Stereo Odometry – A Review of Approaches

Niko Sünderhauf, Peter Protzel
Department of Electrical Engineering and Information Technology
Chemnitz University of Technology
09111 Chemnitz
Germany
{niko.suenderhauf, peter.protzel}@etit.tu-chemnitz.de

***Abstract*— Estimating its ego-motion is one of the most important capabilities for an autonomous mobile platform. Without reliable ego-motion estimation no long-term navigation is possible. Besides odometry, inertial sensors, DGPS, laser range finders and so on, vision based algorithms can contribute a lot of information. Stereo odometry is a vision based motion estimation algorithm that estimates the ego-motion of a stereo camera through its environment by evaluating the captured images.**

In this paper, we want to give an integrated overview of stereo odometry and the accompanying literature. We want to emphasize the fact that stereo odometry is a chain of several single subprocesses where each relies on its predecessor's results. A variety of exchangable methods for each of these subprocesses is available. The key to a more accurate and efficient stereo odometry lies in an integrated analysis of its single subprocesses and the many algorithms available.

I. INTRODUCTION

First of all, why do we need stereo odometry in robotics? Classical odometry that calculates the robot's movements from counting the revolutions of the robot's wheels is often deceived by wheel slippage, especially in outdoor terrain. GPS information may not always be available with the desired quality. Be it because the robot is operating in areas where GPS transmissions can not be received like in forests or mines or because the robot happens to be operating on a planet that has not yet been equipped with a set of GPS satellites, like on Mars. Cheng et al. [1] give an interesting insight into the importance of stereo odometry during NASA's MER missions with the rovers Spirit and Opportunity. Other recent applications of stereo odometry on different types of robots in different environments can be found in [2] [3] [4] [5] [6] [7] [8] [9].

Stereo odometry can determine the ego-motion of the stereo camera in all 6 degrees of freedom that are possible in a 3D world: 3 for translation and 3 for rotation.

The process of stereo odometry follows a certain scheme:

1) Acquire a pair of images from the left and right camera of our stereo rig.
2) Find interest points (you may also call them features or landmarks if you prefer) in the images.
3) Calculate the 3D coordinates of those interest points.
4) Match the interest points between images taken from different viewpoints.
5) Use the matches to calculate the motion (which means combined translation and rotation) between the two viewpoints.

A very comprehensive and excellent textbook regarding stereo vision is Hartley and Zisserman's *Multiple View Geometry in Computer Vision* [10]. Many basic but also many advanced techniques and algorithms are discussed there.

II. FINDING AND MATCHING INTEREST POINTS

Finding interest points in the images is the first step on our road to stereo odometry. Several approaches and operators are known and we want to give a short discussion about them.

The operators used to find interest points have to fulfill certain demands regarding stability and repeatability. What is meant by that? Suppose the projection of a distinct worldpoint (let it be the corner of a desk) is marked as an interest point in one image. Now after the camera undertook a certain motion (including translation and rotation) the same scene is seen from another viewpoint and projected onto an image by the camera. Of course we expect our operator to find the projections of the very same world points to be interest points again, even if we look at the scene from a different viewpoint. The same is true if the camera just stands still and does not move. We certainly do not want the interest points to flicker around in the image.

The most commonly used operator for finding interest points is the well known Harris operator [11] which has been proven to be very stable in the above sense by Schmid et al. [12]. The standard way to match these interest points between two images is using a similarity measure of its neighboring pixels, for instance by determining the correlation between them. See Martin & Crowley [13] for a comparison of several correlation techniques. Another even simpler way to measure the similarity of two neighborhoods in two images is to use a sum-of-absolute-differences approach where the corresponding pixels from the two images are subtracted pairwise and the absolute difference of their greyvalues is summed up. The computation of this SAD is certainly faster then a naive implementation of a correlation based method, although Nister et al. [2] argue that their normalized correlation implementation using MMX instructions is as fast as SAD.

All these rather simple techniques work well in cases where the viewpoint between two images did only change a little,

so that no large scale changes or large rotations occured. If that is the case, however, the standard Harris operator [11] does not perform well anymore. Dufournaud et al. [14] introduced a scale adaptive version of the Harris operator. The application of this new operator to match feature points between two images that differ by a signigicant scale change up to factor 6 is described in [15]. This paper also presents a descriptor that is invariant to rotation and illumination changes. The presented matching results are quite impressive, although no discussion regarding the performance of the proposed method compared to similar techniques is given in the paper. Jung [16] proposed a sophisticated point matching algorithm based on the scale adaptive harris operator. The points are not matched pairwise, but groupwise. This leads to very stable and reliable matches. This approach has also been used in [5] and for SLAM on an airship [17].

