Distance error correction for time-of-flight cameras

Peter Fuersattel[a,b], Christian Schaller[b], Andreas Maier[a], and Christian Riess[a]

[a]Pattern Recognition Lab, Friedrich-Alexander University Erlangen-Nuremberg,
Martensstrasse 3, Erlangen, Germany
[b]Metrilus GmbH, Henkestrasse 91, Erlangen, Germany

ABSTRACT

The measurement accuracy of time-of-flight cameras is limited due to properties of the scene and systematic errors. These errors can accumulate to multiple centimeters which may limit the applicability of these range sensors. In the past, different approaches have been proposed for improving the accuracy of these cameras.

In this work, we propose a new method that improves two important aspects of the range calibration. First, we propose a new checkerboard which is augmented by a gray-level gradient. With this addition it becomes possible to capture the calibration features for intrinsic and distance calibration at the same time. The gradient strip allows to acquire a large amount of distance measurements for different surface reflectivities, which results in more meaningful training data. Second, we present multiple new features which are used as input to a random forest regressor. By using random regression forests, we circumvent the problem of finding an accurate model for the measurement error. During application, a correction value for each individual pixel is estimated with the trained forest based on a specifically tailored feature vector.

With our approach the measurement error can be reduced by more than 40% for the Mesa SR4000 and by more than 30% for the Microsoft Kinect V2. In our evaluation we also investigate the impact of the individual forest parameters and illustrate the importance of the individual features.

Keywords: Time-of-Flight, Systematic Errors, Random Forests

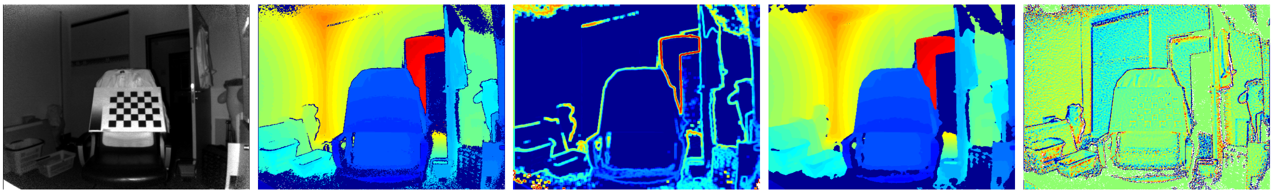
1. INTRODUCTION

Time-of-flight cameras acquire 3-D scene information at high framerates. This enables new opportunities for improving existing systems and developing entirely new applications. For example, the probably most well-known time-of-flight camera Kinect V2 is used both in entertainment systems and for research projects. More recently, time-of-flight sensors have also been integrated in mobile devices, like for example in Google's Tango Phablet.

However, current time-of-flight cameras provide only limited depth accuracy. The total depth error may accumulate to multiple centimeters.[1] These errors can be categorized into scene-dependent and systematic errors. A manufacturer has little to no influence on scene-dependent errors. However, even the most recent cameras still exhibit the typical systematic errors. In the last years, researchers aimed to develop error compensation methods to reduce the impact of these errors. Some approaches aimed at modeling the physical error sources,[2] while others model the resulting errors with parametric and non-parametric models.[3] More recently, the community started to exploit machine learning approaches[4] to model the error behavior more accurately. From an industry perspective, an ideal error compensation method shall include a) effective reduction of measurement errors, b) time-efficient, automated calibration, c) no additional hardware requirements and d) real-time capability.

The presented method uses random regression forests for modeling the errors of time-of-flight cameras. By using this model-free approach we avoid the problem of finding a representative error model for time-of-flight cameras. This decision also makes it feasible to use this method for different range cameras as the error model is not optimized for a particular camera. Based on the error model, we reducing several characteristic errors: temporal noise, the absolute distance error, and the so-called amplitude related error.

Further author information:
Peter Fuersattel: E-mail: peter.fuersattel@fau.de, Telephone: +49 9131 85 27882



(a) Amplitudes A (b) Distances D (c) Variance of D (d) Median of D (e) Difference from median D

Figure 1: Images of some selected features calculated on a regular checkerboard scene. Corresponding pixel values from different feature images are concatenated to form the feature vector.

In this work, we address two important aspects of error correction methods. First, we introduce a new calibration pattern which makes it possible to capture more meaningful training data without requiring the user to capture a specific distance calibration data set. Furthermore, we present several novel features for compensating measurement errors of range cameras.

The feature vector is specifically tailored to our application, and consists of distances, amplitude, statistical measures and filter outputs which can be calculated efficiently. In our experiments we evaluate different features which include: mean and median values of distances and amplitudes, deviations from the median, standard deviations, partial derivatives, distances from the image center as well as filtering results of guided image filters. A small subset of these features is shown in Figure 1.

The proposed calibration pattern is based on a checkerboard, but is augmented by an additional gray-level gradient strip. With this new calibration pattern both the intrinsic calibration and the depth calibration are performed. Distance measurements of the gray-level gradient allow capturing distances for a wide range of surface reflectivities which results in more meaningful training data. Based on the known geometry of the calibration pattern and the intrinsic camera parameters, the ground truth distances are obtained. In summary, with the correct calibration pattern, the training data for the error compensation is generated fully automatically with no additional manual effort during intrinsic calibration.

The presented method effectively reduces the amplitude-related error, the absolute measurement error as well as measurement noise. In our evaluation we study the performance of the proposed method for two time-of-flight sensors: the Mesa SR4000 and the Microsoft Kinect V2. Our experiments show that the mean measurement error its standard deviation can be reduced by more than 40% for both range cameras. Furthermore, we analyze the impact of the training parameters of the proposed method and evaluate the importance of the individual features. Finally we demonstrate how a smaller subset of the proposed features can be used to achieve comparable accuracy but with less computational burden.

In section 2 we present related work. The proposed method is described in section 3. Our experiments and a discussion of the results is presented in section 4. Section 5 summarizes and concludes this work.

