Probabilistic Sensor Model Based Weighting for TSDF based Surface Reconstruction Algorithms

Daniel Johannes Hettegger

Abstract Modern surface reconstruction algorithms often rely on truncated signed distance fields (TSDF) which use a weighted update step to incorporate multiple depth scans into the internal representation. This work aims to improve the aforementioned weighting step in the state of the art, by constructing a sensor noise model for the Kinect v2 based on the pixel locations and depth values of the depth scans and incorporating it in the TSDF updating step. This method produces surface reconstructions with less noise and increased robustness to noisy depth scans compared to its fixed weight predecessor.

Keywords RGB-D · Surface Reconstruction · TSDF · Kinect v2 · Noise Model · Kinect Fusion

1 Introduction

The introduction of consumer grade depth sensors, with one of the first being the Microsoft Kinect in 2010 shortly followed by Intel's RealSense technology and other comparable products, at relatively inexpensive price points has made the technology widely available to a broad range of users. The technology these sensors are based on is often either structured light (SL) or time of flight (ToF) which have existed for several decades [6]. The wide availability if these sensors to consumers and researchers has enabled the development of algorithms that use RGB-D images to reconstruct and represent surface structures of real environments in digital applications.

This enables a whole new field of use cases that incorporate real time surface reconstruction ranging from fields like 3D scanning and modeling used for example by artists, architects or city planners, to consumer oriented recreational software such as games that incorporate movements of a player or the environment as was the original use case of the Microsoft Kinect, to Robotics use cases for example simultaneous localization and mapping (SLAM) to map and navigate previously unknown terrain and many future technologies like augmented and virtual reality which can use these algorithms to enable interactions with real world environments and scenes.

While many approaches to scene reconstruction exist, they are often based on a similar processing pipeline [6]. Especially influential works in the field [1], [3] make use of a truncated signed distance field (TSDF) to represent surfaces as zero crossings. These approaches use a simple weighting formula to incorporate multiple scans into one coherent function. This work aims to further the current state of the art by extending the updating process for TSDF based algorithms by incorporating a sensor noise model and therefore increasing the reconstruction accuracy.

This paper is structured in the following sections: Section 2 discusses related works that have laid the groundwork for modern surface reconstruction. Section 3 explains the proposed approach which tries to extend on previous works. Section 4 shows and discusses experimental results this approach and compares it to the state of the art. Finally section 5 summarizes the outcome of this work as well as provide an outlook on possible future work related to this approach.

2 Related Work

The recent survey by Zöllhofer et al. on surface reconstruction from RGB-D data [6] gives an excellent

Daniel Johannes Hettegger
Technical University of Munich
E-mail: daniel.hettegger@tum.de

overview on the development of the technology and its current state of the art.

Static scene surface reconstruction has been researched quite extensively since the late 20th century with one especially influential work by Curless and Levoy in 1996 called "A Volumetric Method for Building Complex Models from Range Images" [1], which has laid the ground work for many modern approaches by introducing a truncated signed distance field (TSDF) to represent surfaces based on zero crossings in voxel grids.

A paper released in 2011 called "Kinect Fusion: Real-Time Dense Surface Mapping and Tracking" by Newcombe and co-authors [3] has incorporated many of the previous ideas and optimized it for modern hardware such as general purpose graphical processing units (GPGPUs) and made the algorithm capable of updating its internal representation in real time.

Maier-Hein, Lena et. al. showed in their work "Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error" [2] that a sensor noise model based approach dramatically increased accuracy. The ICP algorithm is often used in surface reconstruction for the localization of the camera frame.

In 2015 Sarbolandi et al. investigated the effects of different noise sources while scanning with the two Kinect versions in their published work: "Kinect range sensing: Structured-light versus Time-of-Flight Kinect" [4]. The findings in this paper are used in our approach to create a per pixel noise model for RGB-D scans using Kinect v2, which can be used to predict the reliability of certain pixels in the scan and weight them accordingly when incorporating it into the model.

3 Methodology

The weighting step is a crucial part of the TSDF based surface reconstruction algorithm. Many implementations use the standard uniform weighting for every depth scan. This work extends on the weighting step already established in the state of the art, the details can be found in the work of Curless and Levoy [1].

