

Extracting Sensor Specific Noise Models

Nicolas Grossmann *

Supervised by: Dr. Stefan Ohrhallinger †

Institute of Visual Computing and Human-Centered Technology
Technical University Vienna
Vienna / Austria

Abstract

With the growing number of consumer-oriented depth sensors like the KinectV2 or the newly released Phab2Pro, the question of how precise these sensors are arises. In this work we want to evaluate the average noise in the generated depth measurements in the axial direction and the lateral directions. As part of a two-part project this work will view the noise's development with varying distance and angle. We will present and evaluate two empirical models describing the noise behavior, with the first being derived from solely this part's measurements and the second one being a combination of the previous model and the model of Köppel et al. [7]. These derived models can be used in a post-processing step to filter the generated depth images. We evaluate our models through statistical and experimental testing.

Keywords: noise model, surface reconstruction, sensor noise

1 Introduction

Over the last years several customer-oriented depth sensors arrived on the market. These sensors allow to record the distances to the closest objects, which is particularly useful for segmentation and recognition purposes. The Kinect series, from Microsoft, consists of such sensors. Although it was created for gaming purposes it was also used by businesses and scientists. This resulted in many papers dealing with the use of the Kinect in robotics [2], human- [13], gesture- [9] and object-detection [6], medical visualization [4] and many more. Just recently this technology found its way into smartphones. The Phab2Pro, from Lenovo, uses a sensor similar to the one in the KinectV2, but in a much smaller size. The provided data is used for augmented reality apps. The problem with most of these affordable depth sensors is that they are prone to have high amounts of noise (measurement errors) in their generated depth images. Errors in the depth measurements may lead to problems for different applications, for example, reconstructed surfaces and

objects may have lots of small bumps or holes. One successful way to reduce this problems is filtering the measurement results using an empirically derived model of the sensor noise.

This paper presents one part of a two-part project, trying to estimate elaborate empirical noise models for both the KinectV2 and the Phab2Pro. Our main contributions are as follows:

- Presenting an extraction algorithm for planar targets in depth images
- Extracting sensor noise in axial and both lateral directions
- Estimating empirical models describing the noise based on the target's distance and rotation
- Extending the estimated models by a weighted combination with a second model
- Validation of the derived models by measuring a well-defined object

This paper is structured in such a way that in Section 2 the previous work done in this research field will be briefly described. Following in Section 3 the sensor technologies, their error sources and the ways to analyze these errors are described. In Section 4, we present the experimental setups and the processing steps to estimate the noise models. After that, we present the results in Chapter 5 together with the estimated models for the experiments of this paper and the one which resulted from the combination with the model of another project. Furthermore, we show the results of a simple evaluation where we analyzed the quality of our models by comparing them to preceding models in the literature and by evaluating them with a simple experiment. Finally, in Chapter 7, we discuss the results of our work and how it could be improved.

2 Related Work

One of the first papers that deals with the problem of noisy KinectV1 depth measurements was written by Khoshelham et al. [6]. They showed the need of a

*nicolas.grossmann@cg.tuwien.ac.at

†ohrhallinger@cg.tuwien.ac.at

calibration procedure, but also statistically analyzed the measurement error in regards to the distance between the sensor and a planar surface. Based on this paper, Nguyen et al. [8] extended the previous experiment with the KinectV1 to additionally estimate the noise in regards to the angle of the surface. The resulting data was then used to extend the KinectFusion reconstruction pipeline which led to improved results. That this type of experimental setup is also applicable for the KinectV2 was shown by Fankhauser et al. [3], who evaluated the suitability of the KinectV2 as a sensor for mobile robots. Their work extended the previous experimental setup of distance and angle with an additional influx of sunlight.

While these papers provided the basis for large parts of this project: namely the experimental setup and the estimation of the noise model based on target distance and rotation, the final model was created through a collaboration. Köppel et al. [7] estimated in their project the effects of the target's distance and image position on the noise of both the KinectV2 and the Phab2Pro. By measuring a 3D checkerboard pattern at several different locations in the view frustum of the sensor, they were able to extract the image position dependent noise in axial and lateral direction.