2. RELATED WORK

The errors of time-of-flight cameras have been investigated in several studies,[1,5,6] and are typically categorized in systematic and scene-dependent errors. Systematic errors are explained with characteristics that are inherent to the camera technology. In contrast, scene dependent errors include effects that are caused by the environment, for example sunlight, low scene reflectivity, transparent or specular surfaces or multi-path.

There exist several approaches that aim at reducing the systematic errors based on individual camera parameters or observations from the scene. Lichti et al.[7] and Chow et al.[2] suggest a physically motivated error model for pixel-wise error correction. The parameters of these models are obtained via nonlinear optimization. Both methods use information of derived from amplitude and distance images. Kuznetsova and Bosenhahn[3] use non-parametric kernel regression to model the error. The proposed model predicts the error based on amplitude, distance and the distance of the particular pixel position to the principal point. Karel et al. proposed an method that reduces internal scattering via deconvolution with an estimate of the point spread function of the camera's lens.[6]

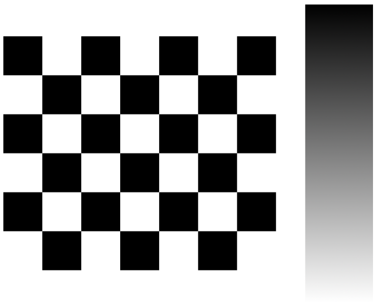


Figure 2: Checkerboard pattern with intensity gradient to provide training data that reflects the dependency between reflectivity and measurement error.

Reynolds et al.[8] used random regression forests to calculate confidence values for individual distance measurements of time-of-flight cameras. Song et al.[9] continued this approach and extended the set of features to make more accurate confidence predictions. Ferstl et al.[4] suggest to use a similar pipeline as presented by Reynold et al.[8] for calculating pixel-wise error correction terms. In their work, the authors suggest to exploit the color images of the Creative Senz3D to detect the calibration patterns more accurately.

Approaches for the reduction of scene-dependent errors typically focus on multi-path effects. Some methods exploit phase information captured at two or more modulation frequencies to filter multi-path effects mathematically.[10,11] Son et al.[12] use neural networks to compensate for multi-path errors for specific applications. The authors demonstrate how the impact of multi-path can be reduced for a robotic arm setup.

In our proposed method, we chose a machine learning approach to avoid the problem of finding an accurate model of the measurement error. Compared to previous approaches, we introduce new features which have not been used in the context of range camera calibration.

3. RANDOM FORESTS FOR DISTANCE ERROR CORRECTION

In this section we present more detailed information on the proposed method and provide an overview how it can be used to calibrate a time-of-flight camera. We divide the algorithm in multiple subsequent steps:

1. Intrinsic calibration of the camera
2. Training data generation
3. Feature calculation
4. Training of the random regression forest.

3.1 Intrinsic Camera Calibration and Data Acquisition

Intrinsic calibration is a common first step when setting up a new application that requires range cameras. We propose a special pattern for this task (see Figure 2). This pattern contains the typical planar checkerboard pattern which is commonly used for intrinsic calibration.[13] Additionally, it also contains a gray-level gradient which will later be used for the generation of training data. Thus the pattern is used both for intrinsic but also for range calibration.

The first step is the acquisition of amplitude and distance images that show the calibration pattern at different poses and distances. For the best results, these images should fulfill the following, mild conditions, of which most are already required for an accurate intrinsic calibration

- The pattern poses should vary with respect to rotation and translation to sufficiently constrain the nonlinear optimization problem of the intrinsic calibration.

- Calibration features should be distributed throughout the whole field of view to allow the estimation of an accurate lens model.
- The calibration patterns should be captured in the application's volume of interest to ensure that the calibration data reflects this scenario. This is particularly important for training data generation.
- Next to the amplitude images, which are used for intrinsic calibration, also the distance images need to be stored as they will be used for range calibration.

Once a sufficiently large data set has been captured, the corners of the checkerboard can be detected with any adequate detector. In our experiments we used the detector by Placht et al.[14] These calibration features and the known dimensions of the checkerboard can be used as input for intrinsic calibration. There exists a large variety of publicly available implementations, for example the Camera Calibration Toolbox[*] or OpenCV[†]. Both implementations are based on Zhang's method[13] and return the pinhole camera model parameters as well as the parameters of a lens model. In this work we use Π to describe the complete set of intrinsic camera parameters. The operator $\Pi(\cdot)$ will be used to project 3-D world coordinates $\mathbf{x}$ onto the 2-D image plane $\mathbf{p}$.

3.2 Training Data Generation

Random regression forests require labeled data for training. As labels, we use the difference between the measured pattern distance and the ground truth pattern distance. To compute the ground truth distances, we use the detected calibration features from the intensity image, the known pattern dimensions and the intrinsics Π. In detail, we obtain for each inner checkerboard corner a 3-D position. To obtain dense 3-D points for the complete pattern, we fit a plane segment P to these coordinates and intersect P with the viewing rays of all pixels that belong to the checkerboard. More formally, this backprojection is

$$d^* = \|\mathbf{x}\| = \left\|\Pi^{-1}(\mathbf{p}, P)\right\| \quad , \tag{1}$$

where the inverse projection $\Pi^{-1}(\mathbf{p}, P)$ takes two parameters, namely the 2-D image coordinate $\mathbf{p}$ and the plane P. From the intrinsics and the 2-D point coordinate, the direction of the viewing ray can be obtained. We calculate the 3-D world coordinate of the point on the pattern by intersecting this ray with P. Finally, calculating the norm of $\mathbf{x}$ results in the reference distance d^* from the camera to the checkerboard at a pixel $\mathbf{p}$. The measurement error e at a pixel position within the area of the checkerboard can now be computed with

$$e = d^* - d \quad . \tag{2}$$

For feature calculation and training we store the measurement error for pixels that belong to the calibration pattern.

