Modeling Kinect Sensor Noise for Improved 3D Reconstruction and Tracking

Chuong V. Nguyen
CSIRO
Canberra, Australia
chuong.nguyen@csiro.au

Shahram Izadi
Microsoft Research
Cambridge, United Kingdom
shahrami@microsoft.com

David Lovell
CSIRO
Canberra, Australia
david.lovell@csiro.au

Abstract

We contribute an empirically derived noise model for the Kinect sensor. We systematically measure both lateral *and* axial *noise distributions, as a function of both distance and angle of the Kinect to an observed surface. The derived noise model can be used to filter Kinect depth maps for a variety of applications. Our second contribution applies our derived noise model to the KinectFusion system to extend filtering, volumetric fusion, and pose estimation within the pipeline. Qualitative results show our method allows reconstruction of finer details and the ability to reconstruct smaller objects and thinner surfaces. Quantitative results also show our method improves pose estimation accuracy.*

1. Introduction

Researchers in many areas including computer vision, augmented reality (AR), robotics and human computer interaction (HCI) have begun to embrace the Kinect as a new commodity depth sensor. Whilst compelling due to its low-cost, the output depth maps of the Kinect sensor are noisy. Researchers have begun to explore the sensor noise characteristics of the Kinect [5, 9] and shown an increased variance in reported depth as the distance between the sensor and observed surface increases.

However, this prior work has only quantified *axial* noise as a function of distance to the observed surface. In this paper we contribute a new, empirically derived noise model for the Kinect sensor. We quantify how Kinect noise has both an axial and *lateral* component, which can vary based on distance and *angle* to the observed surface. As shown in Figure 1, we systematically measure both lateral and axial noise distributions and fit a suitable parametric model to each. These noise models can be used to filter Kinect depth maps for a variety of applications.

As an application of our derived noise model we extend KinectFusion [4, 7], a system that creates real-time 3D reconstructions using a moving Kinect sensor. We show improved reconstructions can be achieved by explicitly modelling the Kinect sensor noise and applying this to various stages of the KinectFusion pipeline. Qualitative results show finer details in the acquired reconstructions and the ability to reconstruct smaller objects, whilst quantitative results show improvements in pose estimation accuracy.

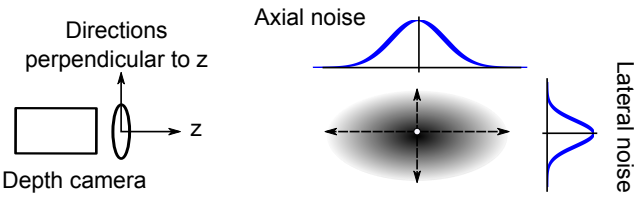


Figure 1. We propose a 3D noise distribution of Kinect depth measurement in terms of *axial* (z-direction) and *lateral* (directions perpendicular to z) noise.

2. Experimental analysis of sensor noise

Kinect sensor noise is the difference between measured depth and ground truth. Previously, axial noise has been estimated by comparing the depth measurement of a planar target (a door) against the mean depth of the measured surface perpendicular to the sensor's z-axis [5]. This showed increased variance of depth measurements with increasing distance from sensor to surface, but did not explore surface angle or lateral noise.

Lateral noise is not so readily measurable: ideally this would be the point spread function (PSF) recovered from the Kinect sensor. However, given the Kinect only produces noisy depth measurements, recovering the full PSF is challenging. We propose that the PSF can be approximated from the derivative of the sensor's step response function, which can be observed at depth discontinuities. When observing the edges of a planar target sensed with the Kinect, we can see that straight-edges appear as zig-zag lines, and the distribution of the pixels along these edges gives an indication of the step response function. We therefore approximate the PSF as the distribution of absolute distances from the observed edge pixels to a fitted straight edge of a planar target. From our experiments, we have found this simple to

978-0-7695-4873-9/12 $26.00 © 2012 IEEE
DOI 10.1109/3DIMPVT.2012.84

Authorized licensed use limited to: CSIRO Information Technology Services. Downloaded on August 16,2024 at 05:20:52 UTC from IEEE Xplore. Restrictions apply.

implement and more accurate than direct estimation of the step response. The standard deviation (STD) of lateral noise is simply the STD of the edge distance distribution.

2.1. Experimental setup

Figure 2 summarizes our setup for measuring lateral and axial noise probability distribution functions (PDFs) for a Kinect sensor. We used a planar target rotating freely around a vertical axis in front of a fixed Kinect (Figure 2a). All experiments in this paper used a Kinect for Windows sensor set to run in "near-mode" with depth sensing between 0.4m to 3.0m.

The principal axis of the depth camera is defined as the z-axis and the target rotation axis is approximately perpendicular. Lateral noise is extracted from the pixels along the vertical edges of the depth map (Figure 2b, front view). Axial noise is obtained from the differences between raw depth and a fitted plane (Figure 2c, top view).