3 Theory

Both, the KinectV2 and the Phab2Pro, are time-of-flight sensors. [10] They estimate the distance of objects by measuring the travel time of infrared light pulses. Compared to other depth sensing technologies, like structured-light or laser-based ones, this architecture has several advantages. Firstly it is rather simple without the need of moving parts or a wide aperture and secondly depth sensors using this principle generally have rather high frame rates, some even above 100 frames per second. But this architecture also has some problems, namely its susceptibility to other infrared light sources like the sun or other time-of-flight sensors. Additional factors like temperature of the camera [1] and reflectiveness of the measured objects may also influence the results.

This work mainly focuses on analyzing the systematic error (which remains after a calibration procedure) and its relation to the distance and rotation of an object. To calculate an error model we use a stochastic approach where the error is calculated based on the difference between measurement results and ground truth information. Other works have shown that this approach is suitable for time-of-flight sensors. [3] The systematic noise has axial and lateral components that need to be analyzed separately as previously shown by Nguyen et al. [8]

The axial noise describes the depth measurement error along the z-axis. In general, it describes the average dif-

ference between the measured depth and the actual depth (as seen in Figure 1). This type of error is determined by measuring a flat surface, fitting an ideal plane through the measurement points and then calculating the difference between the measurements and the assumed surface plane. [8] Because the noise is assumed to have a normal distribution (Nguyen et al. [8] and Fankhauser et al. [3]), the standard deviation can be used as a characterizing parameter.

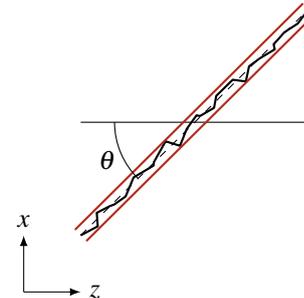


Figure 1: Axial noise along the face of the planar target

The lateral noise is a measure for the pixel deviation along the two image axes x and y. It describes how much the measured point is vertically or horizontally misplaced compared to the actual point on the object. To measure this, the plane is placed in such a way that its border points are measured as straight lines. The difference between the presumed border line and the measured one is the lateral noise (as seen in Figure 2). This noise can be measured for the x- and the y-direction. [8] Previous works showed that the resulting distribution is only partially described by a normal distribution, but because many frameworks assume this distribution, the standard deviation is used as a measure. [3]

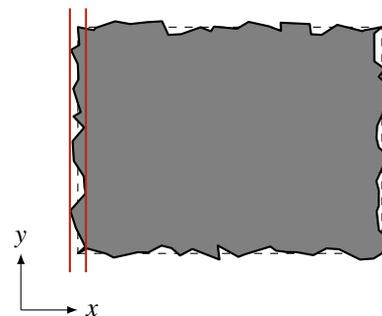


Figure 2: Lateral noise alongside the edge of the planar target

4 Method

In this section we will describe the experimental setups we used in our work, as well as the processing steps we took to extract the valuable data from the measurements and how we calculated the noise models.

4.1 Experimental Setup

Our test setup (as seen in Figure 3) was nearly identical to the ones used by Nguyen et al. [8] and Fankhauser et al. [3], which were based on Khoshelham et al. [6]. As the target a wooden press board with a faint white surface was mounted onto a tripod. The board itself was placed on a rotation device, which is used in photography to rotate a camera to a certain angle. This allowed us to extract the angular noise characteristics by rotating the plane vertically from 0° to 80° . The camera in our setup was placed on a wooden construct that in turn was standing on a desk in front of the fixed target. We used a metal tape ruler as a rail for our camera construct while we moved it from 0.9m to 3.0m in intervals of 0.1m to capture the distance dependent noise characteristics. Before the measurements were started, the camera was running for at least 30 minutes to reduce temperature dependent errors [1]. After that, the test procedure started at 0.9 m at 0° . At each setting of distance z and angle θ 200 images were taken. 100 images with the camera in a horizontal position and 100 in a vertical position to capture the lateral noise in both x- and y-direction.

