Real-time Rotation Estimation for Dense Depth Sensors in Piece-wise Planar Environments

Yi Zhou, Laurent Kneip and Hongdong Li

***Abstract*— Low-drift rotation estimation is a crucial part of any accurate odometry system. In this paper, we focus on the problem of 3D rotation estimation with dense depth sensors in environments that consist of piece-wise planar structures, such as corridors and office rooms. An efficient mean-shift paradigm is developed to extract and track planar modes in the surface normal vector distribution on the unit sphere. Robust and piece-wise drift-free behavior is achieved by registering the bundle of planar modes from the current frame with respect to a reference frame using a general ℓ_1-norm regression scheme. We furthermore add a memory scheme to the regular birth and death of modes, which further compensates accumulated rotational drift when previously discovered modes are revisited. We discuss the robustness issue and evaluate our algorithm on both custom synthetic as well as real publicly available datasets. Our experimental results demonstrate high robustness and effectiveness of the proposed algorithm.**

I. INTRODUCTION

3D depth sensors such as RGB-D cameras are a popular alternative to classical cameras for the purpose of autonomous navigation and robotic perception. Active sensors are particularly advantageous when it comes to structures with homogeneously colored surfaces, textureless environments, or even operation in darkness. The point clouds produced by these sensors come in metric scale. They can be used directly to perform point registration via the iterative closest point method (ICP) [1], thus resulting in motion estimation in absolute scale. However, ICP-based motion estimation is either too easy to get trapped in local minima, or too computationally expensive to meet the requirements of real-time application. Considering the fact that rotational drift is an important part of the inaccuracy of position estimation, the goal of this paper is to develop an efficient and piece-wise drift-free 3D rotation estimation method for RGB-D cameras operating in man-made environments.

Our approach relies on surface normal vectors, which can be extracted directly from point clouds, and convey rich geometric information for applications like scene segmentation and object classification [2], structure and pose estimation [3], [4], and even grasping or manipulation [5]. Normal vector distributions typically contain a special structure due to the vast availability of planar surfaces in man-made environments. These structural regularities notably lead to modes in the normal vector density distribution.

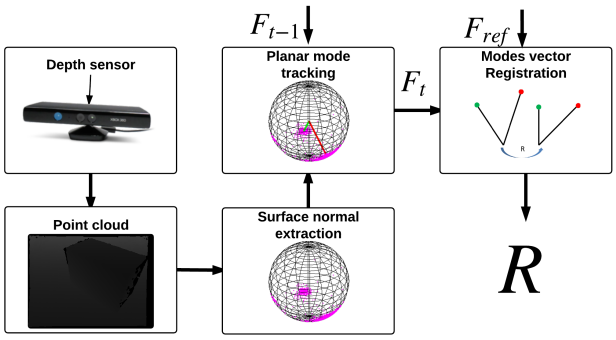


Fig. 1. Overview of the proposed 3D rotation estimation algorithm for depth cameras in piece-wise planar environments.

Rotation estimation for depth cameras by exploiting the organized structure of surface normal vector distributions has been studied previously. However, existing works are limited to either strict Manhattan World (MW) environments [6], [7] or the further relaxed Mixture of Manhattan Frames (MMF) case [8]. Following the idea of [7], [8], we also exploit surface normal vector distributions, but extend it to the more general case of piece-wise planar environments with arbitrary pieces of slanted planes.

The contribution of this paper is three-fold:

- Assuming that there are several dominant planes in the environment, we present a non-parametric method for discovering and tracking planar modes in the density distribution of the surface normal vectors. It is a mean-shift algorithm that operates on the unit sphere, and avoids the need of estimating the parameters of a complete explicit model of the density distribution function.
- Second, we present a robust and piece-wise drift free rotation estimation method which solves the joint registration of pairs of corresponding planar modes in a general ℓ_1-norm regression scheme. This algorithm works robustly with up to 50% of badly tracked modes.
- We introduce a basic memory scheme that remembers dying planar modes. We show that the memory is capable of further compensating drift when previously visited planar structures are reobserved. This functionality has similarities with loop closures in classical SLAM.

The result is a simple but accurate, robust and highly

All the authors are with the College of Engineering and Computer Science, Australian National University. {yi.zhou, laurent.kneip, hongdong li}@anu.edu.au. Hongdong Li is also with National ICT Australia (NICTA) Canberra Labs. The research leading to these results is supported by the ARC Centre of Excellence for Robotic Vision. The work is furthermore supported by ARC grants DP120103896 and DE150101365.

Yi Zhou acknowledges the financial support from the China Scholarship Council for his PhD Scholarship No. 201406020098. We also acknowledge the help from Juan David Adarve for creating the synthetic dataset.

efficient strategy for online tracking of the rotation of a depth camera. Our paper is organized as follows: We conclude the introduction by reviewing the related work. Section II declares all mathematical notations used in this paper as well as all underlying assumptions. Section III presents the core of our method. Section IV finally gives a performance and robustness analysis on both synthetic and real datasets. We conclude with a summary and a discussion about potential future work.

Related work: Online rotation estimation is related to odometry or motion estimation in general. We limit the discussion to solutions that utilize active sensors such as LIDARs and RGB-D cameras because we only use depth information in this work. The most commonly used method is given by the ICP algorithm [1] which performs registration through iterative minimization of the sum of squared distances between spatial neighbors in two point clouds. Classical ICP based methods are prone to local minima as soon as the displacement increases and thus the point cloud structure is subjected to too intensive changes. In order to tackle situations of large view-point changes, the community has therefore investigated globally optimal solutions to the point set registration problem, such as [9]. These methods are however inefficient and thus not suited for real-time application on CPU. Even the most recent local ICP methods [10], [11] achieve real-time frame rate for sub-VGA resolution only (e.g. 320×240 pixel).

