Real-Time Visual Odometry from Dense RGB-D Images

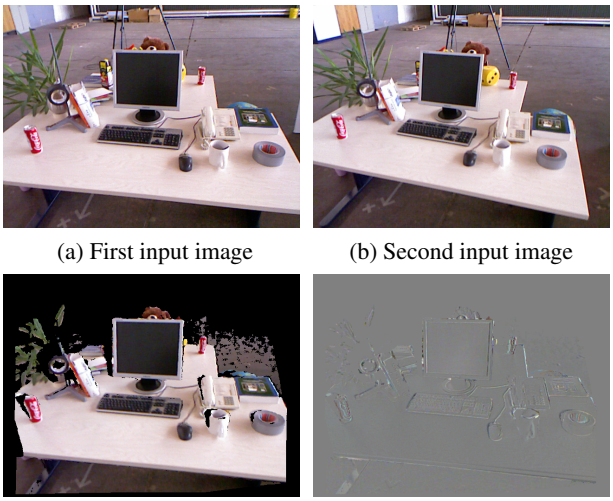
Frank Steinbrücker Jürgen Sturm Daniel Cremers
Department of Computer Science, Technical University of Munich, Germany
{steinbrf,sturmju,cremers}@in.tum.de

Abstract

We present an energy-based approach to visual odometry from RGB-D images of a Microsoft Kinect camera. To this end we propose an energy function which aims at finding the best rigid body motion to map one RGB-D image into another one, assuming a static scene filmed by a moving camera. We then propose a linearization of the energy function which leads to a 6×6 normal equation for the twist coordinates representing the rigid body motion. To allow for larger motions, we solve this equation in a coarse-to-fine scheme. Extensive quantitative analysis on recently proposed benchmark datasets shows that the proposed solution is faster than a state-of-the-art implementation of the iterative closest point (ICP) algorithm by two orders of magnitude. While ICP is more robust to large camera motion, the proposed method gives better results in the regime of small displacements which are often the case in camera tracking applications.

1. Introduction

Visual odometry, i.e., the problem of tracking the pose of a robot purely from vision, has a long history in the fields of computer vision and robotics [6, 4]. To estimate the motion of a robot, often laser scanners have been used. The laser scans are then often matched using variants of the iterative closest point (ICP) algorithm [1, 7]. The general idea is to iteratively assign correspondences between the points of the two scans and to register them. In practice, this alternation is prone to local minima as the registration tends to reinforce a possibly suboptimal initial point correspondence. This limitation is somewhat alleviated by extensions which perform a point-to-plane assignments (rather than point-to-point) [8]. In contrast, many state-of-the-art approaches using monocular camera images extract key-points and match them with previous frames [3, 9, 2] using a sequence of processing steps including descriptor matching, RANSAC and bundle adjustment. While the reduction to sparse key-points speeds up computation time enormously, much relevant information about the scene is lost. Newcombe et al. [5] recently presented a novel approach to dense visual odometry and 3D surface reconstruction from monocular image streams through extensive GPGPU parallelization. As we will show in this paper, RGB-D sensors like the Microsoft Kinect open novel ways to compute visual odometry directly from the input data which significantly reduces the computational costs.



(a) First input image (b) Second input image

(c) Warped second image (d) Difference image

Figure 1: We propose an energy minimization approach to estimate the camera motion between RGB-D images (a)+(b). The idea is to compute the rigid body motion which optimally transforms the second image (c) into the first, i.e. the difference image (d), computed for locations of reliable depth, should be zero (=gray).

1.1. Contribution

We propose an energy minimization approach for visual odometry from dense RGB-D images. The key idea is to tackle the underlying inverse problem by minimizing the backprojection error: Our goal is to find a rigid body transformation $g \in SE(3)$ representing the camera motion such that the registered second image exactly matches the first. We approximate the minimizer of this non-convex energy by sequential convex optimization: We linearize the energy and solve the arising normal equation for the 3D twist coordinates representing the desired rigid body motion. Since

the linearization only holds for small twists, we apply a coarse-to-fine approach to cope with larger camera motions.