To improve the weighting of multiple depth scans, a sensor noise model for the depth camera is created. In this work the Kinect v2 will be used, which utilizes ToF technology.
In [4] Sarbolandi and his co-authors provide a in depth comparison of the sensor noise of the two Microsoft Kinect depth sensors. Based on the data provided in that work, the construction of a noise model was possible. To estimate the quality of a given depth scan pixel at a certain depth, one can model the standard deviation σ at every point in the image. Figure 1 shows the standard deviation of 3 different pixel regions at a measured depth [4]. One can observe that the center regions of the image are subject to less noise in comparison to the corner pixels. Furthermore one can see from the plot that the standard deviation rises almost linearly with the depth in logarithmic scale, which can be approximated with an exponential function.

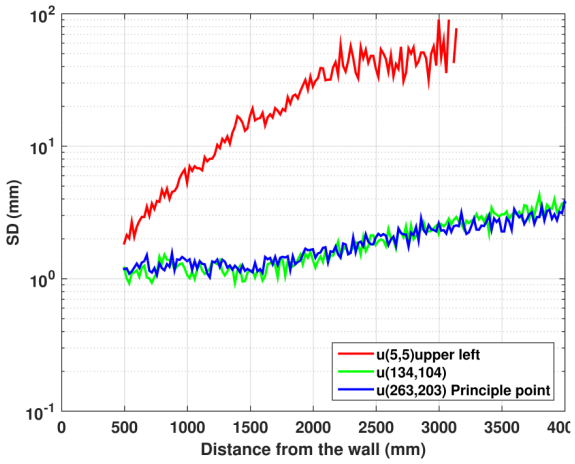

Fig. 1 Standard deviation of scans with the Kinect 2 of a flat surface at 3 pixel regions dependent on depth. Source: Adapted from [4]

To approximate the standard deviation of the Kinect v2 sensor at any given pixel location, the first step is approximating the standard deviation at the known pixel locations, as described by the curves seen in figure 1. The first function $\sigma_{\text{center}}(d)$ approximates the standard deviation σ for pixels in the center region with depth d, while the second function $\sigma_{\text{corner}}(d)$ approximates the standard deviation of pixels in the corner region. Both functions are based on a simple exponential function of the form:

$$\sigma(d) = p \cdot e^{\lambda \cdot d} \tag{1}$$

The parameters p and λ can be determined by simply inserting two points that lie on the desired function and solving the resulting set of equations. Table 1 contains said parameter values for the center and corner regions.

After modeling the standard deviation for these two regions, the next step is to create a model for the whole range of the depth image. To achieve this, a few assumptions are made which will be explained in the following

Function	p	λ
$\sigma_{\text{center}}(d)$	1.0316	0.000305
$\sigma_{\text{corner}}(d)$	0.8111	0.001805

Table 1 Parameter values for estimation of standard deviation of corner and center regions.

paragraph.
Firstly, it is assumed that the noise model is symmetric around the center of the image. This implies that the function that approximates the standard deviation at a certain pixel location in the depth image is only dependent on the distance from the center D_c, rather than the full x and y coordinates.
Secondly, when consulting figure 1, one can observe that the standard deviation increases with the depth at the same rate for the center pixel (263,203) and intermediate pixel (134,104), while the corner pixel shows a greatly increased standard deviation. Therefore it is assumed that the standard deviation in the center of the image as well as the region in the shape of a circle with a radius D_t of 170 pixels around the principal point is described by the function $\sigma_{\text{center}}(d)$.

To approximate the standard deviation in the area between the center and corner regions, a linear interpolation of the parameters p and λ is performed, dependent on the distance from the circular region in the center described above.

With the assumptions mentioned above the final formula for the estimation of the standard deviation can be constructed.

$$\sigma(d, D_c) = p(D_c) \cdot e^{\lambda(D_c) \cdot d} \tag{2}$$
$$p(D_c) = o_p + k_p \cdot D_c \tag{3}$$
$$\lambda(D_c) = o_\lambda + k_\lambda \cdot D_c \tag{4}$$

Where $o_p, o_\lambda, k_p, k_\lambda$ are the parameters of the linear interpolation and D_c is the pixel distance from the inner circle, computed from the pixel distance $D(x, y)$ of the current pixel to the principal point of the depth scan and the radius of the circle D_t centered on the principal point in which the standard deviation is considered to be homogeneous. D_c is computed as follows:

$$D(x, y) = \left|\left| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{center} \\ y_{center} \end{bmatrix} \right|\right|_2 \tag{5}$$
$$D_c = \max(0, D - D_t) \tag{6}$$

Table 2 shows the parameter values for the approximation of the standard deviation σ.

