Ego-motion estimation using rectified stereo and bilateral transfer function

Giorgio Panin and Nassir W. Oumer

German Aerospace Center (DLR)
Institute for Robotics and Mechatronics
Münchner Straße 20, 82234 Weßling

Abstract. We describe an ego-motion algorithm based on dense spatio-temporal correspondences, using semi-global stereo matching (SGM) and bilateral image warping in time. The main contribution is an improvement in accuracy and robustness of such techniques, by taking care of speed and numerical stability, while employing twice the structure and data for the motion estimation task, in a symmetric way. In our approach we keep the tasks of structure and motion estimation separated, respectively solved by the SGM and by our pose estimation algorithm. Concerning the latter, we show the benefits introduced by our rectified, bilateral formulation, that provides at the same time more robustness to noise and disparity errors, at the price of a moderate increase in computational complexity, further reduced by an improved Gauss-Newton descent.

1 INTRODUCTION

Visual odometry, or ego-motion estimation, is concerned with the estimation of one's own velocities into an unknown, mainly rigid environment, through sequences obtained from one or more cameras fixed on the moving body. In this context, motion is usually constrained to planar (3-dof) or full 6-dof, under the assumption of a rigid scene with a few independently moving items (such as pedestrians, cars) acting as an external disturbance, which are detected and factored out of the estimation procedure.

Model-based approaches, such as [1], use pre-defined models of shape and appearance to be sought in the image, and provide an efficient and robust estimation of absolute pose and motion. However, such methods require an apriori model of the object, which in many scenarios may not be available.

Feature-based methods use a combination of feature detection, matching, tracking, triangulation and pose estimation from corresponding points. Among such techniques, [2] uses RANSAC and iterative pose refinement for stereo and monocular odometry. A similar monocular technique, which minimizes drift using a local bundle adjustment, was presented in [3]. Integration of other sensory modalities, such as GPS or IMU, also allows robustly coping with fast motion as in the real-time 3D modeler [4]. A disadvantage of feature-based techniques is that errors incurred at intermediate processing stages propagate to a higher

level, and that for reliable tracking a sufficient number of features should be available at each frame, which is not always the case.

Direct methods instead use all possible information from the image, including weak gradient regions, to estimate pose and structure of a scene or an object, by minimizing a photometric error rather than a geometric distance between features. As discussed in [5], a major advantage is that feature extraction and matching are not required, while a very large set of measurements are simultaneously available (one per pixel) providing generally more precise and robust performances.

Related techniques, using appearance models and optical flow [6], employ an extended planar pattern for tracking. However, the underlying assumptions about motion (for example a planar homography [7]) or camera model (for example affine cameras [8]) are usually strong, so that they apply to a restricted class of scenes or objects.

A recently developed approach minimizes intensity errors between consecutive image pairs from calibrated stereo sequences [9], and can be considered between model-based and image-based approaches. In this context, dense stereo matching is used to obtain a reference model for motion estimation, that may be built off-line (from a set of key-frames) or updated at each frame. This model consists of a dense point cloud, including color and disparity information. A point transfer function based on the quadrifocal tensor function allows rigid motion estimation directly on the next stereo pair, with full 6-dof. The approach handles arbitrary 3D structures, and improves the convergence domain with respect to planar region-based methods, since the whole image is used for the registration task.

In this paper we improve the above mentioned approach in some important aspects. Firstly, we introduce a symmetric transfer error, simultaneously projecting points forwards and backwards over time, where the Jacobian of the inverse transformation is easily obtained in the Lie algebra setting, and it is computed once per frame using an inverse compositional formulation. Due to the fact that stereo matching on consecutive frames does not provide fully overlapping point clouds, as explained in Section 4.1, this scheme integrates additional information for a more accurate motion estimation. This formulation also respects the symmetry of the problem, since by inverting the image sequence we exactly obtain the inverse motion estimates.

Secondly, instead of trifocal tensors we utilize rectified images, and stereo triangulation in homogeneous coordinates. This results in a simpler formulation, clearly isolating computation into off-line (structure) and on-line (re-projection) terms, while keeping intermediate quantities within a good numerical range. Stereo matching is done here with the semi-global matching (SGM) algorithm [10] using mutual information.

