to construct photo realistic forgeries. Figure 5.11 shows two example images from the dataset.

Performance Evaluation We compare five variants of the method. Throughout this section, we manually annotated the faces by marking a bounding box around the face In the classification stage, we use a five-fold cross validation protocol, an SVM classifier with an RBF kernel, and classical grid search for adjusting parameters in training samples [Bish 06]. Since each image provides a different number of feature vectors, we also use a proportional weight of classes to equalize them in the training stage. Let w_{orig} represent the number of feature vectors extracted paired faces in non-manipulated (pristine) images during training, and w_{manip} represent the number of feature vectors extracted from paired faces of composite images also during training. To use a proportional class weighting, we set the weight of non-manipulated

image class to $w_{\text{manip}}/\left(w_{\text{orig}}+w_{\text{manip}}\right)$ and the weight of composite image class to $w_{\text{orig}}/\left(w_{\text{orig}}+w_{\text{manip}}\right)$.

As for the experiments, we compare these five experimental setups:

- **SASI-IIC:** we extract SASI-features from an IIC-based illuminant map. The SASI descriptor is calculated over the Y channel from the YC_bC_r color space. We configure SASI algorithm as presented in [Pena 12][2]. O

- **SASI-Gray-World:** we calculate gray world illuminant maps using $n_{\text{GW}}=1$, $\tau_{\text{GW}}=1$ and $\sigma=3$. The SASI descriptor is extracted from gray world IMs using the same configuration as SASI-IIC.

- **HOGedge-IIC:** we compute the HOGedge descriptor on the IIC-based illuminant map. For the HOGedge descriptor, it is necessary to adjust some parameters. The (empirically determined) best parameters are: edge detection is performed on the Y channel of the YC_bC_r color space, with a Canny lower bound of 0 and an upper bound of 10. The square region for edge point filtering is set to 32×32 pixels. Furthermore, we use 8-pixel cells without normalization in HOG, and 100 visual words for both the original and the tampered images (i. e. the dictionary consists of 200 visual words).

- **HOGedge-Gray-World:** this configuration is similar to HOGedge-IIC. We compute gray world illuminant maps with the same parameters as above, $n_{\text{GW}}=1$, $\tau_{\text{GW}}=1$ and $\sigma=3$. The empirically determined best performing parameters for HOGedge-Gray-World were the same as for HOGedge-IIC, with one exception: the size of the visual word dictionary is set to 75 visual words from each class (thus, the dictionary contained 150 visual words).

- **Metafusion:** We implemented a late fusion algorithm as explained in Section 5.2.2.4 using SASI-IIC, SASI-Gray-World, and HOGedge-IIC. HOGedge-Gray-World is excluded from the input methods, due to its weaker performance (see the evaluation below).

Figure 5.12 depicts a ROC curve of the performance of all methods using bounding box annotations. A user was required to click the corners of bounding boxes to crop out the faces. We used sensitivity and specificity to assess the accuracy on doctored and original images, respectively. The area under the curve (AUC) is computed to obtain a single numerical measure for each result.

Metafusion performs best, resulting in an AUC of 85.3%. In particular for high specificity (i. e. few false alarms), the method yields a much higher sensitivity compared to the other variants. Specifically, in a real forensic scenario, when an analyzed photograph is classified as composite using this variant of the method, it provides high confidence about the image authenticity. This kind of confidence is an important initial step when an expert needs to decide about image authenticity, decreasing the quantity of necessary future work.

The second best variant is SASI-Gray-World, with an AUC of 84.0%. In particular for a specificity below 80.0%, the sensitivity is comparable to Metafusion. SASI-IIC

[2]We gratefully thank the authors for the source code.

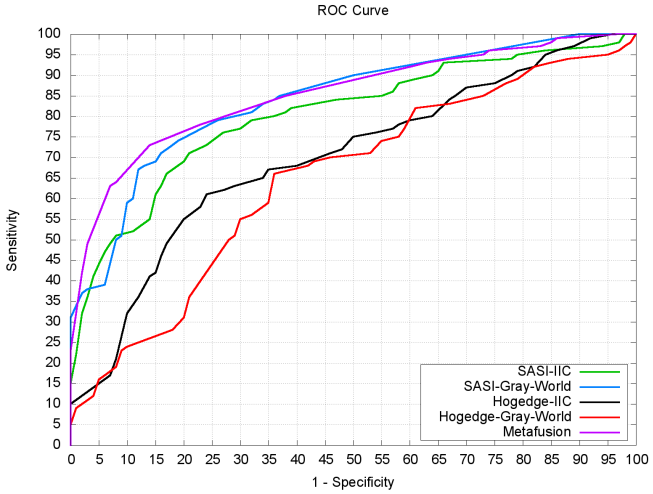


Figure 5.12: Comparison of all algorithm variants. Metafusion performs best.

achieved an AUC of 80.3%, followed by HOGedge-IIC with an AUC of 69.9% and HOGedge-Gray-World with an AUC of 64.7%.

5.2.3 Discussion

We presented two directions to exploit illuminant color as a forensic cue. The manual approach strictly follows a principled, physics-based pipeline. The automated method relieves the user from a technically challenging, potentially subjective assessment of the illuminant maps. Thus, both approaches have their validity, and can be seen as complementary: for a first check, the automated method can be applied, and then, for a more detailed analysis, regions of interest can be manually examined. For manual assessment, a more detailed analysis should be established in future work. For instance, it would be interesting to investigate consistency criteria directly in inverse intensity-chromaticity space.

Besides the work of Gholap and Bora [Ghol 08] and Wu and Fang [Wu 11], illumination color has not yet been addressed in image forensics. Thus, also the presented work here can be seen as a prototypical proof-of-concept, with several open questions that deserve further investigation. We discuss one issue here in greater detail, as we consider it most relevant in relation to the previous chapter on illuminant estimation, and refer the reader for remaining future work to chapter 6.

One interesting point which was not not been further investigated due to large resource requirements is the relation of illuminant estimation to source identification. This question is indirectly raised through the work by Deng *et al.* [Deng 11]. In this paper, the authors exploit the fact that applying the same white-balancing algorithm twice to an image does not change the result. Thus, if a the white-balancing method of a camera is known, the camera type can be identified if repeated application of the camera's white-balancing algorithm leads to identical images. However, camera white-balancing algorithms are often protected as intellectually property of the manufacturer. Thus, Deng *et al.* generalize these findings by applying different

white-balancing algorithms to a training set of images, and compute image quality metrics [Avci 03, Eski 95] as feature vectors. A support vector machine can then be used to classify unknown images by their camera type.

Considering the close relation between illuminant color estimation and white balancing, which property of a spliced image is actually exploited — the difference in the (physical) illumination, in the sense of Chap. 4, or different white balancing algorithms of the source cameras used for capturing the images? In the first case, a physics-based method is preferred, due to its rigorous formulation. In the latter case, statistical methods could show better performance, as camera white-balancing algorithms are often variants of gray world or white patch methods. Thus, statistical methods can be expected to better model the in-camera color processing than physics-based methods. We consider this question the most interesting for developing future, more robust color-based cues for forensic algorithms.

5.3 Illumination Direction

The direction of the incident light has been proposed as a forensic cue by Johnson and Farid [John 07a]. The distribution of the direction of incident light, together with the surface normals of an object of interest, are combined in a potentially powerful geometric algorithm for exposing image manipulations.

A user is required to mark the contours of an object. The proposed method by Johnson and Farid computes then the brightness distribution over the contour normals. Brightness distributions from different object within an image can then be compared, e. g. by computing the correlation. Like most high-level forensic methods, this algorithm does not depend on digital imagery. It can as well be applied on analogue photographs or even paintings.

However, in our experiments, it turned out that this method is relatively difficult to apply in practice. One reason are the relatively strict constraints that are imposed on the contour. Regions of self-shadowing, non-convex edges and specularities have to be excluded. Furthermore, the contour normals must span a minimum of about 130 degrees, and the contour must consist of uniform albedo.

As shown in Fig. 5.13, real-world images often do not satisfy these conditions. In the left picture, to compare the both pedestrians, the pose and different layers of cloth make it difficult to apply the method. In the right image, a small angle of the surface normals per cloth make a reliable recovery of the illuminant direction also challenging. Both cases are discussed in greater detail in Sec. 5.3.5.2.

In this section, we aim to extend the applicability of this method by investigating intrinsic image decomposition as a way to relax the constraint of uniform albedo. We first present the method by Johnson and Farid in Sec. 5.3.1. Previous work in intrinsic image decomposition is presented and discussed in Sec. 5.3.2. We propose a new method, *intrinsic contour estimation* (ICE), in Sec. 5.3.3. It is specifically tailored for the application with the basic algorithm. A preliminary comparative evaluation has been conducted. The results that were obtained so far are presented and discussed in Sec. 5.3.5.

Figure 5.13: Failure cases for (our implementation of) the algorithm by Johnson and Farid. In both cases, it is not possible to find a uniform albedo, piecewise contour around the object that spans more than 130 degrees. Left picture courtesy of Ed Yourdon [Your 08].

5.3.1 Basic Method for Comparing Lighting Environments

Ramamoorthi and Hanrahan [Rama 01] showed that the irradiance (i. e., incident intensity) on a Lambertian sphere can be recovered from observations of the brightness distribution on the surface of the sphere. The solution is obtained by representing the intensity distribution as spherical harmonics, i. e., spherical coordinates centered at the object. Additionally, Basri and Jacobs [Basr 03] showed that different distributions of incident lights on Lambertian surfaces can be relatively accurately represented by a nine-dimensional subspace.

Putting these two findings together, Johnson and Farid proposed the baseline method for lighting environments as a forensic cue [John 07a]. The authors propose to use a nine-dimensional subspace of the spherical harmonics to estimate the distribution of incident light on convex objects. For two or more objects in the scene, the lighting environments can be correlated. If the difference of two lighting environments exceeds a threshold, the image is assumed to be manipulated.

In the next paragraphs, we derive this algorithm more formally, following the presentation in [John 07a]. Assume that we operate on an object of constant albedo under Lambertian reflectance, and the camera response function is linear. Then, the amount of incident light on a surface patch is identical to the observed intensity, up to an unknown multiplicative factor (see [John 07a]). Assume furthermore, for simplicity, that we operate on a grayscale image. With these assumptions, let $p(\boldsymbol{\nu}(\boldsymbol{x}))$ the intensity of a pixel $\boldsymbol{p}$. As the albedo is assumed to be constant, differences in the intensity depend only on the surface normal $\boldsymbol{\nu}(\boldsymbol{x})$ at pixel $\boldsymbol{x}$. Thus,

$$p(\boldsymbol{\nu}(\boldsymbol{x})) = \int_{\Omega} e(\boldsymbol{v}, \boldsymbol{x}) r(\boldsymbol{v}, \boldsymbol{\nu}(\boldsymbol{x})) d\boldsymbol{v} \quad . \tag{5.2}$$

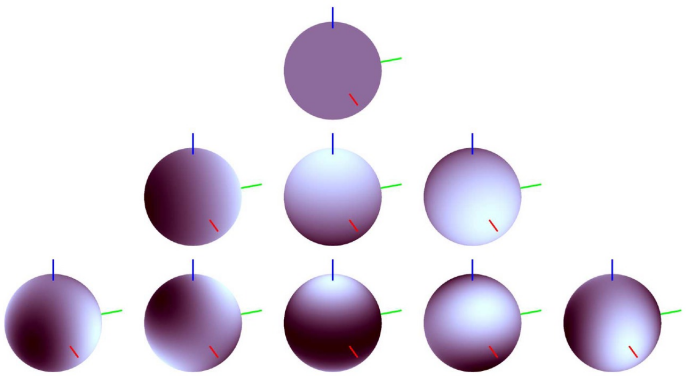


Figure 5.14: Illustration of the spherical harmonics basis functions. Picture courtesy of [John 07a].

	j				
i	-2	-1	0	1	2
0			$\frac{1}{\sqrt{4\pi}}$		
1		$\sqrt{\frac{3}{4\pi}}v_y$	$\sqrt{\frac{3}{4\pi}}v_z$	$\sqrt{\frac{3}{4\pi}}v_x$	
2	$\sqrt{\frac{45}{12\pi}}v_xv_y$	$\sqrt{\frac{45}{12\pi}}v_yv_z$	$\sqrt{\frac{5}{16\pi}}(3v_z^2-1)$	$\sqrt{\frac{45}{12\pi}}v_xv_z$	$\sqrt{\frac{45}{48\pi}}(v_x^2-v_y^2)$

Table 5.1: First 9 basis functions of the spherical harmonics $\eta_{i,j}(\boldsymbol{v})$, for $\boldsymbol{v} = (v_x \;\; v_y \;\; v_z)^{\mathrm{T}}$.

Here, Ω captures all possible angles $\boldsymbol{v}$ of the incident light $e(\boldsymbol{v}, \boldsymbol{x})$ at pixel $\boldsymbol{x}$, and $r(\boldsymbol{v}, \boldsymbol{\nu})$ denotes the surface reflectance function. As we assume constant albedo, the dependence of $r(\boldsymbol{v}, \boldsymbol{\nu})$ to $\boldsymbol{x}$ can be neglected.

The direction of incident light $e(\boldsymbol{v}, \boldsymbol{x})$ can be expressed in terms of spherical harmonics[3] [Rama 01] as

$$e(\boldsymbol{v}, \boldsymbol{x}) = \sum_{i=0}^{\infty} \sum_{j=-i}^{i} h_{i,j}(\boldsymbol{x}) \eta_{i,j}(\boldsymbol{v}) \;\; , \tag{5.3}$$

where $h_{i,j}(\boldsymbol{x})$ are weighting factors, and $\eta_{i,j}(\boldsymbol{v})$ denotes the basis function of the spherical harmonics. The basis functions are illustrated in Fig. 5.14, the actual basis functions up until $i = 2$, for $\boldsymbol{v} = (v_x \;\; v_y \;\; v_z)^{\mathrm{T}}$ are listed in Tab. 5.1. As can be seen from Fig. 5.14, the $\eta_{i,j}(\boldsymbol{v})$ oscilates along the surface of a sphere. The number of period increases with i, the orientation of the oscilation varies with j.

The reflectance function $r(\boldsymbol{v}, \boldsymbol{\nu})$ in Eqn. 5.2, under the assumption of Lambertian reflectance (see Sec. 4.1.1 on page 59), is

$$r(\boldsymbol{v}, \boldsymbol{\nu}(\boldsymbol{x})) = \begin{cases} \boldsymbol{v}^{\mathrm{T}} \cdot \boldsymbol{\nu}(\boldsymbol{x}) & \text{if } \boldsymbol{v}^{\mathrm{T}} \boldsymbol{\nu}(\boldsymbol{x}) > 0 \\ 0 & \text{otherwise} \end{cases} \;\; , \tag{5.4}$$

[3] For a general introduction to spherical harmonics, see e. g. [Groe 96].

i. e. the cosine between the direction of the incident light and the surface normal, or 0 if the surface is not directly illuminated by the light source. Additionally, this intensity can directly be observed with a digital camera, as intensities of Lambertian reflectance do not depend on the position of the observer.

Equation 5.4 can also be rewritten in terms of spherical harmonics. For Lambertian reflectance, the amount of incident light depends only on the angle of the incident light to the surface normal. Thus, most terms cancel, such that (after some calculations), it turns out that

$$r(\boldsymbol{v}, \boldsymbol{\nu}(\boldsymbol{x})) = \sum_{i=0}^{\infty} \hat{r}_i r_{i,0} \left((0 \;\; 0 \;\; \boldsymbol{v}^{\mathrm{T}} \boldsymbol{\nu}(\boldsymbol{x})) \right) \quad . \tag{5.5}$$

depends only on harmonics with $j = 0$ (see [John 07a]). $\hat{r}_i$ incorporate the Lambertian reflectance assumption into Eqn. 5.5.

Equation 5.3 and Eqn. 5.5 can directly be inserted in the illumination model in Eqn. 5.2. Then, after some simplifications (see [John 07a]), $p(\boldsymbol{n})$ becomes

$$p(\boldsymbol{\nu}(\boldsymbol{x})) = \sum_{i=0}^{\infty} \sum_{j=-i}^{i} \sqrt{\frac{4\pi}{2i+1}} \hat{r}_i h_{i,j} \eta_{i,j}(\boldsymbol{\nu}(\boldsymbol{x})) \quad . \tag{5.6}$$

Basri and Jacobs showed that illumination changes of Lambertian surfaces can be modeled by a nine-dimensional subspace [Basr 03]. Thus, if i is limited to $0 \leq i \leq 2$, we obtain the nine basis functions as listed in Tab. 5.1, which leave only a small model error compared to using the full range of $0 \leq i \leq \infty$. However, in practice, it is only in special cases possible to obtain reliable three-dimensional surface normals from a single image[4]. Johnson and Farid employ a trick to make the described scheme feasible for general images by considering only the contour of the objects. Here, under the additional assumption of orthographic projection, the z-component of the surface normals is 0, and the x- and y-components can be estimated by fitting a curve to the object contour. Then, from the nine base functions $\eta_{0,0}$ through $\eta_{2,2}$, only five remain. Additionally, the coefficients $\hat{r}_i$ can be explicitly solved for Lambertian reflectance.

Taking these facts together, Eqn. 5.6 can be directly transferred in a linear system of equations. One line of the equations consists of

$$\begin{aligned} \boldsymbol{p}(\boldsymbol{\nu}(\boldsymbol{x})) = & \xi + h_{1,-1} \frac{2\pi}{3} \eta_{1,-1}(\boldsymbol{\nu}(\boldsymbol{x})) + h_{1,1} \frac{2\pi}{3} \eta_{1,1}(\boldsymbol{\nu}(\boldsymbol{x})) \\ & + h_{2,-2} \frac{\pi}{4} \eta_{2,-2}(\boldsymbol{\nu}(\boldsymbol{x})) + h_{2,2} \frac{\pi}{4} \eta_{2,2}(\boldsymbol{\nu}(\boldsymbol{x})) \quad . \end{aligned} \tag{5.7}$$

where

$$\xi = h_{0,0} \frac{\sqrt{\pi}}{2} - h_{2,0} \frac{\sqrt{5\pi}}{16} \quad . \tag{5.8}$$

Collecting the unknown coefficients ξ, $h_{1,-1} \ldots h_{2,2}$ in $\boldsymbol{h}$, different intensities (observations) in $\hat{\boldsymbol{p}}$, and the remaining coefficients in a matrix $\boldsymbol{M}$, Eqn. 5.7 can be more compactly written as

$$\hat{\boldsymbol{p}} = \boldsymbol{M}\boldsymbol{h} \quad , \tag{5.9}$$

[4] One such special case was demonstrated by Kee and Farid [Kee 10], by fitting 3-D face models to persons in the scenes under investigation.

which can be solved for $\boldsymbol{h}$ with a least squares approach, i. e.

$$\boldsymbol{h} = (\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M})^{-1}\boldsymbol{M}^{\mathrm{T}}\hat{\boldsymbol{p}} \ . \tag{5.10}$$

To avoid effects of noise, a Tikhonov regularization term is added. Thus, the overall objective function is

$$\epsilon_{\text{light}} = \|\boldsymbol{M}\boldsymbol{h} - \hat{\boldsymbol{p}}\|^2 + \lambda_{\text{light}}\|\boldsymbol{C}\boldsymbol{h}\|^2 \ , \tag{5.11}$$

where λ_{light} is a weighting factor, and

$$\boldsymbol{C} = \text{diag}(1 \quad 2 \quad 2 \quad 3 \quad 3) \tag{5.12}$$

denotes a diagonal matrix to enforce a lower energy in the higher order harmonics. The solution to Eqn. 5.11 is

$$\boldsymbol{\eta} = (\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M} + \lambda_{\text{light}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C})^{-1}\boldsymbol{M}^{\mathrm{T}}\hat{\boldsymbol{p}} \ , \tag{5.13}$$

which can directly be implemented. For details, please refer to [John 07a].

For forensic purposes, Johnson and Farid propose to compute the correlation of the coefficient vectors from two lighting environments, $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ as

$$\text{corr}(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2) = \frac{\boldsymbol{\eta}_1^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\eta}_2}{\sqrt{\boldsymbol{\eta}_1^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\eta}_1} \cdot \sqrt{\boldsymbol{\eta}_2^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\eta}_2}} \ , \tag{5.14}$$

where

$$\boldsymbol{Q} = \text{diag}(0 \quad \frac{\pi}{6} \quad \frac{\pi}{6} \quad \frac{15\pi}{512} \quad \frac{15\pi}{512}) \tag{5.15}$$

denotes a diagonal matrix containing the weighting of the coefficients. For a full derivation of the error function, please refer to [John 07a]. In practice, the object contour can be obtained by user annotations. In the quite extensive experimental part of their work, Johnson and Farid demonstrate that these annotated contours should span at least angles of about 130 degrees. If only a smaller part of the contour is available, the solution in Eqn. 5.13 becomes unstable.

Although theoretically compelling, the presented algorithm suffers from a number of assumptions. One relatively strict assumption is that the object albedo is constant. This forces a user to select only contours of the same material. In conjunction with the demand that a contour should span a minimum of 130 degrees, this algorithm is difficult to apply on several real-world images, as shown in Fig. 5.13 on page 136.

To alleviate this constraint, we investigated strategies to neutralize the intensity differences of different albedos. Such a method can act as a preprocessing step to the presented algorithm, and effectively extends the applicability of the method. In Sec. 5.3.2, we investigate algorithms for intrinsic image decomposition, which in theory should accomplish this task. Unfortunately, the results from these methods were not satisfying. Additionally, many of these methods suffer from excessive ressource requirements. Thus, we propose a specific, hand-tailored method in Sec. 5.3.3, called *Intrinsic Contour Estimation*. We conclude this chapter with an evaluation of these approaches in Sec. 5.3.5.

Figure 5.15: Example "teabag2" for intrinsic image decomposition (from [Gros 09]). The input image (left) is separated in two images, one containing the shading component (middle), the other containing the reflectance component (right). In this particular case, the reflectance image still contains slight shading artifacts.

5.3.2 Intrinsic Image Decomposition

The goal of intrinsic image decomposition is to separate a single input image into two images, one containing the shading, the other the reflectance of the scene. Figure 5.15 shows an example decomposition for the image "teabag2" from the publicly available dataset by Grosse *et al.* [Gros 09]. On the left, the input image is shown. In the middle and on the right, the ground truth shading and reflectance images are shown.

Assume that a shading image of reasonable quality can be obtained for a scene under investigation. Then, the lighting environment estimation by Johnson and Farid could directly operate on this shading image. In cases where it is difficult to find suitable contour segments on the original objects, the shading image can make the application of Johnson's and Farid's method possible.