Parameter	value
o_p	1.316
k_p	−0.00315
o_λ	0.000305
k_λ	$9.285\ 10^{-6}$
x_{center}	263
y_{center}	203
D_t	170

Table 2 Parameter values for the computation of $\sigma(d, D_c)$.

Using the approximation of the standard deviation in equation 2 for every pixel at any depth one can now compute the new weight associated with every pixel in the depth image $w_{x,y}(x, y, d)$.

$$w_{x,y}(d, D_c(x, y)) = \frac{1}{\sigma(d, D_c)} \tag{7}$$

The computed weight is proportional to the probability density function of the maximum likelihood solution of a Gaussian distribution with standard deviation σ.

$$p(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{8}$$
$$p_{ML}(x = \mu|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\mu-\mu}{\sigma}\right)^2} \tag{9}$$
$$p_{ML}(x = \mu|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \propto \frac{1}{\sigma} \tag{10}$$

Figure 2 shows the resulting weighting value with respect to the depth d and pixel distance from the principal point D. Figure 3 shows the weighting function applied to a depth scan taken with the Kinect v2 from the CoRBS Dataset [5]. One can observe that the regions that are further away and in the periphery of the image are attributed lower weights, as these areas are prone to higher signal noise.

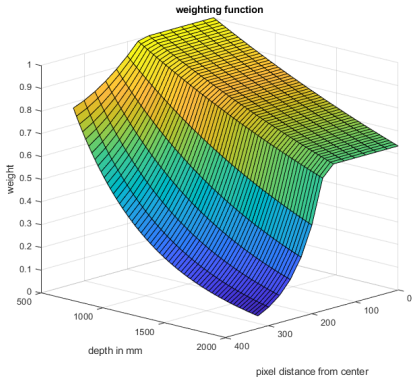

Fig. 2 Weighting function visualized with respect to depth and distance from principal pixel.

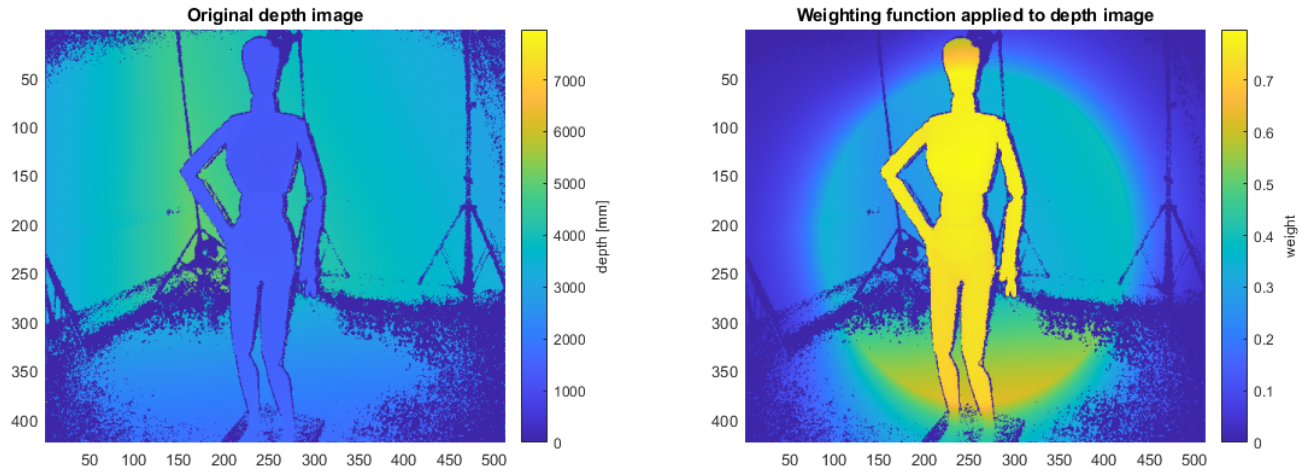

Fig. 3 Weighting function applied to a depth scan taken from the CoRBS Dataset [5].

Fig. 4 Ground truth scan of the puppet model from the CoRBS Dataset [5].

Fig. 5 Ground truth scan of the electrical cabinet model [5].

Fig. 6 Ground truth scan of the desk model [5].

4 Experiments

To compare the algorithm to a standard baseline three models from the CoRBS data set [5] were used. The data set contains depth and color frames recorded from a Kinect V2 Depth sensor as well as a ground truth trajectory of the movements of the camera at any given time. The models that were used are a human sized puppet seen in figure 4 an electrical cabinet seen in figure 5 and a small desk with a monitor and other various items seen in figure 6. As the reconstruction depends on many different factors the actual reconstruction has been performed with the ground truth trajectory of the camera to alleviate any effects that the localization step could have on the reconstruction.