The upcoming of RGB-D cameras has however led to a new generation of 2D-3D registration algorithms that exercise a hybrid use of both depth and RGB information. [12] for instance uses the depth information along with the optimized relative transformation to warp the image from one frame to the next, thus permitting direct and dense photometric error minimization. We evaluate our algorithm on datasets captured by a Microsoft Kinect. We include a comparison of our results to the method presented in [12].

There are some recent works that directly build on top of surface normal vectors. By exploiting the structural regularity of man-made environments, [7] presents a real-time maximum a posteriori (MAP) inference of the local Manhattan Frame (MF). This work heavily relies on GPU resources for a real-time inference of a parametric model, and is furthermore strictly limited to the Manhattan world scenario. More general, non-parametric model estimation is presented in [8], which can handle the arbitrary piece-wise planar case. While strongly related to our work, the method in [8] is more computationally expensive and aims at scene understanding and segmentation rather than accurate rotation estimation.

II. Problem Definition and Prerequisites

Our main assumption is that the environment is static and consists of multiple pieces of planar structures. Under this assumption, the surface normal vectors $\mathbf{N^C} = [\mathbf{n}_1, \dots, \mathbf{n}_M]$ distribute in an organized and distinctive manner on the unit sphere[1]. Given surface normal vectors extracted from point clouds by using the method in [13], our goal is two-fold:

- Discover and keep track of the planar modes $\mathbf{F} := [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N]$ on the unit sphere. $\mathbf{F}$ is a $3 \times N$ matrix which defines a bundle of planar direction vectors $\mathbf{f}_i$. For simplicity, we call $\mathbf{F}$ a *bundle*.
- Estimate the relative rotation $\mathbf{R}$ between the reference and the current frame such that $\mathbf{F}^{cur} \simeq \mathbf{R}\mathbf{F}^{ref}$. $\simeq$ means that the equality is valid up to noise or outliers.

By a reference frame, we understand a frame that is

- the very first frame in the sequence where planar modes are initially discovered.
- a frame further in the sequence selected upon a *bundle update*. During tracking, existing modes may die or new modes may be discovered which leads to a so-called *bundle update*.

III. Normal-vector based Rotation Estimation

The surface normal vectors $\mathbf{N^C}$ of piece-wise planar structures always have some organized distribution on the unit sphere $\mathbb{S}^2$ which can be exploited for tracking the orientation of the depth camera. It is reasonable to assume that these unit vectors $\mathbf{n}_i$ are samples of a probability density function, as they are more likely to be distributed around the normal vectors of the plane pieces. The process of finding these planar direction vectors is therefore equivalent to mode-seeking in this density distribution (i.e. finding local maxima in the density distribution function).

A popular, fast, and notably non-parametric method to seek modes is given by the mean shift algorithm [14]. Given an approximate location for a mode, the algorithm applies local Kernel Density Estimation (KDE) to iteratively take steps in the direction of increasing density. We apply this idea to our unit normal vectors on the manifold $\mathbb{S}^2$ using a Gaussian kernel over conic section windows of the unit sphere. The result is optimal under the assumption that the angles between the normal vectors and their corresponding mode centre have a Gaussian distribution. We track the bundle by simply tracking each individual mode independently. Each mode is tracked by starting from its previous position on the unit sphere. While this means that we allow inter-mode angle variation during tracking the bundle $\mathbf{F}^{cur}$, we follow the mode-tracking by registering the entire bundle with respect to a fixed bundle $\mathbf{F}^{ref}$ in a reference frame, thus avoiding drift-effects.

A. Mean-shift on the unit sphere

The core of our method is a single mean shift iteration for each planar mode given a set of normal vectors on $\mathbb{S}^2$. It works as follows:

- We start by finding all normal vectors that are within a neighbourhood of the considered centre $\mathbf{f}_j$. The range of this neighbourhood is notably defined by the width of our kernel for the KDE. In our case, the window is a conic section of the unit sphere and the apex angle of the cone θ_{window} defines the size of the local window.

[1] Superscript $\mathbf{C}$ denotes that the surface normal vectors are described in the coordinate system of the sensor.

Relevant normal vectors $\mathbf{n}_i$ for mode j need to lie inside the respective cone, and thus pass the condition

$$\angle(\mathbf{n}_i, \mathbf{f}_j) < \frac{\theta_{\text{window}}}{2}. \tag{1}$$

Let us define the index i_j which iterates through all $\mathbf{n}_i$ that fulfill the above condition.