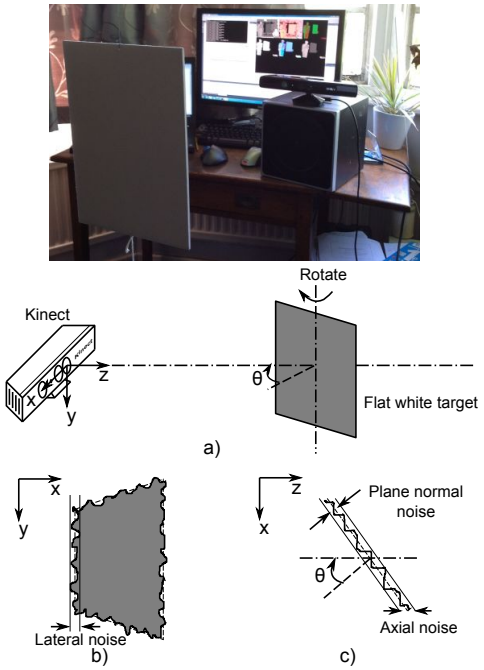


Figure 2. Experimental setup to measure noise distributions of Kinect sensor. a) Photo and schematic representation of experimental setup, with Kinect capturing depth maps of a rotating target. b) The 2D projected depth map of the planar target with lateral noise along one edge highlighted. c) A top-down cross-section of the depth map revealing noise distribution along z-axis and normal to the target plane.

2.2. How to extract axial and lateral noise

Figure 3 shows how to extract lateral noise (top row) and axial noise (bottom row) from a depth map as function of plane angle θ. Figure 3a-c show how to isolate the vertical edges of the observed target and obtain the edge-pixel distance distribution to calculate the lateral noise STD σ_L. Detected edge pixels are shown as thick lines.

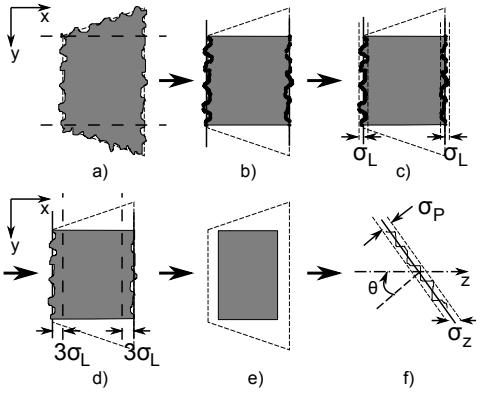


Figure 3. Processing of depth map to extract noise components. a) Crop horizontal edges b) extract edge pixels and fit straight lines to vertical edges. c) Extract distance from edge pixels to the fitted lines to calculate σ_L. d) Crop vertical edges by $3\sigma_L$. e) Fit plane to remaining rectangle depth map. f) extract the rotation angle θ and σ_z.

For axial noise measurement, the vertical edges of the depth map are trimmed $3\sigma_L$ to remove all lateral noise as shown in Figure 3d. The remaining depth map region is fitted to a plane to estimate the plane angle and depth noise along the z-axis as shown in Figure 3e-f. Plane fitting can be performed using orthogonal distance regression.

Figure 4 shows examples of the depth maps at different z positions and plane angles. Detected edges and bounding box are highlighted for illustrative purposes to indicate how σ_L and σ_z are computed.

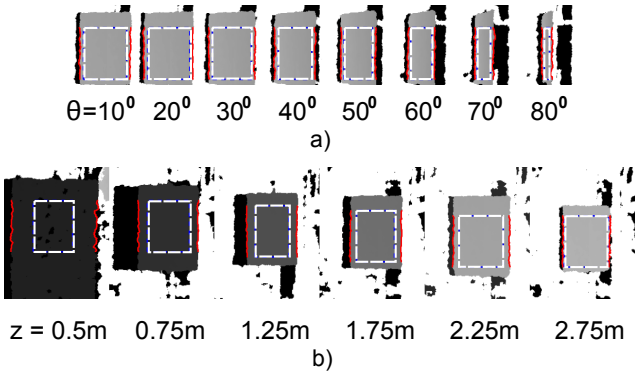


Figure 4. Detected edges (solid red lines) to measure lateral noise and bounding box (white dashed line) to crop the depth map before plane fitting at different angles (a) and different distances (b). Invalid depth measurements shown in black.