3.3 Features for Regression

The random forest is trained such that it finds a correction value e^* based on up to 23 different features. These features are either directly available from a pair of amplitude and distance images, or are filter results. The complete set of features is listed in Table 1. The radial distance measures the distance of the current pixel position to the principal point. Statistics are calculated within 3×3 kernels. The guided image filter is applied three times: on the amplitude image, with the amplitude image as guidance and on the distance image with the amplitude and distance image as different guidances. Deviations from the median or guided filtering results are calculated by subtracting the measured value from the respective filtering result.

[*] http://www.vision.caltech.edu/bouguetj/calib_doc/

[†] http://opencv.org/

Table 1: Features which are evaluated in this work.

Directly available:	Distance, amplitude, radial distance
Derivatives:	$\frac{d}{dx}$, $\frac{d}{dy}$, $\frac{d^2}{dxdy}$ of the amplitude, $\frac{d}{dx}$, $\frac{d}{dy}$, $\frac{d^2}{dxdy}$ of the distance
Statistics:	Mean, variance, median of amplitude/distance, deviation from median distance/amplitude
Filters:	Guided image filter applied to amplitude/distance image, deviations from filter results

3.4 Training the Random Regression Forest

In this section we describe how random regression forests[15] can be used to model the error characteristics of time-of-flight cameras. The proposed error model M approximates the measurement error e^* based on a feature vector $\mathbf{f}$

$$e^* = M(\mathbf{f}) \ . \tag{3}$$

During training, each of the N_t decision trees is trained separately on a randomly selected subset of the training data. At each node of the decision tree the data is split by a applying splitting criterion to one of the components of the feature vector, e.g. a simple threshold. The splitting criterion is selected such that the homogeneity of the resulting two datasets (with respect to the current feature) is maximized. This process continues recursively until the maximum tree depth N_d is reached. The output of each tree is defined as the average of the data at the terminal node. After training the result of the model is given by the average response of all trees in the forest.

This training process directly leads to an important property of random forests: measuring the importance of the individual features can be calculated by counting which features have been selected how many times. Once training has finished, the number of occurrences of individual features at certain tree levels can be computed directly by traversing the trees.

4. EVALUATION

Different aspects of the proposed method are evaluated in this section: its capability for compensating measurement errors, the influence of the random forest parameters and the impact of the individual features. In this evaluation, we use the Microsoft Kinect V2 and the Mesa SR4000 to demonstrate the performance of the proposed method.

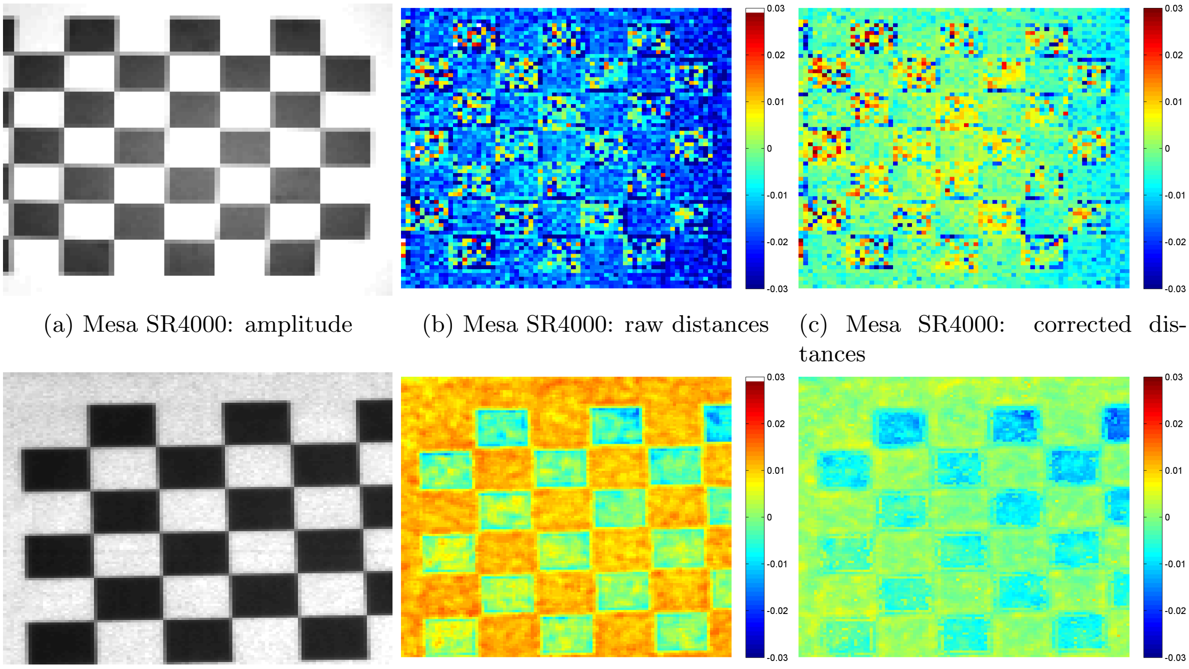
In section 4.1 we describe the used data. Qualitative results for both cameras are shown in section 4.2. The performance of the method with respect to measured distance is evaluated in section 4.3. The importance of the individual features and the performance of the method with only a subset of the proposed features is studied in section 4.4.

4.1 Datasets

With each camera 80 amplitude/distance images have been captured and evenly split into two data sets: one for calibration and one for evaluation. Each image pair shows the proposed checkerboard pattern at different poses and positions.

The checkerboard pattern has 5 × 6 inner corners with a quad length of 59.2 mm. The images are captured such that the patterns are distributed throughout the whole field of view. The distance between the camera and the pattern ranges from 0.7 m to 1.4 m for the Mesa SR4000 and from 1.25 m to 2.2 m for the Kinect V2.

The intrinsic parameters are calculated on the calibration data set, and are used to obtain ground truth distance information for training data generation. For the Mesa SR4000 camera we deactivated internal filtering and fixed the integration time to 5 ms.



(a) Mesa SR4000: amplitude

(b) Mesa SR4000: raw distances

(c) Mesa SR4000: corrected distances

(d) Kinect V2: amplitude

(e) Kinect V2: raw distances

(f) Kinect V2: corrected distances

Figure 3: Qualitative results for the Mesa SR4000 and the Kinect V2. The images show a small section of a checkerboard before and after error correction with the proposed method ($N_t = 15, N_d = 5$). Errors are given in meters.

