σ-DVO: Sensor Noise Model Meets Dense Visual Odometry

Benzun Wisely Babu *
Worcester Polytechnic Institute, Worcester, MA

Soohwan Kim†
Bosch Research, Palo Alto, CA

Zhixin Yan‡
Bosch Research, Palo Alto, CA

Liu Ren§
Bosch Research, Palo Alto, CA

ABSTRACT

In this paper we propose a novel method called σ-DVO for dense visual odometry using a probabilistic sensor noise model. In contrast to sparse visual odometry, where camera poses are estimated based on matched visual features, we apply dense visual odometry which makes full use of all pixel information from an RGB-D camera. Previously, t-distribution was used to model photometric and geometric errors in order to reduce the impacts of outliers in the optimization. However, this approach has the limitation that it only uses the error value to determine outliers without considering the physical process. Therefore, we propose to apply a probabilistic sensor noise model to weigh each pixel by propagating linearized uncertainty. Furthermore, we find that the geometric errors are well represented with the sensor noise model, while the photometric errors are not. Finally we propose a hybrid approach which combines t-distribution for photometric errors and a probabilistic sensor noise model for geometric errors. We extend the dense visual odometry and develop a visual SLAM system that incorporates keyframe generation, loop constraint detection and graph optimization. Experimental results with standard benchmark datasets show that our algorithm outperforms previous methods by about a 25% reduction in the absolute trajectory error.

Keywords: Visual SLAM, Dense Visual Odometry, Camera Pose Tracking, 3D Reconstruction, Augmented Reality

1 INTRODUCTION

Visual SLAM (Simultaneous Localization and Mapping) is one of the fundamental problem in augmented reality to build a map of an unknown environment and localize camera poses based on it. In particular, the accuracy of camera pose estimation is very important for augmented reality applications because users can immediately recognize the discrepancy between virtual and real objects even with a small amount of tracking errors.

Visual odometry which estimates relative camera poses between two adjacent frames, is used in the front-end of visual SLAM. The estimates are sent as inputs to the back-end and corrected through an optimization process. There are two types of approaches for visual odometry; *sparse* and *dense*. Sparse visual odometry extracts visual features from images and estimates camera poses based on the correspondences. Dense visual odometry, on the other hand, makes full use of all the pixels in images and finds the optimal camera poses based on the photometric and geometric consistency.

With the introduction of affordable RGB-D cameras in the last decade such as Microsoft Kinect and Intel RealSense, it has become increasingly popular to use such sensors for visual SLAM in indoor environments [17, 2]. An RGB-D camera captures both color and depth images from the scene. The color image provides texture information that can be used for photometric consistency, while the depth image provides structure information that can be used for geometric consistency.

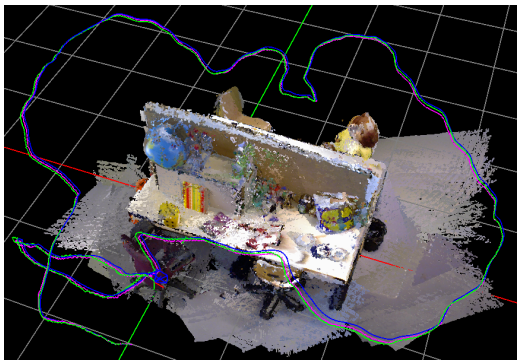


Figure 1: Results of our σ-DVO SLAM with an RGB-D dataset [24]. Notice that our σ-DVO (green) is close to the ground truth (blue). After pose optimization in the back-end, our σ-DVO SLAM (magenta) becomes more aligned with the ground truth.

Previously, Kerl et al. [9, 10] proposed Dense Visual Odometry (DVO) where they analyzed photometric and geometric errors and modeled them with t-distribution. By doing that, they could underweigh outliers and thus achieve more robust camera pose estimation. However, their experimental results in [9] only suggested that the photometric errors of pixels in a gray-scale image follow t-distribution. There is no direct evidence to support that geometric errors share the same distribution.

In this paper, we propose a novel method called σ-DVO for dense visual odometry using a probabilistic sensor noise model as shown in Figure 1. Our motivation is to improve the robustness of the dense visual odometry by adopting a more rigorous representation of weights for each pixel in RGB-D images. For each pixel, noise is introduced in the observation process. Hence, we argue that directly modelling the cause, which is the sensor noise, instead of modelling the result, which is the residuals, is a better approach to solve the optimization problem. For example, when introduced to a new room, in order to localize ourselves, we would prefer to give higher importance to objects nearby because our depth estimation ability drops nearly exponentially as the distance increases. Similarly, the accuracy of an RGB-D camera decreases because of higher triangulation error or weaker illumination quality with distance. Furthermore, we observe that the sensor noise model is valid with geometric errors, but not with photometric errors. Hence, we propose a hybrid approach which utilizes t-distribution for photometric errors and a sensor noise model for geometric errors. Experimental results with benchmark datasets show that our σ-DVO outperforms DVO by about a 25% reduction in the absolute trajectory error.

The main contributions of this paper are:

1. We apply a probabilistic sensor noise model for robust RGB-D based dense visual odometry and propagate the uncertainty in the observation to the residuals by linearization.

*e-mail: bpwiselybabu@wpi.edu. The work was performed during his internship at Bosch Research, USA.

†e-mail: soohwan.kim2@us.bosch.com, Benzun and Soohwan are co-first authors with equal contributions, and Soohwan is the corresponding author of the paper.

‡e-mail: zhixin.yan@us.bosch.com

§e-mail: liu.ren@us.bosch.com

2. We propose a hybrid approach which combines t-distribution for photometric errors and a sensor noise model for geometric errors into a single optimization problem based on rigorous linearization analysis.
3. We extend our σ-DVO to σ-DVO SLAM by incorporating keyframe generation, loop constraint detection and graph optimization.

The rest of the paper is organized as follows. Section 2 provides an overview of existing visual SLAM algorithms. Section 3 defines the dense visual odometry problem mathematically. We propose our σ-DVO in Section 4 and extend it to σ-DVO SLAM in Section 5. Experimental results with benchmark datasets are provided in Section 6. Finally, We discuss benefits and limitations of our method in Section 7 and conclude the paper in Section 8.

2 Background

Early camera pose estimation had been studied in the computer vision community by Sturm and Triggs [25] as the Structure from Motion problem. During the same period research in robotics Leonard and Durrant-Whyte [13] had formulated pose estimation as a filtering problem. One of the early real-time camera pose estimation algorithms was presented by Davison [4]. He used an extended Kalman filter to estimate camera poses from features points. It was limited to texture-rich environments over a small area.