The paper is organized as follows: in Sec. 2 we present the stereo-based estimation framework. The two sub-problems of structure and motion estimation are dealt with in Sec. 3 and 4, respectively. Afterwards, Sec. 5 presents experimental results on simulated and real stereo sequences, compared with ground-truth

trajectories. Sec. 6 concludes the paper, mentioning possible improvements to this system.

2 PROBLEM STATEMENT

In the following, we denote by l, r the left and right camera of the stereo rig, and by k the temporal frame index. Given four arbitrary (3×4) camera matrices $P_{l,r}^{k-1,k}$, two corresponding points in homogeneous coordinates $\mathbf{x}_{l,r}^{k-1}$ that satisfy the epipolar constraints (i.e. back-project to the same 3D point) can be transferred forwards in time to the corresponding points $\mathbf{x}_l^k$ and $\mathbf{x}_r^k$, by means of the respective trifocal tensors [9]. When the stereo rig is calibrated, but motion between $k-1$ and k is unknown, both tensors can be parametrized by the rigid transformation $T_l^{k-1,k}$ of the left camera frame, as well as differentiated through the Lie algebra of the Euclidean group.

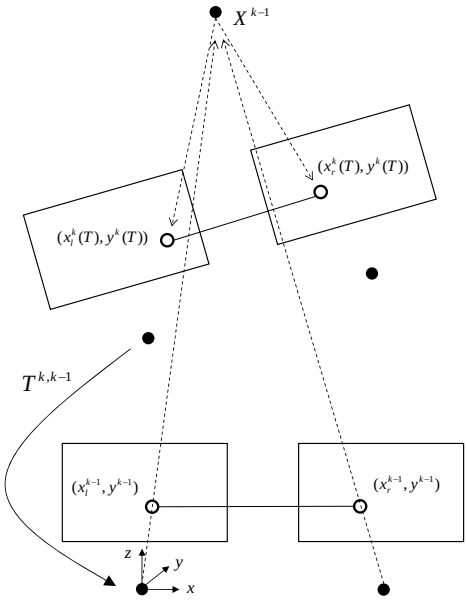

Fig. 1. Forwards re-projection after 3D triangulation.

For almost parallel configurations, we simplify the formulation and address normalization issues by considering only rectified images, and casting the trifocal transfer into a mere forwards reprojection (Fig. 1), given by:

1. *(2D-3D)* Structure estimation from the pair at $k-1$, by triangulation in homogenous coordinates

2. *(3D-2D)* Motion estimation, by minimizing photometric error of re-projected points at k

so that the point cloud is computed out of the motion estimation loop.

3 STRUCTURE ESTIMATION

As a starting point, we need a rectified stereo pair at time $k-1$. This can be done using knowledge about the external and internal camera matrices, through an accurate calibration procedure [11, Chap.7][12], for example using a planar chessboard pattern.

Once that camera parameters are known, stereo rectification [11, Chap.11] consists in rotating both cameras around the respective center, until the Cartesian frames are aligned, and the transformation becomes a pure horizontal translation. Furthermore, it requires to align also the projection planes, so that internal parameters become equal.

This is equivalent to apply a planar *homography* to the left and right image

$$H_l = K_l R_l K_l^{-1};\ H_r = K_r R_r K_r^{-1} \tag{1}$$

where $H_{l,r}$ are functions of the internal camera parameters $K_{l,r}$ and the rotation matrices $R_{l,r}$ that align the two camera frames[1].

In the end, we obtain the following projection matrices:

$$P_l = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix};\ P_r = \begin{bmatrix} f & 0 & p_x & -fT_x \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

where the rectified parameters are given by the common focal length f, the principal point (p_x, p_y), and the horizontal *baseline* T_x, expressed in metric units (e.g. mm).