4.2 Qualitative Examples

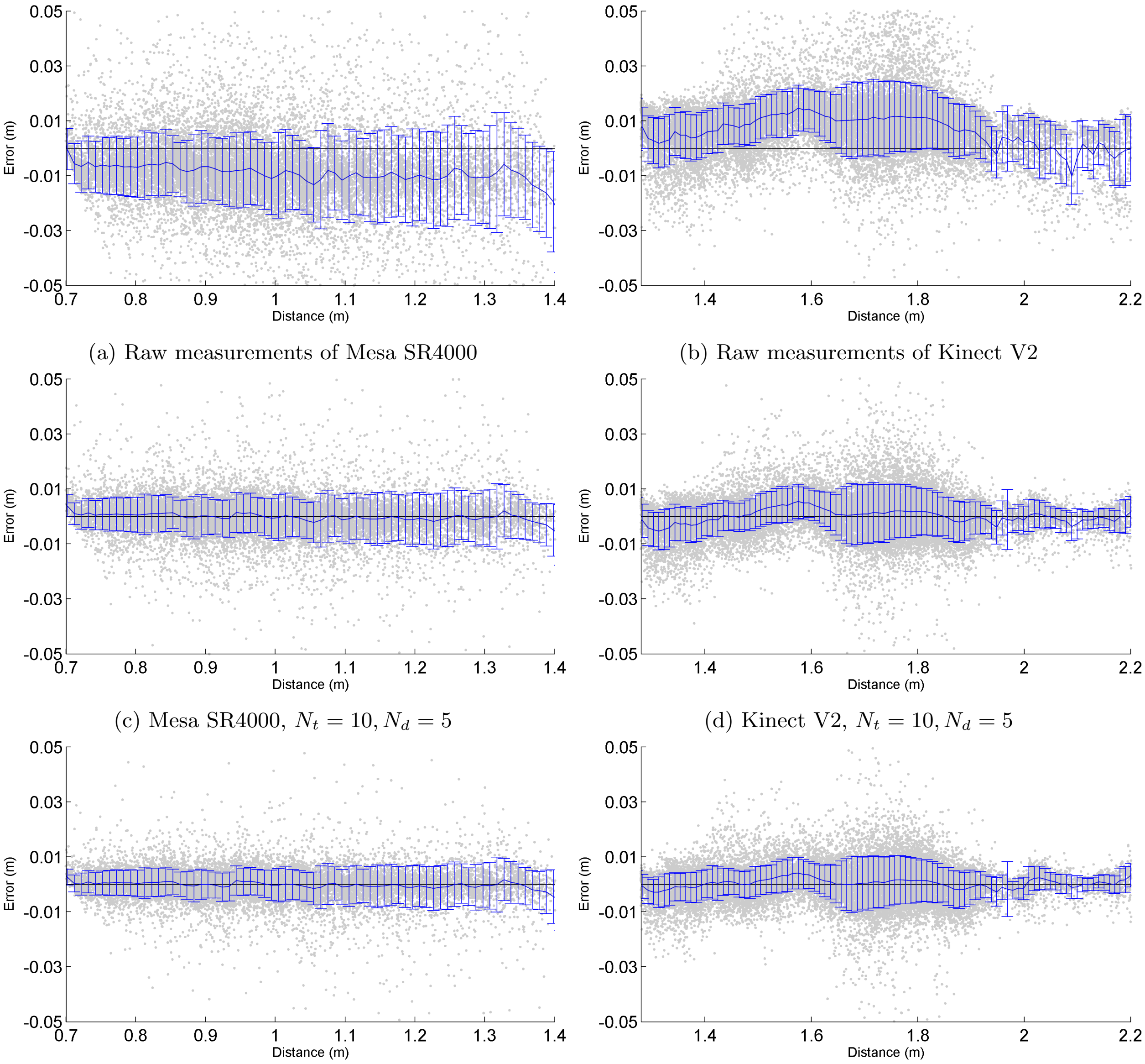
The proposed method effectively reduces amplitude-related measurement errors within the checkerboard pattern. The examples shown in Figure 3 show the difference from the ground truth distances. Each row of the figure shows the evaluated region in the amplitude image, the difference between the raw measurements and the ground truth and the difference between the error-compensated distances and the ground truth. In both examples the overall measurement error and the reflectivity-related errors are reduced. In the Mesa SR4000 example, the mean measurement error in the pictured area accounts for -1.55 cm. After applying the proposed method, the error has been reduced to -0.23 cm. For the Kinect V2 the accuracy can be improved from 0.68 cm to -0.13 cm.

4.3 Performance for Error Correction

In this section, we show how the distance accuracy is improved throughout the calibrated distance range. The accuracy for each camera is evaluated on the respective evaluation data set. In Figure 4 the results for the individual cameras are shown. In these figures individual point measurements are plotted as gray dots. The blue line indicates the mean measurement error and the standard deviation within 1 cm wide bins. In this experiment, two different parameter sets for the random regression forests have been used: ($N_t = 15, N_d = 5$) and ($N_t = 40, N_d = 8$).

Before error correction, the absolute measurement error for the Mesa SR4000 is approximately 1 cm. This offset is removed effectively by the proposed error correction method, independent of the chosen parameters. Furthermore, the standard deviation of the error can be reduced by more than 40 %. This effect becomes more distinct as larger values for N_t and N_d are chosen. Nonetheless, the impact of increasing the forest parameters is comparably small.