2.1 Sparse Methods

Following the work of Davison [4], most of the research in the last decade has been focused on feature based SLAM. Klein and Murray [11] proposed Parallel Tracking And Mapping (PTAM) which reformulated the filtering problem into a batch optimization problem. By using parallel processing capabilities they were able to partition tracking and map management into two separate problems. Even though PTAM was limited to a small area, it introduced ideas like keyframe selection and relocalization that are essential for robust tracking. Recently, Mur-Artal and Tardós [16] presented ORB-SLAM which is a versatile sparse feature based SLAM algorithm for monocular, stereo and RGB-D cameras. ORB-SLAM is able to handle large areas but does not generate a dense map of the world which is essential for augmented reality applications.

2.2 Dense Methods

Compared to sparse feature based methods, dense methods use all the pixels in an image for pose estimation. This results in more robust and accurate pose estimates but at a higher computation cost. Newcombe et al. [18] presented Dense Tracking And Mapping (DTAM) which used a computationally expensive global optimization approach to perform pose estimation. While DTAM could generate accurate camera trajectories and provide a dense mesh from a single RGB camera, the computational limitations created by primal-dual optimization process made it only applicable to small areas. Later, Newcombe et al. [17] proposed a dense surface tracking approach for RGB-D cameras called KinectFusion. It was also limited by the map size and relied extensively on GPUs for map management and representation. To overcome the map limitations of KinectFusion, Stückler and Behnke [23] presented multi-resolution surfel based SLAM which used OctoMaps for map marginalization and optimization. Whelan et al. [26] suggested improvements to the dense SLAM by using deformation based loop closures on the map structure. Recently, Whelan et al. [28] extended the deformation based loop closure to a surfel based mapping algorithm that was able to handle both large and small areas. Most map based optimization methods are computationally expensive and often require GPUs but our approach uses a simpler map representation that is lightweight for camera tracking in augmented reality applications.

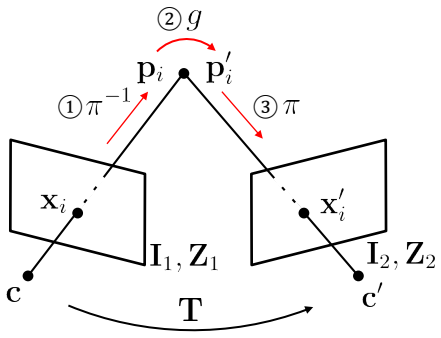


Figure 2: Overview of RGB-D based dense visual odometry. (1) Back-project a 2D point $\mathbf{x}_i$ in Image 1 to a 3D point $\mathbf{p}_i$ given its depth, (2) transform it to a 3D point $\mathbf{p}'_i$ from the previous camera coordinates $\mathbf{c}$ to the current ones $\mathbf{c}'$ given the relative camera pose $\mathbf{T}$, and (3) project it onto Image 2, ending up with $\mathbf{x}'_i$. Now, we optimize the relative camera pose $\mathbf{T}$ by minimizing the intensity and depth errors of two corresponding points $\mathbf{x}_i$ and $\mathbf{x}'_i$.

2.3 Robust Methods

Kerl et al. [9, 10] formulated the dense SLAM algorithm as a probabilistic optimization for the camera pose that minimized photometric errors and geometric errors. In their work, they showed t-distribution to be a good choice for weighting the residuals and to reduce the impacts of outliers in dense visual odometry. Gutirrez-Gmez et al. [7] extended the use of t-distribution to an inverse depth formulation of the dense visual odometry algorithm. Forster et al. [6] introduced Semi-direct Visual Odometry (SVO) for a monocular system. They used the concept of depth filtering to improve the depth estimates. Unlike our approach, none of these methods account for the sensor noise in RGB-D cameras.

2.4 Sensor Noise Based Methods

Maimone et al. [14] applied sparse visual odometry to the Mars rover. They modeled the sensor noise in the image space to improve visual odometry formulation. Segal et al. [21] introduced a generalized formulation of ICP (Iterative Closest Points) called GICP in which the sensor model could be integrated. However, it only considered the geometric mis-match in the optimization framework. Ruhnke et al. [20] showed that by considering the noise of the sensor during model optimization, the accuracy of the generated mesh could be improved. Our approach draws parallels from these previous approaches and introduces the sensor noise in 3D into the dense visual odometry problem. A more detailed survey on current state of camera based tracking for augmented reality has been presented by Marchand et al. [15]

3 Dense Visual Odometry using an RGB-D Camera

In this section we define the problem of RGB-D based dense visual odometry and formulate mathematical equations and notations which will be used throughout the paper.

3.1 Preliminary

The objective of visual odometry is to estimate the ego motion of the camera between two consecutive frames using visual information. In contrast to sparse visual odometry, where visual features are extracted and the camera pose is estimated from matched correspondences, dense visual odometry makes full use of observations. In other words, based on the assumption that between two consecutive frames, there is little change in the scene structure and the lighting condition, we find the optimal camera pose which minimizes the photometric and geometric errors.

Figure 2 describes how to compare intensities and depths of two consecutive frames in three steps. For any arbitrary 2D point $\mathbf{x}_i$ in the previous image coordinates which is associated with a depth value $Z_1(\mathbf{x}_i)$, we first back-project it to a 3D point $\mathbf{p}_i$ in the previous camera coordinates with the back-projection function, $\mathbf{p}_i = \pi^{-1}(\mathbf{x}_i, Z_1(\mathbf{x}_i))$. Given a relative camera pose $\mathbf{T}$ between two frames, we transform $\mathbf{p}_i$ to $\mathbf{p}'_i$ in the current camera coordinates. Then, the corresponding 2D point $\mathbf{x}'_i$ in the current image coordinates can be determined by projecting the transformed 3D point with the projection function, $\mathbf{x}'_i = \pi(\mathbf{p}'_i)$. Finally, we compute the relative camera pose by minimizing both intensity and depth errors during optimization. For more details about the the projection and back-projection, please refer to [8].

The camera trajectory that we wish to estimate lies in the class of rigid body motions formed by *special Euclidean group* SE(3). Thus, the relative pose of the camera between two consecutive frames can be expressed as a 4x4 transformation matrix $\mathbf{T}$ which includes $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^{3\times 1}$ as the *rotation matrix* and *translation vector*, respectively. Then, the mapping between two 3D points $\mathbf{p}_i$ and $\mathbf{p}'_i$ in two camera coordinates is defined as $\mathbf{p}'_i = g(\mathbf{p}_i, \mathbf{T}) = \mathbf{R}\,\mathbf{p}_i + \mathbf{t}$. In order to estimate the relative camera pose, we introduce an optimization problem which minimizes an energy function. We apply a six dimensional minimal representation of the relative pose $\mathbf{T}$ using *twist* ξ coordinates given by Lie algebra $\mathfrak{se}(3)$.

