Plane-based Odometry using an RGB-D Camera

Carolina Raposo†
http://arthronav.isr.uc.pt/~carolina
Miguel Lourenço†
http://arthronav.isr.uc.pt/~mlourenco
João P. Barreto
http://www.deec.uc.pt/~jpbar
Michel Antunes
http://www.isr.uc.pt/~michel

Institute of Systems and Robotics,
Faculty of Sciences and Technology,
University of Coimbra,
3030 Coimbra, Portugal

Abstract

Odometry consists in using data from a moving sensor to estimate change in position over time. It is a crucial step for several applications in robotics and computer vision. This paper presents a novel approach for estimating the relative motion between successive RGB-D frames that uses plane-primitives instead of point features. The planes in the scene are extracted and the motion estimation is cast as a plane-to-plane registration problem with a closed-form solution. Point features are only extracted in the cases where the plane surface configuration is insufficient to determine motion with no ambiguity. The initial estimate is refined in a photo-geometric optimization step that takes full advantage of the plane detection and simultaneous availability of depth and visual appearance cues. Extensive experiments show that our plane-based approach is as accurate as state-of-the-art point-based approaches when the camera displacement is small, and significantly outperforms them in case of wide-baseline and/or dynamic foreground.

1 Introduction

Visual odometry is the process of estimating the motion of a robot using the input of a single or multiple cameras attached to it. It has important applications in robotics, for control and navigation in the absence of an external reference system. Research has been made in order to tackle this problem using RGB cameras [8, 10]. However, these methods face significant challenges including the reconstruction of textureless regions. RGB-D sensors, such as the Microsoft Kinect and the Asus Xtion Pro Live, cope with this issue since they provide the 3D geometry and the visual appearance of the scene simultaneously.

Recently, odometry methods that take advantage of the depth and color information provided by RGB-D sensors have been developed [9, 13]. They run in real-time and provide accurate estimations for high frame rate acquisitions and moderate sensor velocity. However, they are not able to properly cope with large displacements between consecutive frames. In [13], it has been experimentally shown that the performance of the method degrades as the

† The authors assert equal contribution and joint first authorship.

frame interval increases, which is equivalent to decreasing the frame rate of the acquisition, or increasing the sensor velocity.

We propose a new odometry method which uses both depth and color information for extracting planes from the scene, and the relative pose estimation between consecutive frames is cast as a plane registration problem. In the absence of the minimum number of required planes, 2D point correspondences are extracted for finding the remaining degrees of freedom (DOF). This procedure allows the method to cope with large baselines, since it only requires that there exists a few plane and/or point correspondences.

The algorithm uses an hierarchical scheme for selecting the correspondences, in the sense that it favors plane correspondences, and points are only extracted if strictly necessary, i.e., if the requirement for the minimum number of planes in the scene is not satisfied. This leads to a more robust estimation because extracting point correspondences in the presence of wider baselines is more difficult. As a final step, we perform the refinement of the initial estimation by minimizing the photometric error. The algorithm estimates the entire motion of the sensor using only the information acquired from pairs of consecutive frames, and no prior knowledge is considered.

We validated our approach on three image sequences from a recently proposed dataset [14], as well as on a sequence acquired at high resolution by our Kinect device. For all the sequences, the performance of our approach was compared to the state-of-the-art method presented in [9]. We found that we achieve similar accuracy for small camera displacements, significantly outperforming [9] in the presence of wide baselines.

Notation: Matrices are represented by symbols in sans serif font, e.g. G, and image signals are denoted by symbols in typewriter font, e.g. $\mathtt{I}$. Vectors and vector functions are typically represented by bold symbols, and scalars are indicated by plain letters, e.g. $\mathbf{x} = (x,y)^\mathsf{T}$ and $\mathbf{f}(\mathbf{x}) = (f_x(\mathbf{x}), f_y(\mathbf{x}))^\mathsf{T}$. The symbol $\sim$ denotes an equality up to scale.