Another, maybe better known approach towards scale invariant feature descriptors is SIFT (scale invariant feature transform) introduced by David Lowe in [18] and [19]. Both SIFT and the scale adaptive Harris operator use scale-space approaches [20] to achieve the desired scale invariance. First successful tests of using SIFT features in a stereo odometry framework have been described by Se et al. [21]. Mikolajczyk et al. [22] compared SIFT against other feature descriptors and proved the expected high stability of SIFT.

A recent novel approach to detect rotation and scale invariant features has been proposed and evaluated by Bay et al. [23]. These so called SURF (Speeded Up Robust Features) are found to be superior to SIFT regarding both computational time and stability after first tests conducted by the authors and also outperform the affin invariant interest point detector proposed in [24].

Other interesting current developements like the work of van de Weijer et al. [25] search for ways to incorporate color information into the interest point detection and matching process, as other methods simply work on a single grayvalue channel.

As can be seen from this short list of different interest point detectors and descriptors, a huge variety of algorithms is available. Usually, a certain tradeoff between quality and performance needs to be found. The standard method of using Harris features and a simple SAD matching strategy is very fast compared to more sophisticated approaches like SIFT or SURF. On the other hand these rather simple strategies may spawn many false or inaccurate matches so that more time has to be spent for outlier removal or during optimization of the motion estimates later on.

III. ERROR MODELING IN STEREO VISION

Given the two images of a stereo camera, several methods are known to calculate the 3D coordinates for certain pixels in the images. Scharstein and Szeliski [26] give a comprehensive overview of so called dense stereo algorithms that try to calculate 3D information for every pixel in the image. We want to take a short look at sparse stereo, i.e. we review how to calculate 3D information for a single world point only.

In the last section we repeated a variety of algorithms for identifying and matching certain feature points between two images. We can use these algorithms to find corresponding points between the left and right image in a stereo camera. Given the corresponding projections $\mathbf{x} = (u, v)^T$ and $\mathbf{x}' = (u', v')^T$ of a world point $\mathbf{X}$ in the two images of a stereo camera, how can we restore its 3D coordinates? First, the internal and external camera parameters (focal length f, principal point $(p_x, p_y)^T$, and length of the baseline between the two camera centers B) have to be known. This can be achieved by calibration (see [27] for exhausive material and Matlab resources regarding camera calibration). Second, the images have to be free of disturbances from lens effects and rectified in a way that the two image planes are coplanar and the pinpoint camera model [10] can be applied.

If we then write $d = u - u'$, we can use the following homogeneous equation to get the world point $(X/s, Y/s, Z/s)^T$ from an homogeneous point in so called disparity space $(u, v, d, 1)^T$:

$$\begin{pmatrix} X \\ Y \\ Z \\ s \end{pmatrix} = Q \begin{pmatrix} u \\ v \\ d \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & -p_x \\ 0 & 1 & 0 & -p_y \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{-1}{B} & 0 \end{bmatrix} \begin{pmatrix} u \\ v \\ d \\ 1 \end{pmatrix} \tag{1}$$

Here, u and v are the image coordinates of $\mathbf{x}$ in the *left* image. d is the *disparity*, $d = u - u'$, the difference of the u-coordinates (horizontal image dimension) of $\mathbf{x}'$ in the right image and $\mathbf{x}$ in the left image. The length of the baseline is given by B.

If we write this matrix multiplication explicitly and transform the result into non-homogeneous coordinates, we yield:

$$\begin{aligned} X &= -\frac{B(u - p_x)}{d} \\ Y &= -\frac{B(v - p_y)}{d} \\ Z &= -\frac{Bf}{d} \end{aligned} \tag{2}$$

These equations describe the world point $\mathbf{X}$ where the rays from the left camera center through $\mathbf{x} = (u, v)^T$ and from the right camera center through $\mathbf{x}' = (u', v')^T$ intersect. Unfortunately, image sensors are of a discrete nature, they are build from discrete pixels. So there are no intersecting rays but intersecting pyramid-like bodies in space. The intersection of these bodies is not a single point as one might believe after seeing the above equations, but another body in space. The real coordinate of the sought 3D worldpoint can be anywhere inside this body. See figure 1 for illustration.