3.2 Non-linear Least Squares

Given the camera projection model and rigid body motion, the photometric and geometric errors of each pixel are defined as

$$\mathbf{r}_i = \begin{bmatrix} r_i^I \\ r_i^Z \end{bmatrix} = \begin{bmatrix} I_2\left(w(\mathbf{x}_i;\xi)\right) - I_1(\mathbf{x}_i) \\ Z_2\left(w(\mathbf{x}_i;\xi)\right) - \left[g\left(\pi^{-1}(\mathbf{x}_i, Z_i), \xi\right)\right]_Z \end{bmatrix}, \quad (1)$$

where the warping function $w(\mathbf{x}_i;\xi) = \pi\left(g\left(\pi^{-1}(\mathbf{x}_i, Z_i), \xi\right)\right)$, $Z_i = Z_1(\mathbf{x}_i)$, and $[\cdot]_Z$ denotes the z component of the vector. Refer to the Lucas-Kanade algorithm [3] for this optimization framework.

Our objective is to find the relative camera pose which minimizes the weighted sum of squared errors as

$$\hat{\xi} = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{n} \mathbf{r}_i^\intercal \mathbf{W}_i \mathbf{r}_i \ , \quad (2)$$

where n is the total number of valid pixels, and $\mathbf{W}_i \in \mathbb{R}^{2\times 2}$ denotes the weights for different error types.

Since the energy function is non-linear with respect to the relative camera pose ξ, the Gauss-Newton algorithm [19] is usually applied to find the optimal solution numerically. The update formula and the normal equation for Eq. (2) are

$$\xi_{k+1} = \xi_k + \Delta\xi, \quad \sum_{i=1}^{n} \mathbf{J}_i^\intercal \mathbf{W}_i \mathbf{J}_i \Delta\xi = -\sum_{i=1}^{n} \mathbf{J}_i^\intercal \mathbf{W}_i \mathbf{r}_i \ , \quad (3)$$

where the Jacobian matrix is defined as

$$\mathbf{J}_i = \begin{bmatrix} \frac{\partial r_i^I}{\partial \xi_1} & \cdots & \frac{\partial r_i^I}{\partial \xi_6} \\ \frac{\partial r_i^Z}{\partial \xi_1} & \cdots & \frac{\partial r_i^Z}{\partial \xi_6} \end{bmatrix} \in \mathbb{R}^{2\times 6} \ . \quad (4)$$

3.3 Robust Camera Pose Estimation

It is well known that non-linear least squares are sensitive to outliers. Thus, in sparse visual odometry, outlier rejection techniques such as RANSAC are usually employed to enhance the estimated pose accuracy. In dense visual odometry, on the other hand, we elaborate the weights in the energy function to improve the robustness of the optimization against outliers.

3.3.1 Normal Distribution

We first start with viewing the least square errors in a probabilistic way. In fact, Eq. (2) is equivalent with maximum likelihood estimation where each residual is independent and follows an identical Gaussian distribution,

$$\hat{\xi}_{\mathrm{ML}} = \underset{\xi}{\operatorname{argmax}} \sum_{i=1}^{n} \log p\left(\mathbf{r}_i \mid \xi\right) \ , \quad (5)$$

where $p(\mathbf{r}_i \mid \xi) = \mathcal{N}(\mathbf{0}, \Sigma)$, and Σ denotes a covariance matrix of the zero mean normal distribution. Note that this corresponds to the case where $\mathbf{W}_i = \Sigma^{-1}$ in Eq. (2).

Since it is known that maximum likelihood is prone to overfitting, a prior distribution on the relative camera pose can be imposed, leading to maximum a posteriori,

$$\hat{\xi}_{\mathrm{MAP}} = \underset{\xi}{\operatorname{argmax}} \sum_{i=1}^{n} \log p\left(\mathbf{r}_i \mid \xi\right) + \log p\left(\xi\right) \ . \quad (6)$$

The prior distribution is usually defined as a Gaussian distribution based on IMU data or a constant velocity model. Here, we use the previous camera pose as a prior since we assume the system update is fast enough. Henceforth, we omit the prior term for brevity and focus on the likelihood term only.

3.3.2 Student's t-Distribution

Kerl et al. [10] proposed DVO where they analyzed the actual values of the residuals $\mathbf{r}_i$ and showed that a t-distribution explains the residuals better than a Gaussian distribution does. So, they employed a zero-mean t-distribution for the likelihood function,

$$p\left(\mathbf{r}_i \mid \xi\right) = \frac{\Gamma\left((\nu+2)/2\right)}{\Gamma(\nu/2)\,\nu\pi\sqrt{|\Sigma_t|}} \left(1 + \frac{1}{\nu}\mathbf{r}_i^\intercal \Sigma_t^{-1} \mathbf{r}_i\right)^{-(\nu+2)/2} \ , \quad (7)$$

where ν is the number of degrees of freedom and Σ_t denotes the covariance matrix of the t-distribution.

Then, the maximum likelihood estimation of Eq. (5) can be rewritten as

$$\hat{\xi}_{\mathrm{DVO}} = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{n} w_i\, \mathbf{r}_i^\intercal \Lambda \mathbf{r}_i \ , \quad (8)$$

where $w_i = (\nu+2)/(\nu + \mathbf{r}_i^\intercal \Lambda \mathbf{r}_i)$, and Λ is the scaling matrix between photometric and geometric residuals. Note that this corresponds to the case where $\mathbf{W}_i = w_i \Lambda$ in Eq. (2).

In the case of a Gaussian distribution, all the pixels have the same weights as shown in Eq. (5). This can be problematic when there are existing outliers, because they have the same impact on the optimal solution. However, because the probability density function of t-distribution quickly drops as the input moves away from the mean, the weights of outliers become much lower than those of inliers, and so the DVO results are robust to outliers.

4 Our Approach: σ-DVO

The main limitation of DVO is that it only relies on the residual values to determine whether some pixel observations are important (inliers) or not (outliers) without considering the physical observation process. In this section, we introduce σ-DVO which overcomes DVO's limitation by applying a probabilistic sensor noise model and propagating the linearized uncertainty to residuals.

