Direct Iterative Closest Point for Real-time Visual Odometry

Tommi Tykkälä Cédric Audras Andrew I. Comport
CNRS/I3S
2000, Route des Lucioles
surname@i3s.unice.fr

Abstract

In RGB-D sensor based visual odometry the goal is to estimate a sequence of camera movements using image and/or range measurements. Direct methods solve the problem by minimizing intensity error. In this work a depth map obtained from a RGB-D sensor is considered as a new measurement which is combined with a direct photometric cost function. The minimization of the bi-objective cost function produces 3D camera motion parameters which registers two 3D surfaces within a same coordinate system. The given formulation does not require any predetermined temporal correspondencies nor feature extraction when having a sufficient frame rate. It is shown how incorporating the depth measurement robustifies the cost function in case of insufficient texture information and non-Lambertian surfaces. Finally the method is demonstrated in the Planetary Robotics Vision Ground Processing (PRoVisG) competition where visual odometry and 3D reconstruction results are solved for a stereo image sequence captured using a Mars rover.

1. Introduction

In the general SLAM problem, camera pose and environment structure are estimated simultaneously and incrementally in real-time using a combination of sensors. A SLAM approach is interesting in a wide range of robotics applications where a precice map of the environment does not exist or it is inconvenient to store. Recently camera based SLAM approaches have been an important and active area of study [6], [11]. Typically they rely on feature-based approaches, where a sparse set of points is extracted per image frame and the points are matched temporally based on their feature descriptors. The SLAM problem is then often solved by filtering motion and 3D point parameters from measured 2D motion using an Extended Kalman Filter or by performing local bundle adjustment.

Direct SLAM methods avoid the feature extraction process completely by basing the estimation on raw image data [10]. This is achieved by formulating SLAM problem as partial or full image registration task between subsequent frames. Direct methods for both non-planar and planar environment geometries have been studied [4] [16]. The main advantage of direct methods is that they minimize true error based on the actual measurements. They can also be made statistically robust due to the redundancy in information and they are scalable for real-time performance [13].

In this paper the core issue of 3D visual odometry is considered as a 3D surface registration problem where dense structure measurements from different time instants are registered into the same reference coordinate system for obtaining 3D motion parameters. The approach differs from previous dense approaches in a way that it aims to minimize not only photometrical error but also depth image error simultaneously. As the pixel intensities and depth measurements are not compatible, the correct balance between the two components have to be found. This results into a nonlinear bi-objective cost function which is minimized using iteratively re-weighted least-squares optimization.

In section 2 the different approaches to visual odometry are discussed and the context is given for the work. In section 3 the cost function is presented with an overview of the minimization process. Section 4 covers a set of details which are important when implementing the algorithm correctly. Section 5 shows improvement in the pose tracking accuracy and robustness using simulation and section 6 describes how the proposed cost function works in a practical visual odometry task using a sequence provided by Planetary Robotics Vision Ground Processing project. Finally conclusions are made in section 7.

2. Previous work

The traditional method for surface registration is Iterative Closest Point algorithm (ICP) which alternates between finding temporary point correspondencies and updating motion parameters until the system converges [2]. The computational efficiency of this method depends on the amount of points to be matched. ICP does not scale very well as the structure estimates become more dense. A common performance improvement is to use kd-tree for near-

978-1-4673-0063-6/11/$26.00 ©2011 IEEE

est neighbor searches, but also further improvements have been suggested for aiming the method at real-time applications [15]. The outlier matches are a problem for ICP and commonly RANSAC is used for finding a subset of correct matches [8]. The performance of ICP improves when the 3D points are matched by an effective 2D feature descriptor such as SIFT [12].

Instead of minimizing the Euclidian distance between 3D point pairs, direct approaches can be used where the photometrical difference between a registered image pair is used as an error metric. Direct methods require an explicit 3D model or planarity assumption (e.g. homography) for generating warped images to be matched with the reference image. Direct 3D model based approaches have turned to be useful also in settings where explicit models are not available [5, 7]. In this case the 3D model is replaced by dense structure measurements that can be obtained by using a RGB-D sensor such as a stereo camera with a state-of-the-art dense matching technique or alternatively LIDAR, projection of structured light patterns (infra-red or visible light), sonar, etc. For indoor settings, Microsoft Kinect and the Asus Xtion Pro provide good results.