The raw distance-dependent errors of the Kinect V2 are shown in Figure 3d. Similarly as with the Mesa SR4000, the mean measurement error can be reduced significantly. In contrast to the Mesa SR4000, it is not possible to adjust internal parameters like temporal smoothing or spatial filtering. Instead, the Kinect V2 processes all distance measurements internally and returns only the filtered measurements. This results in a



(a) Raw measurements of Mesa SR4000

(b) Raw measurements of Kinect V2

(c) Mesa SR4000, $N_t = 10, N_d = 5$

(d) Kinect V2, $N_t = 10, N_d = 5$

(e) Mesa SR4000, $N_t = 40, N_d = 8$

(f) Kinect V2, $N_t = 40, N_d = 8$

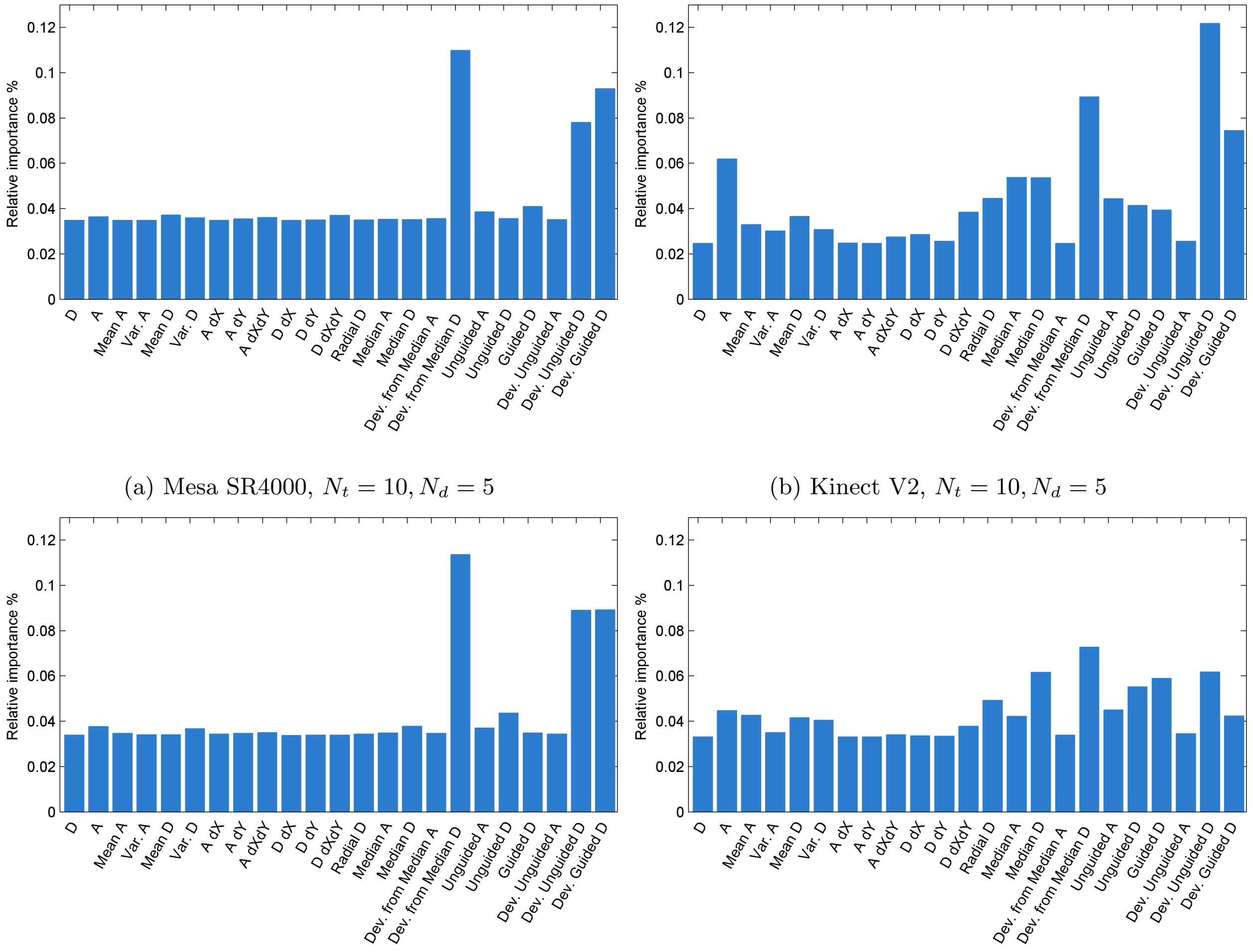
Figure 4: Comparison of measurement error before and after error correction for the Mesa SR4000 and Kinect V2. (a) and (b) show the original measurement error. (c) to (f) show the remaining errors after compensation for different forest parameters.

comparably small standard deviation of the raw range measurements. However, it is still possible reduce the standard deviation by more than 30 %, depending on the range of interest.

4.4 Influence of the Individual Features

In this section, we analyze the contribution of the individual features. The relative importance of a feature can be calculated by counting which feature is selected how often at the nodes of the forest. In Figure 5 the relative contributions of all features are visualized for two sets of forest parameters for both evaluated cameras.

With the Mesa SR4000, the deviation from the median and from the guided-image filtered distance images has the largest influence. This can be observed for both sets of parameters.