Although many interest point detectors return image coordinates with sub-pixel resolution, these coordinates are pertubed by a certain ammount of noise. This noise can

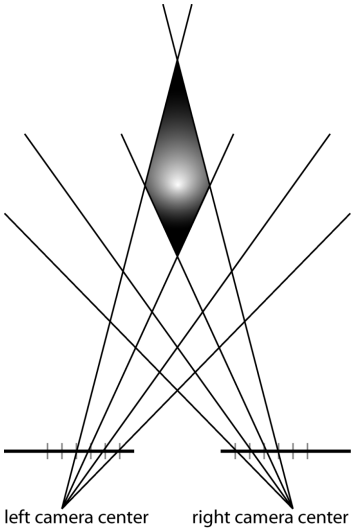

Fig. 1. The triangulation error under central projection arising from the discrete nature of imaging devices. The triangulated world point might be anywhere inside the shaded diamond area.

be modeled to be Gaussian with mean $\mu = (u, v)^T$ and covariance matrix

$$\Sigma_{pixel} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \tag{3}$$

where the σ are the standard deviations in pixel coordinates in x and y direction respectively.
Writing (2) as a vector function $\mathbf{f}$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{f}_{(u,v,u',v')^T} = \begin{pmatrix} -\frac{B(u-p_x)}{u-u'} \\ -\frac{B(v-p_y)}{u-u'} \\ -\frac{Bf}{u-u'} \end{pmatrix} \tag{4}$$

we can set the covariance matrix of the input vector to be a stacked version of (3):

$$\Sigma = \begin{pmatrix} \Sigma_{pixel} & \mathbf{0} \\ \mathbf{0} & \Sigma_{pixel} \end{pmatrix} = \begin{pmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_x^2 & 0 \\ 0 & 0 & 0 & \sigma_y^2 \end{pmatrix} \tag{5}$$

As $\mathbf{f}$ is a nonlinear function of a random vector with mean $(u, v, u', v')^T$ and covariance Σ, the covariance matrix of the 3D point calculated by $\mathbf{f}$ is given according to [10] as

$$\Sigma_{3D} = J\Sigma J^T \tag{6}$$

where J is the Jacobian matrix of $\mathbf{f}$ evaluated at $(u, v, u', v')^T$:

$$J = \begin{pmatrix} \frac{B}{u-u'} - \frac{B(u-p_x)}{(u-u')^2} & 0 & \frac{B(u-p_x)}{(u-u')^2} & 0 \\ \frac{-B*(v-p_y)}{(u-u')^2} & \frac{B}{u-u'} & \frac{B(v-p_y)}{(u-u')^2} & 0 \\ \frac{-Bf}{(u-u')^2} & 0 & \frac{Bf}{(u-u')^2} & 0 \end{pmatrix} \tag{7}$$

An early attempt to include such an error model of stereo vision into motion estimation algorithms has been published by Matthies & Shafer [28]. As [28] and [29] point out, this gaussian error model is just an approximation of the real random pertubation. However, this approximation is good enough to be used in a maximum likelihood approach of stereo odometry presented in [4]. We will come back to that method during the next section of this paper.

IV. SIMPLE ESTIMATION OF MOTION PARAMETERS

In the last sections we reviewed how interest points can be found and matched between two images. Section III showed how to calculate the 3D coordinate of a world point given its two projections. We saw how an Gaussian error model can be retrieved.
Suppose we are given two sets of rigid 3D-points $X = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ and $Y = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n\}$ where $\mathbf{X}_i$ and $\mathbf{Y}_i$ are the 3D-coordinates of the i-th interest point *before* and *after* the motion. In the ideal case, there is a unique solution for the motion parameters R and t so that

$$\mathbf{Y}_i = R\mathbf{X}_i + t \qquad \forall i \tag{8}$$

However, as both X and Y will be disturbed by some ammount of noise, we formulate: What is the translation t and rotation R that transforms X into Y so that the mean squared error ϵ^2 given by

$$\epsilon^2_{(R,t)} = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{Y}_i - (R\mathbf{X}_i + t)\|^2 \tag{9}$$

becomes minimal?
This problem of determining the relative motion that transforms a set of 3D-points into another is well known in the computer vision community. The literature knows several solutions, like [30], [31], [32], [33] or [34].