In this work the 3D registration is defined as a direct image-based minimization task. The objective is to incorporate depth measurements into a recent image-based visual odometry technique [4]. The combination produces an image-based method which is comparable with the standard ICP technique, but has advantages in computational requirement, precision and robustness. Contrary to ICP, all data is stored in images which makes it possible to avoid expensive nearest neighbor searches in 3D space. As the cost function is directly image-based, it is therefore robust and precise. Similar bi-objective minimization has been experimented recently with Inertial Measurement Unit (IMU) and a hand-held camera [14].

3. The direct ICP cost function

The motion is defined mathematically as $\mathbf{x} \in \mathbb{R}^6$ which generates 4×4 matrix $\mathbf{T}(\mathbf{x}) \in \mathbb{SE}(3)$ using the matrix exponential function. All previous motions are stacked into a base transformation matrix $\widehat{\mathbf{T}}$ and thus the problem is always to estimate the next camera motion increment. The transformations $\mathbf{T}$ are defined to map points from the reference camera coordinate frame into the current camera coordinate frame. The pose and the trajectory of the camera can be estimated by minimising a non-linear least squares cost function

$$C(\mathbf{x}) = \mathbf{e}_{\mathcal{I}}^T \mathbf{W}_{\mathcal{I}} \mathbf{e}_{\mathcal{I}} + \lambda^2 \mathbf{e}_{\mathbf{Z}}^T \mathbf{W}_{\mathbf{Z}} \mathbf{e}_{\mathbf{Z}}, \qquad (1)$$

where

$$\begin{aligned} \mathbf{e}_{\mathcal{I}} &= \mathcal{I}\left(w(\boldsymbol{\mathcal{P}}^*; \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}})\right) - \mathcal{I}^*\left(w(\boldsymbol{\mathcal{P}}^*; \mathbf{I})\right) & (2) \\ \mathbf{e}_{\mathbf{Z}} &= \mathbf{Z}\left(w(\boldsymbol{\mathcal{P}}^*; \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}})\right) - [\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}\boldsymbol{\mathcal{P}}^*]_z, & (3) \end{aligned}$$

and $\mathbf{W}_{\mathcal{I}}$ and $\mathbf{W}_{\mathbf{Z}}$ are the diagonal weight matrices obtained from a M-estimator. $\mathcal{I} : \mathbb{R}^2 \Rightarrow \mathbb{R}$ is a color brightness function and $\mathbf{Z} : \mathbb{R}^2 \Rightarrow \mathbb{R}$ is a depth function. Reference variables and functions are denoted by *. $w(\boldsymbol{\mathcal{P}}^*; \mathbf{T})$ is the warping function which transforms and projects reference 3D points $\boldsymbol{\mathcal{P}}^*$ into a new view using 4×4 transformation matrix $\mathbf{T}$ and the intrinsic matrix $\mathbf{K}$. The exact formula is

$$w(\boldsymbol{\mathcal{P}}^*; \mathbf{T}) = N(\mathbf{K}(\mathbf{R}\boldsymbol{\mathcal{P}}^* + \mathbf{t})), \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \qquad (4)$$

where $N(\mathbf{p}) = (p_1/p_3, p_2/p_3)$ dehomogenizes point $\mathbf{p}$. An image-based saliency map is used to for selecting 3D points $\boldsymbol{\mathcal{P}}^* = \{\mathbf{P}_1^*, \mathbf{P}_2^*, \ldots, \mathbf{P}_n^*\}$ where n is the number of points [13]. λ is a constant which defines the gain for the depth component. Operator $[\,]_z$ selects third row from given input vector/matrix. This cost function form does not require any predefined temporal correspondencies as they are solved iteratively via estimating the camera pose increments.

3.1. Bi-objective minimization

Since $C(\mathbf{x})$ is a non-linear function of the unknown pose parameters, it has to be linearized with $\mathbf{x}$ for iterative minimization. With linearization it is assumed that function is locally continuous, smooth and differentiable.

This results in a Jacobian of the form

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \mathbf{J}_1 \\ \lambda \mathbf{J}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\mathcal{I}} \mathbf{J}_w \mathbf{J}_{\mathbf{T}} \\ \lambda(\mathbf{J}_{\mathbf{Z}} \mathbf{J}_w \mathbf{J}_{\mathbf{T}} - [\mathbf{J}_{\mathbf{T}}]_z) \end{bmatrix}, \qquad (5)$$

where $\mathbf{J}_{\mathcal{I}}$ and $\mathbf{J}_{\mathbf{Z}}$ are the image and depth gradients with respect to pixel coordinates of dimension $n \times 2n$, $\mathbf{J}_w$ is the derivative of perspective projection of dimension $2n \times 3n$, and $\mathbf{J}_{\mathbf{T}}$ represents motion of a 3D point respect to motion parameters $\mathbf{x}$ with a dimension of $3n \times 6$.