We validated our approach on image sequences from a recently proposed dataset [10] and compared the performance of our approach to Generalized-ICP (GICP), a state-of-the art implementation of ICP [8]. While we found that ICP is more robust against larger camera displacements, our evaluation shows that our approach provides better results in the regime of small camera motions. Furthermore, our approach is faster than ICP by two orders of magnitude.

2. Visual Odometry from RGB-D Data

Let

$$I_{RGB} : \Omega \times \mathbb{R}_+ \to [0,1]^3, \quad (x,t) \mapsto I_{RGB}(x,t) \tag{1}$$

$$h : \Omega \times \mathbb{R}_+ \to \mathbb{R}_+, \quad (x,t) \mapsto h(x,t) \tag{2}$$

denote the color image data and height field on the image plane $\Omega \subset \mathbb{R}^2$ obtained from an RGB-D sensor at time $t \in \mathbb{R}_+$, where I_{RGB} gives the RGB values and h the depth in meters. From this height field, we can compute a surface S, i.e.,

$$S : \Omega \to \mathbb{R}^3, \quad x \mapsto S(x) \tag{3}$$

$$S(x) = \begin{pmatrix} \frac{(x+o_x)\cdot h(x)}{f_x} & , & \frac{(y+o_y)\cdot h(x)}{f_y} & , & h(x) \end{pmatrix}^\top$$

where $(o_x, o_y)^\top$ denotes the principal point of the camera and f_x and f_y the focal lengths.

As the Kinect sensor has two independent cameras that observe the scene from slightly different positions, we generate the height field h via reprojection of the disparity image of the Kinect into a z-buffer so that $I_{RGB}(x)$ and $h(x)$ refer to the color and depth of the same point in the world. Moreover, for simplicity, we only use the gray values of the color image, i.e., we define $I = (I_R + I_G + I_G)/3$.

Given two consecutive images $I(t_0)$ and $I(t_1)$ with surfaces $S(t_0)$ and $S(t_1)$, we now seek for the rigid body motion $g \in SE(3)$ of the camera between t_0 and t_1. We assume that the scene remains static, i.e., that every surface point has the same color in all camera images it is visible in. Our key idea is now that the rigid body motion g in combination with the surface $S(t_0)$ induces a unique mapping from pixels in $I(t_1)$ to pixels in $I(t_0)$. In the remainder of this paper, we call this mapping the *warp* w.

2.1. Lie Algebra Coordinates of Rigid Body Motion

We represent the six degrees of freedom of a rigid body motion g in the Lie group $SE(3)$ by the vector $\xi = (\omega_1\ \omega_2\ \omega_3\ v_1\ v_2\ v_3)^\top \in \mathbb{R}^6$. It defines a twist

$$\widehat{\xi} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

in the Lie algebra $se(3)$ at time $t \in \mathbb{R}_+$. Assuming that $\xi(t)$ is constant in the temporal interval $[t_0, t_1]$, the rigid body motion $g(t_1)$ is given by the matrix exponential

$$g(t_1) = \exp\big((t_1 - t_0)\widehat{\xi}\big) g(t_0), \tag{5}$$

where g is a 4×4 homogeneous matrix of the form

$$g = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}, \quad \text{with } R \in SO(3),\ T \in \mathbb{R}^3. \tag{6}$$

The Lie group and the Lie algebra are related by the differential equation

$$\frac{\mathrm{d}g}{\mathrm{d}t}(t) = \widehat{\xi}(t) g(t). \tag{7}$$

Rigid body motions g of the camera give rise to respective transformations G of a 3D points $P \in \mathbb{R}^3$:

$$G : SE(3) \times \mathbb{R}^3 \to \mathbb{R}^3, \qquad G(g, P) = RP + T. \tag{8}$$

The respectively transformed surface $G(g, S)$ can be converted to a height map using the projection $\pi : \mathbb{R}^3 \to \Omega$ from 3D space to the image plane given by:

$$\pi(G) = \begin{pmatrix} \frac{G_1 f_x}{G_3} - o_x & , & \frac{G_2 f_y}{G_3} - o_y \end{pmatrix}^\top \tag{9}$$

Concatenating (3), (8) and (9), we obtain the image warp

$$w_\xi : \Omega \times \mathbb{R}_+ \to \Omega, \quad (x,t) \mapsto w_\xi(x,t) \tag{10}$$

$$w_\xi(x,t) = \pi\Big(G\big(\exp(\widehat{\xi}\cdot(t-t_0)) g(t_0), S(x)\big)\Big)$$

2.2. Maximizing Photoconsistency

In the ideal case that the rigid body motion g is known between the two camera views, the warped second image should exactly match the first image. In practice, of course, the match will never be perfect because of missing values, occlusions and noise.