2.3. Results and models of noise distributions

Experimental results were obtained using A2–A5-sized targets of same surface material and thickness. Targets

Authorized licensed use limited to: CSIRO Information Technology Services. Downloaded on August 16,2024 at 05:20:52 UTC from IEEE Xplore. Restrictions apply.

were placed at z-distances between 0.5m to 2.75m from the Kinect at 0.25m increments. At each distance 1000 depth maps were captured at varying angles θ. This led to approximately 10 depth maps being captured for each unique z-distance and angle. We used an A5-sized plane at distances of approximately 0.5m and 0.75m; A4 at 1.0m and 1.25m; A3 at 1.5m, 1.75m and 2.0m; A2 at 2.25m, 2.5m and 2.75m. This allowed us to maintain a fixed measurement region around the optical center of the Kinect camera for all target sizes and z-distances (Figure 3). We used factory default intrinsic parameters for lens calibration with a field-of-view of 70°, focal length of 585 pixels and principal point at the center of the image.

Measured noise distributions obtained for three example z-distances are shown in Figure 5. As illustrated, the spread of lateral noise distributions (obtained in pixels) does not vary significantly with z-distance. In contrast, the spread of axial noise distributions (obtained in meters) clearly increases with z-distance.

Figure 6 gives an overview of lateral and axial noise over z-distance. The distribution of noise as function of z-distances between 0.75m-2.5m is plotted for angles θ between 10-60°. As shown, the lateral noise increases linearly with z-distance. Axial noise increases quadratically with z-distance. Note we avoid reporting data at the limits of the operating distance of the near-mode Kinect or extreme surface angles, as a significant increase of invalid depth pixels makes robust plane fitting impractical.

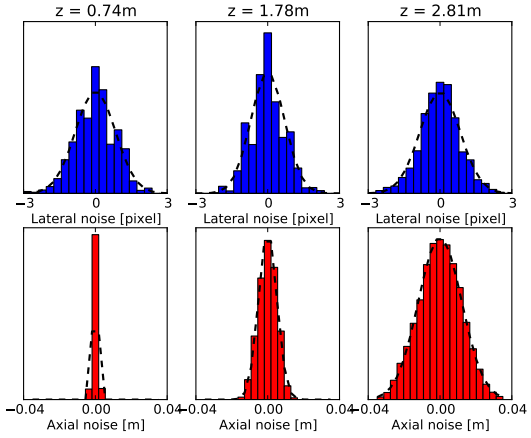


Figure 5. Measured lateral noise (top row) and axial noise (bottom row) measured for three z-distances. Dashed lines show fitted Gaussian distributions. The spread of lateral noise (in pixels) does not vary significantly with increasing z-distance, while the spread of axial noise (in meters) does.

2.3.1 Fitted lateral noise model

Figure 7a shows that lateral noise (in pixels) changes little with z-distance, except close to the limit of the operating range of the Kinect (at 0.5m). Furthermore, lateral noise increases slightly with θ before 70°. When converting from pixel to real-world distances in meters, lateral noise increases linearly with z-depth as shown in Figure 6.

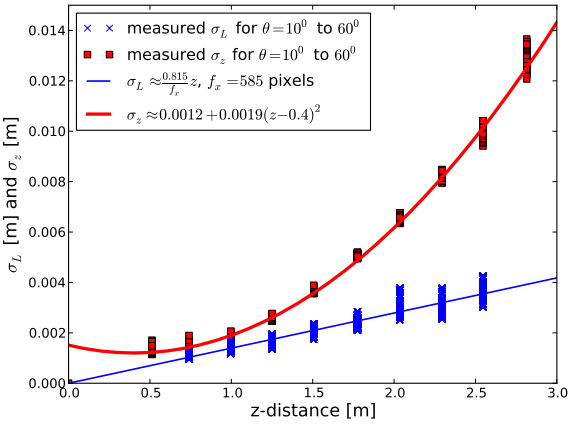


Figure 6. Linear and quadratic fits to the lateral and axial noise components, respectively, as function of z-distances between 0.75m-2.5m plotted for angles θ between 10-60°

We fitted a linear plus hyperbolic model to this data using ordinary least squares regression and excluding spurious data at z-distances less than 0.5m or higher than 2.8m. The hyperbolic term accounts for the rapid growth of the noise towards 90°. Equation 1 gives the standard deviation distribution in pixels, and equation 2 converts this to real world units (meters):

$$\sigma_L(\theta)[px] = 0.8 + 0.035 \cdot \theta/(\pi/2 - \theta) \quad (1)$$

$$\sigma_L(\theta)[m] = \sigma_L(\theta)[px] \cdot z \cdot p_x/f_x \quad (2)$$

where p_x/f_x is the ratio between pixel size and focal length of the Kinect camera (both in either pixel or metric units). σ_L is assumed the same for directions perpendicular to z. The coefficient of 0.035 is found manually. The fitted lateral model is plotted in Figure 7a.