$\mathbf{x}$ is obtained using the pseudo-inverse by

$$\Delta \mathbf{x} = -(\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{W} \begin{bmatrix} \mathbf{e}_{\mathcal{I}} \\ \lambda \mathbf{e}_{\mathbf{Z}} \end{bmatrix} \qquad (6)$$

The increments are updated into the base transformation by $\widehat{\mathbf{T}} \Leftarrow \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}$ as long as it is necessery for reaching convergent condition $\|\mathbf{x}\| < \epsilon$.

3.2. Balancing the cost by λ

Looking at the Hessian matrix

$$\mathbf{H} = \mathbf{J}^T \mathbf{W} \mathbf{J} = \mathbf{J}_1^T \mathbf{W}_{\mathcal{I}} \mathbf{J}_1 + \lambda^2 \mathbf{J}_2^T \mathbf{W}_{\mathbf{Z}} \mathbf{J}_2, \qquad (7)$$

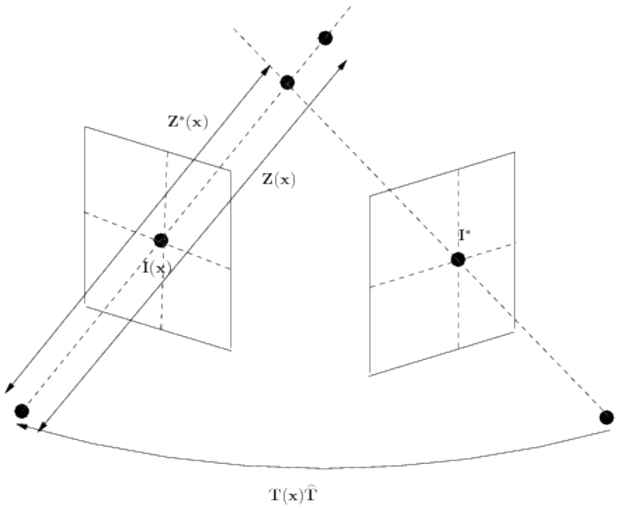

Figure 1. Photometric and depth error are minimized between subsequent image frames. In practise the 3D point associated with color $\mathcal{I}^*$ is transformed and re-projected into the current coordinate system where it is possible to compute $\mathbf{e}_{\mathcal{I}} = \mathcal{I}(\mathbf{x}) - \mathcal{I}^*$ and $\mathbf{e}_{\mathbf{Z}} = \mathbf{Z}(\mathbf{x}) - \mathbf{Z}^*(\mathbf{x})$.

it can be seen how the local curvature of the cost function depends on intensity and depth differentials. The benefit of incorporated depth measurement shows in this form where curvature is gathered from two sources and therefore $\mathbf{H}$ will less likely be singular. $\mathbf{H}$ can be singular in the cases where motion does not infer any appearance changes (e.g. homogeneous regions) or motion infers arbitrary appearance changes (e.g. non-Lambertian surfaces, occlusions). λ acts as a gain for the depth component and it is necessary to adjust it for proper balancing of the error function.

Depending on the metric unit of the 3D coordinate system and the local covariances of the components, one of the components of the residual may be neglible or fully dominate the error. This happens for example when pixel noise is numerically comparable to depth variations. Mathematically wrong choice of λ results in cases $\mathbf{H} \approx \mathbf{J}_1^T \mathbf{W}_{\mathcal{I}} \mathbf{J}_1$ and $\mathbf{H} \approx \mathbf{J}_2^T \mathbf{W}_{\mathbf{Z}} \mathbf{J}_2$, where the fusion does not bring any benefits.

The optimal λ is the one which improves the estimated camera trajectory the most when compared to a purely intensity based cost function. There are mathematical methods such as L-curve based metrics and cross validation for automatic selection [14]. L-curve based selection finds such *Pareto optimal* λ which minimizes both cost components simultaneously by finding the optimal point in the curve whose axices represent the costs of the separate components. In cross validation, on the other hand, a geometry independent λ is learned by finding such parameter which minimizes projection error for an average cost over all *leave-one-out* combinations of the point data.