4.1 RGB-D Sensor Noise Model

Most of the RGB-D cameras such as Microsoft Kinect and Intel RealSense emit infra-red patterns and recover depth from correspondences between two image views with a small parallax. During

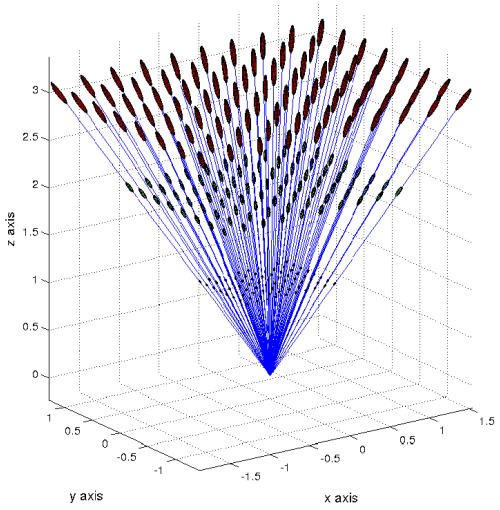


Figure 3: Sensor noise model of an RGB-D camera. The camera is located at the origin and is looking up in the z direction. For each range of 1, 2, and 3 meters, 80 points are sampled. Uncertainty of each point observation is expressed with an ellipsoid.

this process, the disparity is quantized into sub-pixels, which introduces a quantization error in the depth measurement. The noise due to quantization error is defined as,

$$\eta(Z_i) = \frac{q_{\text{pix}} bf}{2} \left[\frac{1}{\text{Rnd}(\frac{q_{\text{pix}} bf}{Z_i} - 0.5)} - \frac{1}{\text{Rnd}(\frac{q_{\text{pix}} bf}{Z_i} + 0.5)} \right] \quad , \quad (9)$$

where q_{pix} is the sub-pixel resolution of the device, b is the baseline, and f is the focal length.

This error increases quadratically with range Z_i, thus preventing the use of depth observations from far objects. The 3D sensor noise of RGB-D cameras can be modeled with a zero-mean multivariate Gaussian distribution whose covariance matrix has the following as diagonal components [20],

$$\sigma_{11}^2 = \tan\left(\frac{\beta_x}{2}\right) Z_i, \quad \sigma_{22}^2 = \tan\left(\frac{\beta_y}{2}\right) Z_i, \quad \sigma_{33}^2 = \eta(Z_i)^2, \quad (10)$$

where the σ_{33}^2 is directed along the ray, and β_x and β_y denote the angular resolutions in x and y directions. Figure 3 describes the RGB-D camera model we use in the paper. Note that the error in the ray direction increases quadratically.

Therefore, each 3D point $\mathbf{p}_i$ in Figure 2 is associated with a Gaussian distribution as shown in Figure 4,

$$p(\mathbf{p}_i) = \mathcal{N}(\bar{\mathbf{p}}_i, \Sigma_i) \quad , \quad (11)$$

where $\Sigma_i = \mathbf{R}_{\text{ray}} \, \text{diag}(\sigma_{11}^2, \sigma_{22}^2, \sigma_{33}^2) \mathbf{R}_{\text{ray}}^\intercal$, and $\mathbf{R}_{\text{ray}}$ denotes the rotation matrix between the ray and camera coordinates.

4.2 Uncertainty Propagation

Recall that the photometric and geometric errors in Eq. (1) are functions of a 3D point $\mathbf{p}_i$. Therefore, we can propagate its uncertainty to the residuals by using linearization. Then, the likelihood function can be expressed as a Gaussian distribution,

$$p(\mathbf{r}_i \mid \xi) = \mathcal{N}(\mathbf{0}, \mathbf{S}_i) \quad , \quad (12)$$

where

$$\mathbf{S}_i = \mathbf{P}_i \Sigma_i \mathbf{P}_i^\intercal + \text{diag}\left(0, [\Sigma_i']_{3,3}\right) \quad , \quad (13)$$

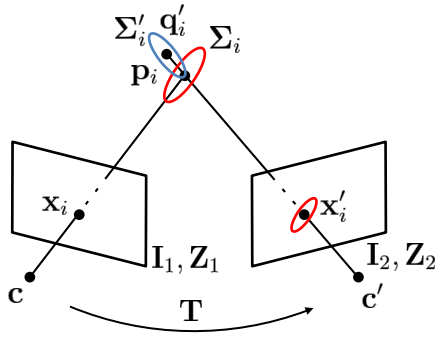


Figure 4: Uncertainty propagation. The uncertainties of back-projected 3D points $\mathbf{p}_i$ and $\mathbf{q}_i'$ are modeled with Gaussian distributions whose covariance matrices are Σ_i and Σ_i', respectively.

$$\mathbf{P}_i^\intercal = \begin{bmatrix} \nabla r_i^I & \nabla r_i^Z \end{bmatrix} = \begin{bmatrix} \frac{\partial r_i^I}{\partial \mathbf{p}_i} & \frac{\partial r_i^Z}{\partial \mathbf{p}_i} \end{bmatrix} \quad . \quad (14)$$

Here, $[\Sigma_i']_{3,3}$ denotes the variance of the back-projected point $\mathbf{q}_i'$ in the z axis of the current camera coordinates as shown in Figure 4.

Then, the maximum likelihood estimation of Eq. (5) can be rewritten as

$$\hat{\xi}_{\text{Sensor}} = \underset{\xi}{\text{argmin}} \sum_{i=1}^{n} \mathbf{r}_i^\intercal \mathbf{S}_i^{-1} \mathbf{r}_i \quad . \quad (15)$$

Note that the single covariance matrix Σ in Eq. (5) is replaced with the individual covariance matrices $\mathbf{S}_i$ in Eq. (12).

Considering the measurement unit difference, we split the individual precision matrix as two square roots $\mathbf{S}_i^{-1} = \mathbf{S}_i^{-1/2} \mathbf{S}_i^{-1/2}$ and normalize it by applying the single precision matrix of the weighted residuals Λ as

$$\hat{\xi}_{\text{Sensor}} = \underset{\xi}{\text{argmin}} \sum_{i=1}^{n} \mathbf{r}_i^\intercal \mathbf{S}_i^{-1/2} \Lambda \mathbf{S}_i^{-1/2} \mathbf{r}_i \quad . \quad (16)$$

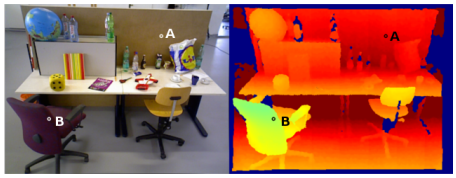
Note that this corresponds to the case where $\mathbf{W}_i = \mathbf{S}_i^{-1/2} \Lambda \mathbf{S}_i^{-1/2}$ in Eq. (2).

4.3 Hybrid Approach

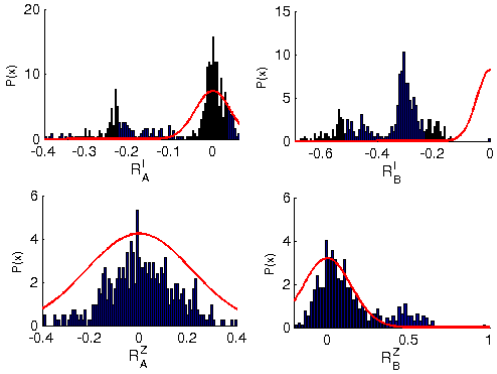
We approximated the non-linear residual functions of Eq. (1) as linear functions around the means using the first-order Taylor expansion and propagated its uncertainty to the residuals. However, this approximation does not hold if the residual functions are highly non-linear. Hence, we performed non-linearity analysis using a Monte Carlo method.