The commonly used image model is typically relatively simple. A pixel $\boldsymbol{p}(\boldsymbol{x})$ is assumed to directly split in a scalar shading component $\tilde{s}(\boldsymbol{x})$ and a reflectance component $\tilde{\boldsymbol{r}}(\boldsymbol{x})$,

$$\boldsymbol{p}(\boldsymbol{x}) = \tilde{s}(\boldsymbol{x})\tilde{\boldsymbol{r}}(\boldsymbol{x}) \ . \tag{5.16}$$

As the righthand side of Eqn. 5.16 contains one variable more than the left hand side, additional constraints are required to find a solution. A common additional constraint is that shading changes smoothly, while transitions in reflectance (due to texture or object boundaries) are typically sharper. Current state-of-the-art intrinsic image decomposition algorithms mainly differ a) in the definition of the boundary between shading and reflectance edges and b) the propagation of these edges across regions without edge information.

As a basic building block, the retinex algorithm is commonly used, originally proposed by Land and McCann [Land 71] (see also Sec. 4.2). The algorithm can be seen as an estimator for the reflectance image: large (color-) edges are preserved, as they are assumed to carry image information, while small differences are suppressed. The difference between the retinex output and the original image can be used as an estimate for the shading image. Given the fact that retinex was originally proposed in 1971, it is surprising that it is still of relevance to the community. It is a simple, versatile method that is broadly applicable. However, one recent criticism with retinex was that its generality prevents it from providing high quality solutions to the in-

trinsic image decomposition problem in particular. Tappen *et al.* [Tapp 05] proposed one of the first "modern" algorithms for intrinsic image decomposition. In this work, the authors propose to find the reflectance and shading edges via classification of the image derivatives. Conflicting classifications are resolved via belief propagation. In a follow-up paper, Tappen *et al.* [Tapp 06] improved over this result by introducing a weighting function to the edges, instead of considering each edge of equal importance. Shen *et al.* [Shen 08] proposed to extend retinex with a constraint that small patches of equal chromaticities should have the same values in the reflectance component $\tilde{\boldsymbol{r}}(\boldsymbol{x})$. Bousseau *et al.* [Bous 09] proposed an interactive approach to intrinsic image decomposition. Thus, the classification of regions of different reflectance is mostly transferred to the user. Based on the user annotations, the estimation of the shading image is used with the so-called matting Laplacian, originally proposed by Levin *et al.* [Levi 08].

Grosse *et al.* [Gros 09] presented a comparison of different techniques for intrinsic image decomposition, together with a carefully captured dataset, consisting of 20 isolated objects. One example image, "teabag2", is seen in Fig. 5.15. The dataset comes with a protocol for evaluating the algorithm, which subsequently became a standard for the community. Recently, Gehler *et al.* [Gehl 11] and Shen and Yeo [Shen 11] developed two algorithms that are based on energy minimization schemes. In the case of Gehler *et al.* [Gehl 11], the objective function is to minimize a sum of three energy terms, which are computed globally on the image. These terms consist of a) a shading prior that penalizes sharp changes in the shading, b) a retinex-based decision function on whether an edge is due to shading or due to reflectance, and c) a reflectance sparsity prior, which rewards a small total number of colors in the reflectance image. The reported quantitative results on the dataset by Grosse *et al.* can be considered state-of-the-art.

The second method, by Shen and Yeo [Shen 11], is in its core also inspired by retinex. The reflectance image is obtained from a theoretically elegant L_1-constrained least squares optimization. Within this optimization function, the image is decomposed with a weighted red-black wavelet (see Uytterhoeven and Bultheel [Uytt 97]). This is a so-called second-generation wavelet, which is subsequently used to group similar spatially connected chromaticities. One of the assumptions by Shen and Yeo is, that these chromaticities share the same reflectance, and all variations within such a region are due to shading. Further terms in the objective function ensure the sparsity of the total number of the reflectances. Quantitative results of this method on the dataset by Grosse *et al.* [Gros 09] are also highly competitive.

5.3.3 Incorporating Geometry with Intrinsic Contours

For this work, we experimented with the last two methods that were presented in the previous subsection, i. e., by Gehler *et al.* [Gehl 11] and Shen and Yeo [Shen 11]. In the case of the method by Gehler *et al.*, the implementation is publicly available and could be adapted to our application. For the method by Shen and Yeo, we aimed to carefully reimplement the method. Validating this reimplementation, however, we were not able to achieve the reported performance on the dataset by Grosse *et al.* [Gros 09]. Thus, we report in this sections only results on the method by Gehler *et al.*.

Figure 5.16: Example results of the method by Gehler *et al.* on data with less constraints than the dataset by Grosse *et al.* [Gros 09]. Left: input image, right: shading image. The shading image still contains considerable brightness differences between skin and shirt.

This method showed also state-of-the art performance on the dataset by Grosse. Unfortunately, we were not able to obtain a similar performance on real-world images. For example, Fig. 5.16 shows the example output of the method by Gehler *et al.* on two of our benchmark images. In both cases, the input images contain large brightness differences between the skin and the shirts of the subjects. The resulting shading images, after applying the intrinsic image decomposition, contained in many cases severe artifacts from the underlying object color (shown right of the input images in Fig. 5.16). Thus, we were not able to use the output of these methods as a preprocessing step for neutralizing the brightness differences induced by different materials. As a consequence, we developed a new method, which is explained in this section.

For extending the range of useful surface normals within the method of Johnson and Farid [John 07a], two additional facts can be used. First, limited information about the image geometry is available, namely the user-annotated contours of the object of interest. Second, we do not require a full shading image. Instead, it suffices if only the contours of an object exhibit the characteristics of a shading image. Thus, an *intrinsic contour* algorithm (in contrast to an intrinsic image algorithm) already serves our purpose. In our proposed method, only the reflectances along the object contour are considered. Thus, our goal is to isolate the shading differences along the contour. This is why we propose to call this approach intrinsic contour estimation.

It turns out that estimates of the contour normals are extremely valuable to the recovery of intrinsic contours. To illustrate this, we assume a single, distant point light source, and two surface patches at the object contour with two different albedos ${}^1\rho_c$ and ${}^2\rho_c$ in channel c. Adopting the simplified Lambertian model from Eqn. 4.4 (see page 60), the two observed pixel intensities in channel c are

$$p_c(\boldsymbol{x}_1) = \cos(\theta_1(\boldsymbol{x}_1)) {e_c}^1\rho_c(\boldsymbol{x}_1) \tag{5.17}$$

$$p_c(\boldsymbol{x}_2) = \cos(\theta_2(\boldsymbol{x}_2)) {e_c}^2\rho_c(\boldsymbol{x}_2) \ , \tag{5.18}$$

Where e_c denotes the illuminant intensity in channel c, and $\theta_1(\boldsymbol{x}_1)$ and $\theta_2(\boldsymbol{x}_2)$ denote the geometry factors of the pixels $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Assuming a single distant light source,

the incidence direction can be assumed to be the same for $p_c(\boldsymbol{x}_1)$ and $p_c(\boldsymbol{x}_2)$. If additionally the contour normals at both points are equal, it follows that

$$\theta_1(\boldsymbol{x}_1) = \theta_2(\boldsymbol{x}_2) \ , \tag{5.19}$$

since θ is, the angle between the direction of the incident light and the surface normal. Using this observation, the differences in $p_c(\boldsymbol{x}_1)$ and $p_c(\boldsymbol{x}_2)$ can be directly explained as albedo differences, i. e.

$$\frac{p_c(\boldsymbol{x}_1)}{p_c(\boldsymbol{x}_2)} = \frac{{}^1\rho_c}{{}^2\rho_c} \ , \tag{5.20}$$

if the surface normals of $p_c(\boldsymbol{x}_1)$ and $p_c(\boldsymbol{x}_2)$ are equal.

5.3.3.1 Algorithm overview

Equation 5.20 can be directly exploited in an algorithm. Assume that a contour has been annotated by the user. We use a polynomial of order two to find the normals on this contour (except of implementation details, this part is identical with the original algorithm by Johnson and Farid [John 07a]). To determine areas of the same albedo, we can either request the user to annotate these differences, or to use an arbitrary clustering algorithm for grouping the pixels by their chromaticities. Another option would be to introduce a gradient-based criterion, similar to prior work in intrinsic image decomposition. In our implementation, we used the k-means clustering algorithm, and set $k = 5$ in the experiments. In Sec. 5.3.5.2, we used manual annotation of the different clusters.

For every group of pixels with approximately similar chromaticity, we create a chart, where the x-axis denotes the orientation of the normals, subdivided in 36 bins (i. e., steps of 10 degrees). At the y-axis, the intensities of the respective pixels is noted. Ideally, for the albedo to be neutralized, all intensities in one bin should be equal. Intensity adjustment can now be conducted, if at least one bin contains pixels from different groups. The correction factor per group is determined from the solution of a linear system of equations. This step is detailed in the next subsection. The normalized intensities are then directly used as input to the method of Johnson and Farid.

5.3.3.2 Intensity Adjustment

Under the assumption of a distance light source, if two diffuse pixels with different albedo share the same surface normal (see Eqn. 5.19), then Eqn. 5.20 shows that the quotient of the pixel intensities corresponds to the quotient of the albedos. Interestingly, the quotient itself is independent of the orientation of the surface normal. Thus, the quotient relates all pairs of pixel intensities that were observed on one albedo to a second albedo, as long as this pair shares the same surface normal.

We use this constraint to set up a linear system of equations. Rewriting Eqn. 5.20, one obtains

$$\frac{p_c(\boldsymbol{x}_1)}{{}^1\rho_c} - \frac{p_c(\boldsymbol{x}_2)}{{}^2\rho_c} = 0 \ , \tag{5.21}$$

Every pair of intensities with the same surface normal can form such a line in a linear system of equations. Note that the unknown albedos ${}^1\rho_c$ and ${}^2\rho_c$ are constant in all

lines. Thus, these equations can be solved for ${}^1\rho_c$ and ${}^2\rho_c$, using e. g. singular value decomposition.

In practice, we relax the task of finding pixels with the same surface normal a bit. To do so, we examined two approaches. First, we subdivided the range of possible angles of the normals in 24 to 36 bins. All pixels from different materials within the same bin were selected to determine the solution for Eqn. 5.21. However, for some test cases, we observed little angular overlap between the surface normals. A sharp binning can lead to situations where Eqn. 5.21 can not be set up, due to the lack of similarly oriented normals. To alleviate this problem, we decided to use a soft threshold as second approach. In this case, a line in Eqn. 5.21 is weighted by a weight w_{ICE}, which is computed from the angle between the surface normals as

$$w_{\mathrm{ICE}} = \begin{cases} \exp\left(\frac{\arccos\left(\boldsymbol{\nu}(\boldsymbol{x}_1)^{\mathrm{T}}\boldsymbol{\nu}(\boldsymbol{x}_2)\right)^2}{\sigma_{\mathrm{ICE}}^2}\right) & \text{if } \arccos\left(\boldsymbol{\nu}(\boldsymbol{x}_1)^{\mathrm{T}}\boldsymbol{\nu}(\boldsymbol{x}_2)\right) \leq 2\sigma_{\mathrm{ICE}} \\ 0 & \text{otherwise} \end{cases}, \quad (5.22)$$

where w_{ICE} denotes a suitably chosen weighting factor. In our implementation, we empirically determined σ_{ICE} an angle of 18.75°. The threshold for setting w_{ICE} to 0 is derived from σ_{ICE} by interpreting it as a cutoff threshold for the tail of Gaussian curves. The area of a Gaussian curve outside a range of two times the standard deviation corresponds to about 5% of the total area, which appeared to be a reasonable threshold for our application. This implies that in order to be able to correct the intensities between two materials, there must be at least one pair of surface normals between two materials whose angle is within a range of $2 \times \sigma_{\mathrm{ICE}} = 37.5$ degrees. For the computation of the intensity correction, we use the green channel of the image, as Johnson and Farid also propose to use the green channel for their method [John 07a].

5.3.4 Dataset

To evaluate our method, we collected a dataset consisting of 10 subjects with a resolution of 3888×2592 pixels. under illumination from different angles. Three light bulbs were attached on a scaffold. Assuming a chest height of $1.5m$ above the ground, the angles between the light sources and the subjects were chosen as 0° (i. e., incident light from the right, mounted at a height of $1.5m$), 45° (i. e., incident light from the top right) and 90° (i. e., incident light from above the subject). Every lighting situation was captured twice, once with only one of these three light sources switched on, and once with additional diffuse room light. To diffuse the room light, a light source was oriented towards the opposite wall of the room (behind the photographer), such that a relatively smooth illumination was obtained. Figure 5.17 shows an example image with and without background illumination. For our evaluation, we excluded the images that were captured without background illumination, for two reasons. First, upon examination of the images, it turned out that contrast is higher and shadows are considerably harder in the images without background illumination. Thus, images with background illumination make a more natural impression. The second, more technical reason to exclude images without background illumination was that all algorithms performed surprisingly well on these images, such that a meaningful comparison was not possible there.

Figure 5.17: Activated background illumination versus single light source. The contrast is considerably lower by additional background light (left) than without (right). At the same time, the left image looks more realistically.

In the top row of Fig. 5.18, samples for the images with background illumination are shown, under angles of 0°, 45° and 90°. In the bottom row, the labelled images for the image under 45° are shown. For each image, we created a full mask (left), in order to make sure that the surface normals point outside of the object, i. e. towards the black part of the mask. In the middle and on the right, two object contours are shown. The contour in the middle satisfies the assumptions of the original method by Johnson and Farid [John 07a]: it denotes a single material, and spans a large angular range for the surface normals. In all cases, we aimed to select the best suited material for the original method. On the right, we additionally marked contours of a second material, the skin of the subject. This contour serves as input to our proposed algorithm. Pictures of all subjects are shown in the appendix in Fig. E.1 on page 191.

One thing to note is that the contours have to be carefully annotated. It is not straightforward for a non-expert to conduct this annotation, due to a number of hidden assumptions. First, only those contours correspond to the computational model that belong to smooth surfaces in three dimensions. For instance, the opening of the short sleeves of the shirt are not smooth in the z-direction, and thus have to be excluded. Additionally, folds and self-shadows on the cloth can lead to severe estimation errors. Textured surfaces, like the hair of the subject in Fig. 5.18 also have to be excluded, as such contours can also lead to high estimation errors. For the same reason, we noted in several cases that also strong hair on the arms of the subjects had to be excluded.

5.3.5 Evaluation

We split the evaluation in two parts. First, we compare the performance of the original method with two variants of the proposed preprocessing, intrinsic image decomposition by Gehler *et al.* [Gehl 11] and intrinsic contour estimation. Then, we investigate less constrained cases, and illustrate the advantages of intrinsic contour estimation.

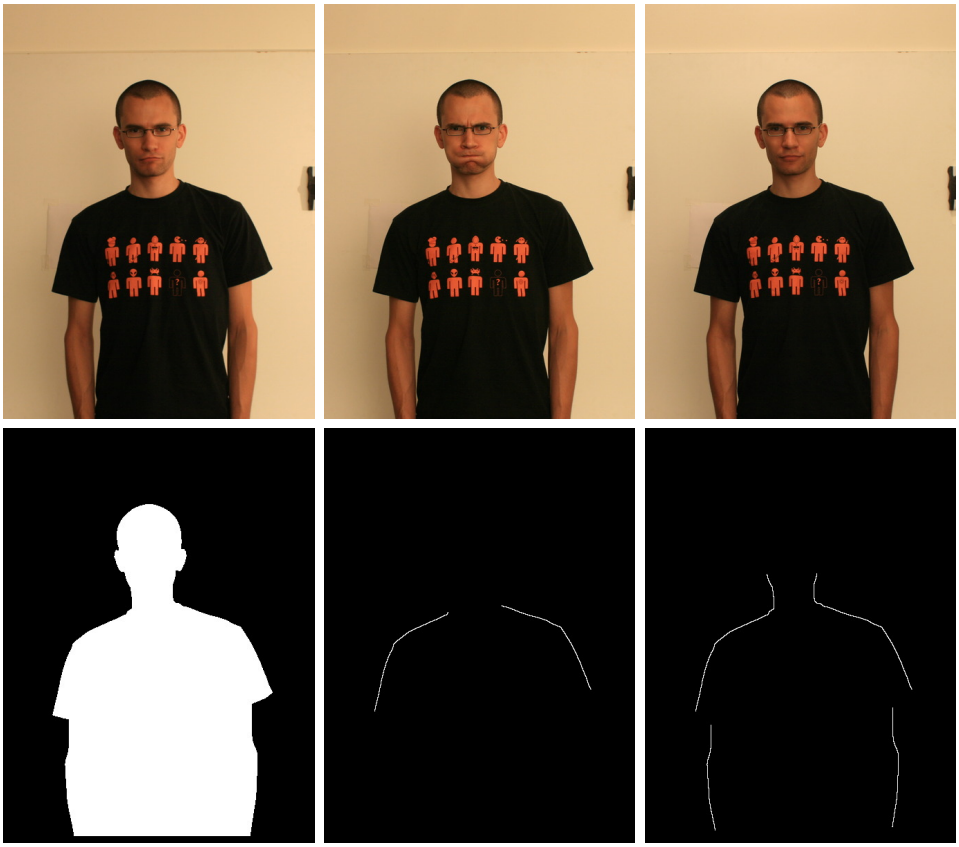


Figure 5.18: Example images from our intrinsic contour dataset. In the top row from left to right: example setup with incident light from 0°, 45° and 90° with activated background illumination. In the bottom row, the labeled data for the middle image is shown.

5.3.5.1 Laboratory Data

For the evaluation of the laboratory data, we downscaled the images to a size of 900×600 pixels. In an initial experiment, we verified that the accuracy of the methods is not significantly influenced, but the runtime of the method by Gehler *et al.* greatly benefits from the downscaling.

We used the original method by Johnson and Farid [John 07a] without any preprocessing, and the same method using as preprocessing steps the intrinsic image composition algorithm by Gehler *et al.* [Gehl 11], and the proposed intrinsic contour estimation. All three methods were applied once on the shorter contours that enclose only a single material, and once on the full contours containing multiple materials. As benchmark images, we used the 3 images per subject with activated background illumination, where the angle of the dominant incident light is 0°, 45° and 90°. Thus, we evaluated in total on $10 \cdot 3 = 30$ images. For quantitative evaluation, we investigated the distribution of the incident light from all directions, which can be directly computed from the output of the method by Johnson and Farid [John 07a].

An example distribution can be seen in Fig. 5.19. On the left, the (single-material) normals for subject 7 under 45° illumination are shown. On the right, the intensity

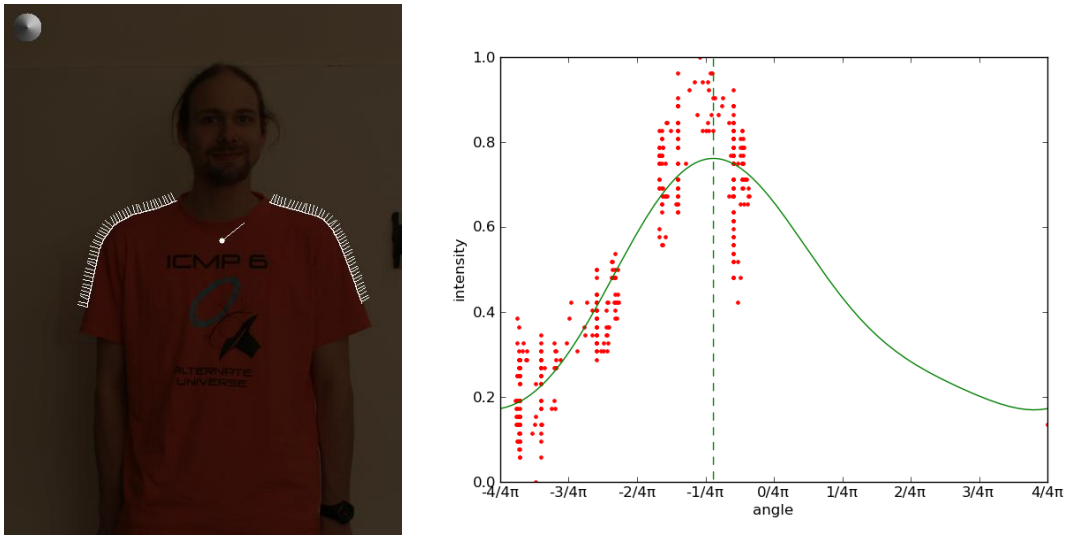


Figure 5.19: Left: Single-material contour normals on subject 7, and reprojected estimated direction of the dominant light source. Right: pixel intensities per angle of the normals (red), and estimated intensity curve of the incident light (green).

distribution per normal angle is plotted in red. The green curve denotes the intensity distribution, as computed with the method by Johnson and Farid. The vertical dashed line indicates the peak of this curve. For visual verification of the result, we reprojected the orientation of the peak in the input image on the left, shown as small stroke on the chest of the subject. This peak should theoretically coincide with the angle of the dominant light source[5]. For the present example, this is almost exactly the fulfilled.

We computed the mean and median errors between the peak in the distribution and the ground truth direction of the dominant illuminant. Additionally, we defined a "successful" recovery of the dominant illuminant as estimates with a lower error than 22.5°, i. e. half of the angular difference between two illuminants.

Table 5.2 shows the per-algorithm performance for this experiment. In the columns on the left, the performance on the shorter contour, consisting of a single material, is shown. All three methods perform similarly, with median errors between 9.1 and 10.9 degrees. Also the number of images where the estimates are a maximum of 22.5° off are approximately the same, ranging from 24 to 26 cases. This validates our preprocessing, in the sense that the preprocessing does not considerably weaken the results if only observations from one material are available. In the right columns, we present the results on multiple materials. Not unexpected, the original method can not handle this case. The 33% of the cases that lie within an angular error of 22° are only by chance correct. However, we consider this experiment interesting as a baseline for the results using the preprocessing by Gehler *et al.* [Gehl 11]. In this case, we obtain for only 3 more images an error of less than 22.5° compared to the original method. This clearly demonstrates that in most cases, this algorithm for intrinsic

[5]Note that this assumption is only approximately correct, as the method by Johnson and Farid assumes distant light sources, which was not realized in our laboratory setup.

	Single-colored contour			Multi-colored contour		
	Median	Mean	Within 22.5°	Median	Mean	Within 22.5°
Original	10.7	13.6	25/30 (83%)			
Gehler	9.1	12.5	**26/30 (86%)**			
ICE	10.9	14.1	24/30 (80%)			
Original				40.2	56.5	10/30 (33%)
Gehler				33.0	50.7	13/30 (43%)
ICE				12.6	13.0	**26/30 (86%)**

Table 5.2: Median and mean angular error on the lighting environment database, and the number images for which the estimation error of the dominant light direction was less than 22° degrees. In the left columns, the best single-colored contour per image is used, in the right columns, mixed-color contours are used.

image decomposition is not able to achieve a sufficiently accurate shading image for our application. Finally, intrinsic contour estimation obtains on multiple materials again comparable results to the single-material cases. The median angular error is slightly increased to 12.6°, but overall 86% of the estimation errors lie within 22.5°. The per-image results of the best variant per method are shown in the appendix, in Tab. E.1 on page 190.