The manual selection of λ can be done by simulating the real application by an image sequence with depth maps and experimenting with different λ values. In many application cases the optimal λ can be fixed only once in the beginning if depth range is known a priori. To automatically determine λ in real-time a robust ratio between the centers of the $\mathcal{I}$ and $\mathbf{Z}$ distributions is proposed such that $\lambda = |\mathrm{median}(\mathcal{I})/\mathrm{median}(\mathbf{Z})|$. By experiments it was noticed that this produces close to manually chosen values for the test sequences used. This formula finds robust scale factor which converts from depth units into pixel units.

4. Details in the implementation

This section presents the details which need to be taken into account when implementing the proposed bi-objective minimization using depth maps.

4.1. Multi-resolution and depth filtering issues

Commonly multiple resolutions are used for increasing the convergence domain. The minimization starts from a low resolution and the solution is refined using a sequence of higher resolution images. For generating a multi-resolution pyramid each layer has to be low-pass filtered and then sub-sampled to avoid aliasing effects. In the case of depth images, low-pass filtering is problematic as it alters the underlying 3D structure. For preventing this, depth values are always sampled at the highest resolution even though matching intensity values are low-pass filtered.

When downsampling images with 2×2 blocks, the subsamples are commonly generated in the middle of each 2×2 block. This needs to be taken into account, when matching points between different layers. In a case of sampling the high resolution depth image, the sample point coordinates are transformed into the higher resolution by $x_h = 2^L x + 0.5(2^L - 1)$, where L is the amount of layers in-between.

4.2. Real-time pixel selection

In each iteration of the minimization, the subset of 3D points which associate with greatest values of $\mathbf{J}$ determine the motion increment $\Delta\mathbf{x}$. Some of the points such as the ones in the homogeneous regions of the image or the depth map do not have any variation during the estimation as $\nabla\mathcal{I} = \mathbf{0}$ or $\nabla\mathbf{Z} = \mathbf{0}$ and thus do not have any influence into $\Delta\mathbf{x}$. This means that computational requirement can be reduced by selecting only the points which contribute to estimation. $\mathbf{J}_1$ and $\mathbf{J}_2$ are both $n \times 6$ matrices whose column vectors are the residual differentials respect to 6 degrees of freedom. Ideally pixel selection is done in a way that each 6 degrees of freedom are equally constrained and thus selection $\mathcal{P}^*$ corresponds to the region of image which covers the majority of the magnitude in each column [13].

Pixel selection can be done efficiently without requirement of sorting the columns $\mathbf{J}$ by neglecting the geometri-

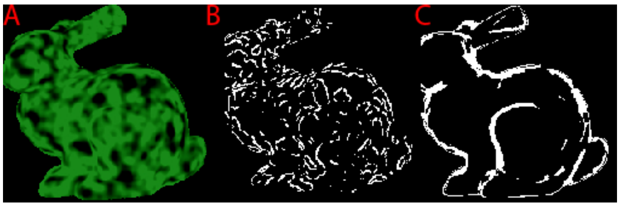

Figure 2. A. Stanford bunny image. B. 15% pixels selected by image gradients. C. 15% pixels selected by depth gradients. Combination selection is required for constraining the motion the most efficiently.

cal component of Jacobian and focusing only on image and depth gradients. Thus reference points $\mathcal{P}^*$ are selected into motion estimation using a 2D selection mask where the selection fitness is the combination gradient magnitude

$$S(x, y) = |\nabla \mathcal{I}_x| + |\nabla \mathcal{I}_y| + \lambda(|\nabla \mathbf{Z}_x| + |\nabla \mathbf{Z}_y|). \quad (8)$$

Selection scores $S(x, y)$ are accumulated into a histogram and a portion of pixels are selected from the end part of the histogram. Histogram is useful to generate as it is trivial to compute such a threshold which selects n best pixels efficiently as explicit sorting of the values is avoided. This improves $O(n\text{log}n)$ requirement into $O(n)$. A comparison of intensity and depth gradient based pixel selections is illustrated in Figure 2.

4.3. Robust weighting of the residuals

M-estimators are used for providing diagonal weight matrices $\mathbf{W}_\mathcal{I}$ and $\mathbf{W}_\mathbf{Z}$ for residuals and Jacobians. The weights damp statistically spurious pixels out from the estimation. The matrices are given by the Tukey weighting function based on a robust statistical distribution of the associated residual [4]. A problem with M-estimators is the computation of median efficiently, which by a naïve approach requires sorting of the residual values per each iteration. The median of $\mathbf{e}_\mathcal{I}$ is determined efficiently from the histogram as the bin in which the total mass of the histogram is halved. This can be done because the elements of $\mathbf{e}_\mathcal{I}$ are typically bounded into discrete range $[-255, 255]$. Unfortunately the range of scaled depth residual $\lambda \mathbf{e}_\mathcal{Z}$ is unlimited and the values are continuous real numbers. For computing the median using using histogram, the values of $\lambda \mathbf{e}_\mathcal{Z}$ are discretized and clamped between $[-255, 255]$.