Subsequent triangulation becomes then a trivial task: let $(x_l, y_l), (x_r, y_l)$ be a pair of corresponding poins, where $y_l \equiv y_r$ is the common coordinate, and let $X = (x, y, z, w)$ be the homogeneous coordinates of the corresponding 3D point, referred to the left camera frame. Then, we have

$$x = \frac{x_l - p_x}{f};\ y = \frac{y_l - p_y}{f};\ z = 1;\ w = \frac{x_l - x_r}{fT_x} \tag{3}$$

where coordinates are defined up to a scale factor, so we are free to choose $z = 1$, that keeps numerical stability and simplifies the computation of inverse-compositional Jacobians (see eq. 15). This representation also allows points at infinity, given by $w = 0$.

[1] The amount of image distortion introduced depends on the convergence angle between optical axes. Therefore, it is best applied to similar and almost-parallel cameras.

In order to perform dense stereo matching we use the state-of-the-art semi-global matching (SGM) algorithm [10],that nicely joins efficiency and robustness properties, through dynamic programming and mutual information. In absence of interpolation, missing disparities will occur over more or less large regions, so that the 3D point cloud will not be 100% dense.

4 MOTION ESTIMATION

The point cloud X^{k-1} is now re-projected on the next frame k by

$$\begin{bmatrix} \bar{x}_l^k \\ \bar{y}_l^k \\ \bar{z}_l^k \end{bmatrix} = P_l T X^{k-1}; \quad \begin{bmatrix} \bar{x}_r^k \\ \bar{y}_r^k \\ \bar{z}_r^k \end{bmatrix} = P_r T X^{k-1} \tag{4}$$

where T is the relative motion of left camera[2] and P_l, P_r are the constant matrices defined in (2), followed by normalization

$$x_l^k = \frac{\bar{x}_l^k}{\bar{z}_l^k}; \; y_l^k = \frac{\bar{y}_l^k}{\bar{z}_l^k} \tag{5}$$

We parametrize motion by using Lie algebras [13]

$$T = \bar{T} \exp\left(\sum_{i=1}^{6} \delta p_i G_i \right) \tag{6}$$

where G_i are the generators, providing a basis for the *tangent space* to the Euclidean group

$$\sum_{i=1}^{6} \delta p_i G_i = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{w} \\ \mathbf{0} & 0 \end{bmatrix} \tag{7}$$

using $[\cdot]_\times$ to denote the (3×3) cross-product matrix, and $\delta \mathbf{p} = [\boldsymbol{\omega}, \mathbf{w}]$ the twist velocity, so that the derivatives at $\delta \mathbf{p} = \mathbf{0}$ (i.e. $T = \bar{T}$) are

$$\frac{\partial}{\partial \delta p_i} \begin{bmatrix} x_l^k \\ y_l^k \\ x_r^k \\ y_r^k \end{bmatrix}_{\delta \mathbf{p} = \mathbf{0}} = \begin{bmatrix} J_n\left(\bar{x}_l^k, \bar{y}_l^k, \bar{z}_l^k\right) P_l \\ J_n\left(\bar{x}_r^k, \bar{y}_r^k, \bar{z}_r^k\right) P_r \end{bmatrix} \bar{T} G_i X^{k-1} \tag{8}$$

for $i = 1, \ldots, 6$, where

$$J_n(\bar{x}, \bar{y}, \bar{z}) = \begin{bmatrix} 1/\bar{z} & 0 & -\bar{x}/\bar{z}^2 \\ 0 & 1/\bar{z} & -\bar{y}/\bar{z}^2 \end{bmatrix} \tag{9}$$

is the Jacobian of the normalization (5), evaluated at $\bar{T}$.

[2] Notice that in this representation we have $T \equiv T_l^{k,k-1}$, i.e. the transformation from *current* to *previous* left camera frame.

Now we can compute the photometric error of a re-projected pair from $k-1$ to k, under the transformation $\bar{T}$

$$\begin{aligned} e_l^k &= I_l^k\left(x_l^k\left(\bar{T}\right), y_l^k\left(\bar{T}\right)\right) - I_l^{k-1}\left(x_l^{k-1}, y_l^{k-1}\right) \\ e_r^k &= I_r^k\left(x_r^k\left(\bar{T}\right), y_r^k\left(\bar{T}\right)\right) - I_r^{k-1}\left(x_l^{k-1}, y_r^{k-1}\right) \end{aligned} \tag{10}$$

where $(x_l^k, y_l^k, x_r^k, y_r^k)$ are given by eq. (3,4,5).