For the remainder of this section, we show the behavior of the proposed algorithm, and some typical failure cases. One pathological case of the original method is the lack of surface normals pointing in the direction of the dominant light source. Figure 5.20 shows such a situation. The dominant light source illuminates subject 4 from an angle of 0°. However, the angles of all surface normals lie between approximately 22.5° and 170°. Consequently, the estimated peak is located at 23.2°, which comes from the fact that the distribution of the normal angles provides no information between −170° and 20°. Reducing this information gap is the main objective of the proposed intrinsic contour estimation. However, using ICE on this image reveals another failure point, this time for the proposed algorithm. Figure 5.21 shows the intensity plots when ICE is applied on the image in Fig. 5.20. Both plots contain two groups of pixels: the pixels from the shirt are plotted in red, the pixels from the skin are plotted in green. On the left, the raw input data is plotted, where the skin pixels appear much brighter than the shirt pixels. Also the range of input normals is extended towards the critical angle of 0°. On the right side, the brightness distribution is shown after applying ICE. Qualitatively, the adjustment looks correct and plausible. However, for a single surface angle, the spread of the intensities was greatly increased for the dark shirt pixels. One reason for that might be that the noise level for cameras is generally higher for very dark pixels. Consequently, the estimate does not improve in this case over the original method. Addressing this issue is subject to future work. In principle, a reasonable way to add greater robustness to noise may be to directly integrate the proposed ICE algorithm into the core of the original algorithm by Johnson and Farid, i. e. into Eqn. 5.13 on page 139.

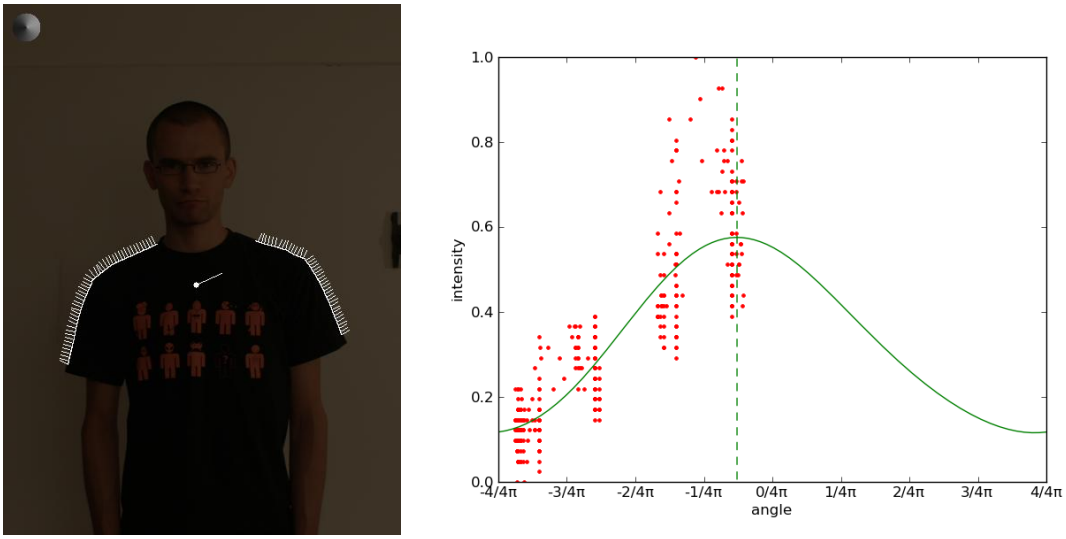


Figure 5.20: Failure case for single-material intensities and the estimated intensity curves. Here, subject 4 is shown under an illumination angle of 0°. However, none of the surface normals points into this direction, which leads to an estimation error of 23.2°.

5.3.5.2 Advantages of Intrinsic Contour Estimation

In practice, annotating contours that provide reliable estimates is surprisingly difficult. When pictures of people in more natural poses are examined, it is often the case that the original algorithm lacks reliable data. Consider the left example image in Fig. 5.13 on page 136. The contours of both persons are partially occluded, mainly due to their motion, bags and additional clothing. From the viewpoint of a user, it is difficult to reliably mark useful contours contours along the subjects, if only a single material may be used. Figure 5.22 shows such example markings. On the left, we limited ourselves to a single material, namely the pullover of the woman, wrapped around the hips, and the shirt of the man. Estimating the direction of the dominant illuminant on these segments yields widely diverging angles. In the right part of the image, a second material was added per person, namely the skin of the woman and the trousers of the man. In this case, the estimates of the dominant light direction lie more closely together, though there is still a considerable error, at least in the estimate for the woman.

We discuss further cases outside of our laboratory data, to show that an important advantage of ICE lies in the robustness, in particular when it comes to user interaction. When setting up the environment for capturing the laboratory dataset, we made also a test image that is shown in Fig. 5.23. In this case, background illumination was activated, and the dominant light source was mounted at an angle of 45°. The second and third image of Fig. 5.23 show the results of annotating either admissible skin regions or admissible shirt regions. The estimation results show a typical drawback of the original method: As in both cases, the normal angles are oriented in exactly opposite directions, the estimated dominant light direction is pointing in the direction of the brighter half of the normals. Thus, a user can not be sure that the direction of the illuminant is correctly estimated, if the surface normals exhibit such

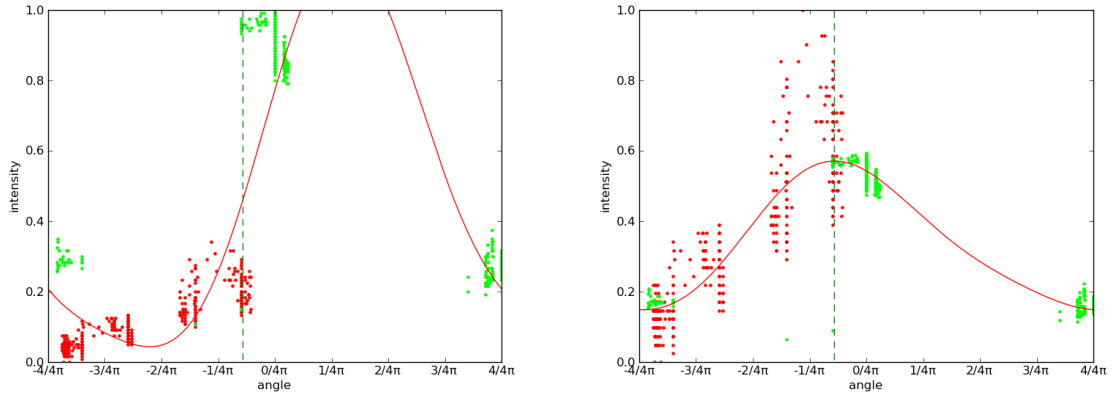


Figure 5.21: Failure case for multi-normal intensities and the estimated intensity curves, using the same image as in Fig. 5.20. Left: shirt pixels are plotted in red, skin pixels are plotted in green. Right: intensity-adjusted plot, using the ICE algorithm. The relatively high noise level in the black pixels of the shirt dominates the skin pixels.

a special structure. Conversely, when incorporating the intrinsic contour estimation, both materials can be added. In this case, normal angles from almost all directions are available.

Figure 5.24 shows a similar case. For the shirt and the jacket, the normals do not point in opposite directions. Instead, the normals of the shirt are apparently pointing away from the light source, while the contour normals on the jacket roughly point towards the light source. However, the angular support for both materials

Figure 5.22: Annotations of single-material contours versus multi-material contours on the image shown in the left image of Fig. 5.13.

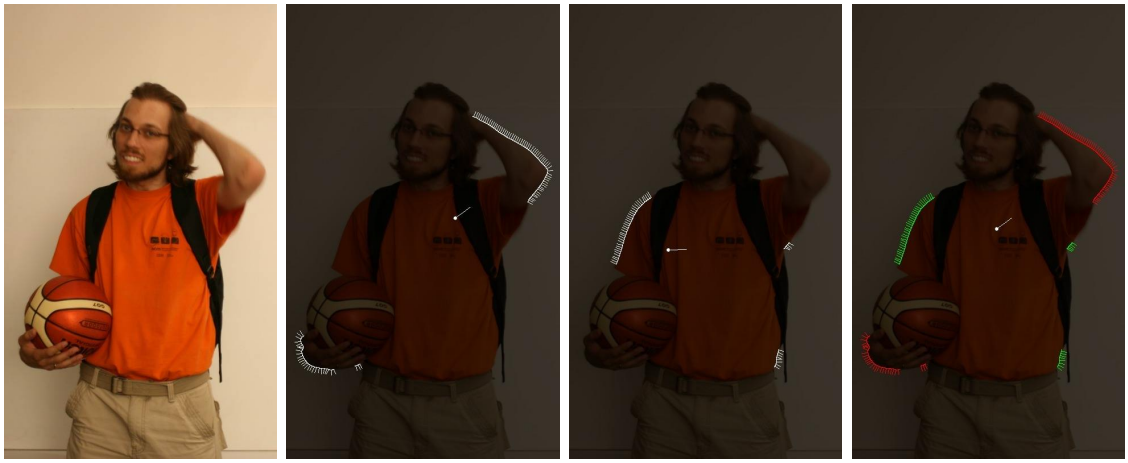


Figure 5.23: Annotations of single-material contours versus multi-material contours. In this case, the user can not make a confident choice for single-material contours (see text for details).

is rather small, and consequently the solution to Eqn. 5.13 on page 139 is severely underconstraint. In this sense, it is not surprising that the estimated direction of the light source is in both cases simply pointing towards the center of the normals. Conversely, if contours from both materials can be selected, the joint normal angles span a range of almost 180°, which greatly increases the confidence in the estimate.

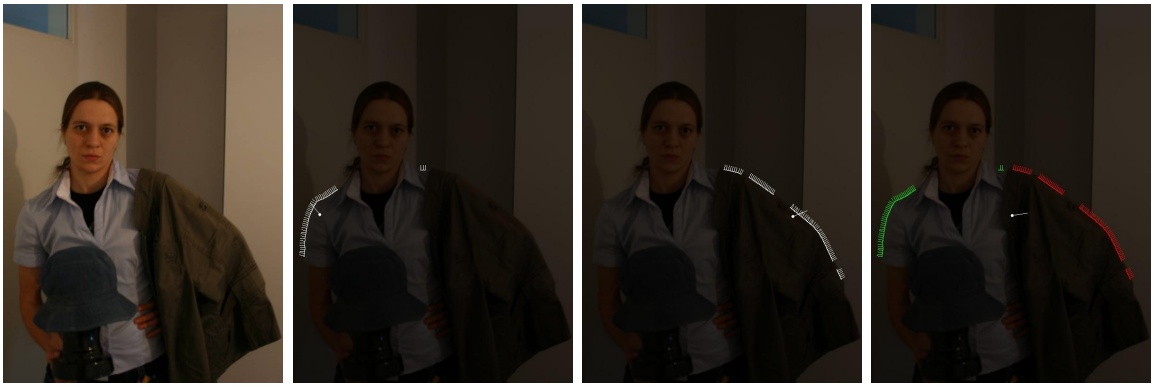


Figure 5.24: Annotations of single-material contours versus multi-material contours. In this case, both the shirt and the jacket provide only small angular support, which also clearly shows in the single-material estimates. The joint estimate, however, can be assumed to be approximately correct.

Chapter 6

Outlook

The results in this thesis open a number of opportunities for further research. We point out a number of aspects, which we consider most promising, or most interesting to the scientific community.

Benchmarking in Image Forensics As image forensics is a relatively new research direction, there barely exists standartized data for benchmarking forensic algorithms. Consequently, most comparatative evaluations are conducted on individually selected benchmark data, which makes it difficult to get an impartial and thorough overview of the expected relative performance of the algorithms.

One reason for this lack of data might be a modeling problem. Every algorithm in image forensics makes implicit assumptions about the behavior and techniques used by some unknown manipulator. For instance, in this work, we assume that additive Gaussian noise, JPEG compression, rotation and scaling approximate typical processing steps of a copy-move forgery. Apart from the fact that we conducted several successful experiments on real copy-move forgeries (not presented in this work), we do not *know* whether our data model is a good approximation of the truth. Thus, it would be interesting to investigate the creation of image manipulations from the viewpoint of a digital artist. For instance, image professionals and amateurs could be recorded while performing a manipulation that is described only on a very high level. For instance, the task could be to "show that politician A had contacts with politician B". The artistic implementation should be completely up to the user. We believe that the research for forensic methods could greatly benefit from the analysis of such data.

Copy-Move Forgery Detection Recent algorithms for copy-move forgery detection are also able to detect copied regions that have undergone affine transformations. As affine transformations are typically implemented as interpolation operations, forensic resampling detectors are in principle also able to detect this subclass of copy-move forgeries. At the same time, resampling detectors are typically computationally much more efficient than copy-move forgery detectors. On the downside, resampling detectors are known to be highly susceptible to noise. Thus, it would be interesting to further investigate reasonable practical application boundaries be-

tween these two approaches. One should also consider creating a hybrid algorithm that incorporates the advantages of both methodologies.

Exploitation of Compression Artifacts Current state-of-the-art algorithms show good performance in discriminating regions with single- and double-JPEG compression. However, work on triple- or quadruple-compression is still largely unexplored. One first step in this direction has been done by Huang *et al.* [Huan 10]. However, further analysis would provide a complete study of compression artifacts. Additionally, it would be highly interesting to combine such an algorithm for n-times compression detection with a theoretical analysis on the discriminability of multiple subsequent recompression steps.

Estimation of the Illuminant Color Color constancy is a classical computer vision problem, in the sense that the separation of material and light color from a single image is severely underconstrained. However, if additional information is available, for instance the surface normals per pixel, the separation of illumination and material can become considerably easier. Interestingly, intrinsic image decomposition, photometric stereo, shape from shading and other computer vision tasks share this property: if one complementary cue is added, then these tasks can be solved with high accuracy.

Consequently, with the increasingly available computational power, it would be interesting to investigate approaches that do not consider color constancy as an isolated problem, but instead aim to jointly solve multiple computer vision tasks at the same time. Barron and Malick [Barr 12] recently proposed a method that aims at a joint solution for color constancy, intrinsic image decomposition and shape-from-shading. However, this work is just a first step, and we assume that the joint investigation of computer vision problems still offers many opportunities for further improvements.

Forensic Exploitation of the Illumination Color Color constancy algorithms are typically bound to restrictive assumptions in order to work properly, like a linear camera function, or non-overlapping color filters. In practice, however, the representation of color is highly dependent on the capturing device. Thus, it is currently unclear whether inconsistencies in the output of illuminant color estimators, are due to differently illuminated scenes, or (for instance) different camera response functions or different camera settings. This question plays a minor role for the result — color descriptors can be used to detect inconsistencies in digital forgeries. However, for the development of improved detection algorithms, this question can become important. Thus, as future work, one may want to investigate the color responses on a dataset captured with different camera models and different camera settings. Up to now, such a dataset does not exist: for instance, the color checker dataset by Gehler *et al.* [Gehl 08] is captured with only two different cameras, and a random selection of motifs. In the Dresden image database [Gloe 10], the same scenes are captured with different cameras and camera settings, but no color chart for objective color analysis is present in these scenes. To foster further research in forensic color cues, we require a dataset that combines the capturing conditions of the

Dresden image database with the presence of a color chart in the scene, as in the Gehler *et al.* [Gehl 08] dataset.

Such a dataset, would also facilitate the investigation of color in a broader context. For instance, one could consider the incorporation of cues on the Bayer-pattern interpolation of the camera (see, e. g., [Pope 05]) or the camera response function (see, e. g., [Ng 09a, Ng 09b]) in order to verify the plausibility of color distributions in the scene under investigation.

Intrinsic Contours for Estimation the Lighting Environment We presented a preprocessing step for the estimation of lighting environments. A direct improvement would be to directly integrate the intrinsic contour algorithm into Eqn. 5.9 on page 138. To achieve this, one could additionally exploit the constraint introduced by Basri and Jacobs [Basr 03] that Lambertian surfaces can be represented in a nine-dimensional subspace. For the albedo coefficients, this finding can be used to constrain the rate of change for intensities with similar surface normals.

An additional promising direction of investigation is the combination of the direction of the illuminant with the illuminant color. Geometry, illumination and surface color provide complementary information. Thus, it is reasonable to assume that a method that unifies both high-level forensic approaches is much more effective than considering each of them in isolation.

Chapter 7

Summary

Digital photography almost completely replaced analogue pictures. At the same time, advanced image processing tools make it straightforward to edit or modify digital images. In court, for police agencies, for insurance or media companies, this raises the challenge of discriminating original images from malicious forgeries. In image forensics, researchers aim to provide computational tools to support human experts in deciding about the authenticity of an image. In many images, one can not assume that a security scheme, for instance a digital watermark, has been embedded by the capturing device. Thus, *blind* image forensics investigates authenticity criteria that can be obtained from either a) detecting artifacts from a particular tampering operation or b) confirming the authenticity of an image by verifying artifacts introduced during the image formation process.

In this thesis, we present methods for both of these directions. We propose and improve statistical cues that indicate a particular tampering operation, and investigate illumination-based, physical properties of the depicted scene. To achieve this, we also developed methods for estimating the color of the light in scenes under inhomogeneous illumination.

The first part of this thesis covers the investigation of algorithms for copy-move forgeries detection (CMFD). In this scenario, it is assumed that one or multiple regions are copied and pasted within the same image. Additionally, it may be the case that the pasted copies have undergone further processing, like rotation, scaling or the addition of noise. Prior to this thesis, a large number of feature sets and algorithms has been proposed for CMFD. The general approach is to extract feature descriptors from local image windows. Then, descriptors with a small distance in feature space are matched. If the regions from where these matched descriptors have been extracted cluster to larger areas, it is assumed that the matches within these areas belong to a copy-move forgery. We review existing work, and cast these methods within a unified CMFD pipeline. Upon thorough investigation of the three most important steps in this pipeline, we propose novel solutions and recommend concrete design choices for the design of CMFD techniques. We first show that matching descriptors with approximate nearest neighbors (instead of lexicographic sorting) improves the recall for 9 out of 13 block-based feature sets by at least 3 points, in the case of the PCA feature set, this improvement reaches even 17 points. For all examined features, the detection precision is slightly decreased. However,

except for the features Hu, SVD and DWT, the loss in precision is smaller than the gain in the recall. Thus, for feature matching, we recommend the employment of the approximate nearest neighbor search.

For grouping matches where the copied parts have been rotated or scaled, we propose the algorithm *Same Affine Transform Selection* (SATS). It simultaneously performs a grouping of feature matches and an estimation of the affine transform between matched feature blocks. Experimental results show that SATS reliably detects copied regions, without raising the false positive rate.

Finally, we conducted a large-scale comparison of 15 proposed feature sets. We used the same 13 block-based features as before, and additionally Sift and Surf as keypoint-based descriptors. Our analysis shows that keypoint-based methods are clearly superior with respect to computational complexity, and if the copied region has undergone large amounts of rotation and scaling. Several block-based features, on the other hand, outperform Sift and Surf in the precision of the detected regions if the source and the copy regions only moderately differ. Consequently, keypoint-based features are well-suited for a quick online screening of a large number of images, while block-based features can be recommended for a more in-depth analysis of single images.

In the third chapter, we present a pattern-recognition approach to automatically exploit the so-called JPEG ghost observation by Farid [Fari 09]. It allows the discrimination of regions that have been once compressed with the JPEG algorithm from regions that have been doubly compressed. Knowing the number of times a region has been compressed can serve as indirect evidence for tampering: assume for instance, that a JPEG-compressed image is retouched and then recompressed in the JPEG format. In this case, the retouched region appears to be only once JPEG compressed (due to the modification of the original image content), while the remainder of the image appears as doubly compressed. The JPEG ghost observation can be used to detect such cases if the second JPEG compression quality is higher than the first one. In our proposed method, we recompress the images with different JPEG quality parameters. The per-region differences between the input image and the recompressed versions can be plotted as curves. We define a six-dimensional feature vector on these curves. Classification results on these features are highly competitive. For instance, we achieve a sensitivity and specificity of more than 0.8 on image windows of 8×8 pixels and a quality difference between first and second compression of only 5 points. If the application scenario allows larger image windows or larger differences in the compression qualities, sensitivity and specificity can be further improved to rates higher than 0.9, using the AdaBoost classifier or Random Forests. These high detection rates show that the tedious, error-prone process of manual assessment, as suggested in the original work [Fari 09], can be reliably replaced by our automated algorithm.

Besides these statistical approaches, we also investigate the physics of illumination as a cue for image tampering. As a foundation for these methods, we thoroughly study methods for estimating the color of the illuminant (see Chap. 4). The biggest contribution of our approach is that, unlike prior work, we do not assume globally uniform illumination. We are among the first researchers to estimate the distribution of non-uniform illumination within a single scene. This task contains two challenges.

First, the number and color of n illuminants has to be estimated (instead of a single illuminant). Second, the spatial distribution of the illuminants has to be determined (which we refer to as the *localization problem*). We approach this multi-illuminant problem in several steps. First, we present two ways to create ground truth for scenes under non-uniform illumination. One approach relies on repainting the scene in gray to obtain pixelwise ground truth. This is only feasible for laboratory scenes, due to the destructive nature of the process. As a second, non-destructive method, we propose the computation of the influence of every illuminant from a set of images. Given multiple captures of the same scene, exposed to different illuminants, we demonstrate how to recover accurate pixelwise ground truth. The resulting images from both approaches are used to evaluate different algorithms for multi-illuminant estimation. First, we investigate whether it is possible to downscale the spatial support for classical color constancy algorithms, and thus obtain a de-facto multi-illuminant method. On our gray-painted dataset, Random Forest Regression on several outputs of single-illuminant estimators achieved a median error of 4.1°.

However, one limitation of this approach is that the spatial neighborhood of pixels is not taken into consideration for the localization problem. To address this, we developed a physics-based method that clusters local illuminant estimates into 3 to 5 coarse regions. A single-illuminant variant of this estimator performs well on the publicly available benchmark datasets by Barnard *et al.* [Barn 02c]. Qualitative results on multi-illuminant scenes look reasonable. One drawback of this approach is that finding the illuminant colors and the solution of the localization problem is disconnected. Thus, we present a third approach based on Conditional Random Fields, which jointly solves the estimation and localization problem. To evaluate this method, we use a novel dataset that is based on the second algorithm for ground truth computation. A comparative analysis to the method by Gijsenij *et al.* [Gijs 12b] and different single-illuminant estimators shows the efficacy of the proposed method. Ultimately, we obtain a median angular error of 2.58° on laboratory data, and a median angular error of 3.32° on real-world images.