A. A solution based on a singular value decomposition

The method described next was published in [35] and bases on [31] and [32] but corrects a mistake which led to wrong results in some degenerated cases.
The idea behind the algorithm is to decouple translation and rotation. The coordinates of the points $\mathbf{X}_i$ and $\mathbf{Y}_i$ relative to their centroid μ_x and μ_y will be equal before and after the transformation. This is simply because the transformation by R and t is an euclidean transformation and does not affect the relative position of the points to each other. They are moved like one *rigid* body. Given this information. we can split the original problem into two parts:

1) Find R to minimize

$$\epsilon^2 = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{Y}_i - R\mathbf{X}_i\|^2 \tag{10}$$

2) Then the translation t is given by $t = \mu_y - R\mu_x$.

The minimization problem in (10) can be solved using the *singular value decomposition* SVD:

1) Calculate the centroids

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i \tag{11}$$

$$\mu_y = \frac{1}{n}\sum_{i=1}^{n} \mathbf{Y}_i \tag{12}$$

2)

$$\Sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{Y}_i - \mu_y)(\mathbf{X}_i - \mu_x)^T \tag{13}$$

3) Let UDV^T be the singular value decomposition of Σ_{xy}, $\mathrm{SVD}(\Sigma_{xy})$.

4)

$$S = \begin{cases} I, & \text{if } \det(U)\cdot\det(V) = 1 \\ \mathrm{diag}(1,1,\ldots,-1), & \text{if } \det(U)\cdot\det(V) = -1 \end{cases} \tag{14}$$

5)

$$R = USV^T \tag{15}$$

6)

$$t = \mu_y - R\mu_x \tag{16}$$

B. A Solution Based on the Essential Matrix

Another solution to the problem of determining the motion parameters R and t can be retrieved using the essential matrix as proposed by [10] and [36].
Given two sets of matched image points $x = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and $y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$ the *fundamental matrix* F is defined by

$$\mathbf{x}F\mathbf{y} = 0 \tag{17}$$

for every pairwise matched $(\mathbf{x}, \mathbf{y})$. An overview of several algorithms for calculating the fundamental matrix can be found in [10]. They should not be reconsidered here. Once the fundamental matrix F is known, one can calculate the *essential matrix* E:

$$E = K^T F K \tag{18}$$

K is the camera calibration matrix which has to be known:

$$K = \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix} \tag{19}$$

where f is the focal length and $(p_x, p_y)^T$ is the principal point. See [10] for details of the underlying pinpoint camera model.
The sought relative motion parameters R and t are contained in the essential matrix and can be retrieved like this: The singular value decomposition of E is given by

$$\mathrm{SVD}(E) = U\,\mathrm{diag}(1,1,0)V^T \tag{20}$$

The translation t is given up to a scale factor and unknown sign by $\pm\mathbf{u_3}$ where $\mathbf{u_3}$ is the third column of U. Rotation matrix R can be either

$$R = UWV^T \tag{21}$$

or

$$R = UW^TV^T \tag{22}$$

where

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{23}$$

So with two possible solutions for each of t and R there are four ambigious solutions and an unknown scale factor to the sought motion parameters. However, these ambiguities can be resolved easily as any arbitrary reconstructed 3D point will be in front of the second camera (described by R and t) only for one of the four solutions.

C. A Maximum Likelihood Solution

In section III we reviewed how to formulate a Gaussian error model for the triangulated 3D worldpoints. The above methods however did not make use of this error model. [4] proposed the following maximum likelihood approach to solve for R and t while taking the uncertainties in the 3D coordinates of the worldpoints into account. This approach has also been used during NASA's MER missions on the rovers Spirit and Opportunity [1].
Reformulating (8) we can write

$$\mathbf{Y}_i = R\mathbf{X}_i + t + e_i \tag{24}$$

where e_i is a zero-mean Gaussian error vector with covariance matrix Σ_i as it has been calculated in (6). The conditional probability for the observations $\mathbf{Y}_i$ given the motion parameters R and t can be written as

$$P_{(\mathbf{Y}_1,\mathbf{Y}_2,\ldots,\mathbf{Y}_n|R,t)} \propto e^{-\frac{1}{2}\sum_{i=0}^{n} r_i^T \Sigma_i^{-1} r_i} \tag{25}$$

which is a Gaussian distribution. Here

$$r_i = \mathbf{Y}_i - R\mathbf{X}_i - t \tag{26}$$

Obviously, minimizing the exponent will result in a maximized probability. Thus one has to solve

$$\min_{R,t}\sum_{i=0}^{n} r_i^T \Sigma_i^{-1} r_i \tag{27}$$

for the maximum-likelihood estimate of R and t. Because of the involved rotation this is a nonlinear minimization problem. We linearize it by taking the first order Taylor expansion with respect to the rotation. We assume Θ_0 to be the initial estimate on the rotation angles and R_0 the corresponding rotation matrix. Deriving from (24) we develop:

$$\mathbf{Y}_i \approx R_0\mathbf{X}_i + J_i(\Theta - \Theta_0) + t + \tilde{e_i} \tag{28}$$