Derivatives of the residual with respect to local motion parameters δp_i are finally obtained, by taking the image gradients and multiplying them by the screen Jacobians

$$\begin{aligned} J_{l,i}^k &= \nabla I_l^k\left(x_l^k, y_l^k\right) \cdot \frac{\partial}{\partial \delta p_i} \begin{bmatrix} x_l^k \\ y_l^k \end{bmatrix}_{\delta \mathbf{p}=\mathbf{0}} \\ J_{r,i}^k &= \nabla I_r^k\left(x_r^k, y_r^k\right) \cdot \frac{\partial}{\partial \delta p_i} \begin{bmatrix} x_r^k \\ y_r^k \end{bmatrix}_{\delta \mathbf{p}=\mathbf{0}} \end{aligned} \tag{11}$$

for $i = 1, \ldots, 6$. At non-integer point coordinates (x, y), the corresponding image values and gradients are obtained by bilinear interpolation from the four nearest neighbors. Furthermore, we set a minimum threshold on image gradients in order to avoid uniform regions, that create ambiguities both for stereo matching and motion estimation.

By putting together all of these quantities in vector form $\mathbf{J}^k, \mathbf{e}^k$, where each row of the $(2n_{k-1} \times 6)$ Jacobian is given by the above derivatives, and n_{k-1} is the number of matching pairs at $k-1$, we can write the normal equations for the linearized LSE problem

$$\left(\mathbf{J}^{k,T}\mathbf{J}^k\right)\delta\mathbf{p} = \mathbf{J}^{k,T}\mathbf{e}^k \tag{12}$$

where $\mathbf{H}_{i,j}^k = \mathbf{J}^{k,T}\mathbf{J}^k$ is the Hessian matrix and $\mathbf{g}_i^k = \mathbf{J}^{k,T}\mathbf{e}^k$ the gradient.

All of these quantities are evaluated at the current $\bar{T}$ that, after solving eq. (12), is updated to $\bar{T} \leftarrow \bar{T} \cdot \exp(\sum_i \delta p_i G_i)$.

As a further speed-up, we apply the *inverse-compositional* method [7]: instead of computing the Jacobian $\mathbf{J}^k$ over I^k at each iteration, we rather evaluate it using ∇I^{k-1} at the identity transform $\bar{T} = I$, and call it $\mathbf{J}_0^{k-1}$, so that

$$\delta\mathbf{p} = -(\mathbf{J}_0^{k-1,T}\mathbf{J}_0^{k-1})^{-1}\mathbf{J}_0^{k-1,T}\mathbf{e}^k \tag{13}$$

where the sign also changes, because the linearized residual now depends on $\bar{T}$ through $I_{l,r}^{k-1}$ instead of $I_{l,r}^k$, as explained in [7]. The notation $\mathbf{J}_0^{k-1}$ may create some confusion, however we preferred it in order to underline that this is a quantity related to the previous frame, and not to the current one.

Dropping the 0 subscript, eq. (11) becomes

$$\begin{aligned} J_{l,i}^{k-1} &= \nabla I_l^{k-1} J_{n,l}^{k-1} P_l G_i X^{k-1} \\ J_{r,i}^{k-1} &= \nabla I_r^{k-1} J_{n,r}^{k-1} P_r G_i X^{k-1} \end{aligned} \tag{14}$$

where image gradients are taken at (x^{k-1}, y^{k-1}), and J_n is also evaluated at $\bar{T} = I$ with X given by (3), so that

$$J_{n,l}^{k-1} = \begin{bmatrix} 1 & 0 & -x_l^{k-1} \\ 0 & 1 & -y_l^{k-1} \end{bmatrix} \tag{15}$$

and similarly for $J_{n,r}^{k-1}$.

Also considering the simple structure of P_l, P_r, this provides a very fast computation. In fact, since the set of pairs is changing at each frame, $\mathbf{J}_0$ must be re-computed at each k. However, the cost of doing only one evaluation becomes negligible, with respect to the iterated point transfer and Gauss-Newton updates.