4.2 Plane Extraction

After the point clouds were captured they were subject to several preprocessing steps to ensure a valid noise estimation. First the image distortion of the cameras was accounted for by applying a standard calibration procedure. For the KinectV2 we used a checkerboard pattern printed on an A4 sized paper as a reference [12] and took several infrared images with the pattern at different positions and rotations. These images were then used by a camera calibration tool to estimate the concrete camera parameters, which in turn allowed us to undistort the depth images. [14] The Phab2Pro removes lens distortions by default [5] in their reconstructed point clouds. After that, we extracted the position of the plane in the measurements to serve as a region of interest for further calculations. This was done by applying an automated pipeline consisting of several image processing steps (which can be seen in Figure 4):

1. **Create Binary Image** - An estimated depth threshold based on plane distance and rotation was applied to transform the image into a binary format.
2. **Mask Out Unused Regions** - A manually selected region was used as a mask to reduce the influence of



Figure 3: Images of the setups for the two different sensors, consisting of a fixed, wooden plane and a moving construct for the sensor

the surrounding objects.

3. **Remove Noise** - The amount of noise in the image was reduced by a hole filling algorithm and by first dilating and then eroding the image.
4. **Erode and Dilate Plane** - A structure element shaped like the plane was calculated and used to erode and dilate the image, effectively removing other objects.
5. **Calculate Bounding Box** - The remaining white region was assumed to be the plane and so its bounding box was extracted.

4.3 Noise Extraction

Using the estimated position of the plane in the depth measurements the lateral and axial noise could be calculated. For the lateral noise the two plane border regions on the left and right side were selected. By combining a threshold and a Canny edge detector the border pixels were extracted. A vertical line was fitted through them and the resulting standard deviation of the pixels was assumed to be the lateral noise $\sigma_L[px]$. As the result was pixel based, it had to be transformed to a metric measure using the estimated camera parameters. The axial noise region was selected by shrinking the bounding box by around 20% to remove possible influence of the lateral noise. The points inside this region were selected and a plane was fitted through them. The axial noise $\sigma_A[mm]$ was then estimated

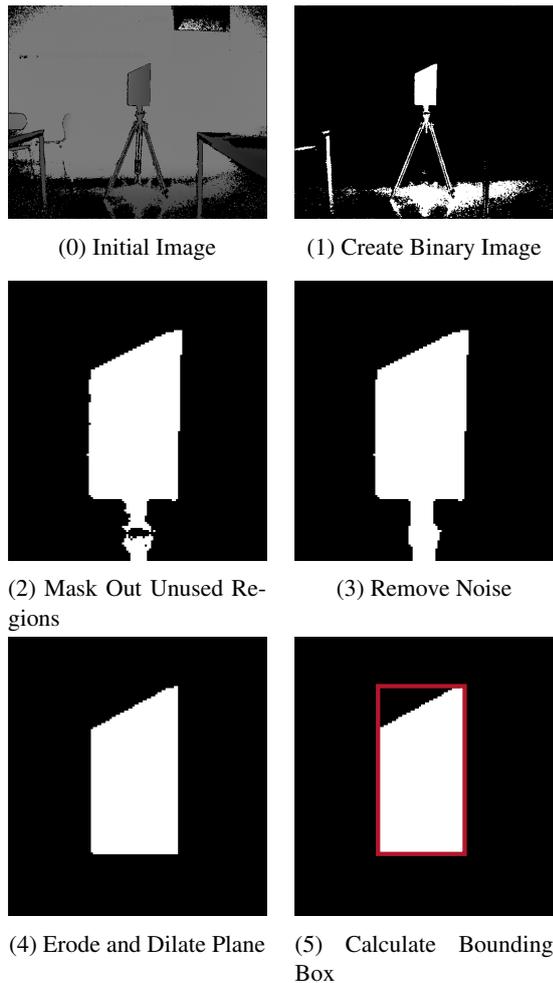


Figure 4: Processing steps of the plane extraction algorithm, starting with a depth image and ending with a bounding box for the planar target

by calculating the standard deviation of the points from the plane.