To compare the uniformly weighted reconstructions to the sensor noise model based surface reconstruction a qualitative visual examination is used, as there are significant visual differences in reconstruction quality.

The following subsections compare the results of a reconstruction with the proposed weighting algorithm versus the resulting reconstruction of a uniformly weighted TSDF fusion update step by using the same input frames and trajectories.

4.1 Reconstruction of a Human Puppet

The data of the scans containing the human puppet are taken from the CoRBS data set [5] and can be found on their website with the H1 identifier.
Figure 7 shows the comparison of the reconstruction of the human puppet without (left hand side) and with (right hand side) the proposed weighting scheme. Overall it can be observed that the reconstruction with the proposed weighting scheme produces smoother surfaces and is less prone to noise in the scans as well as frames that contain the model in the periphery of the image where the scan exhibits significantly increased noise.
In the upper left part of the image a portrait of the puppet can be seen. It can be observed that the head of the puppet is missing a big part of the forehead in the original scan whereas the scan with the newly proposed method contains the whole head. This can most probably be attributed to the fact that the scan of the model contained frames where the head of the puppet is contained in the periphery of depth images where greatly increased noise destroyed previously correctly reconstructed surfaces. Similar effects can be seen on the neck and upper torso of the reconstruction.
When observing the lower left image, the hip region of the puppet can be seen. Comparing the uniformly weighted reconstruction to the reconstruction with the proposed weighting scheme, it is clearly visible that the resulting surfaces are smoother and contain less noisy edges. Above the hand of the puppet a noisy surface on top of the actual back hand is visible in the uniformly weighted reconstruction which is not part of the updated reconstruction. On the hip and lower stomach of the puppet a darker discoloration can be observed, this is due to the fact that non Lambertian surfaces as well as lighting, camera gain or camera white balance changes are not modeled in the color reconstruction of the surfaces.
In the middle of the image the whole puppet reconstruction of the puppet can be observed for comparison.

4.2 Reconstruction of an Electrical Cabinet

The data of the scans containing the electrical cabinet are taken from the CoRBS data set [5] and can be found on their website with the E1 identifier.

Figure 8 shows the comparison of the reconstruction of the electrical cabinet without (left hand side) and with (right hand side) the proposed weighting scheme. The top two images show the whole electrical cabinet from a side angle. Overall it can be seen that TSDF based reconstruction has difficulty reconstructing very thin objects that are observed from both sides, this is to be expected as the surfaces are represented in the zero crossing of the TSDF, and as the thin object is observed from both sides, the field will contain positive distance values on each side of the object and a zero crossing in the TSDF is often not present, which means that the reconstruction algorithm will not produce a surface at that location.
In the lower two images a comparison of a corner of the electrical cabinet is visible. The reconstruction without the newly proposed weighting scheme is missing the left upper part of the frame. In the reconstruction with the proposed weighting scheme this part of the frame is correctly reconstructed, this can again be attributed to depth frames at larger distances as well as peripheral views of the object, which destroy previously correctly reconstructed surfaces in the object.

4.3 Reconstruction of a Desk Scene

The data of the scans containing the desk scene are taken from the CoRBS data set [5] and can be found on their website with the D1 identifier.

Figure 8 shows the comparison of the reconstruction of the desk scene without (left hand side) and with (right hand side) the proposed weighting scheme. In the top left images of the comparison the whole desk scene can be seen. In general it can again be observed that the

Fig. 7 Comparison of the reconstruction of the human puppet without (left hand side) and with (right hand side) the proposed weighting scheme.