- We then project all contributing $\mathbf{n}_{i_j}$ into the tangential plane at $\mathbf{f}_j$ in order to compute a mean shift. Let $\mathbf{Q}$ represent the rotation matrix that rotates $\mathbf{f}_j$ to $[0,0,1]^T$. $\mathbf{Q}$ can be obtained by

$$\mathbf{Q} = \mathbf{I} + [\mathbf{v}]_\times + [\mathbf{v}]_\times^2 \frac{1-c}{s^2}, \tag{2}$$

where $\mathbf{v} = \mathbf{f}_j \times [0,0,1]^T$, $s = \|\mathbf{v}\|$, $c = \mathbf{f}_j^T [0,0,1]^T$, and $[\mathbf{v}]_\times$ is the skew-symmetric matrix of $\mathbf{v}$. Then

$$\mathbf{n'}_{i_j} = \mathbf{Q}\mathbf{n}_{i_j} \tag{3}$$

represents the normal vectors rotated such that the last coordinate is along the direction of $\mathbf{f}_j$. In order for the distances in the tangential plane to represent proper geodesics on $\mathbb{S}^2$ (or equivalently angular deviations), we apply Riemann exponential map. The rescaled coordinates in the tangential plane are given by

$$\begin{aligned} \mathbf{m'}_{i_j} &= \frac{\sin^{-1}(\lambda)\,\text{sign}(n'_{i_j,z})}{\lambda} \begin{bmatrix} n'_{i_j,x} \\ n'_{i_j,y} \end{bmatrix}, \\ \text{where } \lambda &= \sqrt{n'^2_{i_j,x} + n'^2_{i_j,y}}. \end{aligned} \tag{4}$$

Note that—due to the factor $\text{sign}(n'_{i_j,z})$—this projection has the advantage of correctly projecting normal vectors from either direction sense into the same tangential plane.

- We compute the mean shift in the tangential plane

$$\mathbf{s'}_j = \frac{\sum_{i_j} e^{-c\|\mathbf{m'}_{i_j}\|^2} \mathbf{m'}_{i_j}}{\sum_{i_j} e^{-c\|\mathbf{m'}_{i_j}\|^2}}. \tag{5}$$

c is a design parameter that defines the width of the kernel in the tangential plane. It can be derived from θ_{window}.

- To conclude, we transform the mean shift back onto the unit sphere using the Riemann logarithmic map. The update mode $\mathbf{f}_j^\star$ is finally obtained by compensating the rotation $\mathbf{Q}$.

$$\mathbf{f}_j^\star = \mathbf{Q}^T \overline{\begin{bmatrix} \frac{\tan(\|\mathbf{s'}_j\|)}{\|\mathbf{s'}_j\|} \mathbf{s'}_j^T & 1 \end{bmatrix}}^T, \tag{6}$$

where $\overline{[\cdot]}$ returns the input 3-vector divided by its norm.

B. Robust rotation estimation

Once the new location of each mode of the bundle $\mathbf{F}$ has been tracked, the rotation from the reference frame to the current frame can be obtained by applying a least-squares fitting method [15]. Each mode of $\mathbf{F}^{ref}$ and $\mathbf{F}^{cur}$ is regarded as a 3D point. This reduces the problem to finding a rotation $\mathbf{R}$ that minimizes the cost function

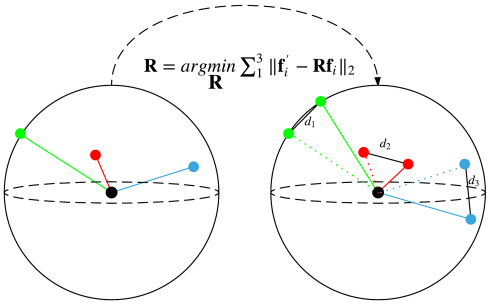


Fig. 2. Illustration of the geometry of the problem. Three modes exist in both the reference view (left) and the current view (right). The chordal distance d_i between each corresponding pair of modes is indicated with a black line segment. The relative rotation from the reference view to the current view is the solution that minimizes the sum of the chordal distances (in a general sense of ℓ_1-norm regression).

$$\begin{aligned} \Sigma^2 &= \sum_{i=1}^{N} (\mathbf{f}_i^{cur} - \mathbf{R}\mathbf{f}_i^{ref})^T (\mathbf{f}_i^{cur} - \mathbf{R}\mathbf{f}_i^{ref}) \\ &= \sum_{i=1}^{N} (\mathbf{f}_i^{curT}\mathbf{f}_i^{cur} + \mathbf{f}_i^{refT}\mathbf{f}_i^{ref} - 2\mathbf{f}_i^{curT}\mathbf{R}\mathbf{f}_i{}^{ref}) \end{aligned} \tag{7}$$

This cost function has a geometric meaning as shown in Fig. 2. Each item of the cost function is the square of the chordal distance between a pair of corresponding modes on the unit sphere. Minimizing Σ^2 therefore is equivalent to finding the closest bundle near $\mathbf{F}^{cur}$ that has same inter-mode angles than $\mathbf{F}^{ref}$, and notably under an ℓ_2-metric (i.e. squared chordal distances).

We apply Arun's method [15]. Minimizing Σ^2 is equivalent to maximizing the third cost term because the previous terms are constant. The original minimization problem therefore turns into maximizing

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^{N} \mathbf{f}_i^{curT}\mathbf{R}\mathbf{f}_i{}^{ref} \\ &= Trace(\sum_{i=1}^{N} \mathbf{R}\mathbf{f}_i{}^{ref}\mathbf{f}_i^{curT}) = Trace(\mathbf{R}\mathbf{H}) \end{aligned} \tag{8}$$

where $\mathbf{H} := \sum_{i=1}^{N} \mathbf{f}_i{}^{ref}\mathbf{f}_i^{curT}$. Let the SVD of $\mathbf{H}$ be $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$. The best rotation matrix is $\mathbf{R} = \mathbf{V}\mathbf{U}^T$. A reflection check is necessary for the case of $det(\mathbf{R}) = -1$. Readers can find the detailed mathematical proof in [15].