2 Related Work

Typical visual odometry systems can be split into 3 steps: (1) feature tracking/matching between images; (2) estimation of the camera motion inside a random-sample based procedure for robustness against outlier matches, and (3) optimization using bundle-adjustment for refining the camera poses. Novel RGB-D sensors, like the Microsoft Kinect, provide dense depth maps in addition to the color images. Odometry systems that operate with such information are, therefore, different from the monocular systems, since depth information can be explored for providing reliable camera poses and 3D reconstructions.

Several researchers have focused on the problem of odometry and SLAM for RGD-D sensors [4, 6, 9, 13, 16]. Endres *et al.* [4] proposed a two-fold SLAM system for RGB-sensors. On the front-end, the spatial relation between adjacent RGB-D images is established by extracting and matching image features. The matches are then used to estimate the relative transformation between sensor poses using a RANSAC-based procedure. The back-end of the SLAM system optimizes the pose observations with a graph-optimization procedure to keep long-term reliable reconstructions. A similar work to [4] was presented by Henry *et al.* [6]. Their approach uses sparse feature matches to compute an initial pose estimate using RANSAC, which is refined using an ICP procedure.

Steinbruecker *et al.* [13] proposed a photo-consistency approach that aims to find the best transformation between two sequential RGB-D frames. For robustness against large image displacements, the optimization is carried from a coarse-to-fine image resolution. This approach was then generalized by Kerl *et al*. [9] by including a probabilistic derivation and by showing how motion priors can be used to further improve the performance of [13]. Kerl *et al*. also study the performance of different outlier weighting functions, concluding that weighting outlier pixels with t-distribution in conjunction with motion priors leads to the best performance.

An important difference in our method is that all the motion estimation is performed pairwise, whereas in the state-of-the-art methods temporal information is often explored to enforce smoothness in the trajectories. As baseline comparison, we adopt the DVO method proposed by Kerl *et al*. [9] since it outperforms previously published algorithms [4, 6, 13].

Closely related with this work is the recent paper by Taguchi *et al*. that has been pre-released on-line [16] and to the best of our knowledge it is the first work that proposes plane-based SLAM for RGB-D sensors. It uses both points and planes as primitives, and the registration of 3D data in different coordinate systems provides the relative pose estimation. Although our method also relies on planes and points for achieving the pose estimation, registration is performed pairwise and not in relation to a global map. Moreover, we only use points if strictly necessary, as opposed to Taguchi's method. Also, our points are not reconstructed, being more robust to measurement errors. Another key difference is the refinement step, where in [16] a bundle-adjustment procedure to minimize error between points (and between points and planes) is performed, whereas in our method photo-consistency is used.

3 Algorithm Overview

We propose a new method for estimating an RGB-D sensor motion from the acquired color image-depth map pairs. For each pair of RGB-D images, two main consecutive steps are performed: an estimation of the sensor's relative pose between the two frames, and a refinement of this initial estimation. The reconstruction of the whole trajectory of the sensor is achieved by using only the pairs of consecutive frames, and does not take into account any prior information.

3.1 Initial Estimation

The initialization step uses corresponding planes extracted from both RGB-D images for determining the relative pose. If a given pair of images does not contain at least three corresponding non-parallel planes, it is not possible to fully determine the transformation between the images. In this case, a local feature detector (SURF [3]) is used for extracting image points, which are used to determine the remaining degrees of freedom. Table 1 shows the number of points and planes used for all the possible cases. The algorithm favors the usage of planes, by using the maximum number of non-parallel planes present in the image pair, and only using points when strictly necessary. Note that the points are not reconstructed, and only 2D points are used. This way, the estimation is less affected by measurement noise.

The initialization step starts by segmenting the planes present in both RGB-D images, which is performed by using the method proposed by Taylor and Crowley [15]. Next, a search for the corresponding sets of planes in both frames is performed hierarchically, meaning that sets with more planes are selected first. As an example, sets of two planes are only selected if there are no corresponding sets of three planes in the pair of frames.