In the fifth chapter, we transfer the insights from illuminant color estimation to image forensics. We propose the use of a variant of the physics-based estimator that has been presented in the previous chapter. Its main advantage is that there are no assumptions on spatial context, and that its physical derivation makes failure cases easier to predict. The core of the method is to estimate the illuminant color locally, and to create a so-called illuminant map by reprojecting the illuminant estimates on the image. Although initial results look promising, one challenge is that the user has to assess the output of the illuminant estimator in order to decide whether the image has been tampered. This requires an human expert, and is therefore not suitable for broad application.

Consequently, we developed a machine-learning algorithm that automates the tampering decision. It operates on the proposed physics-based illuminant estimates, and additionally on local gray world illuminant estimates. To improve the tractability of the results, we limit this investigation to objects of approximately the same material, i. e. faces. Texture descriptors are extracted from the illuminant maps on the facial regions. These descriptors are classified with an SVM-Meta Fusion algorithm. Preliminary results show that an Area Under the Curve (AUC) of 78% can

be obtained, when the user input is limited to specifying a bounding box around a face.

In another study, we investigate the direction of incident light as a cue for image manipulation. Johnson and Farid [John 07a] proposed an algorithm to estimate and compare the directions of incident light within the image plane. A user is required to mark the contours of objects under investigation. The estimated lighting environments are then compared via computing the correlation of spherical harmonics coefficients. Although this method is theoretically strong, it suffers in practice from a number of constraints. One constraint is that for one object, only contours exhibiting the same material can be compared. To relax this constraint, we propose a computational method to bridge the differences in the object material without noteworthy additional user interaction. We call this algorithm *Intrinsic Contour Estimation*. Preliminary results show, that the algorithm works comparably well on single-material regions, but increases the robustness and confidence in the results when the object contours consist of multiple materials.

In summary, forensic researchers aim to provide a set of tools to decide whether an image is original or not. Depending on the situation, different subsets of these tools are applicable to the images under investigation. In this thesis, we provide novel techniques for a broad range of forensic approaches, from statistical tampering artifacts to consistency criteria on the physics of the scene. The algorithms differ in their maturity: while we consider the statistical approaches to be in principle ready for application in practice, the higher level approaches are, due to the largely increased complexity, still in research stage. However, the insights of this work show that the applicability of illumination-based methods can be further improved, towards the high robustness and user-friendliness that is required in practice.

Appendix A

Acronyms

AUC	area under the curve
CC	color constancy
CFA	color filter array
CIE	commission internationale de l'éclairage
CMFD	copy-move forgery detection
CRF	conditional random field
CV	computer vision
DCT	discrete cosine transform
DN	do nothing
DSP	digital signal processor
GE1	first-order gray edge
GE2	second-order gray edge
HOG	histogram of oriented gradients
IIC	inverse-Intensity Chromaticity
JPEG	Joint Photographic Experts Group
MAP	maximum a posteriory
MLP	multi-layer perceptron
MRF	Markov random field
PCA	principal component analysis
RANSAC	random sample consensus
RS-CMFD	rotated-scaled copy-move forgery detection
RGB	red, green, blue (color channels)
ROC	receiver operator characteristics
SATS	same affine transform selection
SD-JPEG	shifted-double JPEG compression
SIFT	scale-invariant feature transform
SVM	support vector machine
WP	White Patch

Appendix B

Notation

Symbol	Meaning
${}^{<index>}_{<algo>}< var >^{<illum>}_{<channel>}$	subscripts and superscripts before and after a variable *<var>* have different meaning: subscript *<channel>* indicates the color channel; superscript *<illum>* denotes the lighting conditions; subscript *<algo>* indicates some applied processing (e. g. Gaussian smoothing); superscript *<index>* denotes the index of the variable if multiple instances of the same variable are considered.
Temporary variables	
i, j	counters
n	upper limit of a counter, or number of elements
$\boldsymbol{a}$, $\boldsymbol{b}$	vectors of coordinates
α, β	matrix element, or other, temporarily required place holder
Images and pixels	
$\mathbf{I}$	image
$\boldsymbol{x}$, $\boldsymbol{x}$'	spatial positions in 2D or 3D
p	grayscale intensity of a pixel (i. e., from an image with one channel)
$\boldsymbol{p}$	color pixel (in this thesis, typically a three-component vector consisting of red, green and blue intensities)
$\boldsymbol{p}(\boldsymbol{x})$	color pixel at position $\boldsymbol{x}$
p_c	intensity of $\boldsymbol{p}$ at channel $c \in \{R, G, B\}$
$\breve{\mathbf{I}}$	image after illumination neutralization (von Kries model)
$\breve{\boldsymbol{p}} = (\breve{p}_R, \breve{p}_G, \breve{p}_B)^T$	pixel after illumination neutralization (von Kries model)
$\boldsymbol{\chi}(\boldsymbol{p}) = (\chi_R(\boldsymbol{p}), \chi_G(\boldsymbol{p}), \chi_B(\boldsymbol{p}))^T$	chromaticity of pixel $\boldsymbol{p}$ (brightness normalization of the pixel color)
Illumination	
λ	wavelength
$e(\lambda)$	spectral power distribution of the illuminant

$\boldsymbol{e}$	illuminant color (in this thesis, typically a three-component vector consisting of red, green and blue intensities)
$\boldsymbol{e}(\boldsymbol{x})$	illuminant color at position $\boldsymbol{x}$
$e_c(\boldsymbol{x})$, $c \in \{R, G, B\}$	red, green or blue illuminant intensity at position $\boldsymbol{x}$
$\tilde{\boldsymbol{e}} = (\tilde{e}_R, \tilde{e}_G, \tilde{e}_B)^T$	estimated illuminant color
$\tilde{\boldsymbol{e}}(\boldsymbol{x})$	estimated illuminant color at position $\boldsymbol{x}$
${}^i\tilde{\boldsymbol{e}}$	atomic illuminant estimate number i, in case that multiple estimates $1 \leq i \leq n$ are required
Material and reflectance	
$q_c(\lambda)$	wavelength-dependent camera response function for channel $c \in \{R, G, B\}$
$\rho(\lambda)$	albedo as a function of wavelength
$\boldsymbol{\rho}$	albedo (in this thesis, typically a three-component vector consisting of red, green and blue intensities)
${}^i\rho_c$	albedo number i in channel c
$\boldsymbol{\rho}(\boldsymbol{x})$	albedo at position $\boldsymbol{x}$
ρ_c	albedo at channel $c \in \{R, G, B\}$
θ	angle between lighting direction and surface normal
$S_d(\lambda)$	diffuse surface reflectance function
$S_s(\lambda)$	specular surface reflectance function
$\boldsymbol{s}^d(\boldsymbol{x})$	simplified diffuse surface reflectance function at position $\boldsymbol{x}$
$\boldsymbol{s}^s$	simplified specular surface reflectance function at position $\boldsymbol{x}$
$m_d(\boldsymbol{x})$, $m_s(\boldsymbol{x})$	geometric scaling for diffuse and specular reflectance at position $\boldsymbol{x}$
Evaluation measures	
n_{TP}	number of true positives
n_{TN}	number of true negatives
n_{FP}	number of false positives
n_{FN}	number of false negatives
prec	precision
rec	recall
spec	specificity
F_1	F_1-score
ϵ_{ang}	angular error between two vectors
$\epsilon_{\mathrm{Euclidean}}$	Euclidean error between two vectors
Chapter 1: Copy-Move Forgery Detection	
τ_{dist}	minimum Euclidean distance between two blocks, such that they can be matched as copy-moved blocks
τ_{minSize}	filtering threshold. If the cardinality of a group of copy-moved matches is smaller than τ_{minSize}, it is ignored
τ_{SATSdist}	maximum Euclidean distance of neighbored blocks for SATS region growing

$\tau_{\text{SATSminSize}}$	filtering threshold. If the cardinality of a SATS-group of copy-moved matches is smaller than $\tau_{\text{SATSminSize}}$, it is ignored
${}^{i}_{\text{M}}f$	ith pair of matched CMFD features
${}^{i}_{\text{M}}f_1$, ${}^{i}_{\text{M}}f_2$	First and second feature vector that belong to the matched pair ${}^{i}_{\text{M}}f$
$\mathcal{H}$	hypothesis set of matches for SATS
$\text{coord}({}^{i}_{\text{M}}f_1)$,	Pixel coordinates from where ${}^{i}_{\text{M}}f_1$ has been extracted
$\|\boldsymbol{a}, \boldsymbol{b}\|_2$	Euclidean distance between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
n_{B}	number of blocks in an image, from which detection features are extracted
n_{CB}	number of blocks in an image where the underlying image content is a copy-move forgery
n_{NH}	Size of the pixel neighborhood where SATS searches for matches
Chapter 2: JPEG Ghost Detection	
$\boldsymbol{I}_{q_1}$	JPEG image, compressed with JPEG quality q_1
$\boldsymbol{I}_{q_1,q_2}$	JPEG image, first time compressed with JPEG quality q_1, then with JPEG quality q_2
$\boldsymbol{D}$	Differences of the same image, with different JPEG compression history
$\boldsymbol{\Delta}$	Blockwise averaged version of $\boldsymbol{D}$
$\mathcal{Q}$	set of JPEG quality factors
${}_{\text{J}}f_i$	ith feature type for JPEG ghost detection
Chapter 3: Illuminant Color Estimation	
$\check{c}_c(\lambda)$ for $c \in \{R, G, B\}$	camera color response function for wavelength λ in channel c
${}_{\sigma}\boldsymbol{p}(\boldsymbol{x})$	color pixel $\boldsymbol{p}$, after Gaussian smoothing with standard deviation σ has been applied to the image
k	intensity scaling, e. g. used for generalized Gray World
$\partial^{n_{\text{GW}}}$	n-th derivative for Gray World
τ_{GW}	order of the Minkovski-norm in generalized Gray World
σ	standard deviation
$\mathbf{I}^{(\text{B};\emptyset)}$	image where only blue light (left side) is activated
$\mathbf{I}^{(\emptyset;\text{B})}$	image where only blue light (right side) is activated
$\mathbf{I}^{(\emptyset;\text{R})}$	image where only red light (right side) is activated
$\mathbf{I}^{(\text{B};\text{R})}$	image where two lights are activated: blue (left) and red (right)
$\boldsymbol{p}^{(\text{B};\emptyset)}(\boldsymbol{x})$	pixel of image $\mathbf{I}^{(\text{B};\emptyset)}$ at position $\boldsymbol{x}$
$\boldsymbol{p}^{(\emptyset;\text{B})}(\boldsymbol{x})$	pixel of image $\mathbf{I}^{(\emptyset;\text{B})}$ at position $\boldsymbol{x}$
$\boldsymbol{p}^{(\emptyset;\text{R})}(\boldsymbol{x})$	pixel of image $\mathbf{I}^{(\emptyset;\text{R})}$ at position $\boldsymbol{x}$
$p_c^{(\text{B};\emptyset)}(\boldsymbol{x})$	pixel intensity at channel $c \in \{R, G, B\}$ of image $\mathbf{I}^{(\text{B};\emptyset)}$

$\boldsymbol{e}^{(\mathrm{B})}$	ground-truth illuminant of the blue light source
$\boldsymbol{e}^{(\mathrm{R})}$	ground-truth illuminant of the red light source
w_{GT}	weighting factor between $\boldsymbol{e}^{(\mathrm{B})}$ and $\boldsymbol{e}^{(\mathrm{R})}$ for ground truth generation
$\boldsymbol{e}^{(\mathrm{B;R})}(\boldsymbol{x})$	interpolated ground-truth illuminant from $\boldsymbol{e}^{(\mathrm{R})}$ and $\boldsymbol{e}^{(\mathrm{B})}$ at position $\boldsymbol{x}$
$w(\boldsymbol{x})$	interpolation weight to compute $\boldsymbol{e}^{(\mathrm{B;R})}(\boldsymbol{x})$
$\theta^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$	geometry of image $\mathbf{I}^{(\mathrm{B};\emptyset)}$ (blue light left)
$\theta^{(\emptyset;\mathrm{R})}(\boldsymbol{x})$	geometry of image $\mathbf{I}^{(\emptyset;\mathrm{R})}$ (red light right)
$k^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$	intensity of image $\mathbf{I}^{(\mathrm{B};\emptyset)}$ (blue light left)
$k^{(\emptyset;\mathrm{R})}(\boldsymbol{x})$	intensity of image $\mathbf{I}^{(\emptyset;\mathrm{R})}$ (red light right)
$\breve{\mathbf{I}}^{(\mathrm{B};\emptyset)}$	illumination-normalized image $\mathbf{I}^{(\mathrm{B};\emptyset)}$
$\breve{\mathbf{I}}^{(\emptyset;\mathrm{R})}$	illumination-normalized image $\mathbf{I}^{(\emptyset;\mathrm{R})}$
$\breve{p}^{(\mathrm{B};\emptyset)}(\boldsymbol{x})$	illumination-normalized pixel intensity of image $\mathbf{I}^{(\mathrm{B};\emptyset)}$ for $c \in \{R, G, B\}$
${}^{i}\boldsymbol{o}$	sensor responses that constitute the canonical gamut
$\mathcal{O} = \{{}^{1}\boldsymbol{o}, \ldots, {}^{n}\boldsymbol{o}\}$	set of sensor responses $\boldsymbol{o}$i
$\mathcal{G}\ (\mathcal{O})$	canonical gamut
$\mathcal{G}\ (\mathbf{I})$	image gamut
$\mathbf{T}^{\boldsymbol{p},\boldsymbol{o}}$	diagonal matrix transform between $\boldsymbol{p}$ and $\boldsymbol{o}$
$\mathcal{M}$	set of possible mappings for gamut mapping
$\breve{\mathcal{M}}$	intersected set of possible mappings for gamut mapping
g_{B}	loss function in Bayesian color constancy
$\hat{\boldsymbol{e}}$	illuminant candidate in Bayesian color constancy
k_{B}	scaling factor in Bayesian color constancy
w_d, w_s	diffuse and specular weighting factors containing geometry and image intensity
$\boldsymbol{\zeta}(\boldsymbol{x})$, $\boldsymbol{\gamma}(\boldsymbol{x})$	diffuse and specular chromaticities
$s_c(\boldsymbol{x})$	reflectance slope in inverse-intensity chromaticity space for channel c
$\mathcal{S}$	set of illuminant estimates
$\mathcal{P}$	set of approximately correct illuminant estimates
$\mathcal{N}$	set of wrong illuminant estimates
$\mathrm{hist}(\mathcal{S})$	histogram of illuminant estimates in $\mathcal{S}$
$\mathcal{F}_i$	superpixel with index i
$\mathcal{R}_{\mathrm{IIC}}$	set of pixels in IIC space
λ_1, λ_2	largest and second largest eigenvalues of a set of points
	set of pixels in IIC space
$\mathcal{G}$	graph
$\mathcal{V}$	vertices of a graph $\mathcal{G}$
$\mathcal{E}$	edges of a graph $\mathcal{G}$
$\mathcal{X}$	discrete random field over the graph $\mathcal{G}$
$\boldsymbol{u}_i$	value for a random variable in $\mathcal{X}$

$\boldsymbol{l}_i$	illuminant label i
$\mathcal{L} = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_n\}$	set of illuminant labels
$\tilde{\mathcal{L}}$	set of estimated illuminant labels
$\breve{\boldsymbol{u}}$	labelling for a random variable in $\mathcal{X}$
$\mathcal{U}$	set of all possible labellings on $\mathcal{X}$
$\mathcal{C}$	clique
$\breve{\boldsymbol{u}}^{\mathcal{C}}$	labelling on a clique C
$\mathcal{C}_{\text{All}}$	set of all cliques $\mathcal{C}$
$\xi^{\mathcal{C}}(\breve{\boldsymbol{u}}^{\mathcal{C}}\|\mathcal{F})$	potential functions
$E(\breve{\boldsymbol{u}}\|\mathcal{F})$	Gibbs energy
$\breve{\boldsymbol{u}}^*$	maximum a posteriori labelling
$w_{\mathcal{F}_i}$	weighting for unary potential in patch $\mathcal{F}_i$
$t(.)$	robust error function for the unary potential
w_{r}	exponent for t
w_{PW}	weight for the pairwise potential in $E(\breve{\boldsymbol{u}}\|\mathcal{F})$
$\phi\left(\boldsymbol{u}_i\|\mathcal{F}_i\right)$	unary potential for a conditional random field
$\psi\left((\boldsymbol{u}_i, \boldsymbol{u}_j)\|(\mathcal{F}_i, \mathcal{F}_j)\right)$	pairwise potential for a conditional random field
$\tilde{\boldsymbol{e}}^{\text{GW}}$	estimated illum color using the generalized gray world algorithm
$\tilde{\boldsymbol{e}}^{\text{IIC}}$	estimated illum color using the physics-based inverse-intensity chromaticity space
$\text{robust}_\sigma(\epsilon_{\text{ang}})$	robust angular error function with dampening parameter σ
w_{Damp}	dampening parameter for the weights of the unary potentials
τ_b, τ_s	Lehman/Palm specularity segmentation parameter
τ_{sp}	specularity threshold per superpixel for physics-based illuminant estimates
w_{sp}	specularity weight per superpixel for physics-based illuminant estimates
w_{UW}	weighting between statistical and physics-based unary potentials
τ_{Sat}	saturation threshold for unusual illuminants
$\delta(i, j)$	unit impulse function on label assignments for neighbored superpixels $\mathcal{F}_i$ and $\mathcal{F}_j$
$\text{bound}(i, j)$	boundary length between two superpixels $\mathcal{F}_i$ and $\mathcal{F}_j$
$h(\mathcal{F}_i, \mathcal{F}_j)$	function to compute pairwise potentials based on the content of superpixels $\mathcal{F}_i$ and $\mathcal{F}_j$
w_{PWsh}	exponential weight for the pairwise potentials
Chapter 4: Illumination Cues in Image Forensics	
w_{orig}	number of feature vectors from original faces
w_{manip}	number of feature vectors from manipulated faces
$\boldsymbol{\nu}(\boldsymbol{x})$	normal vector of point $\boldsymbol{x}$
$p(\boldsymbol{\nu}(\boldsymbol{x}))$	intensity on a constant albedo surface, depending on the surface normal
$r(\boldsymbol{v}, \boldsymbol{\nu})$	surface reflectance function, depending on the direction of incident light and the surface normal
$e(\boldsymbol{v}, \boldsymbol{x})$	incident light at pixel $\boldsymbol{x}$ from angle $\boldsymbol{v}$
$\boldsymbol{v} = (v_x \;\; v_y \;\; v_z)^{\text{T}}$	direction of incident light
$\eta_{i,j}(\boldsymbol{v})$	spherical harmonics basis function

$h_{i,j}(\boldsymbol{x})$	weighting factor for the spherical harmonics
w_{ICE}	weighting factor for pixels with similar surface normals
σ_{ICE}	standard deviation for the computation of w_{ICE}
$\boldsymbol{M}$	estimation matrix for the direction of the incident light
$\boldsymbol{Q}$	weighting matrix for the lighting coefficients

Appendix C

Additional Material on Copy-Move Forgery Detection

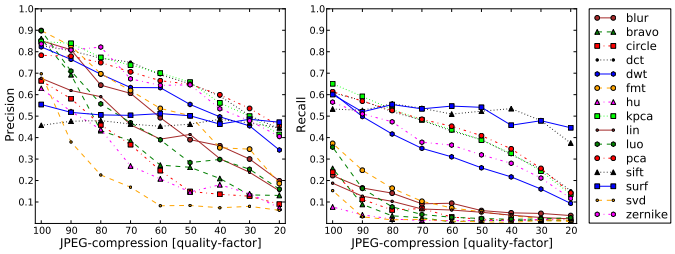
We add several details to the benchmarking of copy-move methods. First, we briefly state our variant of keypoint-based postprocessing. Then, we present our experiments on evaluating the copy-move feature on downscaled images and on the dataset after re-organization in different categories. Finally, we show the benchmark images, sorted by the (qualitative) texture categories *rough*, *smooth* and *structure*.

C.1 Postprocessing of Keypoint-based Methods

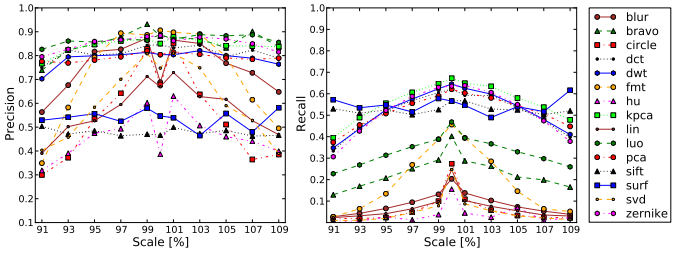
We apply a hierarchical clustering on matched feature pairs. We follow the algorithm by Amerini *et al.* [Amer 11], i. e. we assign each point to a cluster and merge them according to a linkage-method. For the linkage method we chose "single" linkage as it is very fast to compute and as the choice of the linkage-method is not critical [Amer 11]. On the other hand, the stop condition for merging the clusters ("cut-threshold") plays an important role. Here, we did not use the *inconsistency coefficient* as proposed by Amerini *et al.*. Instead, we rely on the distance between the nearest clusters. Two clusters are merged if their distance lies within the cut-threshold. We chose to use a cut-threshold of 25 pixels for SIFT and 50 pixels for SURF. As a special case, if we obtained less than 100 matches, the cut-threshold is raised to 75 pixels. Note that the cut-thresholds are chosen in a defensive way, such that typically multiple smaller clusters remain, which are merged at a later stage of the algorithm.

If a minimum of 4 correspondences connects two of these clusters, we estimate with RANSAC the affine transformation between the points that are connected by these correspondences, analogously to Amerini *et al.* [Amer 11]. In contrast to prior work, we further compute the optimal transformation matrix from the inliers according to the gold-standard algorithm for affine homography matrices [Hart 03, pp. 130]. Transformations with implausibly large or small values are removed via thresholding.

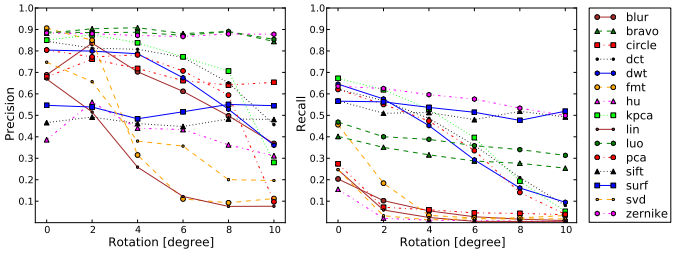
For large images containing large copied areas the transformation matrices of the clusters may often be similar. So, we decided to merge them if the root mean square error (RMSE) between two transformation matrices is below a threshold of 0.03 for the scaling and rotation part of the matrix, and below a threshold of 10 for the



(a) JPEG compression



(b) Scaled copy



(c) Rotated copy

Figure C.1: Recomputed results for scaled images: recall is significantly worse.

translation part. For these merged clusters we reestimate the transformation with the same procedure as above.