For the sake of robustness, we replace the least-squares method with a robust general ℓ_1-norm regression scheme. The new optimization problem becomes

$$\mathbf{R} = \underset{\mathbf{R}}{\text{argmin}} \sum_{i=1}^{n} |\mathbf{f}_i^{cur} - \mathbf{R}\mathbf{f}_i^{ref}| \tag{9}$$

where $|\cdot|$ returns the length of a given vector. The most common tool for solving ℓ_p-norm regression problems with

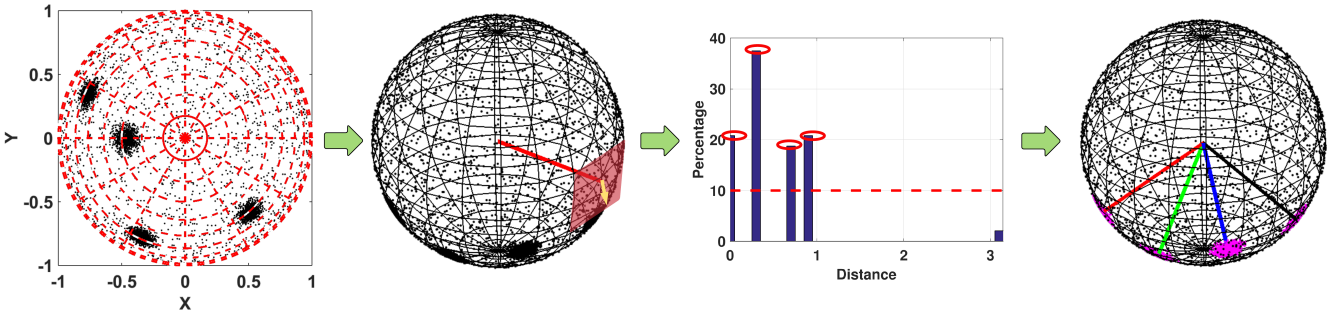


Fig. 3. Initialial mode seeking. The first figure shows the pattern that defines the starting coordinates for the mean-shift clustering. The second figure shows a mean-shift in a tangential plane starting from a given coordinate. The histogram-based non-maximum suppression is shown in the third figure. It splits off mode centres by picking one mode and creating a histogram of rotation distances with respect to all other modes. The final result after non-maximum suppression is shown in the last figure. Four planar modes are found and highlighted with different colors.

an objective function format like Eq. 9 is the iteratively reweighted least squares (IRLS) method [16]. In our case, iterative reweighting is easily done by iteratively finding the rotation matrix $\mathbf{R}_k$ that maximizes

$$\begin{aligned} \mathcal{L} &= Trace(\sum_{i=1}^{N} w_i \mathbf{R}_k \mathbf{f}_i{}^{ref} \mathbf{f}_i^{curT}), \text{ where} \quad (10) \\ w_i &= |\mathbf{f}_i^{cur} - \mathbf{R}_{k-1}\mathbf{f}_i^{ref}|^{-1}. \end{aligned}$$

As this remains a linear problem in each iteration, Arun's method [15] remains applicable. Section IV-B illustrates the benefit of the ℓ_1-extension. The pseudo code of bundle tracking and robust rotation estimation is given in Alg. 1.

C. Initialization and bundle update

We use mean-shift clustering to initialize the algorithm, and thus build on top of our planar mode tracking scheme. The procedure is summarized in Fig. 3. In order to guarantee that the mode-seeking covers the whole space, the unit sphere is divided equally along longitudes and latitudes which gives a set of starting coordinates for the mean-shift tracking. Mean-shift iterations starting from neighboring coordinates may converge to the same mode, which is why we clean the identified set of modes by a histogram-based non-maximum suppression.

New modes may appear or disappear as view-point changes. If the density of surface normal vectors in one mode decreases to less than a designed threshold, the mode is dying and removed from the reference bundle $\mathbf{F}^{ref}$. We find new modes by a mode discovery module, and update the reference bundle $\mathbf{F}^{ref}$ each time a new mode is found[2]. The mode-discovery module continuously monitors the number of surface normal vectors in each cell of the above mentioned grid. If a new mode appears, the number of the surface normal vectors in that direction will grow substantially, thus triggering mean-shift tracking from the center of the cell. Note that this operation is much more expensive than simple mode tracking. We therefore run this monitoring in a separate thread and at a lower frame rate, thus maintaining real-time performance for the actual rotation estimation.

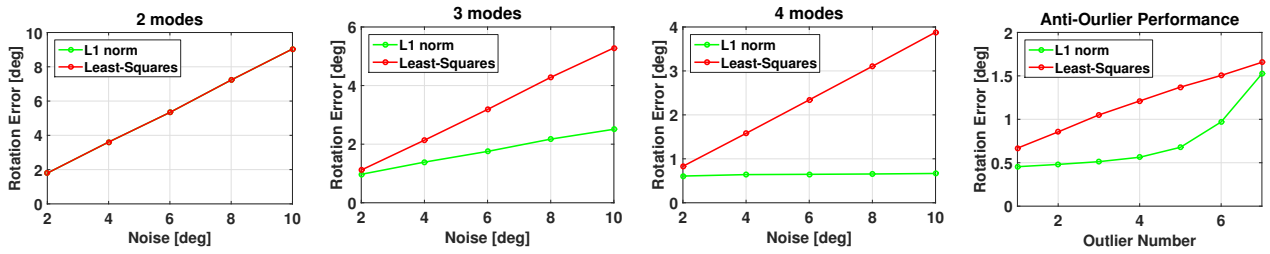
[2]Note that—in order to reduce drift—we simply rotate persisting modes forward rather than replacing them by their tracked equivalent.

