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 mod-
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).

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 $\theta$. 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 $\sigma_L$. Detected edge pixels are shown as thick lines.

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 $\sigma_L$ and $\sigma_z$ are computed.

2.3. Results and models of noise distributions

Experimental results were obtained using A2–A5-sized targets of same surface material and thickness. Targets
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 $\theta$. 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 $70^\circ$, 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 $\theta$ between 10-60$^\circ$. 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.](image)

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 $\theta$ before $70^\circ$. When converting from pixel to real-world distances in meters, lateral noise increases linearly with z-depth as shown in Figure 6.

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$^\circ$. Equation 1 gives the standard deviation distribution in pixels, and equation 2 converts this to real-world units (meters):

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

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

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). $\sigma_L$ 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 $\sigma_z$ varies significantly with z-depth but remains constant at angles less than approximately $60^\circ$ and increases rapidly when the angle approaches $90^\circ$. At distances beyond 2.5m $\sigma_z$ show a noticeable decrease as angle approaches $0^\circ$. 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$^\circ$ 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$^\circ$.

$$\sigma_z(z, \theta) = 0.0012 + 0.0019(z - 0.4)^2, \ 10^\circ \leq \theta \leq 60^\circ$$
There results in a theoretical covariance matrix $\Sigma = (\alpha_i \times \text{covariance matrix})$.

A 2D visualization of the PDF contours for the Kinect, in image space (top) and in 3D space (bottom). Note how the PDFs only expands along the $z$-direction as depth increases in image space.

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 $\theta$ approaches $90^\circ$:

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

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 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.

In the following subsections, a 2D pixel position on Kinect depth map is denoted as $\mathbf{u} = (x, y)$. $D_i(\mathbf{u})$ 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 $\mathbf{v}_i(\mathbf{u}) = D_i(\mathbf{u})\mathbf{K}^{-1}[\mathbf{u}, 1]$. $\mathbf{D}_i$ 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 use the model the noise PDF as a full $3 \times 3$ 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 $\sigma^2_L$ and $\sigma^2_z$. 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 $\sigma_L$ and $\sigma_z$ can be used directly to support edge-aware filtering of depth maps. As $\sigma_L$ is mostly less than 1 from equation 2, a raw depth map $\mathbf{D}_i$ is smoothed using a kernel size of $3 \times 3$ to generate a smoothed depth map $\mathbf{D}_i(\mathbf{u})$. This is described in Listing 1.
A normal vector map $\mathbf{N}_i$ is generated from $\mathbf{D}_i$ where its element $\mathbf{n}_i(u) = (v_i(x + 1, y) - v_i(x, y)) \times (v_i(x, y + 1) - v_i(x, y))$. Then $\mathbf{n}_i(u)$ is normalized to unit length. Surface angle $\theta$ is calculated as angle between $\mathbf{n}_i(u)$ and the camera z-axis, i.e. $\theta = \arccos(|\mathbf{n}_i|)$. The normal vector is expressed in global coordinates as $\sigma^3(u) = \mathbf{R} \mathbf{n}_i(u)$.

### 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 $\theta = \arccos(|\mathbf{n}_i|)$ is less than $70^\circ$ 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 $w_i = \frac{\sigma_x(z_{min}, 0) / \sigma_x}{\sigma_x(z_{min}, 0) / \sigma_x}$ in the relative pose given by:

$$\arg \min_u \sum_{\theta > 0} \| (\mathbf{T}_i \mathbf{v}_i(p) - \mathbf{v}_{i-1}(u)) \cdot \mathbf{n}_{i-1}(u) \cdot w_i \|$$

### 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 $\mu$ 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 $sdf_k$ calculated as in the original KinectFusion algorithm.

```
Listing 1: Smoothing raw depth
1: Assuming $\theta_{mean} = 30^\circ$
2: for each image pixel $u \in$ depth map $D_i$ in parallel do
3: if $D_i(u)$ within depth range then
4: $\sigma_L, \sigma_S \leftarrow $ calculate from $D_i(u)$ and $\theta_{mean}$
5: for each $u_k$ in $3 \times 3$ pixel area around $u$ do
6: $\Delta u \leftarrow |u - u_k|$
7: $\Delta z \leftarrow |D_i(u) - D_i(u_k)|$
8: if $\Delta z <$ threshold then
9: $w_k \leftarrow \exp \left(-\frac{\Delta u^2}{2\sigma_L^2} - \frac{\Delta z^2}{2\sigma_S^2}\right)$
10: else
11: $w_k \leftarrow 0$
12: $\hat{D}_i(u) \leftarrow \sum_{u_k} D_i(u_k) w_k$

Listing 2: Projective TSDF integration
1: for each voxel $g$ on x-y slice of volume in parallel do
2: while sweep along z-axis of volume do
3: $v^g \leftarrow$ convert $g$ from grid to global 3D position
4: $v \leftarrow T_r^{-1} v^g$
5: $p \leftarrow$ perspective project vertex $v$
6: if $p \in$ depth map $D_i$ and $D_i(p) > 0$ then
7: $\text{tsdf}_k \leftarrow \text{tsdf}_{k-1}$
8: $w_i \leftarrow w_{i-1}$
9: for each $p_k$ in $3 \times 3$ area around $p$ do
10: $\theta \triangleleft$ angle(z-axis, $\mathbf{n}_i(p_k)$)
11: $\Delta u \leftarrow ||p \cdot \mathbf{p}_k||$ with sub-pixel accuracy
12: $\Delta z \leftarrow ||D_i(p) - D_i(p_k)||$
13: if $D_i(p_k) > 0$ and $\theta <$ angle threshold then
14: $\sigma_L, \sigma_S \leftarrow$ calculate from $D_i(p_k)$ and $\theta$
15: $sdf_k \leftarrow |||t_i - v|| - D_i(p_k)||$
16: if $sdf_k > -6\sigma_z$ and $\Delta z < 3\sigma_z$ then
17: $\text{tsdf}_k \leftarrow \text{sgn} (sdf_k) \sqrt{1 - e^{-\frac{\Delta z^2}{2\sigma_z^2}}}$
18: $w_k \leftarrow \frac{\sigma_x(z_{min}, 0) \sigma_x}{\sigma_x(p_k)} e^{-\frac{\Delta u^2}{2\sigma_L^2} - \frac{\Delta z^2}{2\sigma_z^2}}$
19: $\text{tsdf}_i \leftarrow (\text{tsdf}_i + \text{tsdf}_k w_k) / (w_i + w_k)$
20: $w_i \leftarrow \min(\text{max weight}, w_i + w_k)$
```

In the original KinectFusion system geometric features could be lost when the accumulated voxel model was updated with noisier 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 $\frac{\sigma_x(z_{min}, 0)}{\sigma_x(p_k)}$ 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).
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.

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° 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
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/surfacer econ/

References