2.3.2 Fitted axial noise model

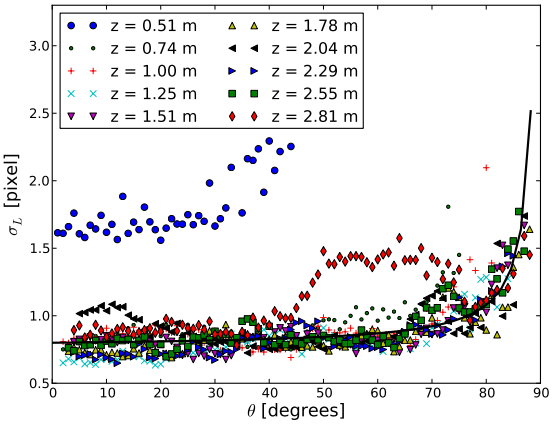
Figure 7b shows that σ_z varies significantly with z-depth but remains constant at angles less than approximately 60° and increases rapidly when the angle approaches 90°. At distances beyond 2.5m σ_z show a noticeable decrease as angle approaches 0°. This is because Kinect's depth resolution decreases at larger distances [5].

We can derive an equation for z-axial noise by firstly assuming it is constant for angles from 10-60° and using linear regression to fit a relationship between z-axial noise and reported depth. Note we again avoid capturing spurious data at extreme angles outside of the range of 10-60°.

$$\sigma_z(z,\theta) = 0.0012+0.0019(z-0.4)^2, 10^\circ \leq \theta \leq 60^\circ \quad (3)$$

Authorized licensed use limited to: CSIRO Information Technology Services. Downloaded on August 16,2024 at 05:20:52 UTC from IEEE Xplore. Restrictions apply.

a)



b)

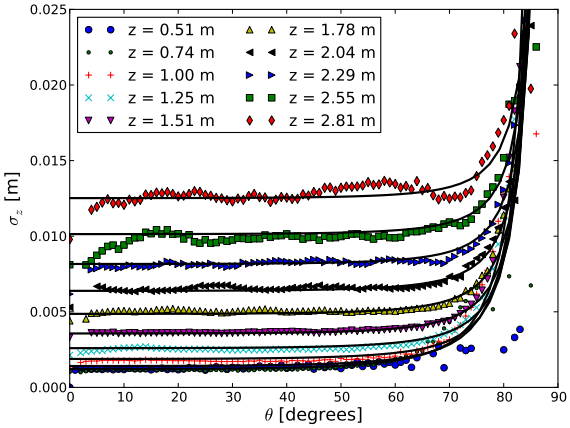


Figure 7. Fitted models plotted over a) lateral and b) z-axial noise.

This equation differs from the axial noise model in [5] as it includes extra terms with coefficients 0.0012 and -0.4 to better fit noise at smaller distances. We then add a hyperbolic term to account for the increase as θ approaches 90°:

$$\sigma_z(z,\theta) = 0.0012 + 0.0019(z-0.4)^2 + \frac{0.0001}{\sqrt{z}}\frac{\theta^2}{\left(\frac{\pi}{2}-\theta\right)^2} \quad (4)$$

The coefficient of $z^{-\frac{1}{2}}$ is found manually. The fitted axial noise model is plotted over the raw data in Figure 7b.

3. Applying our noise model to KinectFusion

To demonstrate the value of our empirically derived Kinect noise model we use it to extend the KinectFusion system. In KinectFusion, depth data from the Kinect camera is integrated into a regular voxel grid structure stored on the GPU. Surface data is encoded *implicitly* into voxels as signed distances, truncated to a predefined region around the surface, with new values integrated using a weighted running average [3]. The global pose of the moving depth camera is predicted using the point-plane variant of the iterative closest point (ICP) algorithm [2]. Drift is mitigated by aligning the current raw depth map with the *accumulated* model (instead of just the previous raw frame). This implicit surface can be extracted either through ray-casting the voxel grid or triangulating a mesh using marching cubes or other variants.

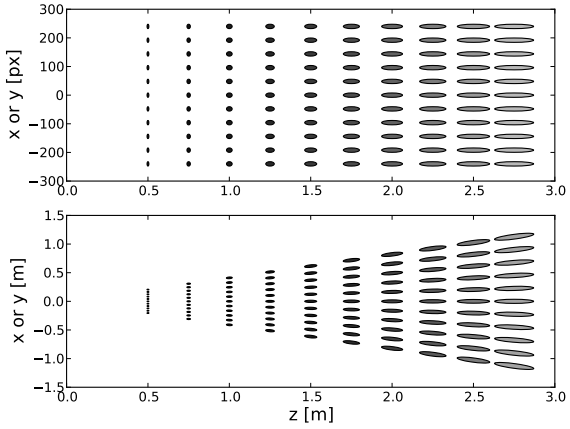


Figure 8. 2D visualization of the PDF contours of Kinect sensor noise distributions in image space (top) and in 3D space (bottom). Each ellipse represents the noise distribution with σ_z and σ_L scaled up by a factor of 20.