J_i is the Jacobian of the rotation for the i-th point evaluated at Θ_0. The error vector $\tilde{e_i}$ now has covariance $\tilde{\Sigma_i} = \Sigma_i^a + R_0\Sigma_i^b R_0^T$ where Σ_i^a is the measurement covariance *after* the motion and Σ_i^b *before* the motion respectively. After this linearization we can solve (27) using a linear method. We now use $r_i = \mathbf{Y}_i - R_0\mathbf{X}_i - J_i(\Theta - \Theta_0) - t$ and $\tilde{\Sigma_i}$ for the covariance.
The reader is referred to [4] and [37] for further details.

A simpler weighted least-squares solution that only takes the volume of the error ellipsoid provided by Σ into account

but not its shape, is given in [1] as well:
Here we solve for R and t by minimizing

$$\min_{R,t} \sum w_i r_i^T r_i \tag{29}$$

where r_i is as defined in (26) and

$$w_i = (\det \Sigma_{\mathbf{X}_i} + \det \Sigma_{\mathbf{Y}_i})^{-1} \tag{30}$$

V. OUTLIER REMOVAL – ROBUST ESTIMATION

In the last section we reviewed three different algorithms that calculated the ego-motion of a stereo camera from two sets of matched points $\mathbf{x}_i$ and $\mathbf{y}_i$ and their corresponding 3D coordinates. All of the above algorithms will only work if all of the matches are correct and there are no outliers in the data. What are outliers in the terms of our motion estimation problem? Remember that we work with pairs of matched image points $(\mathbf{x}_i, \mathbf{y}_j)$. If all matches are correct, or only affected by Gaussian noise, the above algorithms will work well. But if some of the matches are wrong, so that $\mathbf{x}_i$ and $\mathbf{y}_j$ are not the projections of the same world point, the algorithms will fail. We can call these false matches outliers. However, there are robust methods that can cope with outliers in the data. So obviously, the above algorithms can not be used alone, without any sort of improvement, as long as we have to expect outliers in our matching data.
The literature knows a large number of robust estimation schemes. RANSAC [38] may be the best known of them. MSAC or MLESAC [39] or ASRC [40] are examples for more recent developements in this field.
Unfortunately, a comprehensive comparison of these robust estimators in a stereo odometry framework has not been conducted yet. Most robust methods generate a hypothesis of the sought solution (motion parameters in our case) from a small set of data points. The methods presented in section IV and especially the SVD-based method from IV-A are used as hypothesis generators. A scoring function evaluates how good that hypothesis fits to the rest of the data. This is repeated several times, depending on the algorithm. The best hypothesis is eventually returned as the robust solution. As this is an iterative process, the more outlier the data contains, the longer will it take to identify and discard them. Here we see the importance of a good and stable interest point identification and matching. Again, we have to find a certain tradeoff for the overall-process: Using fast interest point identification and matching algorithms is fast, but most likely produces many outliers that have to be filtered out by iterative robust algorithms later. On the other hand, using more sophisticated algorithms in the first place is slower but may hardly produce any outliers at all, so that the robust methods do not have to iterate through the dataset so many times.
Herein lies the need to consider the whole stereo-odometry process when comparing and evaluating single algorithms involved.