Figure 5(a) shows two representative points A and B from a captured image. For each point, we generated 500 samples based on the associated Gaussian distribution of Eq. (11). Then, given the ground truth camera pose, the samples were (1) back-projected, (2) transformed, and (3) projected onto the next image frame as we did in Figure 2. Figure 5(b) depicts the histograms of photometric residuals (upper row) and geometric residuals (lower row) of the samples for the point A (left column) and B (right column). The propagated Gaussian distributions of Eq. (12) were displayed in red. From this analysis, we observe that the linearization of the depth residual is valid while the linearization of the intensity residual is not. This can be explained with the fact that the texture variation in a natural scene is random, but the geometry varies gradually except in some extreme cases like sharp objects.

Therefore, applying the sensor noise model to both photometric and geometric errors degrades the accuracy of the estimated camera poses. Hence, we propose to combine the previous approach of using t-distribution for photometric errors and the sensor noise model



(a) Two representative points A and B are picked in a captured image. For each point, 500 samples were generated based on the sensor noise model for the analysis in (b).



(b) Given the ground truth of the relative camera pose, histograms of the intensity residuals (upper row) and depth residuals (lower row) for the point A (left column) and B (right column) are depicted. The propagated uncertainties of Eq. (12) are colored in red and overlaid on corresponding histograms.

Figure 5: Linearlization analysis. The linearization of the depth residuals is valid, while the linearization of the intensity residuals is not.

of a Gaussian distribution for geometric errors. From Eq. (8) and (15), the maximum likelihood of the hybrid approach is

$$\hat{\xi}_{\text{Hybrid}} = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{n} \mathbf{r}_i^{\mathsf{T}} \mathscr{W}_i^{1/2} \Lambda \mathscr{W}_i^{1/2} \mathbf{r}_i \ , \tag{17}$$

where the weight matrix $\mathscr{W}_i = \text{diag}(\omega_i^I, \omega_i^Z)$, and σ_I^2 is the intensity variance of t-distribution,

$$\omega_i^I = \frac{\nu+1}{\nu + \left(\frac{r_i^I}{\sigma_I}\right)^2} \ , \tag{18}$$

$$\omega_i^Z = \frac{1}{\nabla r_i^Z \Sigma_i^{-1} \nabla r_i^{Z\mathsf{T}} + [\Sigma_i']_{3,3}} \ . \tag{19}$$

Note that this corresponds to the case where $\mathbf{W}_i = \mathscr{W}_i^{1/2} \Lambda \mathscr{W}_i^{1/2}$ in Eq. (2). This hybrid approach is called σ-DVO.

Figure 6 shows the comparison of depth weights between DVO and σ-DVO. The sensor noise based weights have smaller values for objects which are far away, in agreement with the fact that distant objects are more noisy and should be weighted less.

5 Back-end of Visual SLAM

The visual odometry presented in the previous section, provides a pairwise transformation estimate between two image frames. Thus, an incremental odometry estimate can be calculated with respect to the origin. As a result of incremental nature, the estimation errors can accumulate over time, leading to a globally inconsistent camera trajectory. Hence, to reduce the problem of camera drifting, the idea of keyframes [1] is used.

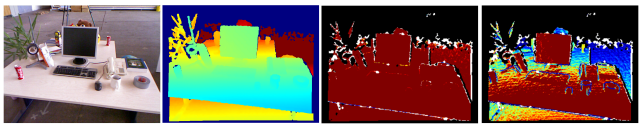


Figure 6: (a) RGB image, (b) Depth image, (c) t-Distribution weights (DVO), (d) Sensor noise based weights (σ-DVO). Observe that the sensor noise based weight decreases as the depth increases.

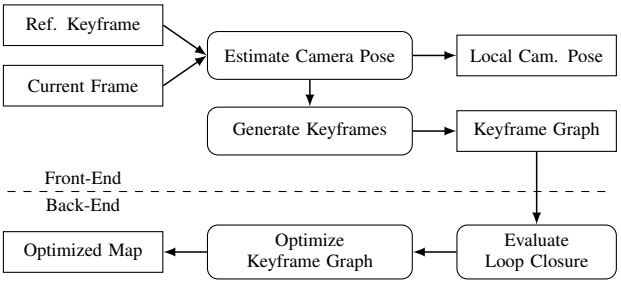


Figure 7: Block diagram of our σ-DVO SLAM system

Figure 7 presents the complete SLAM system consisting of the front-end and back-end. The front-end uses the visual odometry approach to generate a local odometry estimate. Additionally, the front-end also generates keyframes K_i based on the quality of the odometry estimate. The back-end, on the other hand, creates a graph $G \subset \{K_i\}$ using the keyframes generated by the front-end. Additional constraints based on the return (loop closure) to previously visited locations are added to the graph to improve its connectivity. The final graph is optimized with additional constraints to produce the final trajectory.

5.1 Keyframe Generation

The accuracy of the final odometry estimate can be improved by incorporating the keyframe based SLAM back-end. As shown in Figure 8, instead of estimating the camera pose between two consecutive frames f_{n-1} and f_n, the nearest keyframe K_1 and current frame f_n are used in dense visual odometry to reduce camera drifting. When the current keyframe does not contain sufficient information to track, a new keyframe is generated. In our approach, we use two criteria to generate new keyframes. Firstly, we adopt the strategy suggested in [10] for dense SLAM to use the entropy of the camera pose estimate. This strategy generates a new keyframe when the estimated entropy between the keyframe K_i and the current frame f_n falls below a threshold normalized by the largest estimated entropy in the neighborhood. The largest estimated entropy is assumed to be the one between the keyframe K_i and the first frame after K_i.

However, as this scaling is adaptive, it can yield very poor keyframes near turns which have largely changing scenes. Hence, we suggest an additional keyframe generation strategy based on the curvature of the camera trajectory. The curvature $\rho_{i,j}$ between frames i and j is defined as the ratio of the sum of the translation between the frames in the local neighborhood with respect to the translation between the keyframe and the latest frame,

$$\rho_{i,j} = \frac{\Sigma_{i \in N}^{j} \delta_{i,i-1}}{\delta_{i,j}} \ . \tag{20}$$

5.2 Loop Closure

Loop closure provides soft constraints in the graph optimization problem. After optimization, the pose graph is adjusted based on

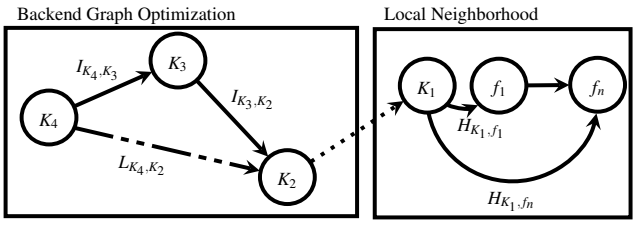


Figure 8: Keyframe graphs. Local neighborhood contains only the recent keyframe K_1 and the frames tracked with respect to it, $(f_1, ..., f_n)$. The keyframes are determined based on the ratio of entropies $H_{K_1,f_1}/H_{K_1,f_n}$. In the back-end the loop constraints L_{K_i,K_j} combined with odometry constraints weighted by I_{K_i,K_j} is optimized.