4.4. Real-time minimization

When considering only the minimization of the intensity part, the Jacobian $\mathbf{J}_1$ can be fully pre-computed assuming a small motion between the frames. The geometrical part $\mathbf{J}_w \mathbf{J}_\mathbf{T}$ represents the 2D optical flow of the 3D structure which is computed only once at the reference image. Also the photometrical part $\mathbf{J}_\mathcal{I}$ can be approximated by the gradients of the reference image. The approximations are based on the fact that if $\widehat{\mathbf{T}}$ represents a small motion and also the optimal increment $|\mathbf{\Delta x}_o|$ is small, the optical flows as well as the gradients of the warped image and the reference image are approximately the same. The residuals are however computed accurately using the current and the reference image for converging into the correct minimum.

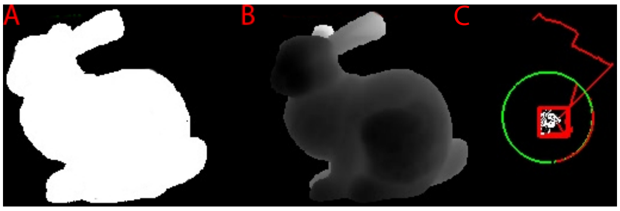

Figure 3. Orbiting sequence around Stanford bunny. A. image from the sequence, B. the corresponding depth map, C. the estimated camera trajectory (red) along with the ground truth (green). Insufficient texturing produces singular $\mathbf{H}$ which causes divergence during pose estimation. This problem is fixed by incorporated depth maps.

When minimizing depth residual, however, precomputation is not possible, because the reference depth values depend also on motion parameterization $\mathbf{x}$. Thus $\mathbf{J}_2$ is updated per each iteration without approximation.

5. Proof of concept by simulation

The robustness and the accuracy improvement were observed using two simulated sequences. In the first sequence, synthetic RGB-D sensor was set to rotate around Stanford bunny, whose image was toggled between textured and plain white. Because intensity based pose estimation relies on image gradients, divergence occurs once in a while when the edge information of the silhouette is not sufficient for tracking (Figure 3). This problem is can be fixed by incorporated depth maps. The depth map matching keeps $\mathbf{H}$ non-singular during tracking and full circle is obtained from camera pose estimation.

In the second sequence, the accuracy improvement is obtained by manually finding an optimal λ for a rendered Itokawa sequence. A single image and depth image of the sequence is illustrated in Figure 4. The estimated trajectories with and without incorporated depth maps are compared with the ground truth trajectory in Figure 5. The figure shows how camera trajectory is improved when using depth data. Non-Lambertian surface properties of Itokawa produces error in purely intensity based minimization.

6. PRoVisG MARS 3D Challenge sequence

The novel cost function is experimented with the sequence provided by the PRoVisG MARS 3D Challenge.

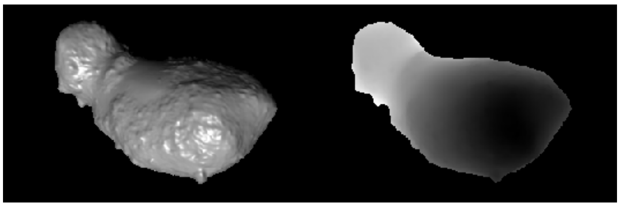

Figure 4. Picture of Itokawa and the corresponding depth map. The reflectance of real Itokawa follows rougly Hapke model instead of Lambertian. This causes problems for intensity based pose estimation as intensity values change as the function of motion parameters. The rendered picture of Itokawa has specular surface for demonstrating the problem.