In the following subsections, a 2D pixel position on Kinect depth map is denoted as $\mathbf{u} = (x, y)$. $\mathbf{D}_i(\mathbf{u})$ is the depth map value at $\mathbf{u}$ retrieved at time frame i. With an intrinsic calibration matrix $\mathbf{K}$, a 3D vertex of the depth value at $\mathbf{u}$ is $\mathbf{v}_i(\mathbf{u}) = \mathbf{D}_i(\mathbf{u})\mathbf{K}^{-1}[\mathbf{u}, 1]$. $\mathbf{D}_i$ therefore results in a single vertex map $\mathbf{V}_i$. A rotation and translation matrix for time frame i is $\mathbf{T}_i = [\mathbf{R}_i, \mathbf{t}_i]$. The vertex position is expressed in global coordinates as $\mathbf{v}_i^g = \mathbf{T}_i\mathbf{v}_i$.

3.1. Data filtering

One approach to filtering Kinect data in 3D space is to the model the noise PDF as a full 3×3 covariance matrix. As shown however, lateral noise is independent of depth distance (Figure 7a), and so a full covariance matrix can lead to redundancy. As a result, we can filter input depth data in image space, using only two variances σ_z^2 and σ_L^2. This allows us to apply more efficient and simpler image-based filtering methods, such as [10] to achieve similar results to full 3D filtering. Figure 8 shows a 2D visualization of the PDF contours for the Kinect, in image space (top) and 3D space (bottom). Note how the PDFs only expands along the z-direction as depth increases in image space.

Thus, the derived σ_L and σ_z can be used directly to support edge-aware filtering of depth maps. As σ_L is mostly less than 1 from equation 2, a raw depth map $\mathbf{D}_i$ is smoothed using a kernel size of 3×3 to generate a smoothed depth map $\tilde{\mathbf{D}}_i(\mathbf{u})$. This is described in Listing 1.

Authorized licensed use limited to: CSIRO Information Technology Services. Downloaded on August 16,2024 at 05:20:52 UTC from IEEE Xplore. Restrictions apply.

Listing 1 Smoothing raw depth

Assuming $\theta_{mean} = 30^\circ$
for each image pixel $\mathbf{u} \in$ depth map $\mathbf{D}_i$ **in parallel do**
  if $\mathbf{D}_i(\mathbf{u})$ within depth range **then**
    $\sigma_L, \sigma_z \leftarrow$ calculate from $\mathbf{D}_i(\mathbf{u})$ and θ_{mean}
    for each $\mathbf{u}_k$ in 3×3 pixel area around $\mathbf{u}$ **do**
      $\Delta u \leftarrow ||\mathbf{u} - \mathbf{u}_k||$
      $\Delta z \leftarrow ||\mathbf{D}_i(\mathbf{u}) - \mathbf{D}_i(\mathbf{u}_k)||$
      if $\Delta z <$ threshold **then**
        $\mathbf{w}_k \leftarrow \exp\left(-\frac{\Delta u^2}{2\sigma_L^2} - \frac{\Delta z^2}{2\sigma_z^2}\right)$
      else
        $\mathbf{w}_k \leftarrow 0$
    $\tilde{\mathbf{D}}_i(\mathbf{u}) \leftarrow \frac{\sum_k \mathbf{D}_i(\mathbf{u}_k)\mathbf{w}_k}{\sum_k \mathbf{w}_k}$

A normal vector map $\mathbf{N}_i$ is generated from $\tilde{\mathbf{D}}_i$ where its element $\mathbf{n}_i(\mathbf{u}) = (\mathbf{v}_i(x+1, y) - \mathbf{v}_i(x, y)) \times (\mathbf{v}_i(x, y+1) - \mathbf{v}_i(x, y))$. Then $\mathbf{n}_i(\mathbf{u})$ is normalized to unit length. Surface angle θ is calculated as angle between $\mathbf{n}_i(\mathbf{u})$ and the camera z-axis, i.e. $\theta = \arccos(|\mathbf{n}_{i,3}|)$. The normal vector is expressed in global coordinates as $\mathbf{n}_i^g(\mathbf{u}) = \mathbf{R}_i\mathbf{n}_i(\mathbf{u})$.

3.2. Weighted ICP for pose estimation

KinectFusion uses point-plane ICP to extract the relative 6-DoF rigid transform that best aligns the set of oriented 3D points derived from the current depth map with the oriented points derived from the previous pose. Correspondences are found using *projective data association* [8]. In addition to the distance and normal thresholds used in projective data association, only points where the angle of normal vector $\theta = \arccos(|\mathbf{n}_{i,3}|)$ is less than 70^0 are used for pose estimation, as points with larger normal angle have significant error (as shown in Figure 7).