Therefore, we propose to compute the rigid body motion which maximizes photoconsistency. More specifically, we compute the twist ξ which minimizes the least-squares error

$$E(\xi) = \int_\Omega \left[I(w_\xi(x,t_1), t_1) - I(w_\xi(x,t_0), t_0)\right]^2 \mathrm{d}x \tag{11}$$

Under the assumption $g(t_0) = id$, the second term of (11) reduces to

$$I(w_\xi(x,t_0), t_0) = I(x, t_0). \tag{12}$$

2.3. Linearization of the Energy

Unfortunately, the energy (11) is not convex in the parameter ξ and therefore finding the minimum is non-trivial.

To overcome this limiation, we approximate both the image at time t_1 and the corresponding warp w by first-order Taylor approximations

$$I(w_\xi(x,t_1),t_1) \approx I(x,t_1) + (w_\xi(x,t_1) - x) \cdot \nabla I(x,t_1) \tag{13}$$

and

$$w_\xi(x,t_1) \approx x + (t_1 - t_0) \cdot \underbrace{\frac{\mathrm{d}\,(\pi \circ G \circ g)}{\mathrm{d}t}}_{=\frac{\mathrm{d}w}{\mathrm{d}t}}\Bigg|_{(x,t_0)} \tag{14}$$

By using the approximations (13) and (14), we get

$$E_l(\xi) = \int_\Omega \Bigg(I(x,t_1) - I(x,t_0) \quad + \nabla I(x,t_1) \cdot (t_1 - t_0) \cdot \frac{\mathrm{d}w}{\mathrm{d}t}(x,t_0) \Bigg)^2 \mathrm{d}x \tag{15}$$

Without loss of generality, we set $(t_1 - t_0) = 1$, since it is only a scalar factor to the minimizing ξ of the linearized energy. Additionally we can assume that the temporal derivative of the image is constant between t_0 and t_1, so we can substitute $I(x,t_1) - I(x,t_0) = \frac{\partial I}{\partial t}$ and obtain

$$E_l(\xi) = \int_\Omega \left(\frac{\partial I}{\partial t} + \nabla I(x,t_1) \cdot \frac{\mathrm{d}w}{\mathrm{d}t}(x,t_0) \right)^2 \mathrm{d}x \tag{16}$$

By means of the chain rule, we can express the total derivative $\frac{\mathrm{d}w}{\mathrm{d}t}$ in (16) as the product of several total derivatives (see (14)):

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{\mathrm{d}\pi}{\mathrm{d}G}\Bigg|_{\pi(G(g(t_0)),S(x)))} \cdot \frac{\mathrm{d}G}{\mathrm{d}g}\Bigg|_{G(g(t_0)),S(x))} \cdot \frac{\mathrm{d}g}{\mathrm{d}t}\Bigg|_{t_0} \tag{17}$$

With this, the energy becomes

$$E_l(\xi) = \int_\Omega \left(\frac{\partial I}{\partial t} + \nabla I \cdot \frac{\mathrm{d}\pi}{\mathrm{d}G} \cdot \frac{\mathrm{d}G}{\mathrm{d}g} \cdot \frac{\mathrm{d}g}{\mathrm{d}t} \right)^2 \mathrm{d}x, \tag{18}$$

where we neglected the evaluation points to improve readability. As a next step, we plug (7) into (18) and obtain

$$E_l(\xi) = \int_\Omega \left(\frac{\partial I}{\partial t} + \nabla I \cdot \frac{\mathrm{d}\pi}{\mathrm{d}G} \cdot \frac{\mathrm{d}G}{\mathrm{d}g} \cdot \widehat{\xi} \cdot g(t) \right)^2 \mathrm{d}x, \tag{19}$$