No. of no parallel planes	3	2	1	0
No. of 2D points	0	1	4	5

Table 1: Possible combinations of the number of points and planes used in the relative pose estimation process.

Two planes are considered parallel if the smallest angle between their normals is below a pre-defined threshold (≈ 6°). If there are no parallel planes in a triple set, then this set is selected, and the triples in the other frame that correspond in relative normal orientation are considered putative matches. The association is carried by computing the three smallest angles between the normals of the planes in each set, sorting them, and computing their difference. If any of these differences is higher than a pre-defined threshold (≈ 3°), the two sets do not correspond. Although many correspondences are discarded with this procedure, erroneous ones may be selected. Thus, the dominant color of each plane is extracted, and only sets with corresponding colors are selected as hypothesis to test.

3.1.1 Relative Pose Estimation

Depending on the number of corresponding non-parallel planes in both frames, the relative pose estimation between the two frames is computed differently. All possible cases are shown in table 1 and, for each case, the transformation between the poses of the sensor is computed as follows.

Three Planes

For the case of two corresponding triplets of planes, a minimal, optimal solution is computed. The registration problem is the one of estimating R and $\mathbf{t}$ such that

$$\Pi_s^{(i)} \sim \begin{bmatrix} \mathsf{R} & \mathbf{0} \\ -\mathbf{t}^\mathsf{T}\mathsf{R} & 1 \end{bmatrix} \Pi_f^{(i)}, i = 1,2,3 \tag{1}$$

verifies, where $\Pi_f^{(i)}$ and $\Pi_s^{(i)}$ are planes in the first and second reference frames, respectively, in homogeneous representation $\Pi_f^{(i)} \sim [\mathbf{n}_{fi} \quad 1]^\mathsf{T}$ (and equivalent for $\Pi_s^{(i)}$). Knowing that points and planes are dual entities in 3D - a plane in the projective space $\mathcal{P}^3$ is represented as a point in the dual space $\mathcal{P}^{3*}$, and vice-versa - equation (1) can be seen as a projective transformation in $\mathcal{P}^{3*}$ that maps points $\Pi_f^{(i)}$ into points $\Pi_s^{(i)}$. It can be shown that R and $\mathbf{t}$ can be determined separately. R is firstly computed by normalizing $\mathbf{n}_{fi}$ and $\mathbf{n}_{si}$, and applying the algorithm from [7] for computing a transformation between two sets of unitary vectors. R can be computed from $N = 2$ point-point correspondences, but $\mathbf{t}$ requires $N = 3$ point-point correspondences to be estimated. Some algebraic manipulation of equation (1) leads to

$$\mathbf{n}_{si}^\mathsf{T}\mathbf{n}_{si}\mathbf{n}_{fi}^\mathsf{T}\mathsf{R}^\mathsf{T}\mathbf{t} - \mathbf{n}_{si}^\mathsf{T}\mathbf{n}_{si} + \mathbf{n}_{si}^\mathsf{T}\mathsf{R}\mathbf{n}_{fi} = 0. \tag{2}$$

Each pair $\Pi_f^{(i)}$, $\Pi_s^{(i)}$ gives rise to a linear constraint in the entries of the translation vector $\mathbf{t}$, which is the solution of a linear system of equations.

Two Planes

In case of existing only two corresponding pairs of planes, the rotation R can be fully determined using Horn's method [7], and two subsequent linear constraints in the form of equation (2) are obtained. Thus, an extra point correspondence $(\mathbf{x}_s, \mathbf{x}_f)$ is needed for computing the full translation vector $\mathbf{t}$. In this case, the epipolar constraint $\mathbf{x}_s^T \mathsf{E} \mathbf{x}_f = 0$, where $\mathsf{E} = [\mathbf{t}]_\times \mathsf{R}$ is the essential matrix, is stacked with the previous equations, forming a linear system of equations whose solution is the translation vector.