(a) Mesa SR4000, $N_t = 10, N_d = 5$

(b) Kinect V2, $N_t = 10, N_d = 5$

(c) Mesa SR4000, $N_t = 40, N_d = 8$

(d) Kinect V2, $N_t = 40, N_d = 8$

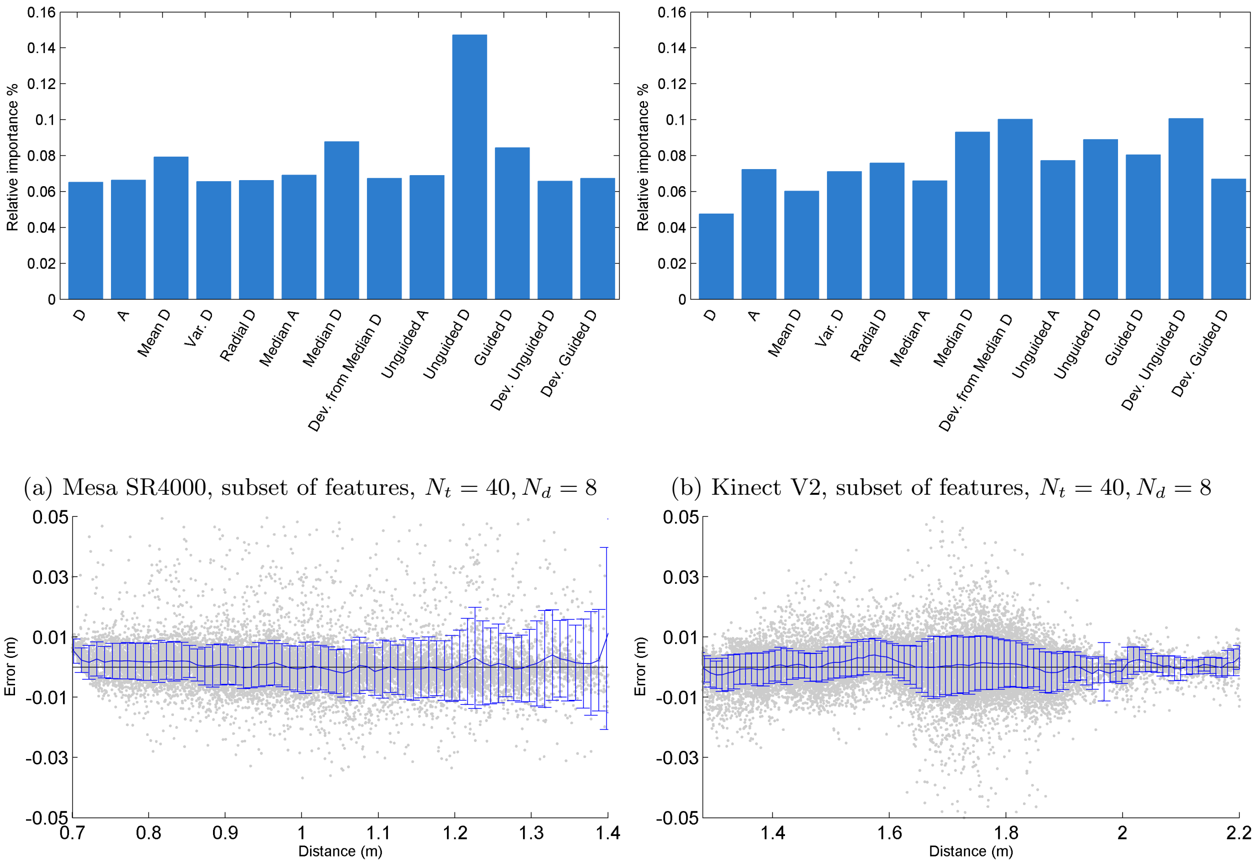
Figure 5: Importance of individual features for different forest parameters. The percentage expresses how often a particular feature is selected across all nodes of the forest. (a) and (b) show the results for $N_t = 10, N_d = 5$, (c) and (d) for $N_t = 40, N_d = 8$.

In the Kinect V2 examples, the deviation from the median and the deviation from the guided-image filtered image are also selected comparably often. However, other features, like the amplitude and the median of distance or amplitude values also affect the error estimate. When increasing the forest size and depth, the importance of prominent features is reduced and less expressive features gain influence.

Based on the previous results we created a new feature vector that contains only the most expressive features. In this study, we trained a forest ($N_t = 40, N_d = 8$) with the features shown in section 6. In figures 6a and 6b the relative importance of the features is plotted. Next we applied the method to the same evaluation data set as in subsection 4.3. For both cameras, the mean measurement error can still be reduced effectively. However, in case of the Mesa SR4000 an increasing standard deviation of the error can be observed as the distance increases. For the Kinect V2, results are comparable the results with the full set of features.

5. CONCLUSION

We presented a method for reducing the measurement errors of time-of-flight cameras. In this work we proposed a novel calibration pattern which allows the user to use the same dataset for both intrinsic and distance calibration. For intrinsic calibration we rely on a checkerboard pattern. For distance calibration, we augment this pattern by



(a) Mesa SR4000, subset of features, $N_t = 40, N_d = 8$

(b) Kinect V2, subset of features, $N_t = 40, N_d = 8$

(c) Mesa SR4000, remaining error after correction

(d) Kinect V2, remaining error after correction

Figure 6: Importance of individual features for different forest parameters. The percentage expresses how often a particular feature is selected across all nodes of the forest. (a) and (b) show the results for $N_t = 10, N_d = 5$, (c) and (d) for $N_t = 40, N_d = 8$.

a gray-level gradient, which makes it possible to obtain training data for a large range of surface reflectivities. We use random regression forests in combination with a specifically tailored feature vector to estimate a model-free representation of the error characteristics of the camera. Based on this new feature vector, accurate error correction terms for each individual pixel can be obtained.

The presented method effectively reduces the amplitude-related error, the absolute measurement error as well as measurement noise. In our evaluation we study the performance of the proposed methods with two time-of-flight cameras. The experiments show that the mean measurement error and its standard deviation can be reduced by up to 40% for the Mesa SR4000 and by 30% for the Microsoft Kinect V2. By analyzing the importance of the individual features, a subset of the most meaningful features can be selected. Finally we demonstrate how this subset can be used to obtain comparable accuracy but with less computational burden.

ACKNOWLEDGMENTS

This work was partly supported by the Research Training Group 1773 “Heterogeneous Image Systems”, funded by the German Research Foundation (DFG).

REFERENCES

[1] Fuersattel, P., Placht, S., Balda, M., Schaller, C., Hofmann, H., Maier, A., and Riess, C., "A comparative error analysis of current time-of-flight sensors," *IEEE Transactions on Computational Imaging* **2**(1), 27–41 (2016).

[2] Chow, J. C. K., Lichti, D. D., Remondino, F., Shortis, M. R., Beyerer, J., and Puente León, F., "A study of systematic errors in the pmd camboard nano," in [*SPIE Optical Metrology 2013*], 87910X (2013).