the edge weights of different constraints in the graph. An erroneous loop constraint can lead to a poorly optimized trajectory. Extending previous loop constraint generation methods such as [10], two additional techniques are used to reduce the impact of wrong loop constraints. Firstly, the loop closure constraints are weighted based on the inverse square of the metric distance between the keyframes that form the loop closure. This is based on the intuition that a loop constraint between far frames is prone to a larger error than frames close to each other. Secondly, occlusion filtering is performed to remove false loop closure constraints. With the geometry information from the depth images, we can filter out points that exceed the maximum possible depth shift. Besides loop closure, the occlusion filtering is also applied in the front-end to improve per frame odometry precision.

5.3 Graph Optimization

On generation of a new keyframe, the back-end graph is updated with the previous keyframe information and a double window graph structure similar to [22] is created as shown in Figure 8. The pose graph in the back-end is optimized using an open source library, g2o [12]. Most of the map representations such as a signed distance [17] or surfels [23] require higher computational demands. Hence, no explicit map representation was performed. A final optimization on the termination of the visual odometry is performed to generate optimized camera trajectory. The σ-DVO algorithm with the improvements in the backend is called σ-DVO SLAM.

6 Experimental Results

Our σ-DVO SLAM algorithm was implemented in a multi-threaded C++ framework. The framework was built on top of the tools provided in the open source implementation of DVO SLAM [1]. All the evaluations were carried out on a workstation with Intel Xeon E5 @ 2.4GHz.

The RGB-D benchmark provided by Technical University of Munich (TUM) [24] is used to evaluate the performance of our algorithm. Since σ-DVO SLAM is intended for indoor AR applications, hand-held SLAM datasets are used. To evaluate the drift in visual odometry we calculate the root mean squares of relative pose errors (RPE, $[m/s]$). Additionally, to evaluate the trajectory error in the complete SLAM system we calculate the root mean squares of absolute tracking errors (ATE, $[m]$). We follow the same definition provided in [24].

6.1 t-Distribution vs. Sensor Noise Model

DVO uses t-distribution on the photometric and geometric residuals to reduce the impact of outliers. In this section, we compare the performance of the linearized sensor noise model as described in Section 4.2 against the existing t-distribution approach. Photometric and geometric residuals are considered separately. A smaller RPE indicates better visual odometry.

Dataset	Intensity Only		Depth Only	
	t-dist.	Linearized	t-dist.	Linearized
fr1/360	**0.167**	0.351	**0.347**	1.780
fr1/desk	**0.058**	0.111	0.282	**0.218**
fr1/desk2	**0.118**	0.119	**0.292**	0.985
fr1/floor	**0.077**	0.104	**0.173**	0.196
fr1/room	**0.085**	0.134	0.356	**0.061**
fr1/rpy	**0.055**	0.079	**0.290**	1.194
fr1/xyz	**0.047**	**0.047**	0.178	**0.050**
fr2/desk	0.042	**0.033**	0.332	**0.057**
fr2/loop	**0.631**	1.052	0.875	**0.584**
fr2/rpy	**0.032**	0.037	0.241	**0.029**
fr2/xyz	0.021	**0.014**	0.191	**0.017**
fr3/office	**0.023**	0.052	0.381	**0.067**

Table 1: Comparison of visual odometry using t-distribution (t-dist.) and the linearized sensor noise model in RPE. In general, the t-distribution performs better for intensity residuals but linearized sensor noise model performs better for geometric residuals.

From Table 1 we observe that there is a reduction in visual odometry drift for geometric residual by using the linearized sensor noise model. The linearized sensor noise model under-performs t-distribution for intensity residuals due to the poor linearization of the sensor noise model. This confirms our linearization analysis in Figure 5.

6.2 DVO vs. σ-DVO in Various Datasets

In this section, the t-distribution based DVO [10] is compared against our approach, σ-DVO. The root mean squared errors in both RPE and ATE are presented. Table 2 shows that our method σ-DVO performs significantly better than t-distribution based DVO in all the datasets. On average, σ-DVO has a 70% reduction in ATE and a 25% reduction in RPE, compared to DVO.

Dataset	DVO		σ-DVO	
	ATE	RPE	ATE	RPE
fr1/360	0.415	0.153	**0.229**	**0.110**
fr1/desk	0.109	0.048	**0.067**	**0.039**
fr1/desk2	0.261	0.074	**0.088**	**0.065**
fr1/floor	0.242	0.070	**0.226**	**0.053**
fr1/room	0.459	0.092	**0.314**	**0.063**
fr1/rpy	0.216	0.065	**0.072**	**0.046**
fr1/xyz	0.102	0.050	**0.052**	**0.036**
fr2/desk	0.561	0.038	**0.184**	**0.016**
fr2/large (with loop)	4.370	0.240	**0.724**	**0.134**
fr2/rpy	0.501	0.039	**0.188**	**0.012**
fr2/xyz	0.497	0.030	**0.188**	**0.010**
fr3/office	0.485	0.044	**0.164**	**0.014**
average	0.684	0.067	**0.208**	**0.050**

Table 2: Comparison between DVO and σ-DVO in various datasets. Our σ-DVO outperforms DVO in all datasets.

6.3 DVO vs. σ-DVO in Different Environment Types

To evaluate the robustness of our approach, we compare its performance on datasets with varying scene texture, geometry and distance. The different environment dataset provided in TUM benchmark was used for this evaluation. As shown in Table 3, our σ-DVO

[1] https://github.com/tum-vision/dvo_slam

outperforms DVO under most scene conditions (18% reduction in RPE). The increased RPE in near datasets is due to occlusion in the scene that was not handled correctly. This has been addressed in σ-DVO SLAM.

Dataset			DVO	σ-DVO
structure	texture	distance		
no	no	far	0.109	**0.041**
no	no	near	**0.142**	0.156
no	yes	far	0.058	**0.046**
no	yes	near	0.025	**0.016**
yes	no	far	0.068	**0.024**
yes	no	near	**0.023**	0.139
yes	yes	far	0.094	**0.014**
yes	yes	near	0.039	**0.013**
average			0.069	**0.056**

Table 3: Comparison between DVO and σ-DVO in RPE for different environment types. Our σ-DVO outperforms DVO in most of the types, but underperforms only in near datasets.

6.4 DVO SLAM vs. σ-DVO SLAM

A comparison against DVO SLAM will provide a clearer measure of σ-DVO SLAM performance. In this section we perform comparison in the number of keyframes generated and ATE of both approaches. Ideally smaller number of keyframes with reduced ATE would indicate a more desirable approach.