One Plane

If there is only one corresponding plane between the two frames, only two orientation angles of the rotation matrix R are known. It is thus necessary to find the remaining orientation angle θ, and the translation vector $\mathbf{t}$. This can be done by using one of the 3-point, 4-point or 5-point algorithms described in [5]. In this work we implemented the simple 4-point algorithm, which yielded good results. However, the minimal number of points required in this case is two since, besides knowing two orientation angles of R, a constraint in the entries of $\mathbf{t}$ and θ is known from equation (2). Thus, the 3-point algorithm from [5] can be adapted to this particular case, and the transformation (up to scale) can be found from two point correspondences.

The 4-point algorithm determines the relative pose transformation up to a scale factor. Thus, the constraint (2) was used for finding the scale factor, and obtaining the true translation vector.

No Planes

In this case, no information about the relative pose of the sensor between the two frames can be extracted from plane correspondences. Thus, all 5 DOF (up to a scale factor) must be determined from point correspondences. We used Nister's solution [12] which is a minimal solution since it uses 5 point correspondences.

For each pair of corresponding sets of planes found in the RGB-D image pair, and by using the necessary number of point correspondences, a transformation matrix is computed. If more than one estimation is obtained, a search for the best one must be performed. In the perspective case, two images $\mathbf{q}_f$ and $\mathbf{q}_s$ of two planes Π_f and Π_s, respectively, are related by an homography $\mathbf{q}_f \sim \mathsf{H}\,\mathbf{q}_s$ of the form:

$$\mathsf{H} = \mathsf{K}\left[\mathsf{R} + \mathbf{t}\frac{\tilde{\mathbf{n}}_f^\mathsf{T}}{d_f}\right]\mathsf{K}^{-1}, \tag{3}$$

where K represents the camera intrinsics, $d_f = 1/||\mathbf{n}_f||$ the distance of the plane to the origin of the reference frame, and $\tilde{\mathbf{n}}_f$ represents the unitary normal vector. The transformation that best correlates the image intensities of the segmented planes is selected for further refinement.

3.2 Pose Refinement with Photo-consistency

The relative pose estimation carried using the segmented planes can be affected by noise in the plane segmentation step. To refine the initial estimation, we use an intensity-based registration procedure. By performing a normalization of $\mathbf{q}_f \sim \mathsf{H}\,\mathbf{q}_s$ to non-homogeneous coordinates, we can define a 2D warping function $\mathbf{w}(\mathbf{q}_f\,; \mathbf{p}) = \Psi\left(\mathsf{K}\left[\mathsf{R} + \mathbf{t}\frac{\tilde{\mathbf{n}}_f^\mathsf{T}}{d_f}\right]\mathsf{K}^{-1}\mathbf{q}_f\right)$, with Ψ denoting the normalization to non-homogeneous coordinates, and $\mathbf{p}$ being the warping parameter vector that encodes 3 parameters for camera rotation, 3 for translation, and 3 for the plane structure.

Given the 2D warping function $\mathbf{w}(\mathbf{q}_f;\mathbf{p})$, it is possible to define a cost function describing the sum of squared differences between the pixels of a planar patch in the reference and incoming image:

$$\varepsilon = \sum_{\mathbf{q}_f \in \mathcal{N}} \Big[\mathtt{I}_s(\mathbf{w}(\mathbf{q}_f\,;\mathbf{p})) - \mathtt{I}_f(\mathbf{q}_f) \Big]^2, \tag{4}$$

where $\mathcal{N}$ denoting a plane integration region. Since an initialization $\mathbf{p}$ of the parameters vector is already known from previous steps, we iteratively solve for $\delta\mathbf{p}$ increments on the warp parameters, with equation (4) begin approximated by