4.4 Model Estimation

Using the resulting noise estimations, empirical models can be drawn from the data. For the first model we used the distance z and surface rotation θ as input parameters. The second model was created by adding two parameters x and y for the image position of a measurement. It was created by combining the first model with the model created by Köppel et al. [7]. We defined a good model as being both able to describe the underlying data as close as possible, while still remaining general enough to be able to predict unmeasured values. For the first constraint fitting-quality measures like the RMSE were used. The second constraint is not easily verifiable, for it we tried to have a function with an as low as possible degree and without too high jumps between measured values.

5 Results

The first important result of our work is the recognition rate of our plane extraction algorithm, we estimated this based on the number of times the plane was detected. As seen in Figure 5, the recognition rate for both cameras is 100% for angles between 0° and 60° . For 70° , the KinectV2 and Phab2Pro recognition start to suffer from errors, which grow bigger the larger the distance gets. Looking at the raw data it was clear that this problem was due to the high noise at these positions. For each position the average detected region was chosen for further calculations. This allowed us to use all measurements for a position as long as any region was detected at that position. For positions where no region was detected due to the high noise, the noise was assumed to approach infinity.

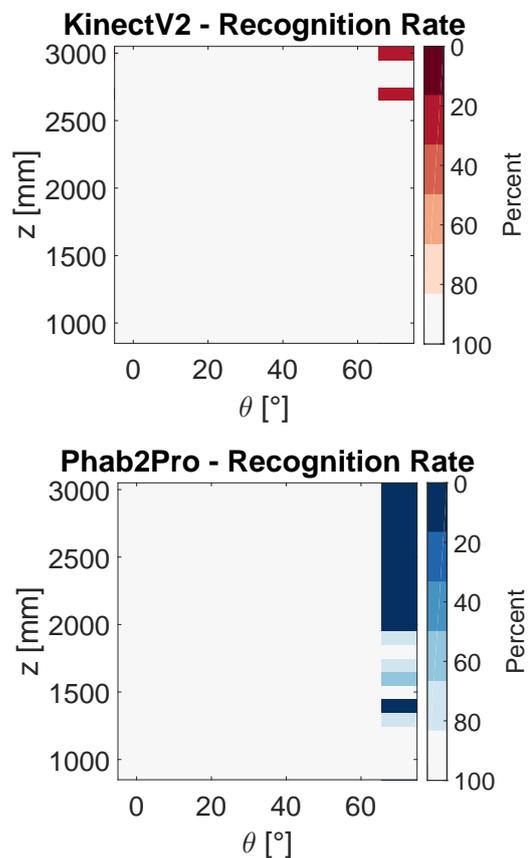


Figure 5: Comparison of the recognition rate of our algorithm for the KinectV2 and Phab2Pro on the complete dataset

Because the measures used in this work, like the standard deviation and the noise models calculated from them, are based on the assumption that the systematic error is normally distributed, one of our next steps was to validate this claim. This was done by both looking at the histograms (seen in Figure 6) of the calculated axial and

lateral noise and calculating the number of measurements, whose noise distributions correspond to a normal distribution based on the Kolmogorow-Smirnow-Test [11]. We tested for a normal distribution at a significance level of 5%. For the axial noise distribution this shows that around 80% of the data is describable through a normal distribution, with most of the not normally distributed measurements lying at 70° . The lateral noise for both the x- and the y-direction do not show clear signs of a normal distribution. Both of these effects were already observed by Fankhauser et al. [3].