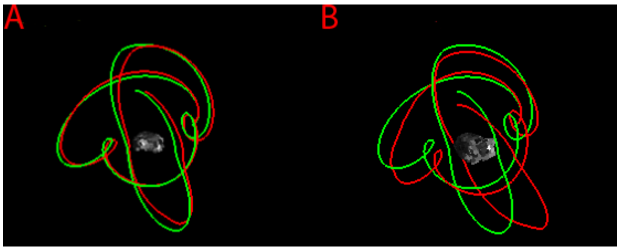

Figure 5. Simulated Itokawa asteroid sequence. A. Complex camera trajectory is more accurate when matching also depth maps. The green curve represents the ground truth camera trajectory and the red curve is the estimated one. B. trajectory of purely intensity-based minimization for comparison.

In the task setting, 3D trajectory and reconstruction is automatically determined from a sequence of stereo images. The tasks are divided into dense matching, camera pose estimation and 3D reconstruction. The following subsections describe the approach and the results.

6.1. The Mars Rover sequence

The given sequence consists of 35 stereo images in 1280×1024 resolution, which have been captured using a moving mars rover. The cameras have 64° angle and the baseline of the stereo camera is 10 centimeters. The estimated length of the rover trajectory is 7.8 meters. The frame rate of the sequence is relatively low compared to the vehicle motion and the images do not contain large quantity of texture details as the sand covers most part of the views. However 3–10 small rocks are visible in the view in all of the frames. There are no major lighting effects in the sequence. An example intensity image and a depth image of the right stereo view along with the corresponding residuals is illustrated in a Figure 6.

6.2. Dense stereo matching

For dense matching semi-global block matching (SGBM) is used. As a global method, semi-global block matching is computationally efficient and real-time implementations relying on FPGA exist [1]. The semi-global block matching finds smooth disparity maps by minimizing the following energy function

$$E(\mathbf{d}) = \sum_{\mathbf{p}} C(\mathbf{p}, d_{\mathbf{p}}) + \sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{p}}} P_1 \delta(|d_{\mathbf{p}} - d_{\mathbf{q}}| - 1) \quad (9)$$

$$+ \sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{p}}} P_2 \delta(|d_{\mathbf{p}} - d_{\mathbf{q}}| - k), \quad (10)$$

where $\mathbf{p} \in \mathbb{R}^2$ are center points of the pixels in the image grid and $d_{\mathbf{p}}$ the disparity values associated with them. $\mathcal{N}_{\mathbf{p}}$ is a set of points containing the 8 neighbor points of $\mathbf{p}$. $C(\mathbf{p}, d)$ is the block matching cost for $\mathbf{p}$ and P_1 and P_2 are the constant discontinuity penalties for unit jumps and bigger ($k > 1, k \in \mathbb{Z}$) [9]. Birchfield-Tomasi metric is used for the block matching due to computational efficiency and the fact that the sequence does not have major lighting variations [17]. For conditions where lighting effects exist BRIEF descriptor can be used instead of direct image comparison [3]. The minima of E is obtained by dynamic programming where the independent costs of 8 incoming 1D directions are maintained per each point. The aggregation of the directional costs is done in two passes where, the first pass aggregates the costs propagated from upper and left sides and the second pass aggregates the lower and right side directions. Finally the disparities which have the smallest sum cost of 8 directions are selected. Penalty constants P_1 and P_2 need to satisfy $P_1 < P_2$ for penalizing slanted surfaces and discontinuities appropriately.

The images are low-pass filtered and downsampled into 320×240 resolution using a multi-resolution pyramid because pixel noise has to be filtered out and the minimization of the proposed cost function works more efficiently with smoother gradients. The baseline of the stereo used is small and thus the matching can be done using discrete disparity range of $[0, 32]$. The disparity values are refined into subpixel accuracy as a post-process. Finally the disparity map obtained is converted into a depth map for matching the cost function.

6.2.1 Noise in depth

Especially local stereo matching methods produce often false discontinuities which are a potential problem for depth minimization as the greatest depth gradients have the greatest influence in the final pose estimate. This is why using a global matcher, such as SGBM, is important because smooth maps are produced by penalizing discontinuities. SGBM assigns discontinuities implicitly whenever image matching starts to fail. When the discontinuities are set locally in the wrong place, small deviations may occur which are filtered out as a post-process by using 5×5 median filter.

6.3. Pose estimation

The proposed cost function is minimized for finding the pose parameters. A typical use case for the proposed cost function is in application where the frame rate is high enough for using $\widehat{\mathbf{T}} = \mathbf{I}$ as the initial guess. For the given sequence the frame rate is low and an initial guess must be obtained by other means, such as by SIFT correspondencies. Figure 6 illustrates the convergent solution of intensity and depth minimization. Real-time performance is reached by approximating Jacobian computations and by using pixel selection. Figure 7 shows how Tukey window based M-estimator rejects $20 - 40\%$ of all selected pixels. Pixel selection was set to 50% which produces 38400 points in 320×240 resolution.