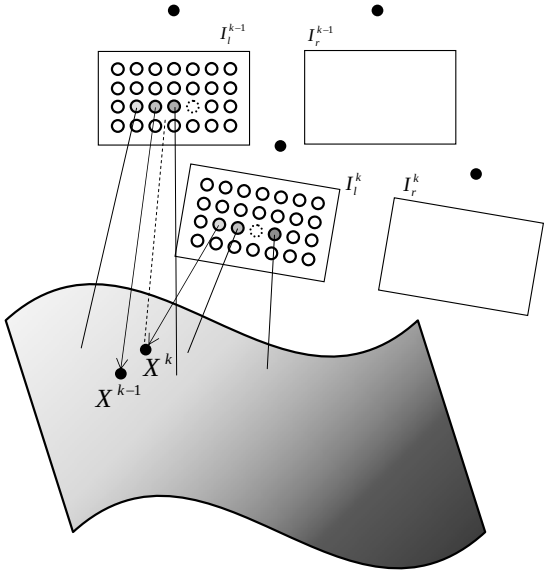

Fig. 2. Two point clouds at adjacent frames do not exactly overlap, neither in space nor in brightness, because of image resolution, noise, stereo matching errors, and occlusions due to motion. Therefore, a symmetric transfer error can improve motion estimation accuracy.

4.1 SYMMETRIC TRANSFER ERROR

The previous result applies to one-directional point transfer, where the previous pair $I_{l,r}^{k-1}$ plays the role of a template, reprojected onto $I_{l,r}^k$. Therefore, we may also add a *backward*-transfer term, parametrized by the inverse transformation T^{-1}, where the current stereo pair $I_{l,r}^k$ is the template, reprojected onto $I_{l,r}^{k-1}$.

In this way we have a symmetric error, roughly requiring twice the amount of computation for the motion estimation part[3].

The main motivation for that lies in the different point clouds $\{X^{k-1}\}, \{X^k\}$ sampled at consecutive frames (see also Fig. 2): in fact, due to many factors such as image resolution, occlusions on boundaries, brightness changes, shading effects, missing (or mis-matched) stereo disparities etc., the two point clouds will never exactly overlap, neither in spatial nor in brightness values, already for a small inter-frame motion. Therefore, they provide two different sets of measurements for the estimation of T, with a higher accuracy and robustness with respect to the mono-lateral case.

To make an interesting comparison, in a feature-based approach the symmetric formulation could be applied to the *geometric* re-projection error, as mentioned in [11, Chap. 4.2.2]. However, in that case temporal matching has already been performed through optical flow, so that the 4-tuple of corresponding points (at least under ideal conditions of an exact localization) is supposed to come from the *same* 3D point $X^k \equiv X^{k-1}$, and therefore no significant benefit is observed by adding the backwards term.

The backwards re-projection error is then

$$\begin{aligned} e_l^{k-1} &= I_l^{k-1}(\bar{T}^{-1}) - I_l^k \\ e_r^{k-1} &= I_r^{k-1}(\bar{T}^{-1}) - I_r^k \end{aligned} \tag{16}$$

and inverse-compositional Jacobian

$$\begin{aligned} J_{l,i}^k &= -\nabla I_l^k J_{n,l}^k P_l G_i X^k \\ J_{r,i}^k &= -\nabla I_r^k J_{n,r}^k P_r G_i X^k;\ i = 1, \ldots, 6 \end{aligned} \tag{17}$$

The opposite sign of eq. (17) is because the backwards-projection Jacobian is computed on the tangent space at $\bar{T}^{-1}$. In fact, we have $(\bar{T}\delta T)^{-1} = \delta T^{-1}\bar{T}^{-1}$, where the inverse of the exponential matrix is $(e^M)^{-1} = e^{-M}$, with $M = \sum_{i=1}^6 \delta p_i G_i$, and $\bar{T} = I$, so that G_i are simply replaced by $-G_i$.