Algorithm 1 Bundle tracking and rotation estimation.

function BundleTracking($\mathbf{N^C}$,$\mathbf{F}^{ref}$,$\mathbf{F}^t$)
$\mathbf{F}^{t+1} = \emptyset$
for each $\mathbf{f}_i^t$ **do**
  if $\mathbf{f}_i^t$ is not dying **then**
    $\mathbf{f}_i^{t+1} \leftarrow$ Mean-shift based mode update.
    Push back $\mathbf{f}_i^{t+1}$ to $\mathbf{F}^{t+1}$
  end if
end for
if numel($\mathbf{F^{t+1}}$) < 2 **then**
  return []. ▷ Tracking lost.
end if
$w_i = 1, i = 1, 2, ..., N$ ▷ N = number of mode pairs.
while $\mathbf{R}$ does not converge **do**
  $\mathbf{H} = \sum_{i=1}^{N} w_i \mathbf{f}_i^{ref} \mathbf{f}_i^{t+1\,T}$
  $\mathbf{U_R \Sigma_R V_R^t} \leftarrow$ svd($\mathbf{H}$)
  $\mathbf{R} = \mathbf{V_R U_R}^T$ ▷ Validity Check, see [15].
  $w_i = \frac{1}{\max(\delta, |\mathbf{f}^{t+1} - \mathbf{R}\mathbf{f}^{ref}|)}, i = 1, 2, ..., N$ ▷ δ is a small number
end while
if New born mode appears **then**
  Push back $\mathbf{f}^\star$ to $\mathbf{F^{t+1}}$
  Update $\mathbf{F}^{ref} \leftarrow \mathbf{F^{t+1}}$
end if
return $\mathbf{R}$,$\mathbf{F}^{t+1}$,$\mathbf{F}^{ref}$.
end function

D. Memory function

Instead of simply removing dying modes, we keep forecasting their direction in the current frame using the estimated rotation (even if no normal vectors are currently associated to it). We call the set of inactive modes a mode memory. If a new planar piece is discovered, and the newborn mode is close to an inactive mode in the memory, we reactivate this mode rather than replacing it with a new one. This association compensates drift since the mode became inactive (and notably about the axis that this mode corresponds to). We will see in Section IV-C that this reduces long-term drift.



(a) 2 modes case. (b) 3 modes case. (c) 4 modes case. (d) Anti-outlier performance.

Fig. 4. Robustness of the rotation estimation. (a) (b) and (c) compare the performance of the least-squares and the ℓ_1-norm regression based methods for the case of 2, 3 and 4 modes, respectively. Note that in (a), the red line and the green line coincide with each other. The horizontal axes of (a), (b), and (c) denote the standard deviation of the noise that is imposed on the "badly tracked mode". (d) demonstrates the outlier resilience of the two methods for an increasing outlier fraction (10 modes in total). All the results (rotation error under each noise level and outlier number) are the average of 1000 trials with combination of arbitrary bundle structure and groundtruth rotation.

IV. Experimental Evaluation

Now we proceed to the evaluation of the presented algorithm. We start by explaining the parameter values chosen in our experiments. Then a dedicated simulation experiment is presented showing the importance of the general ℓ_1-norm regression scheme towards the robustness of the rotation estimation. We also test the algorithm on a custom synthetic dataset which demonstrates the piece-wise drift-free property and long-term drift resilience with activated memory function. Finally we evaluate the proposed algorithm on a set of publicly available, real datasets and compare our results directly to another two state-of-the-art depth camera tracking solutions.

A. Parameter configuration

The apex angle of the conic section corresponding to the width of the kernel for the mode tracking is set to 40° during initialization, and 20° during tracking. By using a larger apex angle in initialization, it is more likely that more seeking trials starting from different coordinates in a neighborhood would converge to the same local maximum which will be picked as a mode in the following. The reduction of the cone apex angle in tracking is justified by the assumption that the orientation of the bundle does not change too much under smooth motion. Each iterative mean-shift procedure terminates once the angle between two successive updates falls below a threshold angle θ_{converge}, which we set to 1°. The factor c in Eq. 5 is set to 20. Mean-shift updates are furthermore required to have a minimum number N_{min} of surface normal vectors within the conic window, which is set to 10% of the total number of surface normal vectors. N_{min} is also the threshold for checking dying modes.

B. Simulation experiments

We provide a dedicated simulation to show that our algorithm can work robustly in a situation where some of the modes are badly tracked. The first part of this simulation consists of a series of 3 experiments during which we perform a registration of bundles with 2, 3, and 4 modes. In each experiment, all the modes are perturbed by Gaussian noise. In addition, an elevated amount of noise is added to one of the modes only, which simulates a situation in which the tracking of that particular modes fails. The case of disturbed surface normal vector measurements may happen for various reasons, including heavily inclined planar pieces, a reflection on a smooth surface, or a moving element in the scene. We each time compare the performance of our general ℓ_1-norm regression scheme to that of the original least-squares method in [15]. It can be seen in the Fig. 4(b) and (c) that our method maintains robustness while the original method deteriorates. It is worth noting that the general ℓ_1-norm regression based method cannot help if only two plane pieces are present in the scene (cf. Fig. 4(a)). It is not possible to solve for the rotation with less than two robustly perceived planar modes being observed, as this represents the minimal case.