$$\varepsilon = \sum_{\mathbf{q}_f \in \mathcal{N}} \Big[\mathtt{I}_s(\mathbf{w}(\mathbf{q}_f\,;\mathbf{p}+\delta\mathbf{p})) - \mathtt{I}_f(\mathbf{q}_f) \Big]^2 \approx \sum_{\mathbf{q}_f \in \mathcal{N}} \Big[\mathtt{I}_s(\mathbf{w}(\mathbf{q}_f;\mathbf{p})) + \nabla \mathtt{I}_s \frac{\partial \mathbf{w}}{\partial \mathbf{p}} \delta\mathbf{p} - \mathtt{I}_f(\mathbf{q}_f) \Big]^2. \tag{5}$$

By differentiating ε with respect to $\delta\mathbf{p}$, we obtained a closed form solution for $\delta\mathbf{p}$:

$$\delta\mathbf{p} = \mathcal{H}^{-1} \sum_{\mathbf{q}_f \in \mathcal{N}} \Big[\nabla \mathtt{I}_s \frac{\partial \mathbf{w}(\mathbf{q}_f\,;\mathbf{p})}{\partial \mathbf{p}} \Big]^{\mathsf{T}} \Big(\mathtt{I}_f(\mathbf{q}_f) - \mathtt{I}_s(\mathbf{w}(\mathbf{q}_f\,;\mathbf{p})) \Big), \tag{6}$$

with $\mathcal{H}$ being a 1st order approximation of the Hessian matrix [1], and the parameter vector being additively updated $\mathbf{p}^{k+1} \leftarrow \mathbf{p}^k + \delta\mathbf{p}$ at each iteration k. For robustness against a noisy camera motion initialization, we use a coarse-to-fine registration framework. We build an image pyramid by down-sampling the original image by factors of 2 (we use 3 pyramid levels). We start by optimizing the parameters at the coarsest level. After convergence (or a maximum number of iterations is reached), the resulting parameters are used to initialize the next pyramid level. The algorithm proceeds until the original image resolution is reached.

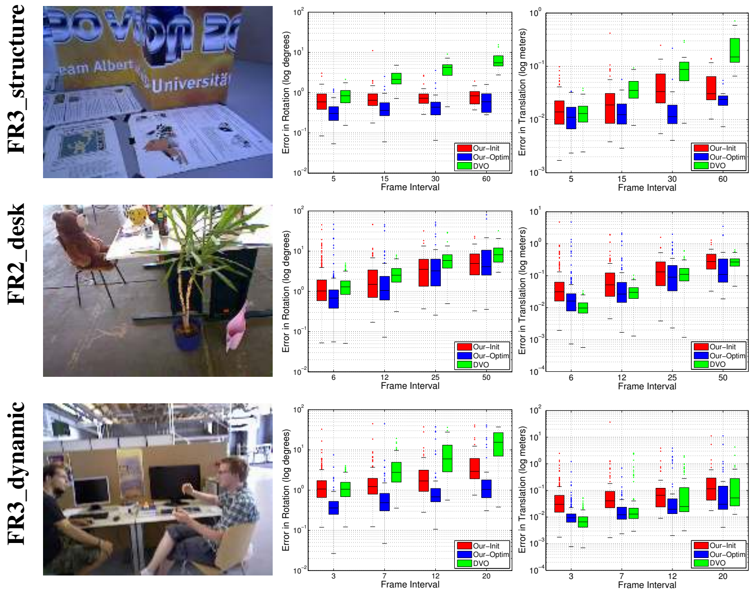
As explained in [2, 11], if only one plane is available it is impossible to estimate the 9 parameters of $\mathbf{p}$ from the 8 non-linear constraints of the homography. In such cases we fix the initial depth of the plane, and optimize the remaining 8 parameters of the warping function. In cases of multiple plane optimization, where the camera extrinsic parameters are the same for all the segmented planes, we adopt two different warping functions [11] that enable to estimate the camera motion globally for all the features being tracked:

$$\mathsf{H}^{(1)} = \mathsf{K}(\mathsf{R} + \frac{\mathbf{t}}{d_{f_1}} \tilde{\mathbf{n}}_{f1}^{\mathsf{T}})\mathsf{K}^{-1}, \qquad \mathsf{H}^{(i)} = \mathsf{K}(\mathsf{R} + \frac{\mathbf{t}}{d_{f1}} \frac{d_{f1}}{d_{fi}} \tilde{\mathbf{n}}_{fi}^{\mathsf{T}})\mathsf{K}^{-1}, i > 1 \tag{7}$$