The point-plane energy function is linearized by assuming small angle changes from one pose to the next [6]. However, this approximation treats data points of different noise levels equally, and may lead to suboptimal solutions. To avoid this we include a weighting factor $w_i = \sigma_z(z_{min}, 0)/\sigma_z$ in the relative pose given by:

$$\arg\min \sum_{\substack{\mathbf{u} \\ \mathbf{D}(\mathbf{u})>0}} \left\| \left(\mathbf{T}_i\mathbf{v}_i(\mathbf{p}) - \mathbf{v}_{i-1}^g(\mathbf{u})\right) \cdot \mathbf{n}_{i-1}^g(\mathbf{u}) \cdot w_i \right\| \quad (5)$$

3.3. Volumetric depth map fusion

To better accommodate the Gaussian-like axial-noise (Figure 5), we propose a new approximation of the Truncated Signed Distance Function (TSDF) [3] encoded in the voxel grid. This is based on the Cumulative Distribution Function (CDF) of the axial noise PDF.

Figure 9 depicts our new approach for modeling the TSDF. In the original TSDF, a truncation length μ is used to account for measurement uncertainty. We set this to $3\sigma_z$ to relate it to the axial noise distribution. In fact, our experiments show that a truncation length of $6\sigma_z$ enables us to accommodate additional noise due to camera tracking error. To allow for more refinement, truncation length can be adjusted adaptively from $6\sigma_z$ down to $3\sigma_z$ according to decreasing residual error of ICP optimization when the reconstruction quality increases. The pseudo-code excerpt in Listing 2 shows the improved volumetric depth map fusion for KinectFusion.

The TSDF calculation at line 17 of Listing 2 is a modified version of an approximate Gaussian CDF [1] to generate values within $[-1, 1]$. This operates on the SDF value $\mathbf{sdf}_k$ calculated as in the original KinectFusion algorithm.

Listing 2 Projective TSDF integration

1 **for** each voxel $\mathbf{g}$ on x-y slice of volume **in parallel do**
2 **while** sweep along z-axis of volume **do**
3 $\mathbf{v}^g \leftarrow$ convert $\mathbf{g}$ from grid to global 3D position
4 $\mathbf{v} \leftarrow \mathbf{T}_i^{-1}\mathbf{v}^g$
5 $\mathbf{p} \leftarrow$ perspective project vertex $\mathbf{v}$
6 **if** $\mathbf{p} \in$ depth map $\mathbf{D}_i$ **and** $\mathbf{D}_i(\mathbf{p}) > 0$ **then**
7 $\mathbf{tsdf}_i \leftarrow \mathbf{tsdf}_{i-1}$
8 $w_i \leftarrow w_{i-1}$
9 **for** each $\mathbf{p}_k$ in 3×3 area around $\mathbf{p}$ **do**
10 $\theta \leftarrow$ angle(z-axis, $\mathbf{n}_i(\mathbf{p}_k)$)
11 $\Delta u \leftarrow ||\mathbf{p} - \mathbf{p}_k||$ with sub-pixel accuracy
12 $\Delta z \leftarrow ||\mathbf{D}_i(\mathbf{p}) - \mathbf{D}_i(\mathbf{p}_k)||$
13 **if** $\mathbf{D}_i(\mathbf{p}_k) > 0$ **and** $\theta <$ angle threshold **then**
14 $\sigma_L, \sigma_z \leftarrow$ calculate from $\mathbf{D}_i(\mathbf{p}_k)$ and θ
15 $\mathbf{sdf}_k \leftarrow ||\mathbf{t}_i - \mathbf{v}|| - \mathbf{D}_i(\mathbf{p}_k)$
16 **if** $\mathbf{sdf}_k > -6\sigma_z$ **and** $\Delta z < 3\sigma_z$ **then**
17 $\mathbf{tsdf}_k \leftarrow \mathrm{sgn}(\mathbf{sdf}_k)\sqrt{1 - e^{-\frac{2}{\pi}\frac{\mathbf{sdf}_k^2}{\sigma_z^2}}}$
18 $w_k \leftarrow \frac{\sigma_z(z_{min},0)}{\sigma_z}\frac{z_{min}^2}{\mathbf{D}_i^2(\mathbf{p}_k)}e^{\left(-\frac{\Delta u^2}{2\sigma_L^2} - \frac{\Delta z^2}{2\sigma_z^2}\right)}$
19 $\mathbf{tsdf}_i \leftarrow (\mathbf{tsdf}_i w_i + \mathbf{tsdf}_k w_k)/(w_i + w_k)$
20 $w_i \leftarrow$min(max weight, $w_i + w_k$)

In the original KinectFusion system geometric features could be lost when the accumulated voxel model was updated with nosier depth measurements further away from the surface. This is because noisy depth measurements were integrated into the voxel grid with the same weight as measurements closer to surfaces. To avoid this problem, normalized terms are added to the voxel weight (line 18, Listing 2). The exponential term is the noise distribution weight. The term $\frac{z_{min}^2}{\mathbf{D}_i^2(\mathbf{p}_k)}$ adjusts for the spread of the width of 3D noise distribution, covering more voxels as z-depth increases. The term $\frac{\sigma_z(z_{min},0)}{\sigma_z}$ accounts for increased length of the distribution for larger z-depths (see Figure 8).