For each cluster we warp the image according to the estimated transformation matrix and compute a correlation map between the image and its warped version. From here on, we follow the algorithm by Pan and Lyu [Pan 10]. For every pixel and its warped version we compute the normalized correlation coefficient with a window size of 5×5 pixels. To remove noise in the correlation map, it is smoothed with a 7×7 pixels Gaussian kernel. Every smoothed correlation map is then binarized with a threshold of 0.4, and areas containing less than 1000 pixels were removed. Furthermore, areas where no match lies were removed, too. Then, the outer contours of each area is extracted and the inner part is flood-filled, which closes holes in the contours. The output map is the combination of all post-processed correlation maps. As a final verification step, each area from the combined output map is tested if it also has a correspondence which lies in another marked area.

Table C.1: Categorization of the database by object classes.

	Assigned images	
Category	Small copied area	Large copied area
living	giraffe, jellyfish chaos, cattle, swan	four babies
nature	fisherman, no beach, berries,	Scotland, white, hedge, christmas hedge, Malawi, beach wood, tree
man-made	supermarket, bricks, statue, ship, sailing, dark and bright, sweets, disconnected shift, Egyptian, noise pattern, sails, mask, window, writing history, knight moves	horses, kore, extension, clean walls, tapestry, port, wood carvings, stone ghost, red tower
mixed	barrier, motorcycle, Mykene, three-hundred, Japan tower, wading, central park	fountain, lone cat

C.2 Results after interpolated downsampling

Figure C.1 shows the detection results if the input image is downsampled prior to examination. Downsampling saves computational time, but is in general penalized by a worse recall.

C.3 Categorization of the dataset

In the thesis, we report results on the full dataset. However, the performance of the feature descriptors undoubtedly depends on the content of the image and the copied area. With a subdivision of the dataset in smaller categories, we make an attempt towards better explaining the performance differences of the feature sets.

The design of a proper category-driven evaluation in image forensics is still an open problem. We made a first attempt towards a proper categorization using two different approaches. We divided the images in multiple categories, once into object categories and once in texture categories. The results are reported in Sec. C.3.1 and Sec. C.3.2, respectively.

C.3.1 Categorization by Object Classes

We split the images in categories where the copied regions belong to one of the object categories *living*, *nature*, *man-made* or *mixed*. Here, *mixed* denotes copies where arguably multiple object types occur. Table C.1 lists which image belongs to which category. In Sec. C.4, downsampled versions of the images are presented together with the names of the images. The number of images varies between the categories. The smallest category is *living* containing 5 test cases, the largest category is *man-made* (24 test cases). Note that the varying category sizes pose no problem, as

	Image level				Pixel level			
Method	Living	Nature	MM	Mixed	Living	Nature	MM	Mixed
Blur	100.00	97.56	96.00	97.30	64.37	65.83	65.64	64.40
Bravo	100.00	100.00	95.05	94.74	61.82	66.63	65.00	59.59
Circle	100.00	100.00	97.96	94.74	66.09	69.97	71.62	68.91
DCT	100.00	90.91	89.72	90.00	67.27	59.47	68.82	58.86
DWT	100.00	100.00	89.72	87.80	66.77	67.51	68.42	66.64
FMT	100.00	100.00	95.05	94.74	64.85	68.35	69.95	67.98
Hu	86.96	78.43	86.49	76.60	61.36	63.62	64.89	61.50
KPCA	100.00	100.00	91.43	92.31	68.46	68.77	70.56	70.37
Lin	100.00	100.00	97.96	97.30	67.00	67.35	69.57	66.11
Luo	100.00	97.56	96.00	94.74	58.46	65.98	65.03	59.59
PCA	100.00	100.00	89.72	94.74	71.01	67.08	69.88	69.35
Sift	33.33	82.35	88.00	88.89	19.12	58.51	71.87	69.26
Surf	75.00	94.74	90.20	94.12	36.32	73.28	75.97	66.71
SVD	83.33	80.00	85.71	78.26	66.14	60.62	68.71	60.49
Zernike	100.00	100.00	95.05	100.00	67.60	68.59	71.04	68.00
Average	91.91	94.77	92.27	91.75	60.44	66.10	69.13	65.18

Table C.2: Results for plain copy-move per object category at image level (left) and at pixel level (right), in percent. "MM" denotes *man-made*.

we only compute the performance within a category[1]. We used the same parameters as for the evaluations in the main paper. Tab. C.2 shows the F_1 score for plain copy-move forgeries at image level (left) and at pixel level (right). Here, "MM" stands for *man-made*. Note that all four categories perform comparably at pixel level. However, at image level, most feature sets perform best for *living* and *nature*.

The Figures C.2, C.3, C.4 and C.5 show the results for *living*, *nature*, *man-made* and *mixed* under Gaussian noise, JPEG compression, rotation, scaling and joint effects. Again, the evaluation parameters were the same as in the main paper. At pixel level, several feature sets perform better for *nature* than for *living*. However, in several scenarios, the best results are obtained in the categories *man-made* and *mixed*. We assume that this comes from the fact that man-made objects often exhibit a clearer structure, and as such encompass the task of copy-move forgery detection.

C.3.2 Categorization by Texture

We investigated a second categorization of the dataset, this time by texture. The dataset is divided in copied areas providing *smooth*, *rough* and *structured* content. Here, *smooth* and *rough* serves as a distinction of texture properties, which can be approximately seen as low or high entropy in the snippet. The third category, *structured*, refers in most cases to man-made structures, like buildings: regular, clearly pronounced edges and corners.

[1]For instance, for classification tasks, a balanced size of each category can prevent biased results. However, this is not of concern in our copy-move forgery evaluation.

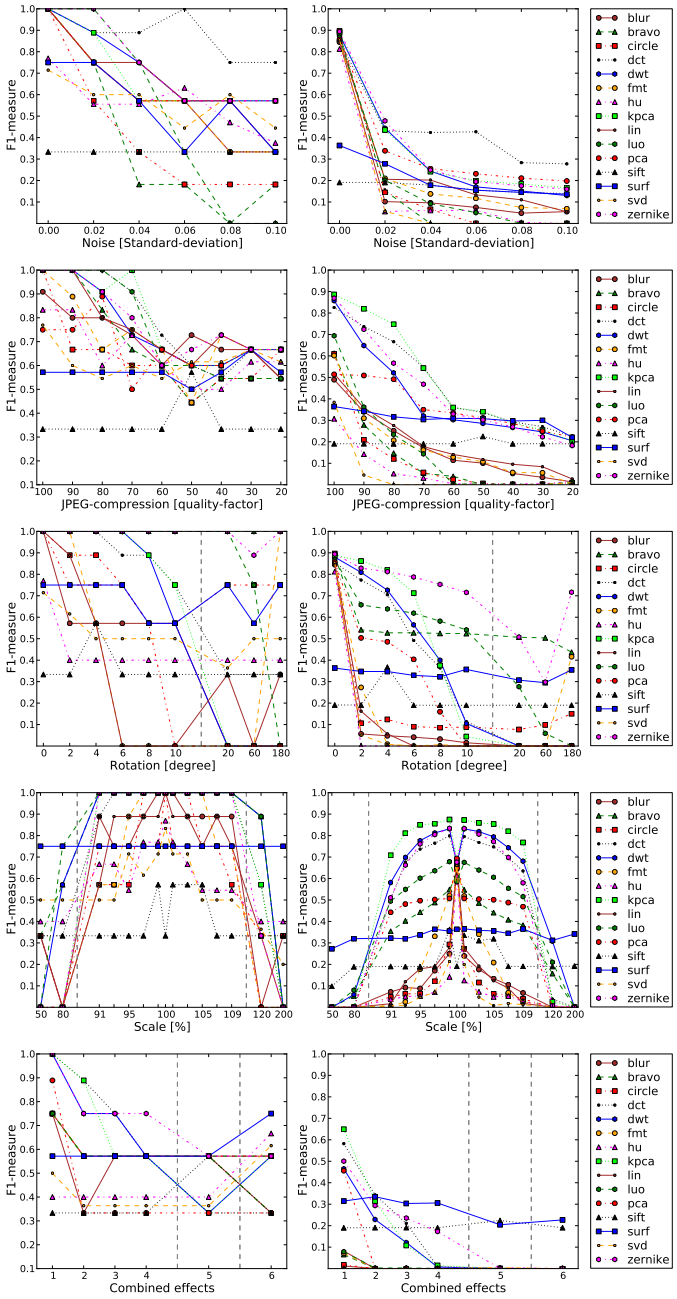


Figure C.2: Performance in the category *living* at image level (left) and pixel level (right).

This categorization was already prepared during the creation of the dataset. We aimed to use a diverse set of scenes, providing various challenges to the detectors. One challenge of real-world forgeries is the fact that we have little control over the creation of the manipulation. As a consequence, the texture categories are not based on quantitative measures. Instead, we used the artists' result on a fuzzy task description, like to "create a copy with little texture". Given the fact that real-world copy-move forgeries are done from artists as well, we found it reasonable to adopt the artists' viewpoint on CMFD.

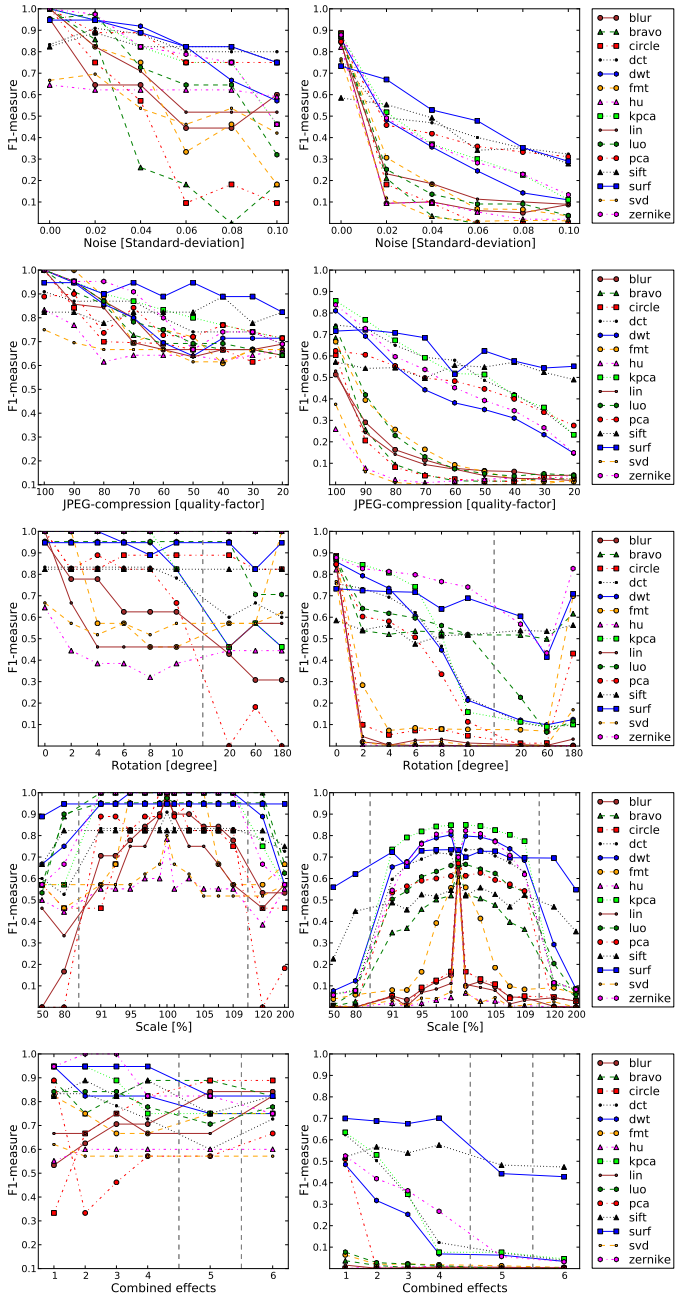


Figure C.3: Performance in the category *nature* at image level (left) and pixel level (right).

Tab. C.3 shows the assignment of images to categories. For our evaluation, we did not distinguish the size of the copied areas. Thus, we compare the three major categories *smooth*, *rough* and *structured*. The number of motifs per category is 17, 16 and 15, respectively.

Tab. C.4, Fig. C.6, Fig. C.7 and Fig. C.8 show the results per category. On the left side, the results are shown at image level, on the right side at pixel level. The most notable performance shift across categories is the relation between keypoint- and block-based methods. SURF and SIFT perform best in *rough* (see Fig. C.7),

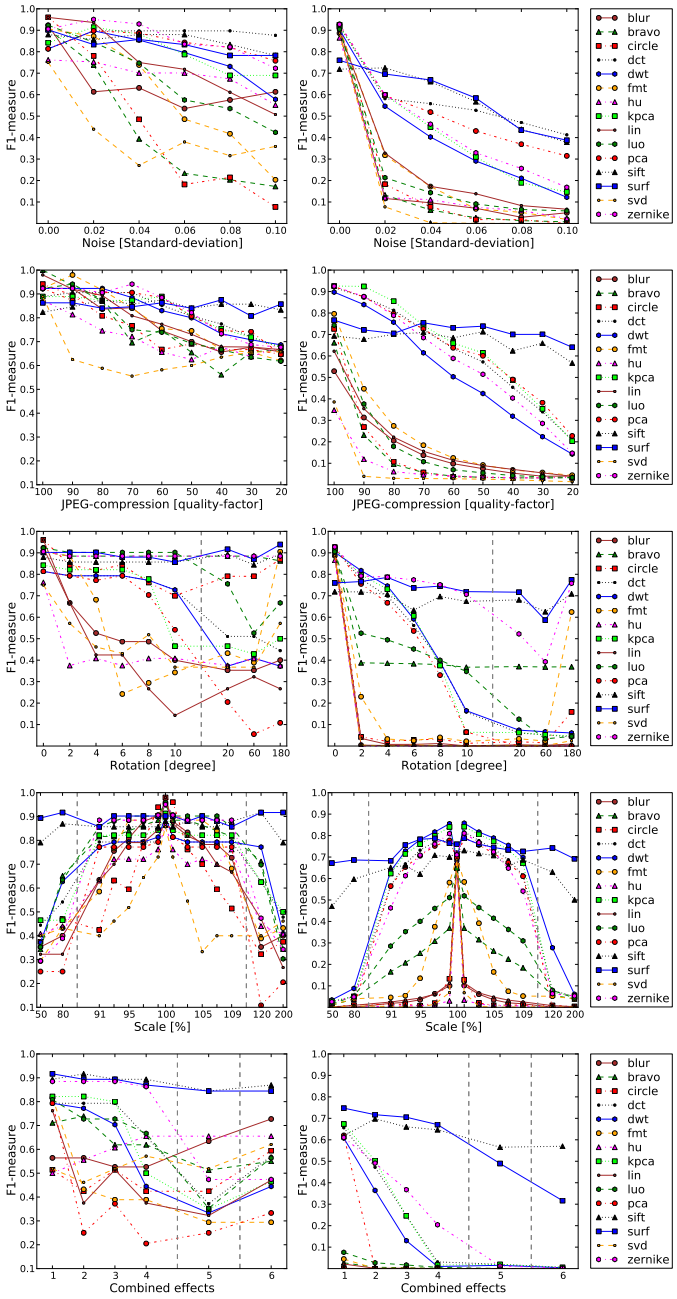


Figure C.4: Performance in the category *man-made* at image level (left) and pixel level (right).

while block-based methods often have an advantage in *smooth* (see Fig. C.6). In the category *structure* (see Fig. C.8), block-based and keypoint-based methods perform similarly well.

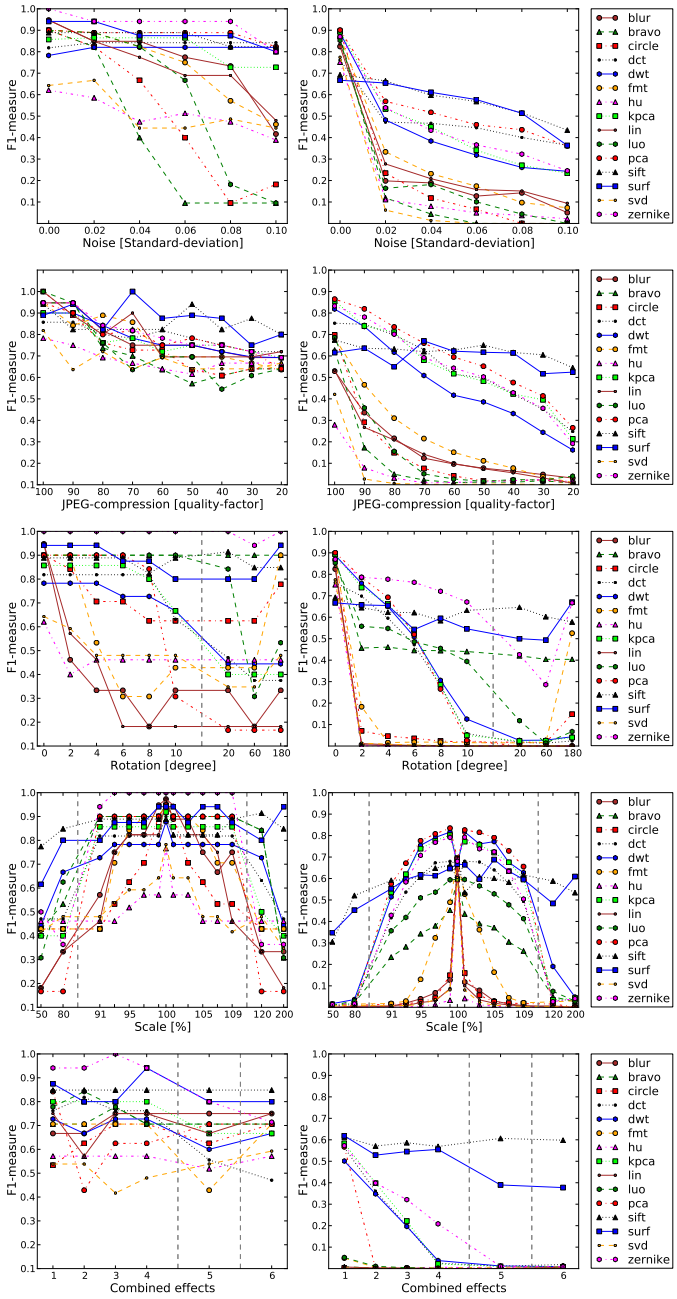


Figure C.5: Performance in the category *mixed* at image level (left) and pixel level (right).

C.4 Base Images in the Forensic Evaluation Framework

The database consists of 48 base images and 87 prepared image regions from these images, called *snippets*. Base images and snippets are spliced, to simulate a close-to-real-world copy-move forgery. During splicing, postprocessing artifacts can be added to the snippets and the final output images. The software to create tampered images and the associated ground truth is written in C++ and is best used with scripts

Table C.3: Categorization of the database by texture properties.

	Assigned images	
Category	Small copied area	Large copied area
Smooth	ship, motorcycle, sailing, disconnected shift, noise pattern, berries, sails, mask, cattle, swan, Japan tower, wading	four babies, Scotland, hedge, tapestry, Malawi
Rough	supermarket, no beach, fisherman, barrier, threehundred, writing history, central park	lone cat, kore, white, clean walls, tree, christmas hedge, stone ghost, beach wood, red tower
Structured	bricks, statue, giraffe, dark and bright, sweets, Mykene, jellyfish chaos, Egyptian, window, knight moves	fountain, horses, port, wood carvings, extension

written in Perl. Within the Perl-scripts, series of output images can be created by iterating over a parameter space. For instance, all spliced images with JPEG compression are obtained by iterating over the JPEG quality parameter space. We call one such parameterization a *configuration*. Upon acceptance of the paper, all images, snippets, code, scripts and configuration files are made publicly available from our web page. Note that with the separate building blocks, it is straightforward to add a copy-move tampering scenario that has not been addressed so far. For instance, assume (hypothetically) that one aims to evaluate instead of Gaussian noise Laplacian noise on the inserted regions. Then, all that is required to the author is to add a Laplacian noise function to the C++ code, and to add a matching configuration to the perl scripts.

Fig. C.9, Fig. C.10 and Fig. C.11 show a preview of the images and the regions of plain copy-move forgeries. Every database entry extends over two rows of images. In the first row, the image containing the "reference" tampered regions are shown. The second row shows the associated ground truth (with white being the copy-source or copy-target regions). Note that several aspects vary over the images. First, the size of the copied regions, second the level of detail in the copied region. Compare e. g. in Fig. C.11 the images in the top row. The statues exhibit relatively clear edges in comparison to the next two images where clouds, or sky and sea are copied, respectively. As presented, the copied regions are meaningful, i. e. either they hide image content, or they emphasize an element of the picture. Note, however, that the software allows the snippets to be inserted at arbitrary positions. Thus, one could equally well create semantically meaningless forgeries. This is implicitly the case when the copied region is rotated and resampled. In such cases, the image content becomes (naturally) implausible.

Table C.4: Assignment of images to the categories *smooth*, *rough* and *structured*.