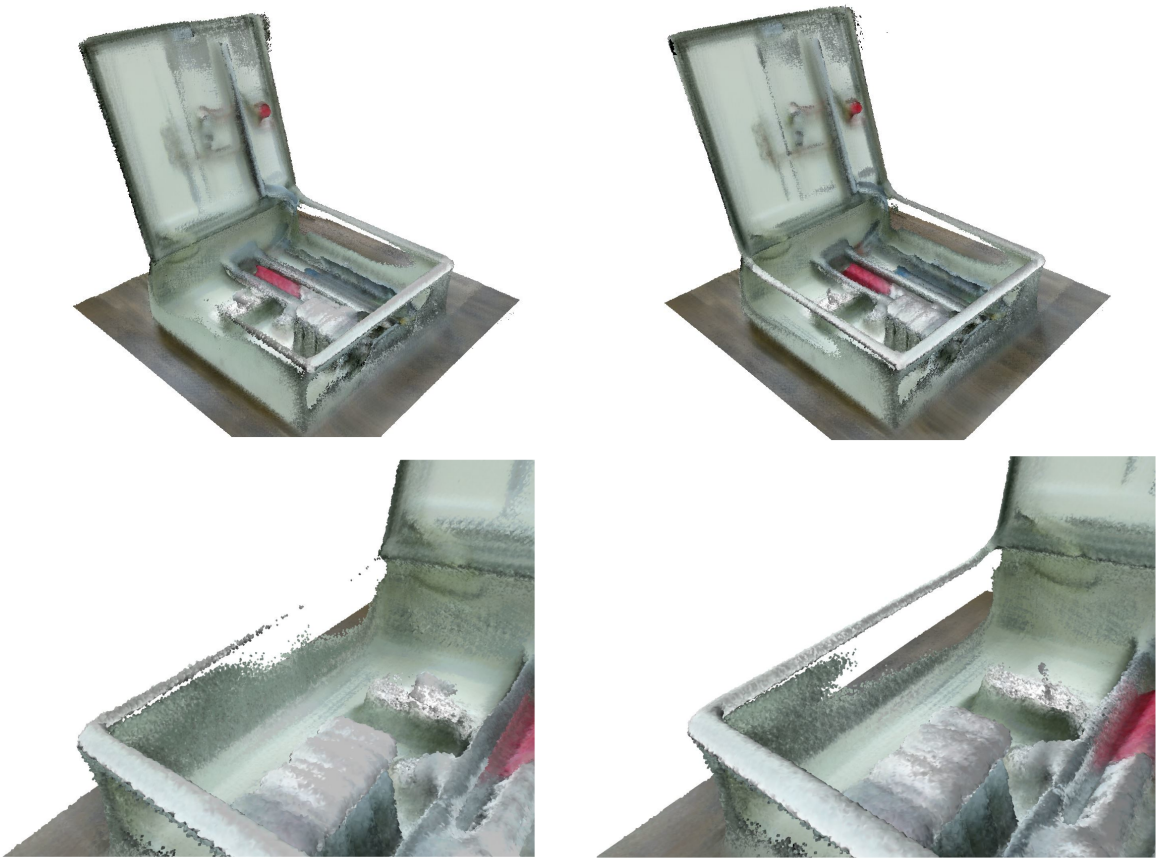

Fig. 8 Comparison of the reconstruction of the electrical cabinet without (left hand side) and with (right hand side) the proposed weighting scheme.

Fig. 9 Comparison of the reconstruction of the desk scene without (left hand side) and with (right hand side) the proposed weighting scheme.

reconstruction with the proposed weighting scheme results in a more complete reconstruction with less noisy surfaces.

In the top right images of the comparison a desk edge is visible. The uniformly weighted reconstruction shows increased noise on the surfaces as well as a more incomplete reconstruction compared to the new weighting scheme. This comparison shows the reconstruction quality improvement of straight edges with the new weighting scheme versus the uniformly weighted reconstruction.

In the lower images a close up examination of the computer monitor is shown. This object is especially interesting as it contains a highly reflective surface which can cause difficulties for the RGB-D camera [4]. This effect results in depth images that are incomplete at surfaces that can not be reconstructed by the sensor. Due to this fact less frames contain actual depth scans for the reflective surface or even contain incorrect depth scans that result from reflections. Comparing the images of the uniformly weighted reconstruction to the reconstruction with the proposed weighting scheme, it can be seen that the uniformly weighted version is missing the middle part of the computer monitor. This can be attributed to the fact that the frames that contained the actual surface of the monitor are overwritten in the TSDF by incorrect scans. An interesting effect is also that an incorrect reconstruction behind the actual monitor is clearly visible in both reconstructions, this can be traced to incorrect depth scans from the sensor, related to the reflective surface of the monitor. As these incorrect depth values are traced through the monitor, the uniformly weighted reconstruction is deleted from the TSDF, while the reconstruction with the newly proposed weighting scheme is more robust to the incorrect depth scans as the surface of the monitor contains already relatively highly weighted surfaces at the correct position. The incorrect depth scan is therefore still visible in the reconstruction as the TSDF behind the monitor is still empty and even values with low weights will be integrated. As an additional remark it has to be said that the scan trajectory does not observe back side of the monitor. By additionally scanning the back side this incorrectly reconstructed surface would be removed, but the front of the monitor would still be missing in the uniformly weighted reconstruction.