Finally, the overall Hessian matrix is

$$\mathbf{H}_{i,j} = \sum_{l=1}^{n_{k-1}} J_{l,i}^{k-1} J_{l,j}^{k-1} + \sum_{r=1}^{n_{k-1}} J_{r,i}^{k-1} J_{r,j}^{k-1} + \sum_{l=1}^{n_k} J_{l,i}^k J_{l,j}^k + \sum_{r=1}^{n_k} J_{r,i}^k J_{r,j}^k \tag{18}$$

that is computed only once per frame, while the gradient

$$\mathbf{g}_i = \sum_{l=1}^{n_{k-1}} J_{l,i}^{k-1} e_l^k + \sum_{r=1}^{n_{k-1}} J_{r,i}^{k-1} e_r^k + \sum_{l=1}^{n_k} J_{l,i}^k e_l^{k-1} + \sum_{r=1}^{n_k} J_{r,i}^k e_r^{k-1} \tag{19}$$

is updated at each step.

[3] Instead, the triangulated structure at frame k, as well as $\mathbf{J}_0^k$, are re-used for the next forwards transfer, from k to $k+1$.

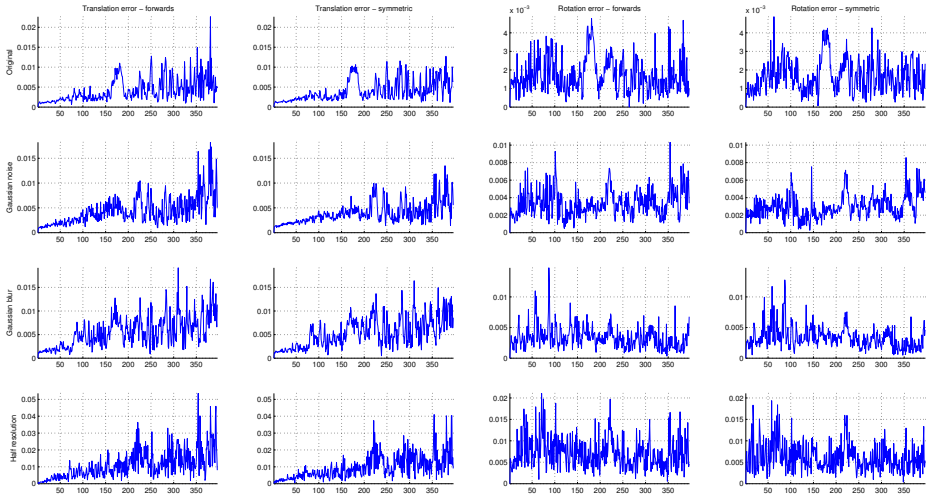

Fig. 3. Frame-to-frame motion estimation errors for different versions of the car sequence, comparing forwards-only with symmetric transfer errors. Translations are given in m, rotation angles in deg.

5 EXPERIMENTAL RESULTS

In order to test our formulation, we performed experiments on sequences involving simulated and real camera images.

For each sequence, comparisons are done with ground truth data, obtained respectively by the simulation environment (hence, exact) or by different sensory data, such as GPS/IMU or robot kinematics. The latter also require knowledge of the rigid transform between the left camera and the sensor, obtained through a *hand-eye* calibration procedure [14], to determine the transformation between the robot TCP and the left camera frame Therefore, accurate ground truth is provided by the direct robot kinematics, through the absolute angular measurements from the joints. Since we do pure frame-to-frame estimation, without keeping a constant structure, we only compare incremental motion $T^{k-1,k}$.

The code has been implemented in C++ on a multi-core (Intel Xeon W3530) CPU with 2.8 GHz and 5 GB RAM, however without exploiting parallelization. Concerning processing times, for both sequences (at VGA resolution 640×480) we observe an average of 0.5 sec/frame using the symmetric transfer error, that decreases to 0.25 sec/frame for the mono-lateral case, as previously explained. However we emphasize that, on a dual-core platform, the symmetric reprojection function can be easily splitted into the two terms (forwards and backwards) at each LM iteration.

The car-driving simulation has been taken from a public dataset of the University of Auckland[4], also related to the work [15]. This sequence consists of

[4] http://www.mi.auckland.ac.nz

396 stereo frames, with a baseline of 0.3 m, showing navigation on a road with textured pavement, and forest trees surrounding it. The road goes briefly uphill and downhill, and makes a brief left turn at the end. Several cars appear in the field of view, from crossroads or running along the opposite lane, resulting in outliers that are detected by the algorithm.