Method	Smooth	Rough	Struct.	Smooth	Rough	Struct.
Blur	91.89	94.12	96.77	83.47	89.52	85.73
Bravo	91.89	91.43	96.77	87.51	91.92	88.67
Circle	97.14	96.97	93.75	88.48	93.86	90.54
DCT	89.47	88.89	85.71	77.90	89.47	88.12
DWT	91.89	91.43	90.91	86.94	91.65	88.06
FMT	91.89	96.97	96.77	86.56	92.13	87.76
Hu	80.95	80.00	81.08	82.13	84.07	82.59
KPCA	91.89	94.12	93.75	87.48	92.83	90.59
Lin	91.89	100.00	100.00	83.82	90.81	85.56
Luo	94.44	91.43	93.75	86.77	90.72	87.79
PCA	91.89	94.12	88.24	86.62	92.68	90.40
Sift	73.33	96.97	78.57	48.18	85.99	55.61
Surf	87.50	93.75	90.32	60.18	79.90	69.11
SVD	79.07	80.00	85.71	75.00	90.87	85.96
Zernike	97.14	96.97	93.75	89.41	91.19	90.32
Average	89.48	92.48	91.06	80.70	89.84	84.45

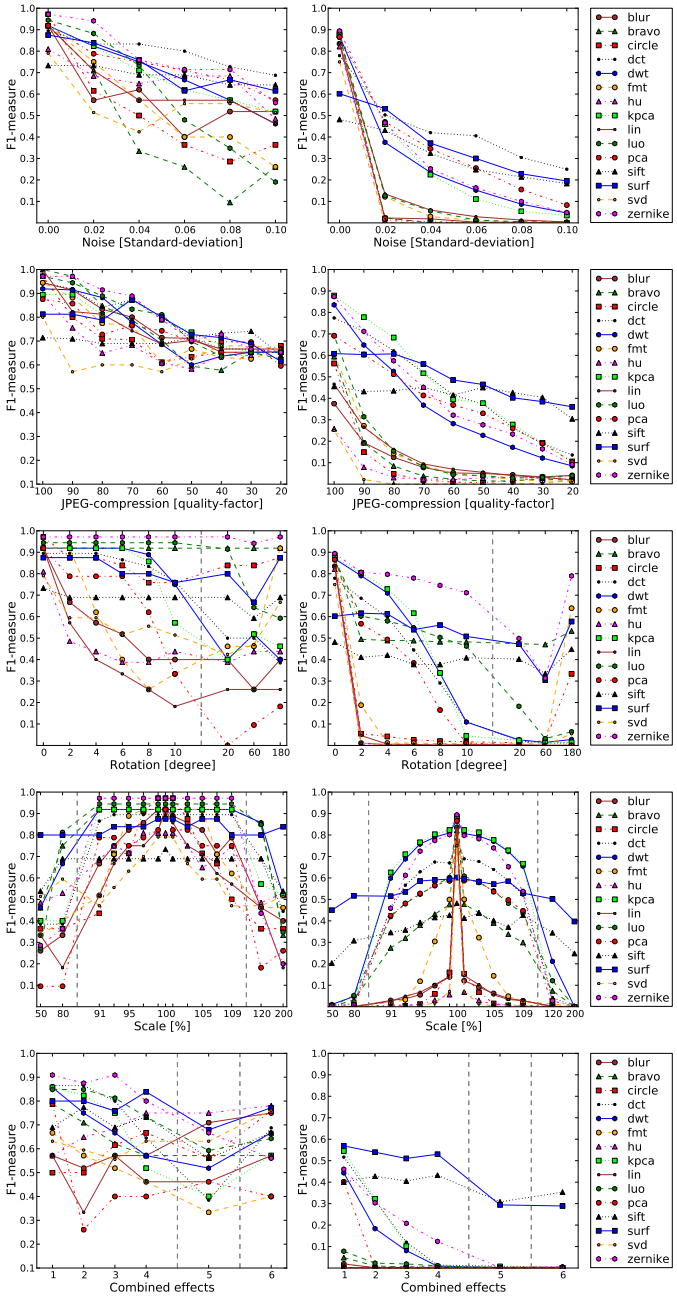


Figure C.6: Performance in the category *smooth* at image level (left) and pixel level (right).

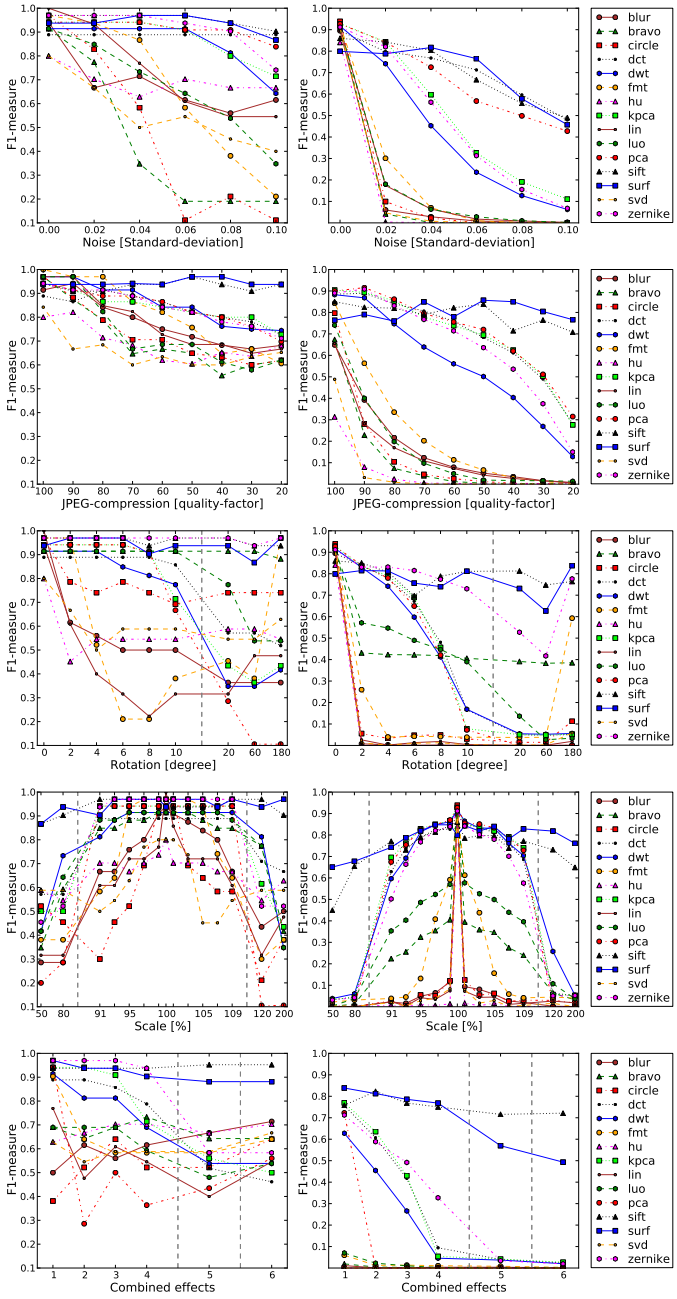


Figure C.7: Performance in the category *rough* at image level (left) and pixel level (right).

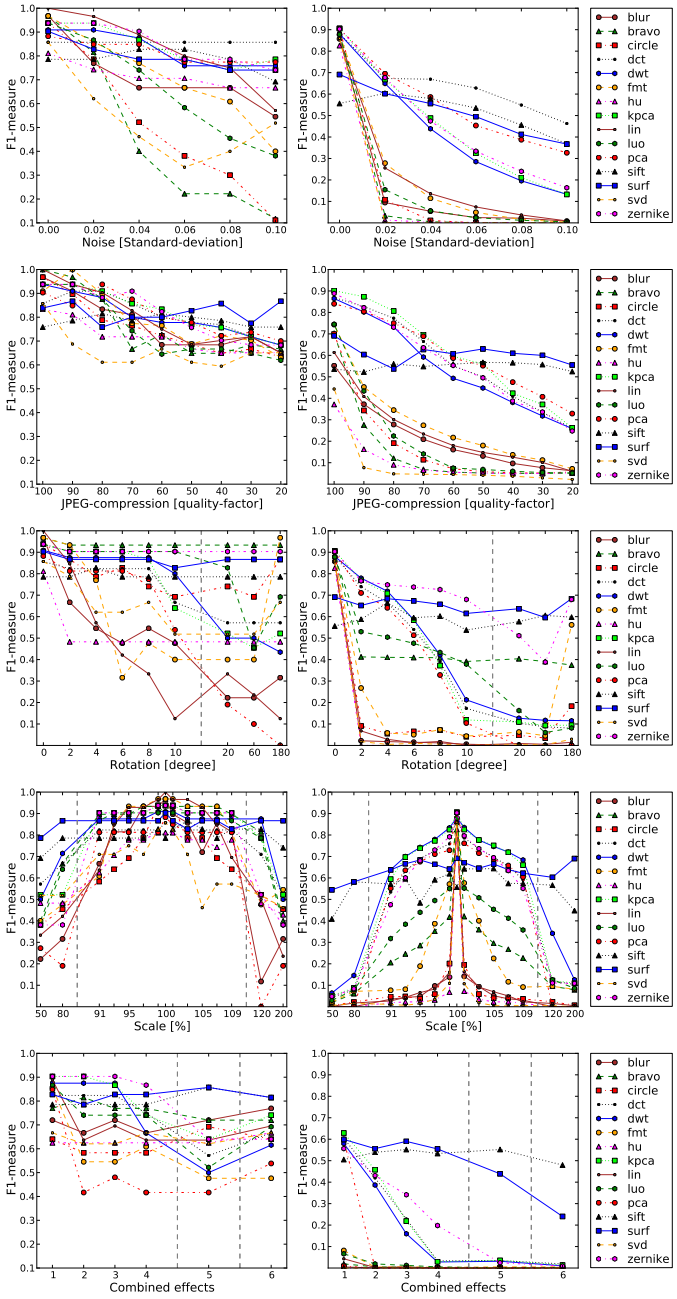


Figure C.8: Performance in the category *structure* at image level (left) and pixel level (right).

Figure C.9: Database images from the category *smooth*, with annotated ground truth for the "reference forgery", i. e. without rotation or scaling of the copied region.

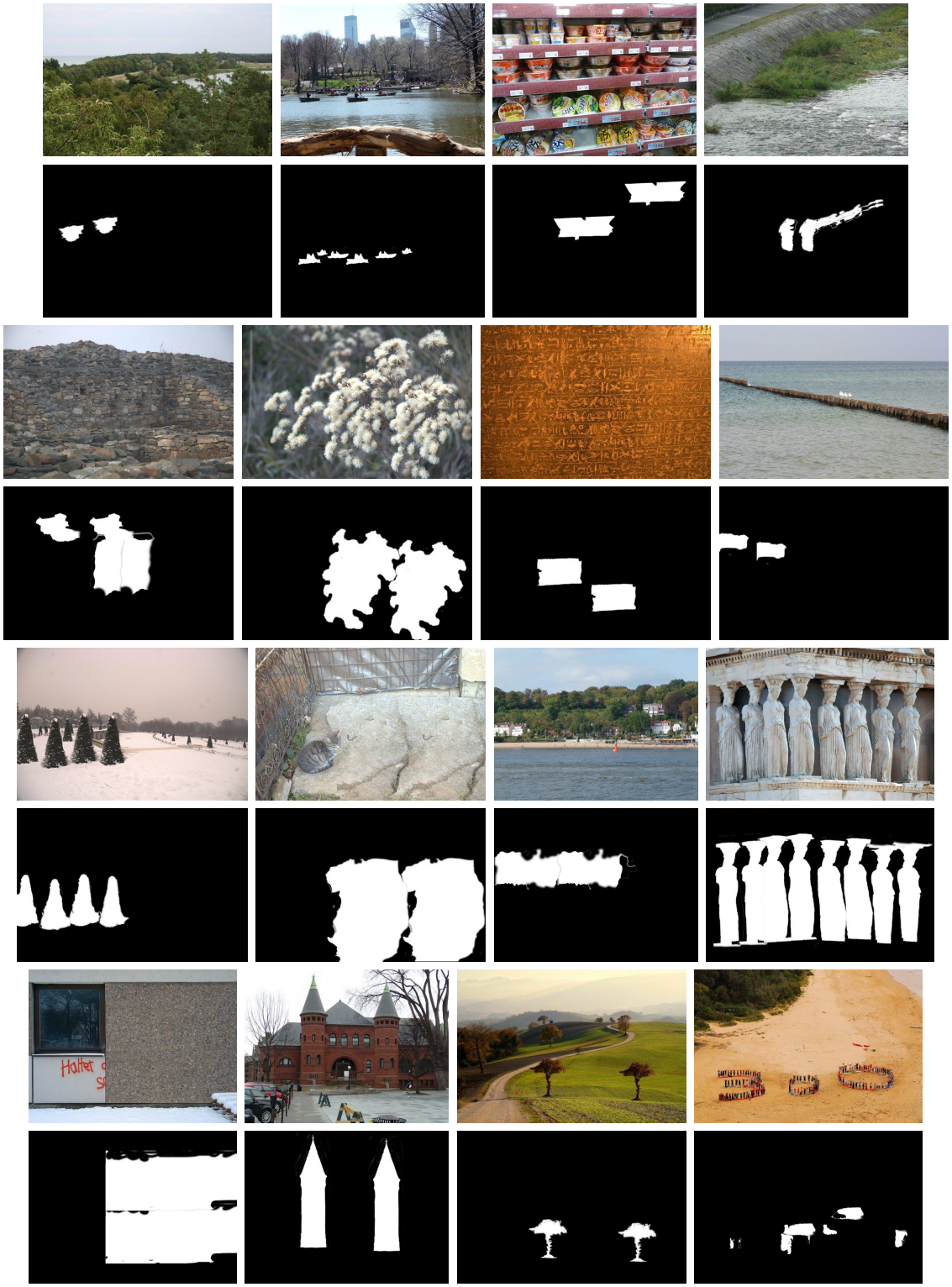


Figure C.10: Database images from the category *rough*, with annotated ground truth for the "reference forgery", i. e. without rotation or scaling of the copied region.

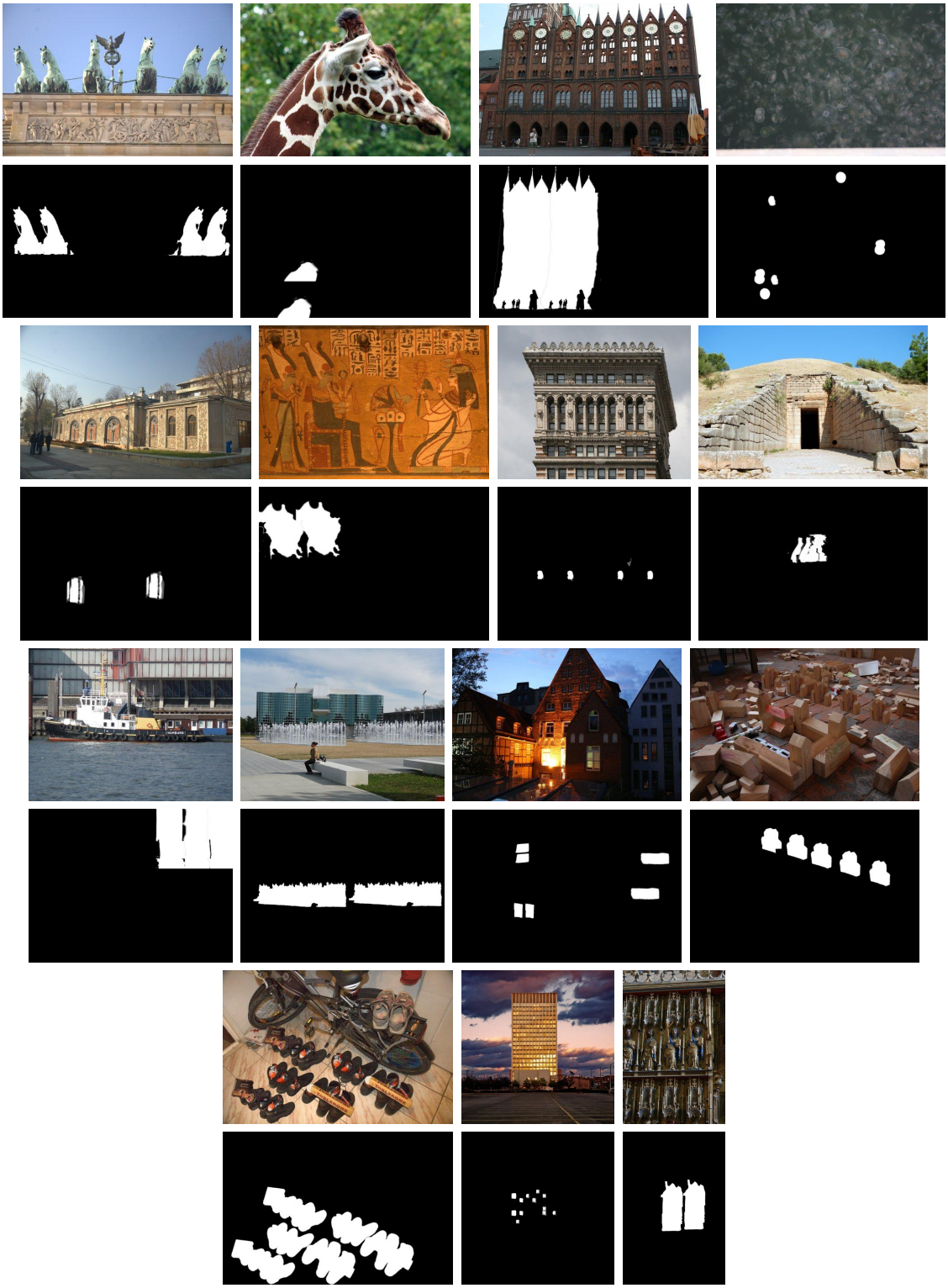


Figure C.11: Database images from the category *structure*, with annotated ground truth for the "reference forgery", i. e. without rotation or scaling of the copied region.

Appendix D

Data for the Analysis of Multi-Illuminant Scenes

Fig. D.1 shows the 11 laboratory scenes for the multi-illuminant dataset. Note that the empty scene "gt" has only been used to estimate the colors of the light sources, not for the evaluation. Every scene is captured under different illuminants from the left side and the right side. In detail, every scene was captured 14 times: six images with only the red, white or blue illuminant on the left side or on the right side (i. e., classical single-illuminant scenes), and eight combinations where on both sides illuminants were switched on (i. e., white-white, white-blue, white-red, blue-white,...). One case, namely red-red, was omitted, as no two red illuminants were available. For the scene "diff", all 14 intermediate images are shown in Fig. D.2.

Figure D.3 shows the 20 real-world scenes. Most of the pictures are indoor scenes. The only two cases with sunlight/shadow illumination are "dark tools" and "orange". Here, our model introduces a slight inaccuracy due to the fact that no full shadow scene was available (see Sec. 4.3.2.5 on page 80 for details).

(a) blue mug (b) pink mug (c) yellow mug

(d) gt (e) lion (f) diffuse

(g) specular and diffuse 2 (h) specular and diffuse (i) medium specularities

(j) blue and pink mug (k) blue and yellow mug

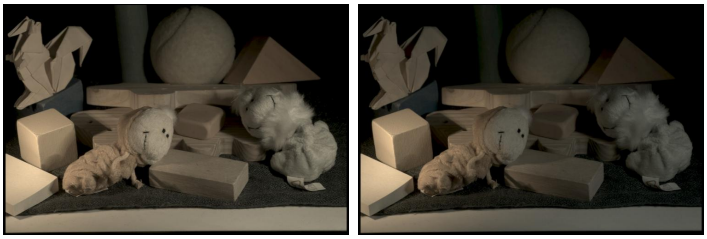
Figure D.1: Scenes from the laboratory multi-illuminant ground truth dataset.

(a) left: ∅, right: w (b) left: ∅, right: r (c) left: ∅, right: b

(d) left: w, right: ∅ (e) left: r, right: ∅ (f) left: b, right: ∅

(g) left: w, right: w (h) left: w, right: r (i) left: w, right: b

(j) left: b, right: w (k) left: b, right: b (l) left: b, right: r



(m) left: r, right: w (n) left: r, right: b

Figure D.2: Scene "diff", under red, white and blue illuminants from the left and the right side. Note that there is no image that is exposed to illuminant "red" from both sides.

(a) all gray (b) bike (c) cameras (d) compressor

(e) dark tools (f) detergents (g) dishes (h) extinguisher

(i) faucet (j) leafs (k) mathy hat (l) metal glass

(m) model (n) orange (o) poster (p) printer paper

(q) rack (r) revistes (s) screens (t) smile

Figure D.3: Scenes from the real-world multi-illuminant ground truth dataset.

Appendix E

Data for Illumination Cues in Image Forensics

We present additionally the per-image results of the evaluation in Fig. E.1. All pictures of the dataset are shown in Fig. E.1.

		Angular Error		
Subject	Angle	Single material	Gehler *et al.* (single mat.)	ICE (multi mat.)
1	0°	**31.2**	19.3	**26.7**
	45°	17.9	16.8	14.7
	90°	1.0	0.3	0.7
2	0°	10.8	9.1	9.8
	45°	10.1	16.1	5.6
	90°	17.5	15.4	**22.8**
3	0°	4.5	6.6	1.0
	45°	18.2	16.8	17.2
	90°	10.5	9.1	14.0
4	0°	**23.2**	15.8	**25.3**
	45°	14.4	8.7	11.9
	90°	7.0	7.0	8.0
5	0°	**33.0**	**30.9**	18.2
	45°	8.7	6.3	1.0
	90°	13.0	14.4	15.8
6	0°	21.0	21.0	5.9
	45°	7.3	2.8	0.7
	90°	13.7	15.4	13.3
7	0°	9.1	8.0	17.2
	45°	5.2	8.4	8.4
	90°	7.7	5.9	7.7
8	0°	**25.6**	**25.3**	20.3
	45°	9.4	8.0	16.1
	90°	**31.9**	**30.2**	**40.4**
9	0°	2.4	2.1	12.3
	45°	7.0	8.7	7.3
	90°	6.3	6.3	9.1
10	0°	6.6	5.6	10.1
	45°	21.7	**23.2**	13.0
	90°	11.6	12.3	14.4

Table E.1: Per-image results of the angular errors, under three different lights. in the lighting environment database. The errors are given for every of the evaluated methods. The failure cases are printed in bold. Intrinsic Contour Estimation (ICE) achieves comparable performance on contours along multiple materials.

Figure E.1: Intrinsic Contour Dataset. The light sources are oriented with angles of 0°, 45° and 90° (i. e., from the right, bottom right, and bottom) towards the subjects. Additionally, background light was switched on.

List of Figures

List of Tables

Bibliography

[Alva 08] J. Álvarez, A. López, and R. Baldrich. "Illuminant-Invariant Model-Based Road Segmentation". In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 1175–1180, Eindhoven, The Netherlands, June 2008.

[Amer 11] I. Amerini, L. Ballan, R. Caldelli, A. D. Bimbo, and G. Serra. "A SIFT-based Forensic Method for Copy-Move Attack Detection and Transformation Recovery". *IEEE Transactions on Information Forensics and Security*, Vol. 6, No. 3, pp. 1099–1110, Sep. 2011.

[Avci 03] I. Avcibaş, N. Memon, and B. Sankur. "Steganalysis using Image Quality Metrics". *IEEE Transactions on Image Processing*, Vol. 12, No. 2, pp. 221–229, Feb. 2003.

[Bago 06] S. Bagon. "Matlab Wrapper for Graph Cut". `http://www.wisdom.weizmann.ac.il/~bagon`, Dec. 2006.

[Bajc 96] R. Bajcsy, S. W. Lee, and A. Leonardis. "Detection of Diffuse and Specular Interface Reflections and Interreflections by Color Image Segmentation". *International Journal of Computer Vision*, Vol. 13, No. 3, pp. 241–272, March 1996.

[Barn 00] K. Barnard. "Improvements to Gamut Mapping Colour Constancy Algorithms". In: *Proceedings of the European Conference on Computer Vision (ECCV 2000)*, pp. 390–403, Dublin, Ireland, June 2000.

[Barn 01] K. Barnard, F. Ciurea, and B. Funt. "Sensor Sharpening for Computational Color Constancy". *Journal of the Optical Society of America A*, Vol. 18, No. 11, pp. 2728–2743, Nov. 2001.

[Barn 02a] K. Barnard, V. Cardei, and B. Funt. "A Comparison of Computational Color Constancy Algorithms – Part I: Methodology and Experiments With Synthesized Data". *IEEE Transactions on Image Processing*, Vol. 11, No. 9, pp. 972–983, Sep. 2002.