5 Conclusions

Probabilistic sensor model based weighting for TSDF based surface reconstruction has resulted in an increase in reconstruction quality. Reconstructed surfaces are smoother and contain less noise. Straight edges are reconstructed in a flatter manner.

The reconstructed surfaces appear to be more robust to

low quality input, as for example when the sensor observes the surface in the image periphery or at greater distances where the depth measurements have increased noise compared to the center regions at closer distances. This result was expected as the approach integrates information about sensor noise into the weighing of the TSDF update and can therefore more accurately integrate depth scans into the internal representation according to the predicted noise of the sensor.
The obvious drawback of such an approach is that accurate knowledge about the sensor has to be available a priory and it can not be applied to arbitrary depth scans for surface reconstruction. However when integrating such a weighting model into a surface reconstruction algorithm, one could easily incorporate a fallback uniform or surface normal based weighting scheme if no information about the sensor noise is available.
While this work shows only the construction of a sensor noise model for the time of flight based Kinect v2 sensor, it is to be expected that this approach will result in a quality and robustness increase for other sensors, including structured light based sensors with respective noise models.
The extension to the surface reconstruction algorithm produced a minor impact on the time needed for the integration of the frame, but the algorithm still proved to be real time capable with appropriate GPU hardware.
The algorithm has been implemented in C++ and in CUDA for modern computing hardware. For use in a robotics setting a ROS 2 node has been implemented which can receive the depth and color images as well as camera poses to perform the surface reconstruction. The node model has been created in Papyrus for Robotics [1], which uses the RobMoSys [2] approach, and then converted to ROS 2 C++ code which as then been extended with the surface reconstruction algorithm. Figure 10 shows the papyrus node that has been created.

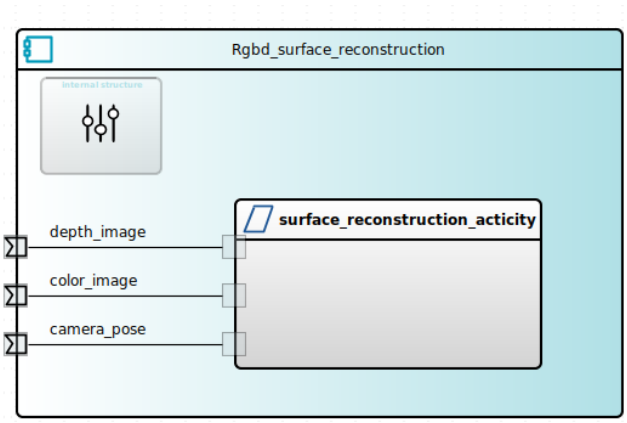

Fig. 10 Papyrus for Robotics node visualization.

5.1 Outlook and Future Work

Future work could focus on the extension of this approach to different sensor hardware, such as a structured light depth sensor or other time of flight depth sensors with their respective noise characteristics.
To improve the accuracy of the sensor noise model incorporating additional information about the scan and sensor, such as temperature scan intensity and surface normal estimation, might increase reconstruction quality even further, future work could focus on finding suitable metrics for this approach and apply them in a comparison to the proposed model.

5.2 Limitations

As this approach heavily relies on the sensor noise model for the deployed depth sensor, it is important that any assumptions made about the sensor noise hold in practice, otherwise the use of such a sensor model might decrease the performance of the surface reconstruction algorithm.

[1] https://www.eclipse.org/papyrus/components/robotics/
[2] https://robmosys.eu

References

1. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 303–312 (1996)
2. Maier-Hein, L., Franz, A.M., Dos Santos, T.R., Schmidt, M., Fangerau, M., Meinzer, H.P., Fitzpatrick, J.M.: Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error. IEEE transactions on pattern analysis and machine intelligence **34**(8), 1520–1532 (2011)
3. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp. 127–136. IEEE (2011)
4. Sarbolandi, H., Lefloch, D., Kolb, A.: Kinect range sensing: Structured-light versus time-of-flight kinect. Computer vision and image understanding **139**, 1–20 (2015)
5. Wasenmüller, O., Meyer, M., Stricker, D.: Corbs: Comprehensive rgb-d benchmark for slam using kinect v2. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–7. IEEE (2016)
6. Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R., Kolb, A.: State of the art on 3d reconstruction with rgb-d cameras. In: Computer graphics forum, vol. 37, pp. 625–652. Wiley Online Library (2018)