Dataset	DVO SLAM		σ-DVO SLAM	
	#KF	ATE	#KF	ATE
fr1/desk	67	0.021	**58**	**0.019**
fr1/desk2	93	0.046	**73**	**0.037**
fr1/room	186	**0.053**	**132**	0.060
fr1/360	126	0.083	**102**	**0.061**
fr3/office	168	0.053	**151**	**0.015**
average	-	0.051	-	**0.038**

Table 4: Comparison of DVO SLAM and σ-DVO SLAM in the number of keyframes (#KF) and ATE.

Table 4 shows that σ-DVO SLAM has an 18.6% reduction in the number of keyframes generated. This is due to the reduced drift in the σ-DVO front-end. At the same time, σ-DVO SLAM produces a 25% reduction in ATE.

The fr3/office dataset in the TUM benchmark is a large dataset with loop closure, which is a suitable candidate to showcase the impacts of drift. Figure 9 shows the comparison of the trajectory generated by our approach with respect to DVO SLAM for the fr3/office dataset. The trajectory generated by our approach closely matches the ground truth and has an ATE of 0.015. As shown in Figure 10, DVO SLAM finds it challenging to track the ground truth trajectory near turns, while σ-DVO SLAM is very close to the ground truth even near turns.

6.5 Comparison with State-of-the-art SLAM

In this section we evaluate the performance of σ-DVO SLAM against the state-of-the-art SLAM algorithms. MRSMap [23] is a surfel based SLAM algorithm that relies on a global map structure for optimization. RGB-D SLAM [5] is a feature based SLAM approach. Both Kintinuous [27] and ElasticFusion [28] are computationally intensive SLAM algorithms that have GPU based implementation. As shown in Table 5, our σ-DVO SLAM performs better than the state-of-the-art SLAM algorithms.

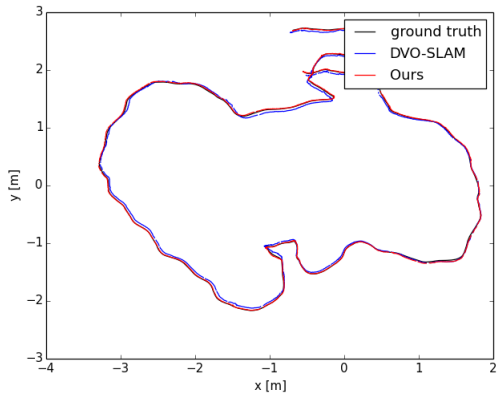


Figure 9: The trajectory of DVO SLAM plotted against the trajectory generated by σ-DVO SLAM for the fr3/office dataset. Our σ-DVO SLAM aligns very closely with the ground truth.

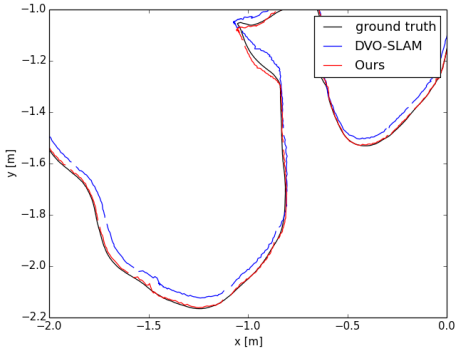


Figure 10: A section of the turn in Figure 9 is highlighted to analyze the trajectory mis-match. DVO SLAM has significant offsets with respect to the ground truth, while our σ-DVO SLAM tracks the ground truth very closely.

6.6 Performance with Intel RGB-D Sensor

In order to test the scalability of our algorithm to different RGB-D cameras, we perform an experiment using a custom dataset collected in a cubicle with Intel RealSense R200. Though Kinect and R200 share similar attributes, they have different working principles and noise characteristics. R200 has significantly higher depth noise and a reduced field of view compared to Kinect. Since we do not have any ground truth measurement for the camera trajectory, we present qualitative results on the performance of σ-DVO SLAM. Figure 11 provides a top view of the map generated by σ-DVO SLAM. The straight walls with right angle corners are preserved in the final point cloud, indicating that an accurate mapping was performed.

The overlap between first frame and last frame can provide a qualitative analysis of drift in the SLAM system. We compare our σ-DVO SLAM against DVO SLAM in Figure 12. With the results of DVO SLAM, we can observe that there is a mis-match on the edges of the whiteboard. This indicates that there is a significant drift in DVO SLAM, which is possibly attributed to the larger noise in R200 compared to Kinect.

6.7 Augmentation Results

In this section we present the augmentation results using σ-DVO SLAM. A qualitative comparison between σ-DVO SLAM and DVO SLAM is presented in Figure 13. We observe that DVO

Algorithm	fr1/desk	fr2/xyz	fr3/office	fr1/360
DVO SLAM [9]	0.021	0.018	0.035	0.083
RGB-D SLAM [5]	0.023	**0.008**	0.032	0.079
MRSMap [23]	0.043	0.020	0.042	0.069
Kintinuous [27]	0.037	0.029	0.030	-
ElasticFusion [28]	0.020	0.011	0.017	-
σ-DVO SLAM	**0.019**	0.018	**0.015**	**0.061**

Table 5: Comparison of the our approach with respect to the existing state-of-the-art SLAM approaches in ATE. Our σ-DVO SLAM performs better in most of the datasets.

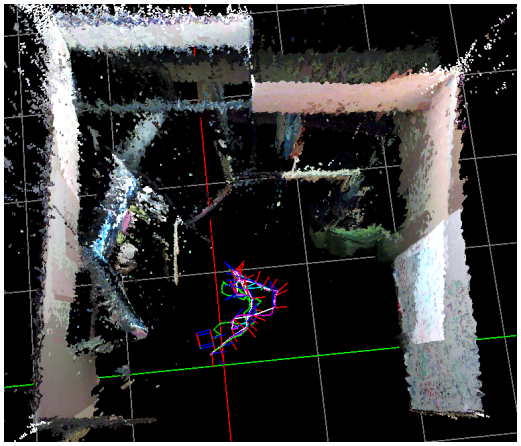


Figure 11: Top view of the map generated by σ-DVO SLAM in a cubicle. The right angle wall indicates that the map is consistent with the real world.

SLAM has more drift in structured but textureless regions. Our σ-DVO SLAM shows more accurate tracking even in such environments due to its ability to make better use of the depth information.