Authorized licensed use limited to: CSIRO Information Technology Services. Downloaded on August 16,2024 at 05:20:52 UTC from IEEE Xplore. Restrictions apply.

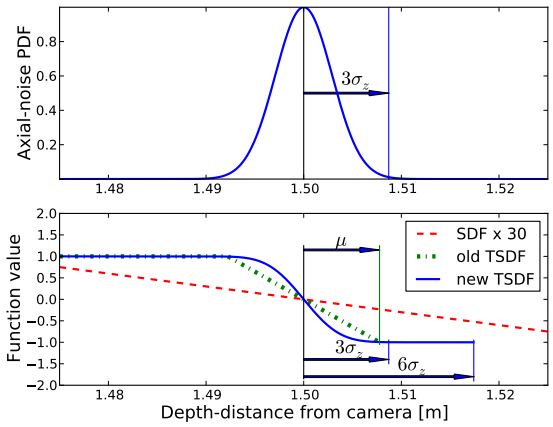


Figure 9. Top: an empirical axial-noise PDF with standard deviation σ_z. Bottom: comparison of SDF, old and new TSDF models generated from depth measurements at 1.5m. (Note: the SDF is scaled 30 times to show the variation.) μ is the truncation length of the original linear TSDF. The new TSDF is a cumulative distribution function based on the axial-noise PDF. While $3\sigma_z$ is equivalent to the truncation length of the original linear TSDF, truncation at $6\sigma_z$ helps accommodate noise due to camera tracking error.

4. Reconstruction and tracking results

To illustrate the effectiveness of the new KinectFusion pipeline, experiments were performed in two challenging scanning situations where objects with thin and fine structures are scanned on a manual turntable as shown in Figure 10. The Kinect was attached to a frame and rotated manually around the table at roughly a fixed distance and angle. Raw depth data was recorded and processed off-line for direct comparison between the original KinectFusion [4, 7] implementation and our extensions. External Vicon tracking is used to provide ground truth tracking information to validate the 6-DoF pose estimation performance. Markers were attached to the body of the Kinect as shown in the left of Figure 10.

4.1. Qualitative Results

Figure 10. Two test scenes scanned by rotating a Kinect around the table as shown in the left figure. Vicon markers for ground truth tracking were attached to the body of the Kinect.

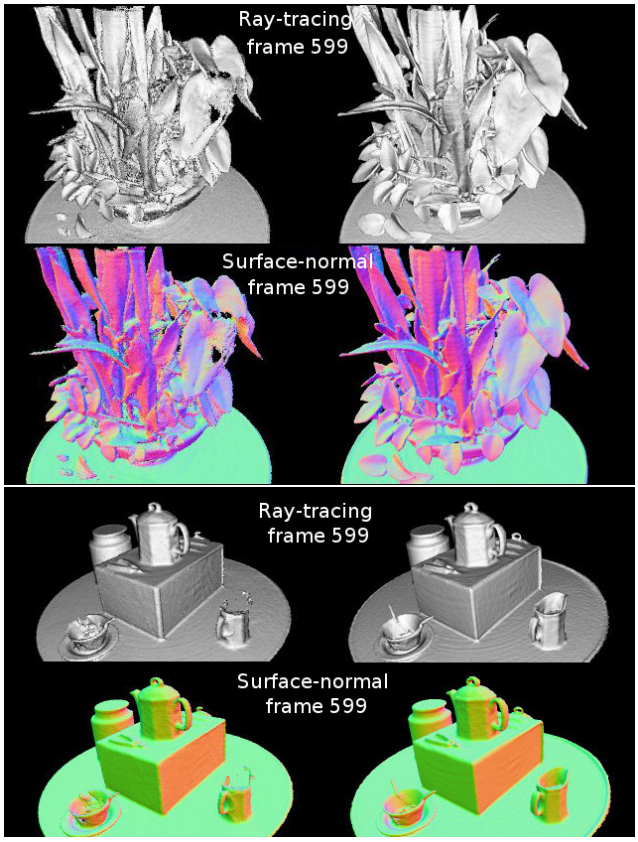


Figure 11. Full 360 reconstruction of two scenes using original KinectFusion (left) and our extensions (right).

Figure 11 shows reconstruction results of the original KinectFusion (left columns) in comparison to results from our extensions (right columns). The reconstruction volume is $1\mathrm{m}^3$ with a resolution of 512^3, centered at 0.8m from the Kinect. The original KinectFusion has more noise and missing structures, and less details. The extended KinectFusion produces reconstructions with higher quality and more detail, especially the leaves and other thin surfaces.