With such parametrization, we end up with a total of $6 \times 3i - 1$ parameters to optimize per frame pair. The parameter updates are computed using the Schur complement to explore the sparsity of the system. For further details on how to compute the parameters, we refer the reader to [2, 11].

4 Results

In this section we conduct two sets of experiments to validate the proposed method in real scenarios. The first set uses a benchmark dataset with ground truth trajectories [14], while in the second we perform a loop-close experiment with large baseline between frames.



No. frames	75
% w/ pts	6.8%
No. planes	3.6
% outliers	3.9%

No. frames	124
% w/ pts	88.4%
No. planes	4.1
% outliers	7.6%

No. frames	110
% w/ pts	24.9%
No. planes	4.3
% outliers	12.3%

(a) Samples images (b) Rotation Error (c) Translation Error

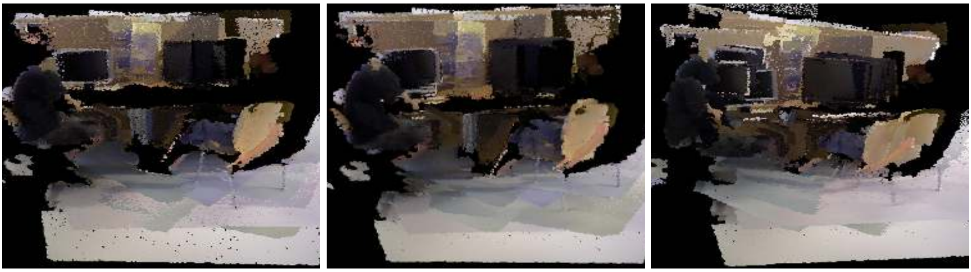
Figure 1: Benchmark validation. (a) shows a sample image of each sequence, (b) and (c) show the rotation and translation error per frame, respectively. The graphics show the performance of the different methods for different frame intervals. The last column shows some statistics regarding each dataset. We show the average number of frames per sequence, the average percentage of cases using points for computing the camera motion, the average number of planes used for registration, and, finally, the average percentage of outlier observations of our optimization step.

4.1 Benchmark Validation

The quantitative evaluation of our method is performed on 3 sequences from the TUM RGB-D dataset [14]. For simulating different camera velocities, we conduct the experiments by leaving out intermediate frames. We evaluate the pairwise camera motion estimations by computing the angular difference between the estimated and ground truth rotation matrix, and the norm of the difference between the estimated and ground truth translation vector. For comparison we use the dense visual odometry (DVO) algorithm proposed by Kerl *et al.*.

Figure 1 shows the results for this controlled set of experiments. The **FR3_structure** dataset is dominated by large support textured planes without any occlusion. We can observe that the DVO algorithm shows good performance for the smallest frame interval. As we increase the baseline between frames, its performance starts to degrade due the larger number of outlier image pixels used for the global image registration. Our method presents an almost constant performance for all the baselines. In this particular dataset, the optimization by plane registration enables to estimate the camera rotation with a median error of less than 0.5 degrees, which is within the measurement error of the sensors used to compute the ground truth camera poses [14].