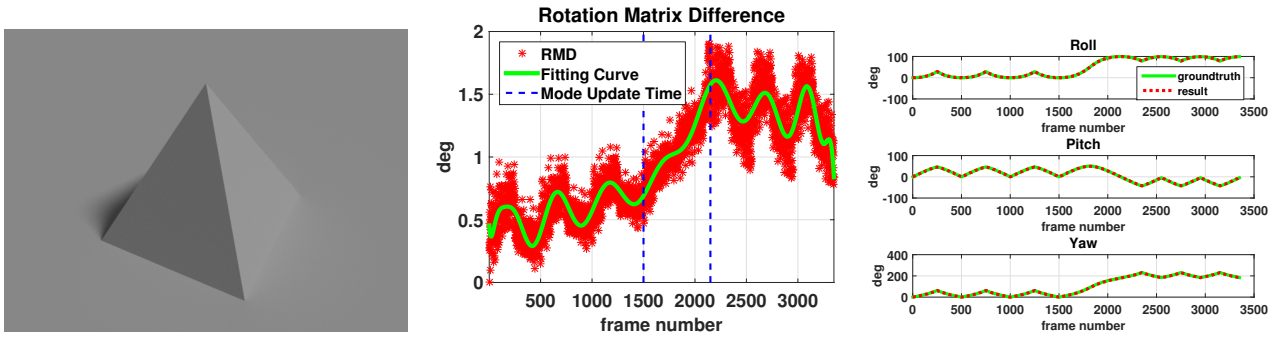
The second part of our simulation experiments is shown in Fig. 4(d), where we register bundles of 10 modes. This experiment evaluates the overall outlier-resilience by perturbing an increasing amount of modes by heavy noise. We compare the performance of our ℓ_1-extension against Arun's original solution. As can be observed, the rotation error stays rather low if at least 50% of the modes are tracked with moderate noise only. This phenomenon confirms the common observation that the ℓ_1-norm scheme can resist up to about 50% of outliers.

C. Evaluation on a synthetic dataset

We created a synthetic dataset using the open-source 3D computer graphics software *Blender* to demonstrate two important properties of our algorithm:

1) Piece-wise drift-free performance between bundle or reference updates.
2) Ability to compensate drift when a previously discovered mode is revisited.

The scene in the dataset is composed of a pyramid with four faces on a ground plane. Two types of sensor motion are added to individually confirm the above two properties. In the first case, the sensor orbits in a back-and-forth fashion around the pyramid while the principal axis of the depth camera keeps pointing towards the centre of the pyramid. In the second case, the sensor orbits smoothly and continuously for



(a) Pyramid. (b) Rotation estimation error. (c) Estimation of roll, pitch and yaw.

Fig. 5. Performance evaluation on the synthetic dataset "Pyramid". (a) shows the synthetic scene which contains a ground plane and the four faces of a pyramid. The rotation estimation error is shown in (b). The estimated roll, pitch, and yaw angles are shown in (c).

several complete loops around the pyramid. The groundtruth depth map and the trajectory of the camera are given each time. Realistic noise is added to the depth map before extracting the surface normal vectors.

The dataset and the results concerning the first property are shown in Fig. 5. The blue dashed lines in Fig. 5 (b) divide the sequence into three parts. They represent the time instants where reference bundle updates happen. We can see that our algorithm returns piece-wise drift free performance in parts 1 and 3 during which no bundle updates happen, meaning that modes are neither dying nor discovered. The drift keeps increasing in the middle part between the dashed lines, where only one planar mode is robustly tracked. As explained in Section IV-B, even the general ℓ_1-norm regression scheme cannot help in this situation because only one planar mode is tracked without gross errors.

The results of the long-term drift experiment are illustrated in Fig. 6. The two subfigures show the rotation estimation performance of the proposed algorithm without and with the mode memory scheme, respectively. In the first figure, the stair-behaviour again shows the piece-wise drift-free performance, however, an accumulated drift over a longer term exists. In the second figure, we can clearly see that the long-term drift stays bounded as soon as at least one of the pyramid surfaces has been revisited for the first time (i.e. after the completion of the first loop).

D. Evaluation on real data

We compare the performance of our method against two state-of-the-art, open-source motion estimation framework for depth cameras, namely DVO [12] and FastICP [10]. All methods are evaluated on two published and challenging benchmark datasets from the ETH RGBD [10], [11] and TUM RGBD [17] series. A qualitative evaluation on the TAMU RGBD [18] dataset is also given (no groundtruth provided). The datasets we picked for evaluation are listed below and the results are summarized in Table I as well as illustrated in Fig 7.

- ETH 1: 0low_0slow_0fly.
- TUM 1: freiburg3_cabinet.
- TUM 2: freiburg4_structure_texture_near.
- TUM 3: freiburg3_structure_notexture_near.
- TUM 4: freiburg3_structure_notexture_far.
- TAMU 1: corridor_A_const.
- TAMU 2: corridor_B_const.