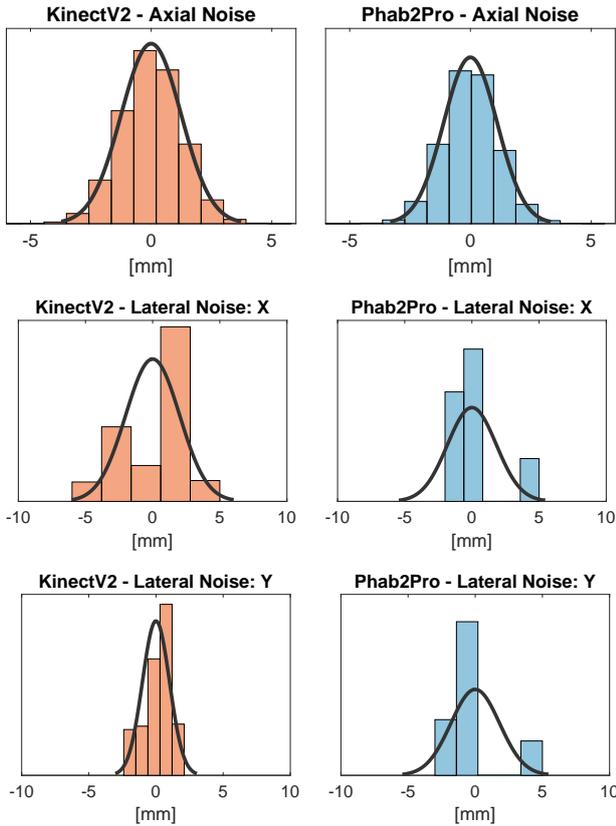


Figure 6: Histograms illustrating the noise distribution at $z = 1$ m and $\theta = 0^\circ$ for both sensors in axial and lateral directions. The line represents the fitted normal distribution.

For each position the noise is calculated based on all measurements. These noise values describe the fluctuation for a measured depth point on a surface. We analyzed how the intensity of noise changes with the two independent variables z and θ . Our axial model can be seen in Equation 1. The axial noise seems to have a low linear growth between 0° and 40° for both sensors. After that, the noise seems to grow rapidly with higher degrees approaching infinity. This behavior was already shown for the KinectV2 by Fankhauser et al. [3], whose model was adapted for our measurements and additionally applied to the Phab2Pro. Their function is a combination of a quadratic part for the distance z and a hyperbolic

part for the rotation θ . While the most coefficients were fitted, as previously explained, the values for e were found manually.

$$\sigma_{z_1} = a + b * z + c * z^2 + d * z^e * \frac{\theta^2}{(\frac{\pi}{2} - \theta)^2} \quad (1)$$

	KinectV2	Phab2Pro
a	2.094	0.3019
b	$-1.099 * 10^{-3}$	$5.712 * 10^{-4}$
c	$4.048 * 10^{-7}$	$6.183 * 10^{-7}$
d	$6.846 * 10^{-7}$	$2.386 * 10^{-5}$
e	1.7	1.47

Table 1: Axial Model Coefficients (Equation 1)

Because of the seemingly random nature of the lateral noise in our measurements, a mathematical model could not be fitted to the data. To circumvent this problem, we decided to calculate the 90-percentile to be used as a plausible upper boundary for the lateral noise in both directions (as seen in Table 2). This value is constant for all distances and rotation angles, as no trend for any variable is visible in the data. It is also notable that the noise in y-direction is lower for both sensors.

	KinectV2	Phab2Pro
σ_{x_1}	2.9110	4.1207
σ_{y_1}	1.9617	3.6665

Table 2: Lateral Models (90-percentile)

The model of this work (Model 1 - σ_{x_1} σ_{y_1} σ_{z_1}) is combined with the model from Köppel et al. [7] (Model 2 - σ_{x_2} σ_{y_2} σ_{z_2} , see [7] for the models equation) by multiplying them with a simple weight function w and adding them up. Although we planned to only average between both models, our evaluation showed that such a model would lead to worse results than Model 1. The reason for this seems to be that the additional noise from a rotated surface outweighs the image position dependent noise component. Because of this, our manually determined weight functions (as seen in Equation 2) assign a higher weight to Model 1 at higher angles.