[Barn 02b] K. Barnard, L. Martin, A. Coath, and B. Funt. "A Comparison of Computational Color Constancy Algorithms – Part II: Experiments With Image Data". *IEEE Transactions on Image Processing*, Vol. 11, No. 9, pp. 985–996, Sep. 2002.

[Barn 02c] K. Barnard, L. Martin, B. Funt, and A. Coath. "A Data Set for Color Research". *Color Research & Application*, Vol. 27, No. 3, pp. 147–151, June 2002.

[Barn 10] M. Barni, A. Costanzo, and L. Sabatini. "Identification of Cut & Paste Tampering by Means of Double-JPEG Detection and Image

Segmentation". In: *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2010)*, pp. 1687–1690, Paris, France, May 2010.

[Barn 97] K. Barnard, G. Finlayson, and B. Funt. "Color Constancy for Scenes with Varying Illumination". *Computer Vision and Image Understanding*, Vol. 65, No. 2, pp. 311–321, Feb. 1997.

[Barn 98] K. Barnard, B. Funt, and B. Burnaby. "Experiments in Sensor Sharpening for Color Constancy". In: *Proceedings of the IS&T/SID Sixth Color Imaging Conference: Color, Science, Systems and Applications (CIC 1998)*, pp. 43–46, Scottsdale, AZ, USA, 1998.

[Barr 12] J. T. Barron and J. Malick. "Color Constancy, Intrinsic Images, and Shape Estimation". In: *Proceedings of the European Conference on Computer Vision (ECCV 2012)*, pp. 57–70, Florence, Italy, Oct. 2012.

[Bash 07] M. K. Bashar, K. Noda, N. Ohnishi, H. Kudo, T. Matsumoto, and Y. Takeuchi. "Wavelet-Based Multiresolution Features for Detecting Duplications in Images". In: *Proceedings of the IAPR Conference on Machine Vision Applications (MVA 2007)*, pp. 264–267, Tokyo, Japan, May 2007.

[Bash 10] M. Bashar, K. Noda, N. Ohnishi, and K. Mori. "Exploring Duplicated Regions in Natural Images". *IEEE Transactions on Image Processing*, March 2010. Accepted for publication.

[Basr 03] R. Basri and D. W. Jacobs. "Lambertian Reflectance and Linear Subspaces". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 2, pp. 218–233, Feb. 2003.

[Batt 09a] S. Battiato and G. Messina. "Digital Forgery Estimation into DCT Domain — A Critical Analysis". In: *Proceedings of the ACM Multimedia Workshop Multimedia in Forensics (MiFor 2009)*, pp. 37–42, Beijing, China, Oct. 2009.

[Batt 09b] S. Battiato and G. Messina. "Digital Forgery Estimation into DCT Domain - A Critical Analysis". In: *Proceedings of the ACM Multimedia Workshop Multimedia in Forensics (MiFor 2009)*, pp. 37–42, Beijing, China, Oct. 2009.

[Bayr 05] S. Bayram, İ. Avcıbaş, B. Sankur, and N. Memon. "Image Manipulation Detection with Binary Similarity Measures". In: *Proceedings of the 13th European Signal Processing Conference (EUSIPCO 2005)*, Divan Antalya Talya, Turkey, Sep. 2005.

[Bayr 09] S. Bayram, H. Sencar, and N. Memon. "An Efficient and Robust Method for Detecting Copy-Move Forgery". In: *Proceedings of the 34th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009)*, pp. 1053–1056, Taipei, Taiwan, Apr. 2009.

[Beig 12] S. Beigpour, C. Riess, J. van de Weijer, and E. Angelopoulou. "Illuminant Color Estimation with Conditional Random Fields". In: *IEEE Transactions on Image Processing*, 2012. Document to be submitted at Nov. 9th.

[Beis 97] J. S. Beis and D. G. Lowe. "Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces". In: *Proceedings of the 10th IEEE Computer Society Conference on Computer Vision and Pattern Recogognition (CVPR 1997)*, pp. 1000–1006, San Juan, Puerto Rico, June 1997.

[Bian 08] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. "Improving Color Constancy Using Indoor-Outdoor Image Classification". *IEEE Transactions on Image Processing*, Vol. 17, No. 12, pp. 2381–2392, Dec. 2008.

[Bian 10] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. "Automatic Color Constancy Algorithm Selection and Combination". *Pattern Recognition*, Vol. 43, No. 3, pp. 695–705, March 2010.

[Bian 12a] T. Bianchi and A. Piva. "Detection of Non-Aligned Double JPEG Compression Based on Integer Periodicity Maps". *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 2, pp. 842–848, Apr. 2012.

[Bian 12b] S. Bianco and R. Schettini. "Color Constancy Using Faces". In: *Proceedings of the 25th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pp. 65–72, Providence, RI, USA, June 2012.

[Bish 06] C. M. Bishop. *Pattern Recognition and Machine Learning. Information Science and Statistics*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Blan 03] V. Blanz and T. Vetter. "Face Recognition Based on Fitting a 3D Morphable Model". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 9, pp. 1063–1074, Sep. 2003.

[Blei 11] M. Bleier, C. Riess, S. Beigpour, E. Eibenberger, E. Angelopoulou, T. Tröger, and A. Kaup. "Color Constancy and Non-Uniform Illumination: Can Existing Algorithms Work?". In: *IEEE Color and Photometry in Computer Vision Workshop (CPCV 2011)*, pp. 774–781, Barcelona, Spain, Nov. 2011.

[Bo 10] X. Bo, W. Junwen, L. Guangjie, and D. Yuewei. "Image Copy-Move Forgery Detection Based on SURF". In: *Proceedings of the 2nd International Conference on Multimedia Information Networking and Security (MINES 2010)*, pp. 889–892, Nanjing, China, Nov. 2010.

[Bous 09] A. Bousseau, S. Paris, and F. Durand. "User-Assisted Intrinsic Images". *ACM Transactions on Graphics*, Vol. 28, No. 5, pp. 130:1–10, Dec. 2009.

[Boyk 01] Y. Boykov, O. Veksler, and R. Zabih. "Efficient Approximate Energy Minimization via Graph Cuts". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, pp. 1222–1239, Nov. 2001.

[Boyk 04] Y. Boykov and V. Kolmogorov. "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, pp. 1124–1137, Sep. 2004.

[Boyk 98] Y. Boykov, O. Veksler, and R. Zabih. "Markov Random Fields with Efficient Approximations". In: *Proceedings of the 11th IEEE Computer Society Conference on Computer Vision and Pattern Recogognition (CVPR 1998)*, pp. 648–654, Santa Barbara, CA, USA, June 1998.

[Brai 86] D. H. Brainard and B. A. Wandell. "Analysis of the Retinex Theory of Color Vision". *Journal of the Optical Society of America A*, Vol. 3, No. 10, pp. 1651–1661, Oct. 1986.

[Brai 97] D. H. Brainard and W. T. Freeman. "Bayesian Color Constancy". *Journal of the Optical Society of America A*, Vol. 14, No. 7, pp. 1393–1411, July 1997.

[Brav 09] S. Bravo-Solorio and A. Nandi. "Passive Forensic Method for Detecting Duplicated Regions Affected by Reflection, Rotation and Scaling". In: *Proceedings of the 17th European Signal Processing Conference (EUSIPCO 2009)*, pp. 824–828, Glasgow, Scotland, UK, Aug. 2009.

[Brav 11] S. Bravo-Solorio and A. K. Nandi. "Exposing Duplicated Regions Affected by Reflection, Rotation and Scaling". In: *Proceedings of the 36th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)*, pp. 1880–1883, Prague, Czech Republic, May 2011.

[Buch 80] G. Buchsbaum. "A Spatial Processor Model for Color Perception". *Journal of the Franklin Institute*, Vol. 310, No. 1, pp. 1–26, July 1980.

[Cann 86] J. Canny. "A Computational Approach to Edge Detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679–698, Nov. 1986.

[Card 02] V. C. Cardei, B. Funt, and K. Barnard. "Estimating the Scene Illumination Chromaticity Using a Neural Network". *Journal of the Optical Society of America A*, Vol. 19, No. 12, pp. 2374–2386, Dec. 2002.

[Cark 03] A. Çarkacioglu and F. T. Yarman-Vural. "SASI: A Generic Texture Descriptor for Image Retrieval". *Pattern Recognition*, Vol. 36, No. 11, pp. 2615–2633, Nov. 2003.

[CASIA] *CASIA Image Tampering Detection Evaluation Database*. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science. `http://forensics.idealtest.org`.

[Chak 12] A. Chakrabarti, K. Hirakawa, and T. Zickler. "Color Constancy with Spatio-Spectral Statistics". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 8, pp. 1509–1519, Aug. 2012.

[Chri 10a] V. Christlein, C. Riess, and E. Angelopoulou. "A Study on Features for the Detection of Copy-Move Forgeries". In: *Proceedings of the 5th Conference Sicherheit, Schutz und Zuverlässigkeit (GI SICHERHEIT 2010)*, pp. 105–116, Berlin, Germany, Oct. 2010.

[Chri 10b] V. Christlein, C. Riess, and E. Angelopoulou. "On Rotation Invariance in Copy-Move Forgery Detection". In: *Proceedings of the 2nd IEEE Workshop on Information Forensics and Security*, pp. 129–134, Seattle, WA, USA, Dec. 2010.

[Chri 12] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou. "An Evaluation of Popular Copy-Move Forgery Detection Approaches". *IEEE Transactions on Information Forensics and Security*, 2012. to appear.

[Ciur 03] F. Ciurea and B. Funt. "A Large Image Database for Color Constancy Research". In: *Proceedings of the IS&T/SID Eleventh Color Imaging Conference: Color Science and Engineering Systems, Technologies, Applications (CIC 2003)*, pp. 160–164, Scottsdale, AZ, USA, Nov. 2003.

[Coff 12] D. Coffin. "dcraw". http://www.cybercom.net/ dcoffin/dcraw/, Oct. 2012.

[Cook 81] R. L. Cook and K. E. Torrance. "A Reflectance Model for Computer Graphics". In: *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1981)*, pp. 307–316, Dallas, TX, USA, Aug. 1981.

[Csur 04] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. "Visual Categorization With Bags of Keypoints". In: *Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004.

[Dala 05] N. Dalal and B. Triggs. "Histograms of Oriented Gradients for Human Detection". In: *Proceedings of the 18th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pp. 886–893, San Diego, CA, USA, June 2005.

[Deng 11] Z. Deng, A. Gijsenij, and J. Zhang. "Source Camera Identification using Auto-White Balance Approximation". In: *Proceedings of the 13th IEEE International Conference on Computer Vision (ICCV 2011)*, pp. 57–64, Barcelona, Spain, Nov. 2011.

[Diri 08] A. E. Dirik, H. T. Sencar, and N. Memon. "Digital Single Lens Reflex Camera Identification From Traces of Sensor Dust". *IEEE Transactions on Information Forensics and Security*, Vol. 3, No. 3, pp. 539–552, Sep. 2008.

[Domi 05] D. Domingo. "Partying at Sonar Night". `http://www.flickr.com/photos/_sml/`, June 2005. Flickr photo collection.

[Drew 00] M. Drew and G. Finlayson. "Spectral Sharpening with Positivity". *Journal of the Optical Society of America A*, Vol. 17, No. 8, pp. 1361–1370, Aug. 2000.

[Drew 09] M. Drew and H. Joze. "Sharpening from Shadows: Sensor Transforms for Removing Shadows using a Single Image". In: *Proceedings of the IS&T/SID Seventeenth Color Imaging Conference: Color, Science, Systems and Applications (CIC 2009)*, pp. 267–271, Albuquerque, NM, USA, Nov. 2009.

[Dyba 07] B. Dybala, B. Jennings, and D. Letscher. "Detecting Filtered Cloning in Digital Images". In: *Proceedings of the 9th Workshop on Multimedia and Security (MM&Sec 2007)*, pp. 43–50, Dallas, TX, USA, Sep. 2007.

[Ebne 09] M. Ebner. "Color Constancy based on Local Space Average Color". *Machine Vision and Applications*, Vol. 20, No. 5, pp. 283–301, July 2009.

[Eski 95] A. M. Eskicioglu and P. S. Fisher. "Image Quality Measures and Their Performance". *IEEE Transactions on Communications*, Vol. 43, No. 12, pp. 2959–2965, Dec. 1995.

[Fari 09] H. Farid. "Exposing Digital Forgeries from JPEG Ghosts". *IEEE Transactions on Information Forensics and Security*, Vol. 1, No. 4, pp. 154–160, March 2009.

[Fari 10a] H. Farid. "A 3-D Lighting and Shadow Analysis of the JFK Zapruder Film (Frame 317)". Tech. Rep. TR2010-677, Dartmouth College, 2010. http://www.cs.dartmouth.edu/farid/downloads/publications/tr10a.pdf.

[Fari 10b] H. Farid and M. J. Bravo. "Image Forensic Analyses that Elude the Human Visual System". In: *Proceedings of the IS&T SPIE Symposium on Electronic Imaging - Media Forensics and Security II*, San Jose, CA, USA, Jan. 2010.

[Fari 11] H. Farid. "Photo Tampering Throughout History". `http://www.cs.dartmouth.edu/~farid/Hany_Farid/PhotoTampering0.html`, 2011. Website last visited Jan. 18th, 2012.

[Felz 04] P. F. Felzenszwalb and D. P. Huttenlocher. "Efficient Graph-Based Image Segmentation". *International Journal of Computer Vision*, Vol. 59, No. 2, pp. 167–181, Sep. 2004.

[Finl 00] G. Finlayson and S. Hordley. "Improving Gamut Mapping Color Constancy". *IEEE Transactions on Image Processing*, Vol. 9, No. 10, pp. 1774–1783, Oct. 2000.

[Finl 01a] G. D. Finlayson, S. D. Hordley, and P. M. Hubel. "Color by Correlation: A Simple, Unifying Framework for Color Constancy". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 11, pp. 1209–1221, Nov. 2001.

[Finl 01b] G. D. Finlayson and G. Schaefer. "Convex and Non-convex Illuminant Constraints for Dichromatic Colour Constancy". In: *Proceedings of the 14th IEEE Computer Society Conference on Computer Vision and Pattern Recogognition (CVPR 2001)*, pp. 598–604, Kauai, HI, USA, Dec. 2001.

[Finl 01c] G. D. Finlayson and G. Schaefer. "Solving for Color Constancy using a Constrained Dichromatic Reflection Model". *International Journal of Computer Vision*, Vol. 42, No. 3, pp. 127–144, May 2001.

[Finl 06] G. D. Finlayson, S. D. Hordley, and I. Tastl. "Gamut Constrained Illuminant Estimation". *International Journal of Computer Vision*, Vol. 67, No. 1, pp. 93–109, Apr. 2006.

[Finl 96] G. D. Finlayson. "Color in Perspective". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 10, pp. 1034–1038, Oct. 1996.

[Fors 90] D. Forsyth. "A Novel Algorithm for Color Constancy". *International Journal of Computer Vision*, Vol. 5, No. 1, pp. 5–36, Aug. 1990.

[Fost 04] D. H. Foster, S. M. Nascimento, and K. Amano. "Information Limits on Neural Identification of Colored Surfaces in Natural Scenes". *Visual Neuroscience*, Vol. 21, No. 3, pp. 331–336, May 2004.

[Frid 03] J. Fridrich, D. Soukal, and J. Lukáš. "Detection of Copy-Move Forgery in Digital Images". In: *Proceedings of the 3rd Digital Forensic Research Workshop (DFRWS 2003)*, Cleveland, OH, USA, Aug. 2003.

[Frie 77] J. H. Friedman, J. L. Bentley, and R. A. Finkel. "An Algorithm for Finding Best Matches in Logarithmic Expected Time". *ACM Transactions on Mathematical Software*, Vol. 3, No. 3, pp. 209–226, Sep. 1977.

[Fu 07] D. Fu, Y. Q. Shi, and W. Su. "A Generalized Benford's Law for JPEG Coefficients and its Applications in Image Forensics". In: *Proceedings of the IS&T SPIE Symposium on Electronic Imaging - Security, Steganography, and Watermarking of Multimedia Contents*, San Jose, CA, USA, Feb. 2007.

[Funt 03] B. Funt and H. Jiang. "Non-Diagonal Color Correction". In: *Proceedings of the 10th IEEE International Conference on Image Processing (ICIP 2003)*, pp. 481–484, Barcelona, Spain, Sep. 2003.

[Funt 04] B. Funt, F. Ciurea, and J. McCann. "Retinex in MATLAB". *Journal of Electronic Imaging*, Vol. 13, No. 1, pp. 48–57, Jan. 2004.

[Funt 98] B. Funt, K. Barnard, and L. Martin. "Is Machine Color Constancy Good Enough?". In: *Proceedings of the European Conference on Computer Vision (ECCV 1998)*, pp. 445–459, Freiburg, Germany, June 1998.

[Gehl 08] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. "Bayesian Color Constancy Revisited". In: *Proceedings of the 21st IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, Alaska, USA, June 2008.

[Gehl 11] P. V. Gehler, C. Rother, M. Kiefel, L. Zhang, and B. Schölkopf. "Recovering Intrinsic Images with a Global Sparsity Prior on Reflectance". In: *Advances in Neural Information Processing Systems (NIPS 2011)*, pp. 765–773, Granada, Spain, Dec. 2011.

[Geus 01] J.-M. Geusebroek, R. van den Boomgaard, A. W. Smeulders, and H. Geerts. "Color Invariance". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 12, pp. 1338–1350, Dec. 2001.

[Geus 03] J.-M. Geusebroek, R. Boomgaard, A. Smeulders, and T. Gevers. "Color Constancy from Physical Principles". *Pattern Recognition Letters*, Vol. 24, No. 11, pp. 1653–1662, July 2003.

[Ghol 08] S. Gholap and P. K. Bora. "Illuminant Colour Based Image Forensics". In: *IEEE Region 10 Conference TENCON (TENCON 2008)*, Hyderabad, India, Nov. 2008.

[Gijs 09] A. Gijsenij, T. Gevers, and J. van de Weijer. "Physics-based Edge Evaluation for Improved Color Constancy". In: *Proceedings of the 25th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, pp. 581–588, Miami Beach, Florida, June 2009.

[Gijs 10a] A. Gijsenij, T. Gevers, and J. van de Weijer. "Generalized Gamut Mapping using Image Derivative Structures for Color Constancy". *International Journal of Computer Vision*, Vol. 86, No. 2–3, pp. 127–139, Jan. 2010.

[Gijs 10b] A. Gijsenij. "Personal communication with the authors". Sep. 2010.

[Gijs 11] A. Gijsenij, T. Gevers, and J. van de Weijer. "Computational Color Constancy: Survey and Experiments". *Transactions on Image Processing*, Vol. 20, No. 9, pp. 2475–2489, Sep. 2011.

[Gijs 12a] A. Gijsenij, T. Gevers, and J. van de Weijer. "Improving Color Constancy by Photometric Edge Weighting". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 5, pp. 918–929, May 2012.

[Gijs 12b] A. Gijsenij, R. Lu, and T. Gevers. "Color Constancy for Multiple Light Sources". *IEEE Transactions on Image Processing*, Vol. 21, No. 2, pp. 697–707, Feb. 2012.

[Gloe 10] T. Gloe and R. Böhme. "The 'Dresden Image Database' for benchmarking digital image forensics". In: *Proceedings of the 25th Annual ACM Symposium On Applied Computing (SAC 2010)*, pp. 1585–1591, Sierre, Switzerland, March 2010.

[Golj 09] M. Goljan, J. Fridrich, and T. Filler. "Large Scale Test of Sensor Fingerprint Camera Identification". In: *Proceedings of the IS&T SPIE Symposium on Electronic Imaging - Media Forensics and Security*, San Jose, CA, USA, Jan. 2009.

[Groe 96] H. Groemer. *Geometric Applications of Fourier Series and Spherical Harmonics*. Cambridge University Press, 1996.

[Gros 09] R. Grosse, M. Johnson, E. Adelson, and W. Freeman. "Ground Truth Dataset and Baseline Evaluations for Intrinsic Image Algorithms". In: *Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV 2009)*, pp. 2335–2342, Kyoto, Japan, Nov. 2009.

[Hale 10] B. Hale. "Spot the difference: How today's airbrushing PC censors decided Churchill could do without his cigar". `http://www.dailymail.co.uk/news/article-1286620/Churchill-non-smoker-How-todays-PC-censors-airbrushed-cigar.html`, June 2010. Article in the Daily Mail, June 15th, 2010. Website last visited Jan. 18th, 2012.

[Hart 03] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2003.

[Hord 06] S. D. Hordley and G. D. Finlayson. "Re-evaluating Color Constancy Algorithm Performance". *Journal of the Optical Society of America A*, Vol. 23, No. 5, pp. 1008–1020, May 2006.

[Hsu 08] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand. "Light Mixture Estimation for Spatially Varying White Balance". *ACM Transactions on Graphics*, Vol. 27, No. 3, pp. 70:1–70:7, Aug. 2008.

[Huan 08] H. Huang, W. Guo, and Y. Zhang. "Detection of Copy-Move Forgery in Digital Images Using SIFT Algorithm". In: *Proceedings of the IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA 2008)*, pp. 272–276, Wuhan, China, Dec. 2008.

[Huan 10] F. Huang, J. Huang, and Y. Q. Shi. "Detecting Double JPEG Compression With the Same Quantization Matrix". *IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 4, pp. 848–856, Dec. 2010.

[Igar 07] T. Igarashi, K. Nishino, and S. K. Nayar. "The Appearance of Human Skin: A Survey". *Foundations and Trends in Computer Graphics and Vision*, Vol. 3, No. 1, pp. 1–95, Nov. 2007.

[Iked 10] R. Ikeda. "Untitled". `http://www.flickr.com/people/i-rocksteady/`, Apr. 2010. Flickr photo collection.

[Jobs 97] D. Jobson, Z. Rahman, and G. Woodell. "A Multiscale Retinex for Bridging the Gap between Color Images and the Human Observation of Scenes". *IEEE Transactions on Image Processing*, Vol. 6, No. 7, pp. 965–976, July 1997.

[John 05] M. K. Johnson and H. Farid. "Exposing Digital Forgeries by Detecting Inconsistencies in Lighting". In: *Proceedings of the 7th Workshop on Multimedia and Security (MM&Sec 2005)*, pp. 1–10, New York, NY, USA, Aug. 2005.

[John 07a] M. Johnson and H. Farid. "Exposing Digital Forgeries in Complex Lighting Environments". *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 3, pp. 450–461, Sep. 2007.

[John 07b] M. Johnson and H. Farid. "Exposing Digital Forgeries through Specular Highlights on the Eye". In: *Proceedings of the 9th International Workshop on Information Hiding (IH 2007)*, pp. 311–325, Saint Malo, France, 2007.