[3] Alina Kuznetsova and Bodo Rosenhahn, "On calibration of a low-cost time-of-flight camera," *IEEE European Conference on Computer Vision Workshops (ECCVW)* (2014).

[4] Ferstl, D., Reinbacher, C., Riegler, G., Rüther, M., and Bischof, H., "Learning depth calibration of time-of-flight cameras," in [*Proceedings of the British Machine Vision Conference 2015*], Xie, X., Jones, M. W., and Tam, G. K. L., eds., 102.1–102.12, BMVA Press, Durham (2015).

[5] Foix, S., Alenya, G., and Torras, C., "Lock-in time-of-flight (tof) cameras: A survey," *IEEE Sensors Journal* **11**(9), 1917–1926 (2011).

[6] Karel, W., Ghuffar, S., and Pfeifer, N., "Modelling and compensating internal light scattering in time of flight range cameras," *The Photogrammetric Record* **27**(138), 155–174 (2012).

[7] Lichti, D. D., Kim, C., and Jamtsho, S., "An integrated bundle adjustment approach to range camera geometric self-calibration," *ISPRS Journal of Photogrammetry and Remote Sensing* **65**(4), 360–368 (2010).

[8] Reynolds, M., Dobos, J., Peel, L., Weyrich, T., and Brostow, G. J., "Capturing time-of-flight data with confidence," in [*2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 945–952.

[9] Song, X., Zhong, F., Wang, Y., and Qin, X., "Estimation of kinect depth confidence through self-training," *The Visual Computer* **30**(6-8), 855–865 (2014).

[10] Dorrington, A. A., Godbaz, J. P., Cree, M. J., Payne, A. D., Streeter, L. V., Beraldin, J. A., Cheok, G. S., McCarthy, M. B., Neuschaefer-Rube, U., Baskurt, A. M., McDowall, I. E., and Dolinsky, M., "Separating true range measurements from multi-path and scattering interference in commercial range cameras," in [*IS&T/SPIE Electronic Imaging*], *SPIE Proceedings*, 786404–786404–10, SPIE (2011).

[11] Freedman, D., Smolin, Y., Krupka, E., Leichter, I., and Schmidt, M., "Sra: Fast removal of general multipath for tof sensors," in [*Computer vision - ECCV 2014*], Fleet, D., ed., *Lecture Notes in Computer Science* **8689**, 234–249, Springer, Cham (2014).

[12] Son, K., Liu, M.-Y., and Taguchi, Y., "Automatic learning to remove multipath distortions in time-of-flight range images for a robotic arm setup," (2016).

[13] Zhang, Z., "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11), 1330–1334 (2000).

[14] Placht, S., Fürsattel, P., Mengue, E. A., Hofmann, H., Schaller, C., Balda, M., and Angelopoulou, E., "Rochade: Robust checkerboard advanced detection for camera calibration," in [*Computer Vision – ECCV 2014*], Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., eds., *Lecture Notes in Computer Science* **8692**, 766–779, Springer International Publishing, Cham (2014).

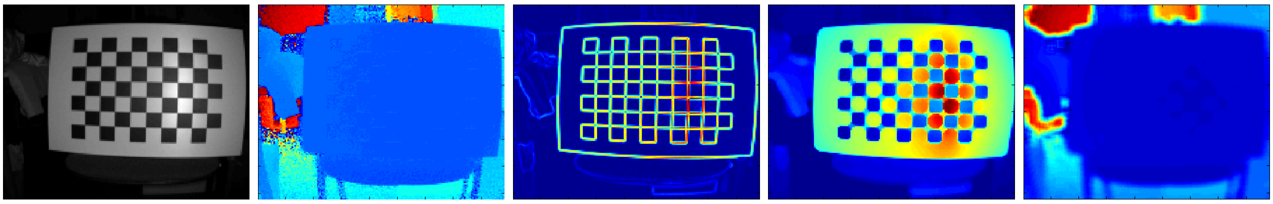
[15] Criminisi, A., "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends® in Computer Graphics and Vision* **7**(2-3), 81–227 (2011).

Supplemental Material for
Distance Error Correction for Time-of-Flight Cameras

Time-of-flight cameras acquire 3D scene information at high framerates. This enables new opportunities for improving existing systems and entirely new applications. Probably the most well-known time-of-flight camera is the Kinect V2, which is used both in entertainment systems, but also has also been adopted by the research community. More recently, time-of-flight sensors have also been integrated in mobile devices, like for example in Google's Tango Phablet.

However, current time-of-flight cameras provide only limited depth accuracy. The total depth error may accumulate to multiple centimeters.[1] Depth errors can be categorized into scene-dependent and systematic errors. A manufacturer has little to no influence on scene-dependent errors. However, even most recent cameras exhibit typical systematic errors. In the last years, researchers aimed to develop error compensation methods to reduce the impact of these errors. Some approaches aimed at modelling the physical error sources,[2] while others model the resulting errors with parametric and non-parametric models.[3] More recently, the community started to exploit machine learning approaches[4] to model the error behavior more accurately. From an industry perspective, an ideal error compensation method shall include a) effective reduction of measurement errors, b) time-efficient, automated calibration, c) no additional hardware requirements and d) real-time capability.

In this work, we propose to use random regression forests for compensating systematic measurement errors of time-of-flight cameras. The method aims at reducing several errors: temporal noise, the absolute distance error, and the so-called amplitude related error. The method makes use of a specifically tailored feature vector that consists of distances, amplitude, statistical measures and filter outputs which can be calculated efficiently. In our experiments we evaluate different features which include: mean and median values of distances and amplitudes, deviations from the median, standard deviations, partial derivatives, distances from the image center as well as filtering results of guided image filters. A small subset of these features is shown in Figure 1.