$$w = \max\left(\frac{w_a - \theta}{w_a} * w_b + w_c, w_c\right) \quad (2)$$

$$\sigma_x = (1 - w) * \sigma_{x_1} + w * \sigma_{x_2} \quad (3)$$

$$\sigma_y = (1 - w) * \sigma_{y_1} + w * \sigma_{y_2} \quad (4)$$

$$\sigma_z = (1 - w) * \sigma_{z_1} + w * \sigma_{z_2} \quad (5)$$

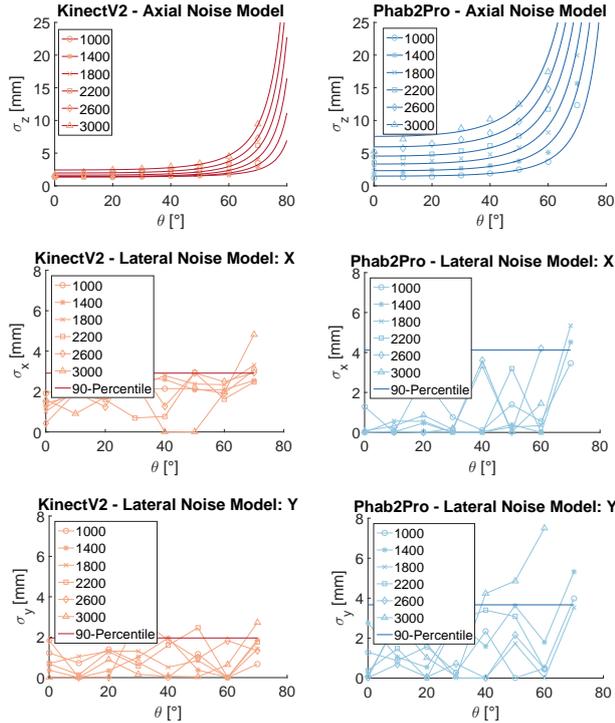


Figure 7: Line Plot of the fitted axial model for both sensors with the measurement data represented by the points. For the measured lateral noise the 90-percentile is used as an conservative estimate.

	KinectV2	Phab2Pro
w_a	29	8
w_b	0.3	0.2
w_c	0	0.3

Table 3: Weight Function Coefficients (Equation 2)

6 Evaluation

As a final step in our work, we evaluated the quality of our produced models and compared our results to the ones from previous works. The quality of our models was estimated in two ways: statistically, by comparing key figures of the fitted models and experimentally, by measuring a simple object in a real-world situation.

6.1 Statistical Evaluation

The statistical evaluation was applied to Model 1. We estimated both the root-mean-square error (RMSE), representing the standard deviation between the measured and the predicted value, and the R-square value, describing how well the amount of variance in the data is described by our model. For the RMSE, values close to 0 are desirable while for the R-square, values close to 1 are optimal. Our models have RMSE values significantly below 1 mm and R-square values close to 1. This means our models are able to estimate the given data rather precisely. For

KinectV2 & Phab2Pro - Weight Function

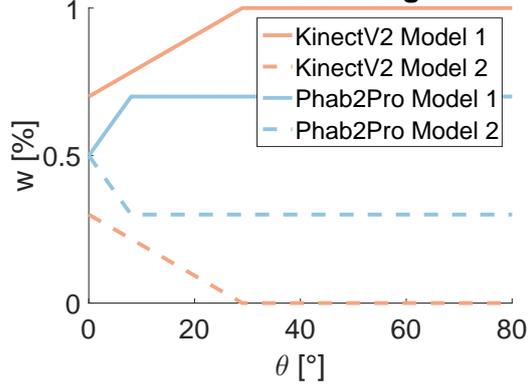


Figure 8: Line Plot of the used weight functions for both sensors.

comparison, the RMSE of Fankhauser et al. [3] was stated as 0.002 mm, which is approximately 30 times lower than ours. The reasons for this are not easily identifiable, but might be a result of different test conditions. By comparing the RMSE of the KinectV2 and the Phab2Pro, one notices that the latter one is nearly four times as high, this is probably due to the lower signal-to-noise ratio of the Phab2Pro.

	KinectV2	Phab2Pro
RMSE [mm]	0.0633	0.2328
R-square	0.9982	0.9992

Table 4: Axial Model Fit Key Figures

6.2 Experimental Evaluation

The second method of evaluation used a simple setup in a real-world situation and was used for our combined model. The setup consisted of a cube of the dimensions 300 mm \times 300 mm \times 300 mm that was measured with both sensors. The cube was placed at different positions, distances and rotations (to show more or less faces) and the sensor was hand held. For each depth image the visible cube faces were extracted. Because of the simplicity of the cube's surface and the fixed positions, the model parameters like rotation and distance could be automatically determined. The measured noise was then compared to the estimations made by our combined model.