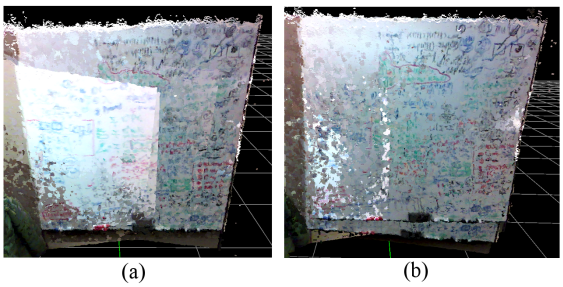
7 Discussion

Based on our evaluation we see that our σ-DVO has better accuracy than existing dense visual odometry algorithms. This improvement in the front-end has a significant impact on the accuracy of the complete σ-DVO SLAM system. We believe that existing RGB-D SLAM algorithms can greatly benefit from the use of the RGB-D sensor noise model. Also using a sensor noise model rather than robust estimators such as t-distribution provides a physical meaning to the residual errors.

The limitations of our current approach lie in the use of linearized propagation of the sensor noise model. Though linearization provides a closed form solution, it fails to provide reasonable approximations for photometric errors. One of the possible improvements is using a nonlinear propagation method such as unscented transform. Also, the current SLAM system does not merge the uncertainty information into a global map. we can improve it by performing more tight integration of the sensor noise model into the back-end of the SLAM algorithm. Finally, our work does not address the issue of re-localization that is essential in AR applications.

8 Conclusion

In this paper we have presented a novel method called σ-DVO for dense visual odometry using a probabilistic sensor noise model. The linearized sensor noise model was incorporated into the optimization framework of dense visual odometry. Based on the insights gained from the linearized propagation of the sensor noise



(a) (b)

Figure 12: Comparison of the whiteboard on the right bottom of Figure 11. The first and the last frames are overlapped to observe how much drift has occurred. (a) The result of our σ-DVO SLAM. The change in color is due to the change in exposure of the camera, but there is no significant drift. (b) The result of DVO SLAM. Observe the mis-match on bottom edges of the whiteboard.

model, we introduced a hybrid visual odometry approach.

We compared the performance between σ-DVO and DVO using the TUM RGB-D dataset. Experimental results show that σ-DVO is more robust to different scene conditions, has a 70% reduction in absolute tracking errors, and a 25% reduction in relative pose errors, compared to DVO. σ-DVO was extended to σ-DVO SLAM which incorporates keyframe generation, loop constraint detection and pose optimization. In comparison with DVO SLAM, σ-DVO SLAM has a 25% reduction in absolute trajectory errors with a 19% reduction in the number of keyframes required. Our σ-DVO SLAM outperforms the state-of-the-art approaches in visual SLAM. In order to check the scalability to different sensors, we applied σ-DVO SLAM to a custom dataset collected with Intel RealSense R200. Qualitative analysis on the dataset indicates that σ-DVO SLAM outperforms DVO SLAM with a reduced drift.

References

[1] M. Agrawal and K. Konolige. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5), October 2008.

[2] C. Audras, A. I. Comport, M. Meilland, and P. Rives. Real-time dense appearance-based slam for rgb-d sensors. In *Robotics and Automation, 2011 Australasian Conference on*, Dec 2011.

[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[4] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410 vol.2, Oct 2003.

[5] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the rgb-d slam system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1691–1696, May 2012.

[6] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[7] D. Gutirrez-Gmez, W. Mayol-Cuevas, and J. J. Guerrero. Inverse depth for accurate photometric and geometric error minimisation in rgb-d dense visual odometry. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 83–89, May 2015.

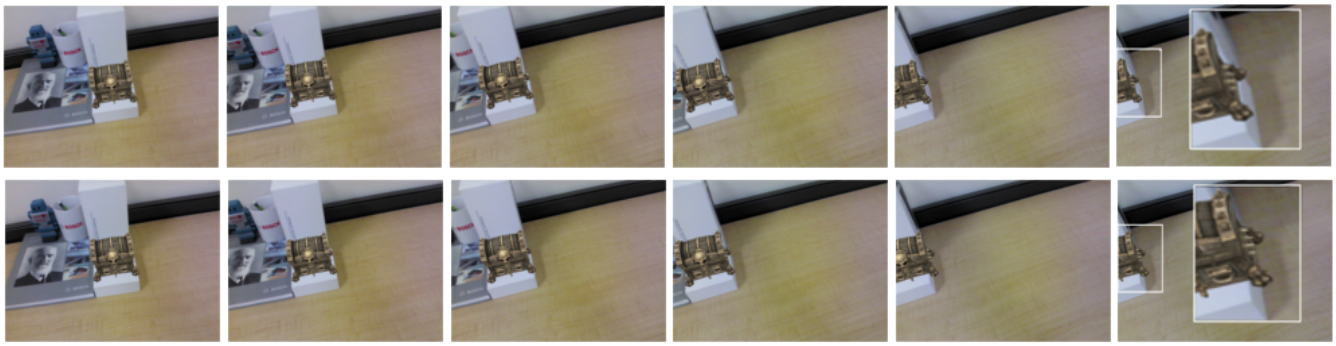


Figure 13: Comparison of the augmentation results between our σ-DVO SLAM (top) and DVO SLAM (bottom). DVO SLAM results in more drift than σ-DVO SLAM as the camera moves to the textureless area.

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[9] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *Int. Conf. on Robotics and Automation*, May 2013.

[10] C. Kerl, J. Sturm, and D. Cremers. Dense visual slam for rgb-d cameras. In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2013.

[11] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234, Nov 2007.

[12] R. Kmmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613, May 2011.

[13] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, pages 1442–1447 vol.3, Nov 1991.

[14] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186, 2007.

[15] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016.

[16] M. J. M. M. Mur-Artal, Raúl and J. D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[17] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proc. of the IEEE Int. Symposium on Mixed and Augmented Reality*, 2011.

[18] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327, Nov 2011.

[19] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. Numerical recipes in c: the art of scientific programming. *Section*, 10:408–412, 1992.

[20] M. Ruhnke, R. Kmmerle, G. Grisetti, and W. Burgard. Highly accurate 3d surface models by sparse surface adjustment. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 751–757, May 2012.

[21] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[22] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double window optimisation for constant time visual slam. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2352–2359, Nov 2011.

[23] J. Stückler and S. Behnke. Multi-resolution surfel maps for efficient dense 3d modeling and tracking. *Journal of Visual Communication and Image Representation*, 25(1):137–147, Jan. 2014. ISSN 1047-3203.

[24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[25] P. Sturm and B. Triggs. *Computer Vision ECCV '96: 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings Volume II*, chapter A factorization based algorithm for multi-image projective structure and motion, pages 709–720. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.

[26] T. Whelan, M. Kaess, J. J. Leonard, and J. McDonald. Deformation-based loop closure for large scale dense rgb-d slam. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 548–555, Nov 2013.

[27] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. Mcdonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *International Journal of Robotics Research*, 34(4-5):598–626, Apr. 2015.

[28] T. Whelan, S. Leutenegger, R. S. Moreno, B. Glocker, and A. Davison. Elasticfusion: Dense slam without a pose graph. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.