The result of $\widehat{\xi} \cdot g(t)$ is a 4×4 matrix, and so the derivative $\frac{\mathrm{d}G}{\mathrm{d}g}$ is a $3\times4\times4$ tensor. To simplify notation, we stack $\widehat{\xi} \cdot g(t)$ as a vector in $\mathbb{R}^{12}$. It can easily be verified that there exists a matrix M_g that fulfills

$$stack(\widehat{\xi} \cdot g(t)) = M_g \cdot \xi. \tag{20}$$

Dataset	Ours	GICP	Improvement
freiburg1/desk	0.0054 m	0.0103 m	1.93x
	0.0065 deg	0.0154 deg	2.37x
freiburg2/desk	0.0020 m	0.0062 m	3.11x
	0.0033 deg	0.0051 deg	1.55x

Table 1: Comparison of the drift per frame of our approach versus GICP on two different datasets. The values give the median. Our approach achieves more than 50% better pose estimates than GICP.

Representing g by its stacked version, we get the final form of our energy equation

$$E_l(\xi) = \int_\Omega \left(\frac{\partial I}{\partial t} + \underbrace{\left(\nabla I \cdot \frac{\mathrm{d}\pi}{\mathrm{d}G} \cdot \frac{\mathrm{d}G}{\mathrm{d}g} \cdot M_g \right)\Big|_{(x,t_0)}}_{=:C(x,t_0)} \cdot \xi \right)^2 \mathrm{d}x. \tag{21}$$

For every pixel x there is a 1×6 constraint $C(x,t_0)$ in the energy. Finding the minimizing ξ for this energy yields solving 6×6 normal equations. Since this is a least-squares problem, it can be easily solved by solving its corresponding normal equation. As the linearization in (15) is only valid for small twists ξ, we apply a coarse-to-fine scheme: We compute a first approximation on a coarse image scale, and iteratively refine this estimate on finer scales.

3. Results

Figure 1 shows a qualitative assessment of our method: The first and second image show the results of a quite large camera pose transformation. As it can be seen on the right side in the warped second image and the difference image of this warped image to the first image, our method still succeeds in estimating this transformation, as the warped image is nearly identical to the first image. The whole video from which these images are taken is included in the supplemental material for this paper. We also included a video of the whole camera tour, recorded in real-time from a third-person camera perspective. The fact that the objects in this video remain at their position shows that the accumulated camera pose is highly accurate.

To evaluate our algorithm quantitatively, we use the publicly available RGB-D dataset by Sturm et al. [10]. This dataset contains RGB-D images from a Microsoft Kinect with synchronized camera poses from an external motion capture system. From the large variety of different sequences, we chose the freiburg1/desk and freiburg2/desk for our experiments as they contain both translational and rotational motions in a typical office environment at different speeds.

We evaluated the drift per frame over these two sequences, see Tab. 1 and Fig. 2(top). Our approach has a median drift of 5.4 and 2.0 mm, while GICP drifts by 10.3 and

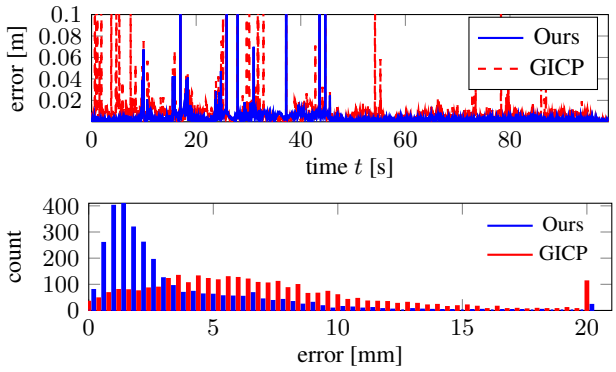


Figure 2: Per-frame error (top) and error histogram (bottom) on the freiburg2/desk sequence. We found that our approach has both a lower median error and fewer outliers in comparison to GICP.