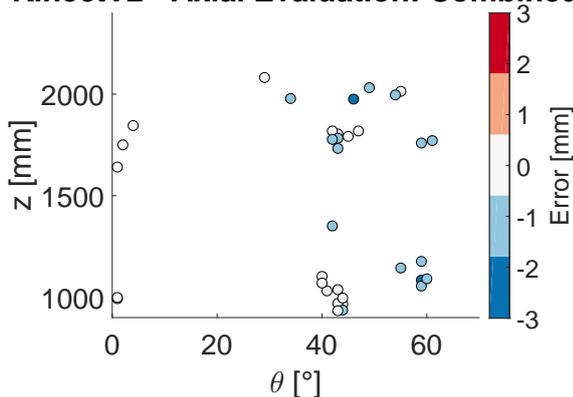
For the axial model we calculated the RMSE of the difference between the measured values and the predicted ones. As for the lateral noise components, we chose to verify, if our model could serve as an upper border by checking, if the noise values are under the predicted values in 90% of the cases. This way of verifying our lateral model was chosen, because the seemingly random

structure of the noise was not predictable with any of our models and was only defined as the 90-percentile of the measurements.

Our evaluation indicated that Model 1 is better at higher angles and Model 2 is better at lower angles. By applying a weight function to both models, we could adapt to this and generated a weighted combined model that is better than its initial components.

The lateral results, seen in Figure 10, show that both directions of our combined lateral models are most of the time above the measured values, thus they can be seen as a good upper border for the lateral noise. More concretely, the percentage of values that are correctly under our border is around 90% in all instances. In contrast to the axial model, the lateral models do not seem to need an additional weight function and could be simply averaged.

KinectV2 - Axial Evaluation: Combined



Phab2Pro - Axial Evaluation: Combined

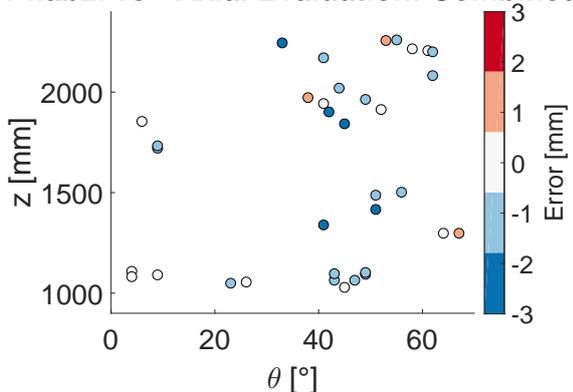


Figure 9: Scatter plots of our experimental evaluation, each point represents one plane in the image with its color showing the error between its actual and its predicted noise

	KinectV2	Phab2Pro
Model 1	0.8947	2.5146
Model 2	1.5197	4.1739
Combined	0.8926	1.7856

Table 5: Axial evaluation results (RMSE [mm]) for both sensors

	KinectV2	Phab2Pro
X - Combined	94.4	93.4
Y - Combined	94.4	94.8

Table 6: Percent of all values below our estimations of the lateral noise calculated by our final lateral models

7 Conclusion

In this paper we presented two different models for estimating the sensor noise in the axial and both lateral directions. Similar models were already calculated for the KinectV2, but as of now we are the first to do this kind of experiment with the Phab2Pro.

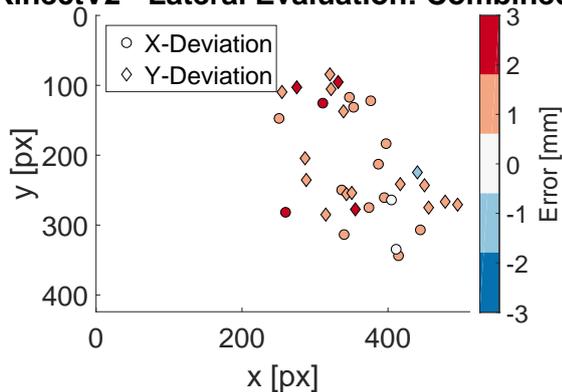
We used a test setup consisting of a rotating planar target that was measured at different distances and rotations to estimate the noise. At each position two measurement series were made, one with the camera in a normal (horizontal) position and one with the camera (vertically) rotated 90° to calculate eventual differences between both lateral directions. Furthermore, we showed an automated pipeline to extract the sensor noise.