6.3.1 Initial guess

$\widehat{\mathbf{T}}$ can be generated by first matching temporally a set of 2D points using SIFT and then minimizing the 2D distance between the warped points and the fixed target points $\mathcal{P}_{\text{sift}}$ in the current image. This smooths the cost function sufficiently for getting closer to solution and after that the local optimization is done using the proposed cost function.

The residual to be minimized for the initial guess generation is thus

$$\mathbf{e}_{\mathbf{G}} = \rho(\mathcal{P}_{\text{sift}} - w(\boldsymbol{\mathcal{P}}^*; \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}})), \tag{11}$$

where ρ produces weighted distances and neglects statistically large displacements. Simple $\rho(x) = \|x\|$ if $\|x\| < \tau$ and otherwise 0, was used in the experiment where τ is a fixed threshold.

6.4. 3D reconstruction

After solving the camera trajectory, all points obtained from semi-global block matching are registered into the coordinate system of the first reference frame (Figure 8).

6.5. Results

The algorithm is executed on Samsung R530 laptop with dual-core Intel i3 CPU (2.13GHz) of which only one core is used. The implementation is written in C++ and relies on Fortran based LAPACK and EXPOKIT routines for linear algebra and matrix exponentials. Figure 9 illustrates the optimization delay in milliseconds per frame with and without depth component. Incorporated depth component adds roughly 50% more delay into computation. As a further optimization step, the algorithm could be run in parallel with both cores for halving the delays. It is unfortunate that the competition sequence has low frame rate and SIFT extraction and matching is required. Computational requirement of using SIFT is not evaluated because as it is considered redundant phase in a real application where sufficient camera frame rate can be selected. Three iterations of initial guess cost function was required after which three multi-resolution layers could be used for convergence. M-estimators reject $20 - 30\%$ of the points by setting zero weights (Figure 7). This rejection does not show in the Figure 9 because the sizes of the matrices used in motion update are not changed. In a real-time application FPGA hardware can be used for computing disparity maps [1]. The other additional phases such as pixel selection and 3D point extraction are $O(n)$ passes for the raw images which are very fast. The quality improvement when incorporating depth component is difficult to evaluate as the ground truth trajectory for the PRoVisG MARS 3D Challenge has not been published. However by observing Figure 8 the camera trajectory and the 3D reconstruction are qualitatively good.

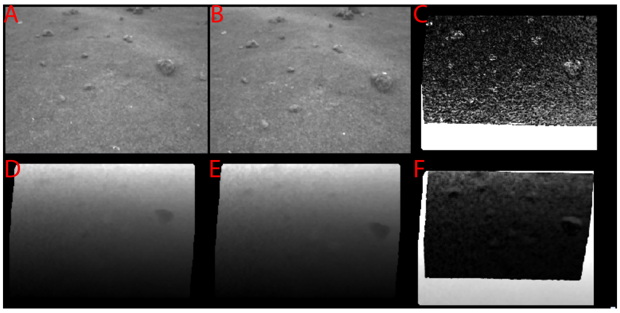

Figure 6. PRoVisG MARS 3D Challenge sequence. The minimization of the proposed cost function is visualized. A) Image 1 B) Image 2 C) The corresponding intensity residual $\mathbf{e}_{\mathcal{I}}(\widehat{\mathbf{x}}_o)$ D) Depth map 1 E) Depth map 2 F) The corresponding depth residual $\mathbf{e}_{\mathbf{Z}}(\widehat{\mathbf{x}}_o)$. The motion parameters $\widehat{\mathbf{x}}_o$ are the result of the proposed minimization scheme. The remaining error is due to interpolation/filtering inaccuracy, occlusions and depth noise.

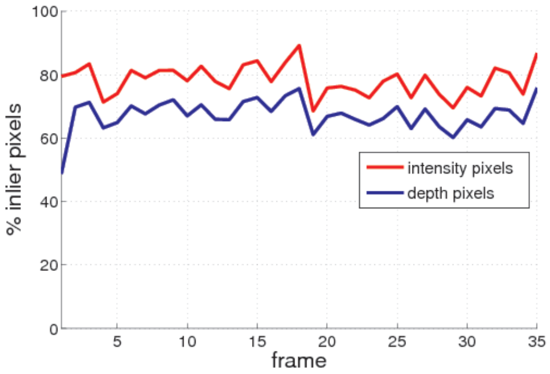

Figure 7. M-estimator functionality illustrated for the rover sequence. Total amount of selected pixels (inliers+outliers) is set to 50% which produces 38400 points in 320×240 resolution. M-estimator is made computationally efficient by using histogram for median computation.