4.2. Quantitative Results

Figure 12 compares angular tracking performance of the original and extended KinectFusion algorithms for the scene shown in Figure 11 (left). The estimation of (yaw) rotation around the turntable is compared against ground truth data from the Vicon across about 600 depth frames. The original KinectFusion has better tracking accuracy at first, however after about 150^0 rotation, the extended version shows improved tracking accuracy. The extended KinectFusion achieves a 2.5% reduction in angular tracking error after a full rotation compared with the original.

5. Conclusion

We have presented a new, empirically derived noise model for the Kinect sensor, which considers both axial and

Authorized licensed use limited to: CSIRO Information Technology Services. Downloaded on August 16,2024 at 05:20:52 UTC from IEEE Xplore. Restrictions apply.

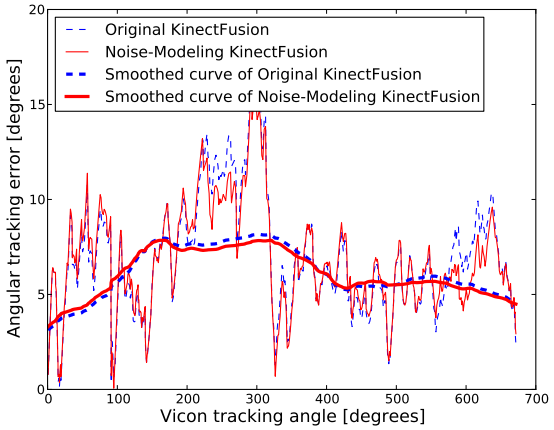


Figure 12. Tracking error around turntable by original and extended KinectFusion algorithms with respect to ground truth data. The error is the magnitude of the difference between yaw angle produced by KinectFusion and the groundtruth by Vicon tracking. The dashed lines are for the angle errors by original KinectFusion before and after smoothing. The continuous lines are for the angle errors with our approach before and after smoothing. The new algorithm reduces tracking error for a full 360 rotation by 2.5%.

lateral noise distributions as a function of distance and angle from sensor to observed surface. Whilst other applications are feasible, we have demonstrated the effectiveness of these derived noise models in improving reconstruction accuracy for KinectFusion.

We have proposed a method of estimating lateral noise PDF using the edge pixels of a planar target. We found that in real world units, lateral noise increases linearly with distance while axial noise increases quadratically. Both lateral and axial noise are fairly independent of surface angle until approximately 70° where they increase rapidly.

We use these derived noise models for improved noise filtering of Kinect data. We also propose extensions to the KinectFusion pipeline to improve reconstruction quality and pose estimation based on these derived noise models. This has been demonstrated by two test scans where thin structures and fine details can pose a significant challenge for 3D reconstruction.

From our experiments we have observed that our new empirically derived noise model has the potential to allow surfaces to converge quicker during reconstruction than the original KinectFusion system. Whilst we need to verify this quantitatively, we have noted that more detailed reconstructions of test scenes could be acquired in less frames with our explicit noise modeling. This was observed during all our experiments (in Figure 11) and will be quantified in our future work. Another rich area for future research is to extend our empirically derived noise model to better accommodate surface material properties such as reflectance and albedo, a research topic in its own right, but one that clearly can impact noise characteristics.

Acknowledgments

The authors wish to thank Matt Adcock of CSIRO ICT; Toby Sharp, Andrew Fitzgibbon, Pushmeet Kohli, Jiawen Chen, David Kim and Otmar Hilliges of Microsoft Research Cambridge; and Paul McIlroy of Cambridge University for their help during this work. This work was supported by Acorn Grant 2011 from CSIRO Transformational Biology.

Supplementary material available at: http://research.microsoft.com/en-us/projects/surfacerecon/

References

[1] K. Aludaat and M. Alodat. A note on approximating the normal distribution function. *Applied Mathematical Sciences*, 2(9):425–429, 2008. 5

[2] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 4

[3] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 4, 5

[4] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, S. Hodges, P. Kohli, J. Shotton, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *ACM UIST 2011*, pages 559–568. ACM, 2011. 1, 6

[5] K. Khoshelham and S. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012. 1, 3, 4

[6] K. Low. Linear least-squares optimization for point-to-plane icp surface registration. Technical report, Technical Report TR04-004, Department of Computer Science, University of North Carolina at Chapel Hill, 2004. 5

[7] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2011*, pages 127–136. IEEE, 2011. 1, 6

[8] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings. Third International Conference on 3-D Digital Imaging and Modeling, 2001.*, pages 145–152. IEEE, 2001. 5

[9] J. Smisek, M. Jancosek, and T. Pajdla. 3d with kinect. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011*, pages 1154–1160. IEEE, 2011. 1

[10] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth IEEE International Conference on Computer Vision, 1998.*, pages 839–846. IEEE, 1998. 4

Authorized licensed use limited to: CSIRO Information Technology Services. Downloaded on August 16,2024 at 05:20:52 UTC from IEEE Xplore. Restrictions apply.