It is necessary to mention that in some cases our algorithm cannot process the entire sequence.This is due to algorithm limitations that are discussed in the following section. In order to remain fair, we evaluate the performance of all algorithms on the same segments of each sequence. We provide root-mean-square (RMS) and median errors $\tilde{e}$ per second for the rotation estimation. The best performing method's error is each time indicated in bold. It can be seen that our method outperforms both FastICP and DVO in most situations. The relatively bad performance of our method on the ETH 1 dataset is related to the low resolution

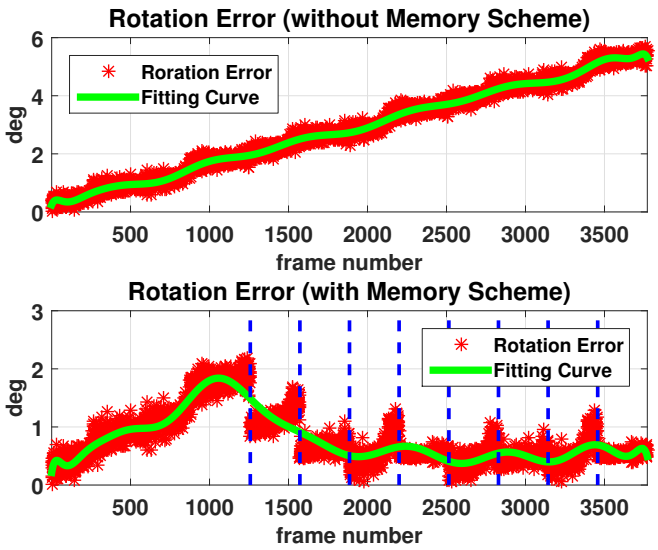


Fig. 6. The rotation estimation performance of the proposed algorithm without and with the mode memory scheme. An obvious step-like curve in the top figure again demonstrates the piece-wise drift-free behavior. The long-term drift compensation is shown in the bottom figure, where the blue dashed lines denote the time instants when planar modes are revisited and accumulated rotational drift gets compensated.

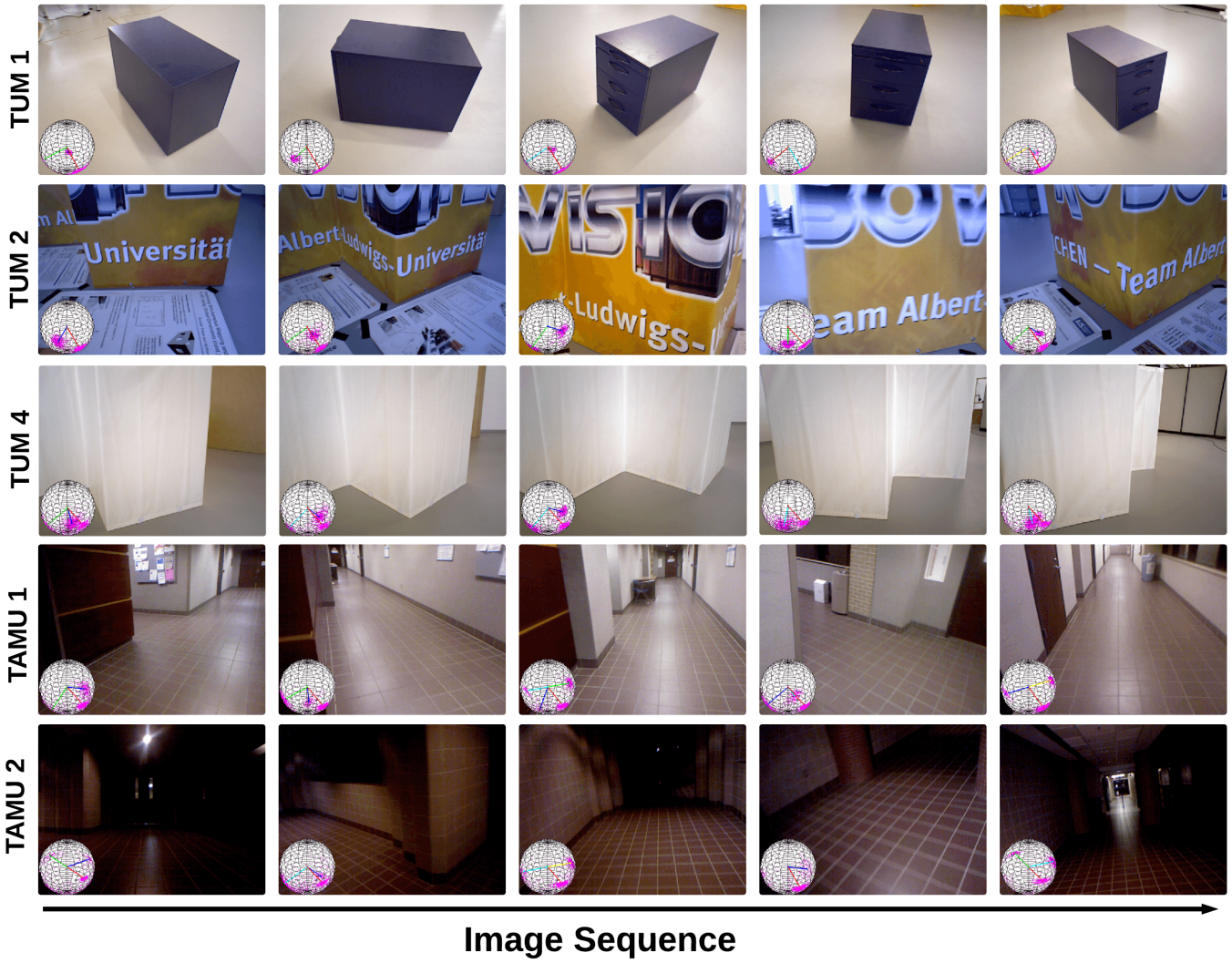


Fig. 7. Illustration of the proposed algorithm running on a set of real datasets. Five short sequences are extracted from each dataset to show the algorithm progress. A unit sphere in the bottom-left corner of each image illustrates the planar mode bundle. Corresponding planes in each image of each sequence are denoted with the same color (e.g. the ground plane is always shown in red). We do not show results of TUM 3 because it has a similar scene as TUM 4. We also don't show images for the ETH 1 dataset because it provides only point clouds.