[Ju 07] S. Ju, J. Zhou, and K. He. "An Authentication Method for Copy Areas of Images". In: *Proceedings of the Fourth International Conference on Image and Graphics (ICIG 2007)*, pp. 303–306, Chengdu, Sichuan, China, Aug. 2007.

[Juel 08] T. Juel. "Knossos". `http://www.flickr.com/people/tjuel/`, Aug. 2008. Flickr photo collection.

[Kang 08] X. Kang and S. Wei. "Identifying Tampered Regions Using Singular Value Decomposition in Digital Image Forensics". In: *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE 2008)*, pp. 926–930, Wuhan, China, 2008.

[Kawa 05] R. Kawakami, K. Ikeuchi, and R. T. Tan. "Consistent Surface Color for Texturing Large Objects in Outdoor Scenes". In: *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV 2005)*, pp. 1200–1207, Beijing, China, Oct. 2005.

[Ke 04] Y. Ke, R. Sukthankar, and L. Huston. "An Efficient Parts-Based Near-Duplicate and Sub-Image Retrieval System". In: *Proceedings of the 12th ACM International Conference on Multimedia*, pp. 869–876, New York, NY, USA, Oct. 2004.

[Kee 10] E. Kee and H. Farid. "Exposing Digital Forgeries from 3-D Lighting Environments". In: *Proceedings of the 2nd IEEE International Workshop on Information Forensics and Security (WIFS 2010)*, Seattle, WA, USA, Dec. 2010.

[Kim 11] K. Kim, J. Bae, and J. Kim. "Natural HDR Image Tone Mapping Based on Retinex". *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 4, pp. 1807–1814, Nov. 2011.

[Klin 88] G. J. Klinker, S. A. Shafer, and T. Kanade. "The Measurement of Highlights in Color Images". *International Journal of Computer Vision*, Vol. 2, No. 1, pp. 7–26, June 1988.

[Klin 90] G. J. Klinker, S. A. Shafer, and T. Kanade. "A Physical Approach to Color Image Understanding". *International Journal of Computer Vision*, Vol. 4, No. 1, pp. 7–38, Jan. 1990.

[Kohl 09] P. Kohli, L. Ladický, and P. Torr. "Robust Higher Order Potentials for Enforcing Label Consistency". *International Journal of Computer Vision*, Vol. 82, No. 3, pp. 302–324, May 2009.

[Kolm 04] V. Kolmogorov and R. Zabih. "What Energy Functions can be Minimized via Graph Cuts?". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 2, pp. 147–159, Feb. 2004.

[Krie 70] J. von Kries. "Influence of adaptation on the effects produced by luminous stimuli". In: D. L. MacAdam, Ed., *Sources of Color Science*, pp. 109–119, MIT Press, Cambridge, MA, USA, 1970.

[Land 71] E. Land and J. McCann. "Lightness and Retinex theory". *Journal of the Optical Society of America*, Vol. 61, No. 1, pp. 1–11, Jan. 1971.

[Land 77] E. H. Land. "Lightness and the Retinex Theory". *Scientific American*, Vol. 237, No. 6, pp. 108–129, Dec. 1977.

[Lang 06] A. Langille and M. Gong. "An Efficient Match-based Duplication Detection Algorithm". In: *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision (CRV 2006)*, pp. 64–71, Quebec City, QC, Canada, June 2006.

[LEE Filt 12] "LEE Filters Worldwide, Central Way, Walworth Business Park, Andover, Hampshire, SP10 5AN, UK". `http://www.leefilters.com/`, Oct. 2012.

[Lee 86] H.-C. Lee. "Method for Computing the Scene-Illuminant Chromaticity from Specular Highlights". *Journal of the Optical Society of America A*, Vol. 3, No. 10, pp. 1694–1699, Oct. 1986.

[Lehm 01] T. M. Lehmann and C. Palm. "Color Line Search for Illuminant Estimation in Real World Scene". *Journal of the Optical Society of America A*, Vol. 18, No. 11, pp. 2679–2691, 2001.

[Levi 08] A. Levin, D. Lischinski, and Y. Weiss. "A Closed-Form Solution to Natural Image Matting". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 2, pp. 228–242, Feb. 2008.

[Li 03] Y. Li, S. Lin, H. Lu, and H.-Y. Shum. "Multiple-cue Illumination Estimation in Textured Scenes". In: *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV 2003)*, pp. 1366–1373, Nice, France, 2003.

[Li 07] G. Li, Q. Wu, D. Tu, and S. Sun. "A Sorted Neighborhood Approach for Detecting Duplicated Regions in Image Forgeries Based on DWT and SVD". In: *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2007)*, pp. 1750–1753, Beijing, China, July 2007.

[Li 08] B. Li, Y. Q. Shi, and J. Huang. "Detecting Doubly Compressed JPEG Images by Using Mode Based First Digit Features". In: *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP 2008)*, pp. 730–735, Cairns, Queensland, Australia, Oct. 2008.

[Lin 01] C.-Y. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui. "Rotation, Scale, and Translation Resilient Watermarking for Images". *IEEE Transactions on Image Processing*, Vol. 10, No. 5, pp. 767–782, May 2001.

[Lin 04] S. Lin, J. Gu, S. Yamazaki, and H.-Y. Shum. "Radiometric Calibration from a Single Image". In: *Proceedings of the 17th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pp. 938–945, Washington, DC, USA, June 2004.

[Lin 09a] H. Lin, C. Wang, and Y. Kao. "Fast Copy-Move Forgery Detection". *WSEAS Transactions on Signal Processing*, Vol. 5, No. 5, pp. 188–197, May 2009.

[Lin 09b] Z. Lin, J. He, X. Tang, and C.-K. Tang. "Fast, Automatic and Fine-grained Tampered JPEG Image Detection via DCT Coefficient Analysis". *Pattern Recognition*, Vol. 52, No. 11, pp. 2492–2501, Nov. 2009.

[Logv 05] A. D. Logvinenko, E. H. Adelson, D. A. Ross, and D. Somers. "Straightness as a Cue for Luminance Edge Classification". *Perception and Psychophysics*, Vol. 67, No. 1, pp. 120–128, Jan. 2005.

[Lu 09] R. Lu, A. Gijsenij, T. Gevers, V. Nedovic, D. Xu, and J.-M. Geusebroek. "Color Constancy using 3D Scene Geometry". In: *Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV 2009)*, pp. 1749–1756, Kyoto, Japan, Nov. 2009.

[Ludw 09] O. Ludwig Junior, D. Delgado, V. Gonçalves, and U. Nunes. "Trainable Classifier-Fusion Schemes: An Application to Pedestrian Detection". In: *Proceedings of the 12th IEEE International Conference on Intelligent Transportation Systems (ITSC 2009)*, St. Louis, MI, USA, Oct. 2009.

[Luka 03] J. Lukáš and J. Fridrich. "Estimation of Primary Quantization Matrix in Double Compressed JPEG Images". In: *Proceedings of the 3rd Digital Forensic Research Workshop (DFRWS 2003)*, Cleveland, OH, USA, Aug. 2003.

[Luo 06] W. Luo, J. Huang, and G. Qiu. "Robust Detection of Region-Duplication Forgery in Digital Images". In: *Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006)*, pp. 746–749, Hongkong, China, Aug. 2006.

[Mahd 07] B. Mahdian and S. Saic. "Detection of Copy-Move Forgery using a Method Based on Blur Moment Invariants". *Forensic Science International*, Vol. 171, No. 2, pp. 180–189, Dec. 2007.

[Maxw 08] B. Maxwell, R. Friedhoff, and C. Smith. "A Bi-Illuminant Dichromatic Reflection Model for Understanding Images". In: *Proceedings of the 21st IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK, USA, June 2008.

[McCa 99] J. McCann. "Lessons Learned from Mondrians Applied to Real Images and Color Gamuts". In: *Proceedings of the IS&T/SID Seventh Color Imaging Conference: Color Science, Systems and Applications (CIC 1999)*, pp. 1–8, Scottsdale, Arizona, US, Nov. 1999.

[Meyl 06] L. Meylan and S. Süsstrunk. "High Dynamic Range Image Rendering With a Retinex-Based Adaptive Filter". *IEEE Transactions on Image Processing*, Vol. 15, No. 9, pp. 2820–2830, Sep. 2006.

[Mozg 10] N. Mozgovaya. "Reuters under fire for removing weapons, blood from images of Gaza flotilla". `http://www.haaretz.com/news/diplomacy-defense/reuters-under-fire-for-removing-weapons-blood-from-images-of-gaza-flotilla-1.294780`, June 2010. Article in Haaretz newspaper, June 8th, 2010. Website last visited Jan. 18th, 2012.

[Muja 09] M. Muja and D. G. Lowe. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration". In: *Proceedings of the 4th International Conference on Computer Vision Theory and Applications (VISAPP 2009)*, pp. 331–340, Lisboa, Portugal, Feb. 2009.

[Myrn 07] A. N. Myrna, M. G. Venkateshmurthy, and C. G. Patil. "Detection of Region Duplication Forgery in Digital Images Using Wavelets and Log-Polar Mapping". In: *Proceedings of the IEEE International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, pp. 371–377, Sivakasi, India, Dec. 2007.

[Ng 04] T. Ng and S. Chang. "A Data Set of Authentic and Spliced Image Blocks". Tech. Rep. 20320043, Columbia University, June 2004.

[Ng 09a] T.-T. Ng and M.-P. Tsui. "Camera Response Function Signature for Digital Forensics — Part I: Theory and Data Selection". In: *Proceedings of the First IEEE International Workshop on Information Forensics and Security (WIFS 2009)*, pp. 156–160, London, England, UK, Dec. 2009.

[Ng 09b] T.-T. Ng and M.-P. Tsui. "Camera Response Function Signature for Digital Forensics — Part II: Signature Extraction". In: *Proceedings of the First IEEE International Workshop on Information Forensics and Security (WIFS 2009)*, pp. 156–160, London, England, UK, Dec. 2009.

[Nizz 08] M. Nizza and P. J. Lyons. "In an Iranian Image, a Missile Too Many". `http://thelede.blogs.nytimes.com/2008/07/10/in-an-iranian-image-a-missile-too-many/`, July 2008. Article in the New York Times blog. Website last visited Jan. 18th, 2012.

[OBri 12] J. F. O'Brien and H. Farid. "Exposing Photo Manipulation with Inconsistent Reflections". *ACM Transactions on Graphics*, Vol. 31, No. 1, pp. 1–11, Jan. 2012.

[Open 12] OpenCV. "Open Computer Vision Library". http://opencv.willowgarage.com, Apr. 2012.

[Ostr 05] Y. Ostrovsky, P. Cavanagh, and P. Sinha. "Perceiving Illumination Inconsistencies in Scenes". *Perception*, Vol. 34, No. 11, pp. 1301–1314, Nov. 2005.

[Pan 10] X. Pan and S. Lyu. "Region Duplication Detection Using Image Feature Matching". *IEEE Transactions on Information Forensics and Security*, Vol. 5, No. 4, pp. 857–867, Dec. 2010.

[Pena 12] O. A. B. Penatti, E. Valle, and R. da S. Torres. "Comparative Study of Global Color and Texture Descriptors for Web Image Retrieval". *Journal of Visual Communication and Image Representation*, Vol. 23, No. 2, pp. 359–380, Feb. 2012.

[Pope 04] A. C. Popescu and H. Farid. "Exposing Digital Forgeries by Detecting Duplicated Image Regions". Tech. Rep. TR2004-515, Department of Computer Science, Dartmouth College, 2004.

[Pope 05] A. C. Popescu and H. Farid. "Exposing Digital Forgeries in Color Filter Array Interpolated Images". *IEEE Transactions on Signal Processing*, Vol. 53, No. 10, pp. 3948–3959, Oct. 2005.

[Qu 08] Z. Qu, W. Luo, and J. Huang. "A Convolutive Mixing Model for Shifted Double JPEG Compression with Application to Passive Image Authentication". In: *Proceedings of the 33th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, pp. 1661–1664, Las Vegas, NV, USA, March 2008.

[RAL gGmb 12] "RAL gemeinnützige GmbH, Siegburger Straße 39, 53757 Sankt Augustin". `https://www.ral-farben.de/`, Oct. 2012.

[Rama 01] R. Ramamoorthi and P. Hanrahan. "On the Relationship between Radiance and Irradiance: Determining the Illumination from Images of a Convex Lambertian Object". *Journal of the Optical Society of America A*, Vol. 18, No. 10, pp. 2448–2459, Oct. 2001.

[Reuter 12] "Reuter Online Shop GmbH, Kühlenhof 2, 41169 Mönchengladbach, Deutschland". `http://www.reuter-badshop.com`, Oct. 2012.

[Ries 09a] C. Riess and E. Angelopoulou. "Physics-Based Illuminant Color Estimation as an Image Semantics Clue". In: *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP 2009)*, pp. 689–692, Cairo, Egypt, Nov. 2009.

[Ries 09b] C. Riess, E. Eibenberger, and E. Angelopoulou. "Illuminant Estimation by Voting". Tech. Rep., Friedrich-Alexander University Erlangen-Nuremberg, Sep. 2009.

[Ries 09c] C. Riess, J. Jordan, and E. Angelopoulou. "A Common Framework for Ambient Illumination in the Dichromatic Reflectance Model". In: *Proceedings of the IEEE Color and Reflectance in Imaging and Computer Vision Workshop (CRICV 2009)*, pp. 1939–1946, Kyoto, Japan, Oct. 2009.

[Ries 10] C. Riess and E. Angelopoulou. "Scene Illumination as an Indicator of Image Manipulation". In: *Proceedings of the 12th International Conference on Information Hiding (IH 2010)*, pp. 66–80, Calgary, AB, Canada, June 2010.

[Ries 11] C. Riess, E. Eibenberger, and E. Angelopoulou. "Illuminant Color Estimation for Real-World Mixed-Illuminant Scenes". In: *IEEE Color and Photometry in Computer Vision Workshop (CPCV 2011)*, pp. 782–789, Barcelona, Spain, Nov. 2011.

[Rose 03] C. Rosenberg, T. Minka, and A. Ladsariya. "Bayesian Color Constancy with Non-Gaussian Models". In: *Advances in Neural Information Processing Systems (NIPS 2003)*, Vancouver and Whistler, BC, Canada, Dec. 2003.

[Ryu 10] S. Ryu, M. Lee, and H. Lee. "Detection of Copy-Rotate-Move Forgery using Zernike Moments". In: *Proceedings of the 12th International Conference on Information Hiding (IH 2010)*, pp. 51–65, Calgary, AB, Canada, June 2010.

[Sabo 11] P. Saboia, T. Carvalho, and A. Rocha. "Eye Specular Highlights Telltales for Digital Forensics: A Machine Learning Approach". In: *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP 2011)*, pp. 1937–1940, Brussels, Belgium, Sep. 2011.

[Scha 04] G. Schaefer and M. Stich. "UCID - An Uncompressed Colour Image Database". In: *Proceedings of the SPIE - Storage and Retrieval Methods and Applications for Multimedia*, pp. 472–480, San Jose, CA, USA, Jan. 2004.

[Scha 05] G. Schaefer, S. D. Hordley, and G. D. Finlayson. "A Combined Physical and Statistical Approach to Colour Constancy". In: *Proceedings of the 18th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pp. 148–153, San Diego, CA, USA, June 2005.

[Schi 00] B. Schiele and J. Crowley. "Recognition without Correspondence using Multidimensional Receptive Field Histograms". *International Journal of Computer Vision*, Vol. 36, No. 1, pp. 31–50, Jan. 2000.

[Schi 96] B. Schiele and J. Crowley. "Object Recognition using Multidimensional Receptive Field Histograms". In: *Proceedings of the European Conference on Computer Vision (ECCV 1996)*, pp. 610–619, Cambridge, England, UK, Apr. 1996.

[Schw 09] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. "Human Detection Using Partial Least Squares Analysis". In: *Proceedings of the 12th IEEE International Conference on Computer Vision (ICCV 2009)*, pp. 24–31, Kyoto, Japan, Nov. 2009.

[Shaf 85] S. A. Shafer. "Using Color to Separate Reflection Components". *Journal Color Research and Application*, Vol. 10, No. 4, pp. 210–218, Winter 1985.

[Shen 08] L. Shen, P. Tan, and S. Lin. "Intrinsic Image Decomposition with Non-Local Texture Cues". In: *Proceedings of the 21st IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, AK, USA, June 2008.

[Shen 11] L. Shen and C. Yeo. "Intrinsic Images Decomposition Using a Local and Global Sparse Representation of Reflectance". In: *Proceedings of the 24th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pp. 697–704, Colorado Springs, CO, USA, June 2011.

[Shi 11] L. Shi and B. Funt. "Re-processed Version of the Gehler Color Constancy Dataset of 568 Images". `http://www.cs.sfu.ca/~colour/data/shi_gehler/`, Jan. 2011.

[Shie 06] J.-M. Shieh, D.-C. Lou, and M.-C. Chang. "A Semi-Blind Digital Watermarking Scheme based on Singular Value Decomposition". *Computer Standards & Interfaces*, Vol. 28, No. 4, pp. 428–440, Apr. 2006.

[Shiv 11] B. L. Shivakumar and S. Baboo. "Detection of Region Duplication Forgery in Digital Images Using SURF". *International Journal of Computer Science Issues*, Vol. 8, No. 4, pp. 199–205, July 2011.

[Spiegel 10] "Der Spiegel". June 2010. No. 23.

[Swai 91] M. Swain and D. Ballard. "Color indexing". *International Journal of Computer Vision*, Vol. 7, No. 1, pp. 11–32, Nov. 1991.

[Taka 09] T. Takai, A. Maki, K. Niinuma, and T. Matsuyama. "Difference Sphere: An Approach to Near Light Source Estimation". *Computer Vision and Image Understanding*, Vol. 113, No. 9, pp. 966–978, Sep. 2009.

[Tan 04] R. Tan, K. Nishino, and K. Ikeuchi. "Color Constancy through Inverse-Intensity Chromaticity Space". *Journal of the Optical Society of America A*, Vol. 21, No. 3, pp. 321–334, March 2004.

[Tan 05] R. T. Tan and K. Ikeuchi. "Separating Reflection Components of Textured Surfaces Using a Single Image". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 2, pp. 178–193, Feb. 2005.

[Tapp 05] M. F. Tappen, W. T. Freeman, and E. H. Adelson. "Recovering Intrinsic Images from a Single Image". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 9, pp. 1459–1472, Sep. 2005.

[Tapp 06] M. F. Tappen, E. H. Adelson, and W. T. Freeman. "Estimating Intrinsic Component Images using Non-Linear Regression". In: *Proceedings of the 19th IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, pp. 1992–1999, New York, NY, USA, June 2006.

[Tomi 91] S. Tominaga. "Surface Identification Using the Dichromatic Reflection Model". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, pp. 658–670, July 1991.

[Toro 07] J. Toro and B. Funt. "A Multilinear Constraint on Dichromatic Planes for Illumination Estimation". *IEEE Transactions on Image Processing*, Vol. 16, No. 1, pp. 92–97, Jan. 2007.

[Uytt 97] G. Uytterhoeven and A. Bultheel. "The Red-Black Wavelet Transform". Tech. Rep. TW271, Department of Computer Science, Katholieke Universiteit Leuven, Dec. 1997. `http://www.cs.kuleuven.ac.be/publicaties/rapporten/tw/TW271.abs.html`.

[Vazq 11] J. Vázquez i Corral. *Colour Constancy in Natural Images through Colour Naming and Sensor Sharpening.* PhD thesis, Universitat Autònoma de Barcelona, March 2011.

[Veda 08] A. Vedaldi and S. Soatto. "Quick Shift and Kernel Methods for Mode Seeking". In: *Proceedings of the European Conference on Computer Vision (ECCV 2008)*, pp. 705–718, Marseille, France, Oct. 2008.

[Veks 10] O. Veksler, Y. Boykov, and P. Mehrani. "Superpixels and Supervoxels in an Energy Optimization Framework". In: *Proceedings of the European Conference on Computer Vision (ECCV 2010)*, pp. 211–224, Hersonissos, Greece, Sep. 2010.

[Wang 09a] J. Wang, G. Liu, H. Li, Y. Dai, and Z. Wang. "Detection of Image Region Duplication Forgery Using Model with Circle Block". In: *Proceedings of the 1st International Conference on Multimedia Information Networking and Security (MINES 2009)*, pp. 25–29, Wuhan, China, June 2009.

[Wang 09b] J. Wang, G. Liu, Z. Zhang, Y. Dai, and Z. Wang. "Fast and Robust Forensics for Image Region-Duplication Forgery". *Acta Automatica Sinica*, Vol. 35, No. 12, pp. 1488–1495, Dec. 2009.

[Weij 05] J. van de Weijer, T. Gevers, and J.-M. Geusebroek. "Edge and Corner Detection by Photometric Quasi-Invariants". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 4, pp. 625–630, Apr. 2005.

[Weij 07a] J. van de Weijer, T. Gevers, and A. Gijsenij. "Edge-Based Color Constancy". *IEEE Transactions on Image Processing*, Vol. 16, No. 9, pp. 2207–2214, Sep. 2007.

[Weij 07b] J. van de Weijer, C. Schmid, and J. Verbeek. "Using High-Level Visual Information for Color Constancy". In: *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil, Oct. 2007.

[Winn 05] J. Winn, A. Criminisi, and T. Minka. "Object Categorization by Learned Universal Visual Dictionary". In: *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV 2005)*, pp. 1800–1807, Beijing, China, Oct. 2005.

[Wu 11] X. Wu and Z. Fang. "Image Splicing Detection Using Illuminant Color Inconsistency". In: *Proceedings of the 3rd IEEE International Conference on Multimedia Information Networking and Security (MINES 2011)*, pp. 600–603, Shanghai, China, Nov. 2011.

[Wysz 82] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons, 1982.

[XRite In 12] "X-Rite". `http://www.xrite.com/`, Sep. 2012.

[Yaho 12] Yahoo! Inc. "Flickr". `http://www.flickr.com/`, Sep. 2012.

[Ye 07] S. Ye, Q. Sun, and E.-C. Chang. "Detecting Digital Image Forgeries by Measuring Inconsistencies of Blocking Artifact". In: *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2007)*, pp. 12–15, Beijing, China, July 2007.

[Your 08] E. Yourdon. "Pedestrians at 72th subway". `http://www.flickr.com/people/yourdon/`, Apr. 2008. Flickr photo collection.

[Zach 12] F. Zach, C. Riess, , and E. Angelopoulou. "Automated Image Forgery Detection through Classification of JPEG Ghosts". In: *Proceedings of the German Association for Pattern Recognition (DAGM 2012)*, pp. 185–194, Graz, Austria, Aug. 2012.

[Zhan 08] J. Zhang, Z. Feng, and Y. Su. "A New Approach for Detecting Copy-Move Forgery in Digital Images". In: *Proceedings of the 11th Singapore International Conference on Communication Systems (ICCS 2008)*, pp. 362–366, Guangzhou, China, Nov. 2008.

Index