We use the source code provided by the authors at https://github.com/tum-vision/dvo



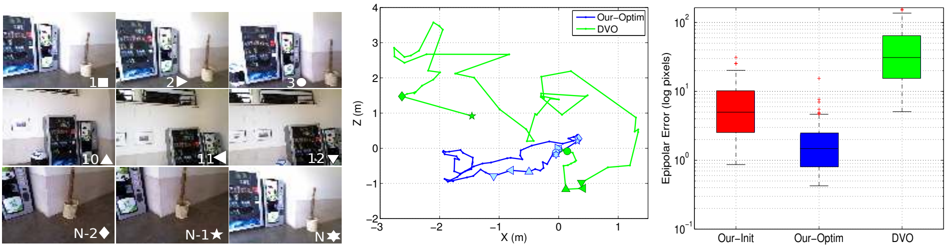
(a) Ground truth (b) Our-Optim (c) DVO

Figure 2: 3D reconstruction for *FR3_dynamic* dataset. The figure shows the 3D reconstruction computed with 2(a) ground truth camera poses, 2(b) our method with optimization, and 2(c) the DVO method. The quality of the 3D reconstruction indicates that our algorithm is more robust than DVO for scenes with dynamic motion.

The **FR2_desk** dataset was acquired in a typical office environment, where planar surfaces present low texture (e.g. tables, monitor and floor). This places some challenges to our local photo-consistency optimization step. We observed this by the larger number of outliers in the box-plots when compared with the DVO algorithm, where photo-consistency is performed using all the available image pixels. By inspection of the results, we observed that the outlier estimations are mainly due to small support planes that, in conjunction with the noise from the initial estimation, do not allow enough overlap between views to successfully perform the registration. Overall, our algorithm performs better than the DVO for large baselines, being consistently better in rotation across all the baselines tested.

Finally, in the **FR3_dynamic** we validate our algorithm in a dynamic scene with two persons moving and partially occluding the surrounding environment. In this sequence, the camera has been rotated along the principal axes, with a minimal translation amplitude (≈ 5 mm between camera poses). We can observe that our method clearly outperforms the DVO algorithm in terms of rotation accuracy across all the baselines, and in translation for the large baseline sequences. Despite of the DVO poor performance in rotation, the algorithm is capable of providing good translation estimations. We believe this is a consequence of the motion *priors* used in the optimization step. Since the translation vector is always very small, the probabilistic filter, due the absence of reliable observations, probably favors the current state keeping the translation vector almost unchanged across frames. Figure 2 shows the 3D reconstruction obtained with the two methods for the sequence with 3 frames of interval, where the DVO provides the lowest error in rotation. Since our algorithm is based on planes, which typically remain static, the camera motion recovery is less error prune and less influenced by the dynamic motion in the scene. This can be clearly seen by the accuracy of the 3D reconstruction where our method reconstructs the 2 existing monitors, while the DVO reconstruction present "phantoms" due to the poor inter-frame registration.

Note that none of the individual estimates where computed using only points. In every pair of frames of all three sequences, the algorithm was able to identify at least one plane correspondence.



(a) Sample Images (b) Trajectory estimation (c) Epipolar Error

Figure 3: Loop closing experiment with N = 59 images. 3(a) shows some images of the sequence, 3(b) shows the estimated trajectories, and (c) compares the epipolar error of the different methods. The trajectory obtained with our method almost closes the loop.

4.2 Loop-closing Experiment

In this experiment we navigate with a hand-held Kinect in a corridor to perform a loop-closed trajectory. This dataset is extremely challenging with difficult illumination conditions (see Fig. 3(a) for some sample images), low texture, and fast camera motion (the images were acquired at 3Hz with a resolution of 1280×1024).

Figure 3(b) shows the trajectory estimation for the different algorithms tested. Our algorithm enables to keep a reliable trajectory estimation, with a consistent smooth transition between frames. The DVO method diverges after the first couple of frames, providing an erroneous trajectory. We believe this happens due to the large baseline between frames, which results in a large number of outlier pixels introduced in the DVO registration process.