The original image sequence is perfecly rectified, and no image noise is present, so that almost an ideal disparity map (dense and accurate) is obtained from the SGM algorithm, limited to a depth range of $z_{min} = 1$m, $z_{max} = 100$m. In this idealized scenario, we obtain good results for both the forwards and symmetric reprojection error (top row of Fig. 3), however we already notice an improvement in accuracy for the translation. The overall frame-to-frame translation error is around 1 cm/frame, for a motion of $50 - 100$ cm/frame. Rotation errors are very low (about 0.005 deg/frame), although the car maintains almost a constant attitude, with the exception of a left turn at the end, of about 0.3 deg/frame.

Subsequently, we tested the same sequence in the presence of image noise, blurring and lower resolution, all of them affecting the estimated disparities and colors. These conditions were obtained, respectively, by adding Gaussian noise with $\sigma = 10$ in the range $[0, 255]$, by Gaussian filtering with $\sigma = 2.5$ pixels, and by sub-sampling to half resolution. As we can see from the related plots in Fig. 3, benefits of the symmetric error become more evident; a combined effect of those disturbances has not yet been tested.

Next, we consider a real camera sequence, recorded by a small-baseline stereo rig (5 cm) mounted onto an industrial manipulator, after performing stereo as well as hand-eye calibration. The sequence consists of 1190 frames, showing a miniature model of the *Neuschwanstein* castle, put on a white table and surrounded by the robot arm, that performs a smooth full 6-dof trajectory. Here we use a depth range of $z_{min} = 0.05$ m, $z_{max} = 2$ m. As a result, the average motion error is about 0.1 mm/frame and 0.02 deg/frame, with the exception of frame 400, where an unpredicted large inter-frame motion caused an error of about 1 cm and 1 deg.

By considering the absolute errors, over a forth-and-back sweep covering about 2 m and 360 deg, using symmetric transfer the estimated trajectory accumulates a maximum drift of 25 mm and 4 deg, both corresponding to roughly 1% final error, which is a good result since no drift reduction (e.g. by means of keyframes selection and wide-baseline matching), is done. Using the forwards-only transfer, the final errors grow up to 35 mm and 5 deg.

6 CONCLUSIONS

We presented an efficient and accurate method for visual odometry in rectified stereo cameras, based on symmetric pixel transfer and photometric error minimization, making use of stereo triangulation in homogeneous coordinates at both consecutive frames. This method naturally handles normalization issues, while efficiently splitting computation between structure and motion estimation, the