From these noise measures we derived an empirical model that used the object’s distance and rotation as parameters. To further improve this model we combined it with the model of Köppel et al. [7], whose model used the image coordinates and the distance of the object as parameters. For the merging of the two separate models, we showed the need for a weight function that decides when to use which model. We used a simple linear function based on the object’s rotation.

Finally, our empirical models were validated, using a simple setup, where a cube is placed in a scene with different distances, positions and rotations. For each face, the local noise is roughly estimated and compared to the predictions of our model. The results of the evaluation indicate that the combined axial model using a weight function is superior to its component models. A similar conclusion can be drawn for both lateral models.

Future work could include attempts to improve the combined model like using a more refined weight function joining the component models. The lateral noise component could be further analyzed to come up with a more concrete model. Furthermore, our generated noise models

KinectV2 - Lateral Evaluation: Combined



Phab2Pro - Lateral Evaluation: Combined

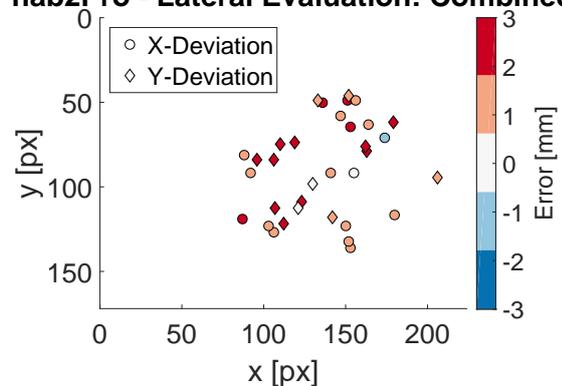


Figure 10: Scatter plots of our experimental evaluation, each point represents one edge in the image with its color showing the error between its actual and its predicted noise

could be used to enhance KinectV2 and Phab2Pro depth data, possibly leading to improved reconstruction results.

References

- [1] Timo Breuer, Christoph Bodensteiner, and Michael Arens. Low-cost commodity depth sensor comparison and accuracy analysis. In *SPIE Security+ Defence*, pages 92500G–92500G. International Society for Optics and Photonics, 2014.
- [2] Riyad A El-laithy, Jidong Huang, and Michael Yeh. Study on the use of microsoft kinect for robotics applications. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 1280–1288. IEEE, 2012.
- [3] Péter Fankhauser, Michael Bloesch, Diego Rodriguez, Ralf Kaestner, Marco Hutter, and Roland Siegwart. Kinect v2 for mobile robot navigation: evaluation and modeling. In *Advanced Robotics (ICAR), 2015 International Conference on*, pages 388–394. IEEE, 2015.
- [4] Luigi Gallo, Alessio Pierluigi Placitelli, and Mario Ciampi. Controller-free exploration of medical image data: Experiencing the kinect. In *Computer-based medical systems (CBMS), 2011 24th international symposium on*, pages 1–6. IEEE, 2011.
- [5] Inc Google. Calibrating your tango device. <https://developers.google.com/tango/hardware/calibration>. Accessed: 2017-04-17.
- [6] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.
- [7] Thomas Köppel. Extracting Noise Models considering X Y and Z Noise. Bachelor’s thesis, TU Wien, Austria, 2017.
- [8] Chuong V Nguyen, Shahram Izadi, and David Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 524–530. IEEE, 2012.
- [9] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 759–760. ACM, 2011.
- [10] Hamed Sarbolandi, Damien Lefloch, and Andreas Kolb. Kinect range sensing: Structured-light versus time-of-flight kinect. *CoRR*, abs/1505.05459, 2015.
- [11] Inc The MathWorks. kstest. <https://de.mathworks.com/help/stats/kstest.html>, 2017. Accessed: 2017-08-07.
- [12] Thiemo Wiedemeyer. Tools for using the kinect one (kinect v2) in ros. https://github.com/code-iai/iai_kinect2, 2017. Accessed: 2017-04-17.
- [13] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 15–22. IEEE, 2011.
- [14] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.