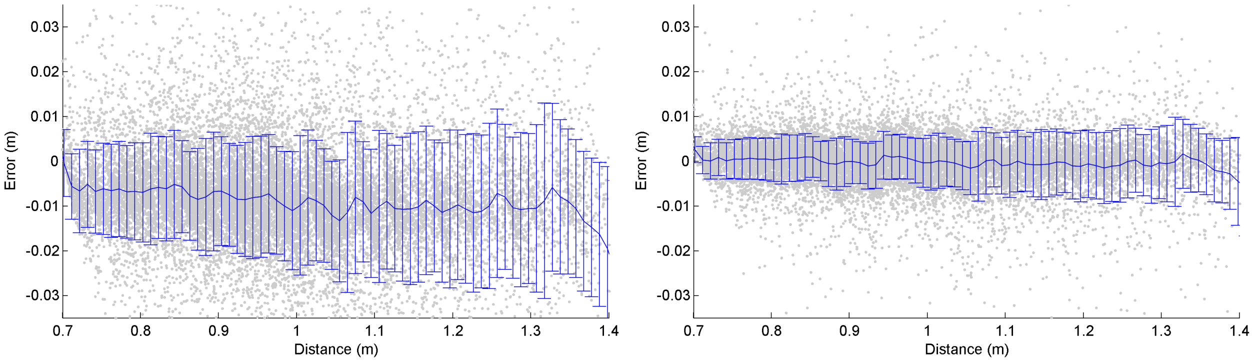
(a) Amplitudes A (b) Distances D (c) Derivative of A (d) mean(A) (e) mean(D)

Figure 1: Images of some selected features calculated on a regular checkerboard scene. Corresponding pixel values from different feature images are concatenated to form the feature vector.

We demonstrate how to train the random regression forest and how to generate training data during the intrinsic calibration process which is performed with well-established methods. The method makes the common, mild assumption that the calibration pattern is planar. However, this only constraint is typically already required for intrinsic calibration.

To perform the calibration, we propose to use an extended checkerboard pattern, which is augmented by an additional gray-level gradient strip. With this new calibration pattern both the intrinsic calibration and the depth calibration are performed. Distance measurements of the gray-level gradient allow capturing a wide range of amplitude values which result in different errors and thus in more meaningful training data.

Based on the known geometry of the calibration pattern and the intrinsic camera parameters, the ground truth position of the pattern plane can be obtained. Ground truth distance values for all pixels within the pattern area can be calculated by intersecting a ray, starting at a particular pixel in the image, with the pattern plane. In summary, with the correct calibration pattern, the training data for the error compensation is generated fully automatically with no additional manual effort during intrinsic calibration.



(a) Distance error without error compensation

(b) Distance error with the proposed method

Figure 2: Error compensation performance for the Mesa SR4000 time-of-flight sensor, evaluated for the range from 70 cm to 140 cm. The solid line shows the measurement error, averaged for 1 cm wide bins. The error bars represent the respective standard deviation.

The presented method effectively reduces the amplitude-related error, the absolute measurement error as well as measurement noise, resulting in a reduction of the mean and standard deviation of the error by more than 40%. An excerpt from our evaluation is shown in Figure 2. The figures show the measurement error for the Mesa SR4000 time-of-flight camera in the range starting from 70 cm to 140 cm without and with the proposed method. Figure 2 shows two effects: a) without our error compensation method, a clear gobal offset of almost a centimeter can be observed, and b) with the proposed method the standard deviation of the measurement error can be reduced efficiently. This is achieved by including spatial features into the error compensation process which effectively performs edge-dependent smoothing.

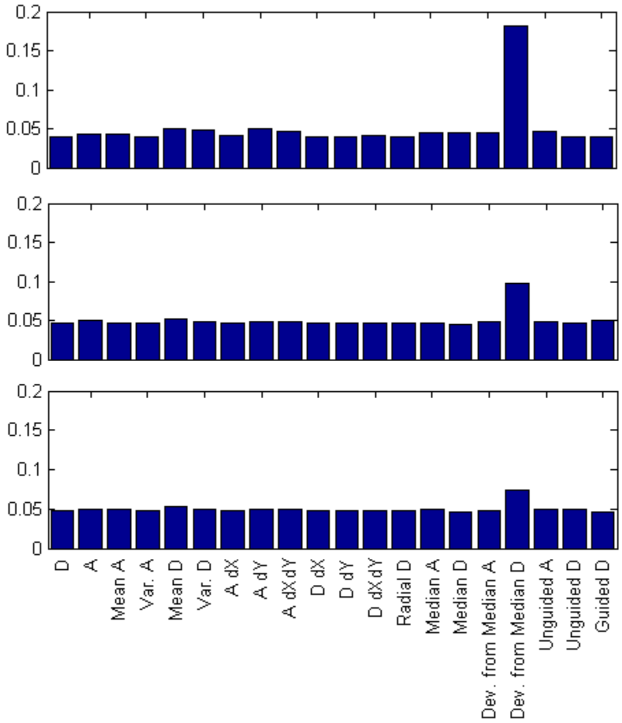


Figure 3: Feature importance for different forest settings. The top figure shows the feature importances for the complete feature set with $N_T = 10$ trees and a maximum tree depth of $N_D = 3$. The center and image show the importances for $(N_T = 10, N_D = 5)$ and $(N_T = 40, N_D = 8)$ respectively.

REFERENCES

1. P. Fuersattel, S. Placht, M. Balda, C. Schaller, H. Hofmann, A. Maier, and C. Riess, "A comparative error analysis of current time-of-flight sensors," *IEEE Transactions on Computational Imaging* **2**(1), pp. 27–41, 2016.
2. J. C. Chow and D. D. Lichti, "A study of systematic errors in the pmd camboard nano," in *SPIE Optical Metrology 2013*, pp. 87910X–87910X, International Society for Optics and Photonics, 2013.
3. A. Kuznetsova and B. Rosenhahn, "On calibration of a low-cost time-of-flight camera," in *Workshop at the European Conference on Computer Vision*, pp. 415–427, Springer, 2014.
4. D. Ferstl, C. Reinbacher, G. Riegler, M. Rüther, and H. Bischof, "Learning depth calibration of time-of-flight cameras," in *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 102.1–102.12, September 2015.