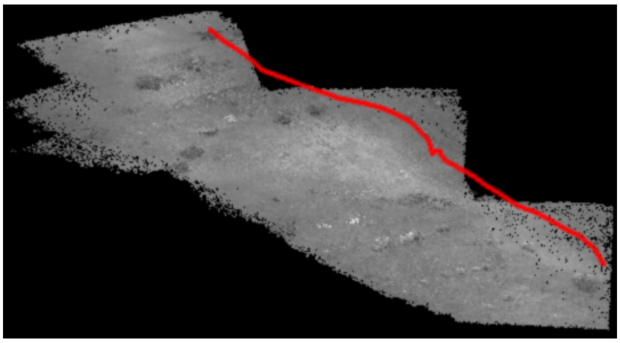

Figure 8. PRoVisG MARS 3D Challenge sequence. The result camera trajectory and 3d reconstruction illustrated. The reconstruction shows the part of ground which is visible during the sequence.

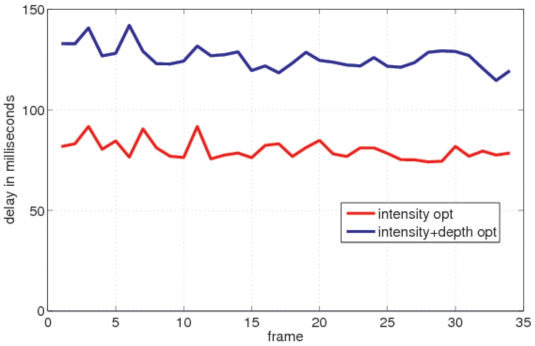

Figure 9. Optimization delay per frame in milliseconds with (blue) and without (red) depth component. Incorporated depth component increases computational requirement by 50%. For full real-time system disparity maps must be generated by external hardware/GPU.

7. Conclusions

In this work depth maps have been incorporated into image-based pose estimation as a new measurement. This increases accuracy in pose tracking when having non-Lambertian surfaces and prevents singular Hessian which can occur due to homogeneous texturing. The improvements are demonstrated in a simulation with known ground truth trajectories. The algorithm avoids costly and error prone feature extraction and matching as in ICP. The downside, however, is the requirement of an additional parameter λ for adjusting the balance between the cost components. Finally the combination cost function is used in PRoVisG MARS 3D Challenge where the minimization produces qualitatively convincing 3D camera trajectory and reconstruction from the given rover stereo sequence.

References

[1] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch. Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation. In F. J. Kurdahi and J. Takala, editors, *ICSAMOS*, pages 93–101. IEEE, 2010.

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

[3] M. Calonder, V. Lepetit, and P. Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision (ECCV)*, 2010.

[4] A. Comport, E. Malis, and P. Rives. Real-time quadrifocal visual odometry. *International Journal of Robotics Research, Special issue on Robot Vision*, 29(2-3):245–266, 2010.

[5] A. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):615–628, 2006.

[6] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1052–1067, 2007.

[7] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):932–946, July 2002.

[8] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communication of the ACM*, 24(6):381–395, June 1981.

[9] H. Hirschmuller. Stereo processing by semi-global matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[10] M. Irani and P. Anandan. About direct methods. *ICCV workshop on Vision Algorithms*, pages 267–277, 1999.

[11] K. Konolige and M. Agrawal. Frameslam: from bundle adjustment to realtime visual mappping. *IEEE Transactions on Robotics*, 24:1066–1077, 2008.

[12] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[13] M. Meilland, A. Comport, and P. Rives. A spherical robot-centered representation for urban navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010.

[14] J. Michot, A. Bartoli, and F. Gaspard. Bi-objective bundle adjustment with application to multi-sensor slam. In *3DPVT*, 2010.

[15] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *International Conference On 3-D Digital Imaging and Modeling*, 2001.

[16] G. Silveira, E. Malis, and P. Rives. An efficient direct approach to visual slam. *IEEE transactions on robotics*, 24:969–979, 2008.

[17] C. Tomasi and S. Birchfield. Depth discontinuities by pixel-to-pixel stereo. *International Conference on Computer Vision*, pages 1073–1080, 1987.