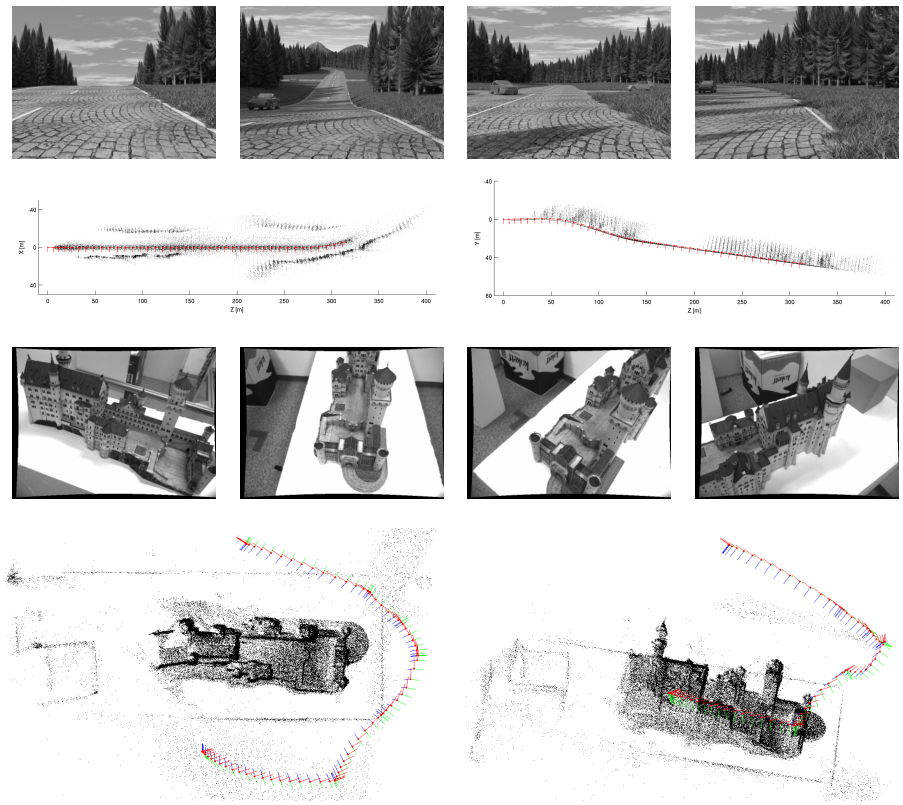

Fig. 4. Reconstructed map (subsets of points, re-projected onto the first camera frame) and left camera trajectories, from two different viewpoints. A few undistorted and rectified frames are also shown.

former executed once per frame, the latter in a Levenberg-Marquardt optimization loop, with outlier rejection and multi-resolution matching.

The current system employs a semi-global matching algorithm for computing dense stereo disparities, that can be accelerated by means of existing GPU [16] or FPGA implementations. The same applies to the odometry algorithm, concerning the computation of per-pixel reprojection errors and Jacobians.

Other issues may concern the robustness of the cost function to photometric outliers, due to shading or specularity effects, global brightness and contrast variations, as well as independently moving objects. In this context, apart from the standard rejection scheme of the present work, further improvements may be obtained by introducing more general cost functions, including explicit modeling of local illumination [17], or mutual information [18].

References

1. A.Comport, E.Marchand, M.Pressigout, F.Chaumette: Real-time markerless tracking for augmented reality: the virtual visual servoing framework. IEEE Trans. on Visualization and Computer Graphics **16** (2006) 615–628
2. Nistér, D., Naroditsky, O., Bergen, J.R.: Visual odometry. In: CVPR (1). (2004) 652–659
3. Mouragnon, E., Dekeyser, F., Sayd, P., Lhuillier, M., Dhome, M.: Real-time localization and 3d reconstruction. IEEE conference of Vision and Pattern Recognition,Washington DC, (**1**)
4. Strobl, K.H., Mair, E., Bodenmüller, T., Kielhöfer, S., Sepp, W., Suppa, M., Burschka, D., Hirzinger, G.: The Self-Referenced DLR 3D-Modeler. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA (2009) 21–28 best paper finalist.
5. Irani, M., P.Anandan: About direct methods. In Proc. Workshop Vision Algorithms:Theory Practice. (1999) 267–277
6. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision (darpa). In: Proceedings of the 1981 DARPA Image Understanding Workshop. (1981) 121–130
7. Baker, S., Matthews, I.: Equivalence and efficiency of image alignment algorithms. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **1** (2001) 1090
8. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1025–1039
9. Comport, A.I., Malis, E., Rives, P.: Real-time quadrifocal visual odometry. I. J. Robotic Res. **29** (2010) 245–266
10. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. **30** (2008) 328–341
11. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
12. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1330–1334
13. Drummond, T., Cipolla, R.: Visual tracking and control using lie algebras. In: CVPR. Volume 02., Los Alamitos, CA, USA, IEEE Computer Society (1999) 2652–2659
14. Strobl, K.H., Hirzinger, G.: Optimal hand-eye calibration. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006, October 9-15, 2006, Beijing, China. (2006) 4647–4653
15. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behavior on synthetic and real-world stereo sequences. In: 23rd International Conference of Image and Vision Computing New Zealand (IVCNZ '08). (2008) 1–6
16. Ernst, I., Hirschmüller, H.: Mutual information based semi-global stereo matching on the gpu. In: Advances in Visual Computing, 4th International Symposium, ISVC 2008, Las Vegas, NV, USA, December 1-3, 2008. Proceedings, Part I. (2008) 228–239
17. Silveira, G., Malis, E., Rives, P.: An effecient direct approach to visual slam. IEEE Transactions in Robotics **24** (2008) 969–979
18. Dame, A., Marchand, E.: Mutual information-based visual servoing. IEEE Trans. on Robotics **27** (2011)