TABLE I

PERFORMANCE COMPARISON ON SEVERAL INDOOR DATASETS.

Dataset	**DVO**		**FastICP**		**Our Method**	
	$\mathbf{rms}(\mathbf{e_R})$	$\tilde{\mathbf{e_R}}$	$\mathbf{rms}(\mathbf{e_R})$	$\tilde{\mathbf{e_R}}$	$\mathbf{rms}(\mathbf{e_R})$	$\tilde{\mathbf{e_R}}$
ETH 1	×	×	**2.030**	**1.749**	2.892	1.920
TUM 1	4.911	4.456	2.849	1.816	**1.582**	**1.054**
TUM 2	**0.938**	**0.740**	×	×	1.572	1.292
TUM 3	10.898	3.888	8.885	4.920	**1.233**	**0.968**
TUM 4	2.209	1.590	3.674	2.497	**0.983**	**0.683**
Average	4.379	2.669	4.360	2.746	**1.652**	**1.183**

of this dataset, which leads to a low-quality surface normal vector result. DVO returns a slightly better performance on the TUM 4 sequence, in which plenty of distinctive texture can be observed. Missing numbers in Table I indicate that the algorithm was not able to successfully process the sequence. Our method handles most of the cases, and remains computationally efficient even on depth images with VGA resolution. Our real-time C++-implementation processes frames at 50 Hz on a laptop with 8 cores. While DVO is real-time capable as well, FastICP quickly drops in computational efficiency as the number of the points increases, and ultimately operates far from real-time on VGA imagery (1 Hz).

E. Limitations and failure cases

As with any method, the proposed algorithm cannot work in any case. Limitations and failure cases are listed as follows:

- The initialisation takes about 1 s. The sensor should not be subjected to substantial motion during this period.
- When only one planar structure is present or can be recognized, the registration of the planar modes based rotation estimator does not work.
- When two planar modes have a small inscribed angle, the mode seeking may converge to the centre of these two modes and mis-recognize them as a single mode. Such bad initialization can affect the sub-sequent mode tracking iterations as well as the rotation estimation.

V. Discussion

This paper presented a highly efficient 3D rotation estimation algorithm for depth cameras in piece-wise planar environments. It shows that by using surface normal vectors as an input, planar modes in the corresponding density distribution function can be discovered and continuously tracked using efficient non-parametric estimation techniques. The relative rotation from the reference view to the current view can be estimated by registering entire bundles of planar modes. Robustness of the bundle registration process is achieved by performing a general ℓ_1-norm regression instead of simply solving a least-squares problem. Piece-wise drift-free performance is achieved as long as no bundle updates happen. The paper furthermore shows that by introducing a mode memory scheme, drift can be avoided even if certain modes are temporally unobserved. Extensive evaluations on simulated, synthetic and real data demonstrate the robustness and effectiveness of the proposed algorithm. Note that our synthetic dataset as well as our code are ready for public release.

The present work unseals an interesting analogy between classical 6 DoF simultaneous localization and mapping (SLAM) of 3D points, and our 3 DoF rotation estimation scheme which shows that—given surface normal vectors—we are able to perform decoupled, simultaneous orientation estimation and mapping of planar modes. In SLAM, long-term drift is eliminated as soon as the 3D points are no longer updated. This corresponds to our drift-free performance in case the reference bundle stays unchanged. Furthermore, our mode-memory scheme has analogies with loop-closure in SLAM, which is well-known to compensate for long-term drift. The analogy with SLAM suggests immediate directions for interesting future work around efficient normal-vector based, decoupled rotation estimation. For instance, we plan to rely on graph-optimization methods leading to a more accurate, multi-frame mode-initialization procedure. Furthermore, the inclusion of appearance information would robustify the reactivation of modes from the memory even in the presence of more significant drift.

References

[1] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[2] X. Wei, S. L. Phung, and A. Bouzerdoum, "Object segmentation and classification using 3-d range camera," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 74–85, 2014.

[3] J. Glover, G. Bradski, and R. B. Rusu, "Monte carlo pose estimation with quaternion kernels and the bingham distribution," in *Robotics: science and systems*, vol. 7, 2012, p. 97.

[4] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1329–1335.

[5] J. Stückler, R. Steffens, D. Holz, and S. Behnke, "Real-time 3d perception and efficient grasp planning for everyday manipulation tasks." in *ECMR*, 2011, pp. 177–182.

[6] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 941–947.

[7] J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher III, "Real-time manhattan world rotation estimation in 3d," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1913–1920.

[8] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher, "A mixture of manhattan frames: Beyond the manhattan world," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 3770–3777.

[9] J. Yang, H. Li, and Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1457–1464.

[10] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart, "Tracking a depth camera: Parameter exploration for fast icp," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3824–3829.

[11] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[12] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3748–3754.

[13] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using rgb-d cameras," in *RoboCup 2011: Robot Soccer World Cup XV*. Springer, 2012, pp. 306–317.

[14] M. Á. Carreira-Perpiñán, "A review of mean-shift algorithms for clustering," *arXiv preprint arXiv:1503.00687*, 2015.

[15] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 5, pp. 698–700, 1987.

[16] wikipedia.org. Iteratively reweighted least squares. [Online]. Available: https://en.wikipedia.org/wiki/Iteratively_reweighted_least_squares

[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[18] Y. Lu and D. Song, "Robustness to lighting variations: An rgb-d indoor visual odometry using line segments," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 688–694.