Finally, we show in figure 3(c) the epipolar geometric error to provide a quantitative error of the pairwise motion estimations. We use SURF to establish putative matches, which are filtered using a RANSAC procedure with the fundamental matrix. The inlier points are used to compute the Sampson distance for each method. We observe that our optimization procedure greatly improves our initial estimations. The epipolar errors obtained with the DVO algorithm justify the erroneous trajectory provided by this method.

Our algorithm was fully implemented in Matlab, taking in average 3 seconds per image pair, while the DVO algorithm runs at 30Hz. We believe that an optimized C++ implementation of our algorithm can achieve more than 10Hz.

5 Conclusions

The advent of commodity RGB-D cameras gave rise to intense research in pipelines for motion estimation using simultaneously dense depth and visual appearance. Unlike previous works that rely in either sparse or dense point features, we propose the use of planes, as an alternative to points, for estimating the camera motion. Plane-based registration is advantageous with respect to point-based registration because: (i) plane-primitives have a more global character, which helps avoiding local minima issues, (ii) scenes are often dominated by large planes, which allow correspondence between wide-baseline frames, (iii) plane primitives are typically in the static background, which improves odometry robustness to possible dynamic foreground, and (iv) the fact that the number of plane-features is much smaller than point-features, favors faster correspondence and scalability under increasing image resolution.

Extensive experiments show that the proposed method is well suited for operation in man-made environments, which are typically dominated by planes. We show that our algorithm outperforms the state-of-the-art algorithm of Kerl et al. [9] for large baselines, while keeping similar performance for small baseline sequences. In particular, we have observed that our algorithm provides better rotation estimations across all the baselines tested, and it is more robust to dynamic motion in the scene. In the future, we will improve our algorithm for dealing with partial plane occlusions by using a robust cost function in the optimization [1], and by including motion priors to increase the performance in small baseline situations.

Acknowledgement

The authors acknowledge the Portuguese Science Foundation (FCT) that generously funded this work through grants SFRH/BD/47488/2008, SFRH/BD/63118/2009 and SFRH/BD/88446/2012. This work has also been supported by the FCT and COMPETE program (co-funded by FEDER) under Project PTDC/EEA-AUT/113818/2009.

References

[1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221 – 255, March 2004.

[2] Simon Baker, Ankur Datta, and Takeo Kanade. Parameterizing homographies. Technical Report CMU-RI-TR-06-11, Robotics Institute, Pittsburgh, PA, March 2006.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3):346–359, June 2008. ISSN 1077-3142.

[4] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *IEEE-ICRA*, May 2012.

[5] Friedrich Fraundorfer, Petri Tanskanen, and Marc Pollefeys. A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In *ECCV*, pages 269–282, 2010.

[6] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *ISER*, volume 20, pages 22–25, 2010.

[7] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, Apr 1987.

[8] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *IEEE-IROS*, pages 3946–3952, 2008.

[9] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *IEEE-ICRA*, May 2013.

[10] Kurt Konolige, Motilal Agrawal, Robert C. Bolles, Cregg Cowan, Martin Fischler, and Brian Gerkey. Outdoor mapping and navigation using stereo vision. In *ISER*, 2006.

[11] C. Mei, S. Benhimane, E. Malis, and P. Rives. Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors. *IEEE-TRO*, 24(6):1352–1364, Dec. 2008. ISSN 1552-3098.

[12] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE-TPAMI*, 26(6):756–777, June 2004. ISSN 0162-8828.

[13] F. Steinbruecker, J. Sturm, and D. Cremers. Real-time visual odometry from dense rgb-d images. In *IEEE-ICCV Workshop on Live Dense Reconstruction with Moving Cameras*, 2011.

[14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE-IROS*, Oct. 2012.

[15] Camillo J. Taylor and Anthony Cowley. Parsing indoor scenes using rgb-d imagery. In *RSS*, July 2012.

[16] Srikumar Ramalingam Yuichi Taguchi, Yong-Dian Jian and Chen Feng. Point-plane slam for hand-held 3d sensors.