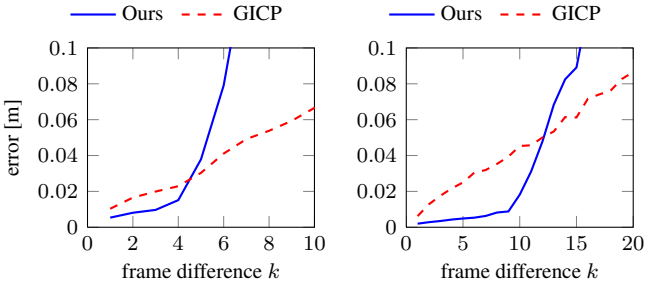
6.2 mm per frame, respectively. We analyzed this result further by computing the error histogram shown in Fig. 2(bottom): This plot confirms that our approach has much lower errors than GICP. Additionally, our approach also has fewer outliers. We found that GICP has in 3.8% of all frames an error larger than 2 cm, while our approach exceeds a 2 cm error only in 0.7% of all 2930 frames of the sequence.

In our next experiment, we simulated larger camera velocities by leaving out intermediate frames. In particular, we matched $I(t)$ and $I(t+k)$ for different $k = 1, \ldots, 20$ and measured for all t the error between our motion estimate and the motion from the ground truth. Fig. 3 shows the median error with respect to k. In particular, we found that our approach outperforms GICP when k is small, i.e., $k < 5$ for freiburg1/desk and $k < 12$ for freiburg2/desk. Note that the average camera speed in freiburg1/desk is much higher than in freiburg2/desk. From this result, we conclude that our approach is well suited for continuous camera tracking, while GICP can better deal with larger displacements.

We also evaluated the run-time of the two approaches. On a single Intel Xeon E5520 CPU with 2.27GHz, we measured that our approach takes 0.08 s, while the standard GICP implementation takes 7.52 s per match. This means that our approach is able provide visual odometry in real-time at 12.5 Hz.

4. Conclusion and Outlook

We introduced an energy-based approach to estimate the rigid body motion of a handheld RGB-D camera for a static scene. The key idea is to represent the rigid body motion in terms of its Lie algebra of twists and to determine the twist which maximizes the photoconsistency of the warped images. To this end, we minimize the non-convex reprojection error by a sequence of convex optimization problems obtained by linearizing the data term and solving the arising normal equations in a coarse-to-fine manner.



(a) freiburg1/desk dataset (b) freiburg2/desk dataset

Figure 3: Pose accuracy under increasing frame differences, i.e., we match $I(x,t)$ and $I(x,t+k)$ for all t. For not too large inter frame differences, the proposed method gives more accurate results while GICP is more robust against larger displacements.

Our plans for future work includes the implementation of a parallelized version of our approach on a GPU. Further, we plan to extend our approach to simultaneous localization and mapping.

Acknowledgements The authors would like to thank Richard Newcombe and Andrew Davision for fuitful discussions.

References

[1] P. J. Besl and H. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992. 1

[2] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Proc. of the Intl. Symp. on Experimental Robotics (ISER)*, Delhi, India, 2010. 1

[3] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, 2007. 1

[4] K. Konolige, M. Agrawal, R. Bolles, C. Cowan, M. Fischler, and B. Gerkey. Outdoor mapping and navigation using stereo vision. In O. Khatib, V. Kumar, and D. Rus, editors, *Experimental Robotics*, volume 39 of *Springer Tracts in Advanced Robotics*, pages 179–190. 2008. 1

[5] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: dense tracking and mapping real-time. In *Proc. of the Intl. Conf. on Computer Vision (ICCV)*, Barcelona, Spain, 2011. 1

[6] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. of the Computer Vision and Pattern Recognition Conference (CVPR)*, Washington, DC, USA, 2004. 1

[7] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. of the Intl. Conf. on 3-D Digital Imaging and Modeling*, Quebec, Canada, 2001. 1

[8] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, 2009. 1, 2

[9] H. Strasdat, J. M. M. Montiel, and A. Davison. Scale drift-aware large scale monocular slam. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, 2010. 1

[10] J. Sturm, S. Magnenat, N. Engelhard, F. Pomerleau, F. Colas, W. Burgard, D. Cremers, and R. Siegwart. Towards a benchmark for RGB-D SLAM evaluation. In *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf. (RSS)*, Los Angeles, USA, 2011. 2, 3