A. Outlier Removal Based on Geometric Constraints

An elegant non-iterative way to identify and remove outliers has been proposed by Hirschmüller et al. [8]. This method is based on certain geometrical constraints. Again consider the world points $X = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ and $Y = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n\}$. In a static environment (i.e. no moving objects observed by the camera) the relative distances between two transformed points are constant before and after the motion. So

$$\|\mathbf{X}_i - \mathbf{X}_j\| = \|\mathbf{Y}_i - \mathbf{Y}_j\| \tag{31}$$

and

$$(\mathbf{X}_i - \mathbf{X}_j)(\mathbf{Y}_i - \mathbf{Y}_j) > \cos\theta \tag{32}$$

where θ was set to be $\frac{\pi}{4}$. If both constraints (31) and (32) are true for the points with indices i and j then both points $\mathbf{X}_i$ and $\mathbf{X}_j$ may be correctly matched. If at least one of the constraints is false, at least one of $\mathbf{X}_i$ and $\mathbf{X}_j$ is not matched correctly. As (31) will almost never hold in reality due to the noise in the 3D coordinates retrieved by triangulations, the authors give a modified version of this constraint, taking an error model similar to the one formulated in section III into account. This error model is the important key to the usability of this approach; it would not be of any practical use without an error model of the 3D coordinates.
[8] also discusses how the complexity of finding the largest self-consistent set of points from X and Y so that the above constraints hold for all combinations of i and j within this set. Solving this seems to be a NP-hard problem, but the authors give a good and fast approximate solution.
Compared to other robust methods for outlier removal presented next, this approach based on general geometric constraints of rigid body movement seems to be superior. A direct comparison has not been conducted yet but the authors tested their approach with a real-world sequence and found the outlier detection taking only 1ms which is very fast.

B. Using RANSAC to Identify and Remove Outliers

RANSAC [38] is used in many implementations of stereo odometry to reject outliers. The work of David Nister [41], [2] but also [3], [9] or [1] and others more are examples of successful use of RANSAC in a visual odometry scheme. The method based on a singular value decomposition from section IV-A can be used as a hypothesis generator for the ransac scheme.
RANSAC is also used to estimate the fundamental matrix which is needed for the essential matrix algorithm from section IV-B. Hartley and Zisserman [10] describe this approach.

VI. CONCLUSIONS AND FURTHER WORK

As we have seen, stereo odometry is a process consisting of a chain of several subprocesses. A huge variety of mostly exchangable algorithms, approaches and theories is available for each of these subprocesses. This mere variety makes exhausting comparisons between different sets of used methods very hard. Although there are some studies that evaluated several approaches within a single subprocess (for instance matching or identification of interest points), a comprehensive combined evaluation of all subprocesses and their mutual side-effects in a stereo odometry framework has

not been conducted yet. At the moment, such a comparison involving the approaches mentioned in this paper is carried out by the authors of this paper, using both simulated and real-world data. We strongly believe that an integrated evaluation regarding every single link in the chain of subprocesses of stereo odometry could help a lot to find better tradeoffs between quality and efficiency in every single subprocess and the ones depending on it.

REFERENCES

[1] Yang Cheng, Mark W. Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. *IEEE Robotics and Automation Magazine*, 13(2):54–62, 2006.

[2] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, pages 652–659, 2004.

[3] Motilal Agrawal and Kurt Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *Proceedings of the International Conference on Pattern Recognition (ICPR06)*, 2006.

[4] C. Olson, L. Matthies, M. Schoppers, and Maimone Maimone. Robust stereo ego-motion for long distance navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-00)*, pages 453–458, Los Alamitos, June 13–15 2000. IEEE.

[5] Thomas Lemaire and Simon Lacroix. Vision-based slam: Stereo and monocular approaches. Technical report, LAAS-CNRS, 2006. submitted to IJCV/IJRR special joint issue.

[6] Niko Sünderhauf, Kurt Konolige, Simon Lacroix, and Peter Protzel. Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle. In Levi, Schanz, Lafrenz, and Avrutin, editors, *Tagungsband Autonome Mobile Systeme 2005*, Reihe Informatik aktuell, pages 157–163. Springer-Verlag, 2005.

[7] Niko Sünderhauf and Peter Protzel. Towards using bundle adjustment for robust stereo odometry in outdoor terrain. In *Proceedings of Towards Autonomous Robotic Systems (TAROS06)*, Guildford, UK, September 2006.

[8] Heiko Hirschmüller, Peter R. Innocent, and Jon M. Garibaldi. Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics. In *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision*, pages 1099–1104, Singapore, December 2002.

[9] Motlal Agrawal, Kurt Konolige, and Luca Iocchi. Real-time detection of independent motion using stereo. In *Proceedings of the IEEE workshop on visual motion*, 2005.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[11] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference 1988*, pages 147–151, 1988.

[12] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[13] J. Martin and L. Crowley. Experimental comparison of correlation techniques. In *Proc. of International Conference on Intelligent Autonomous Systems*, Karlsruhe, March 1995.

[14] Yves Dufournaud, Cordelia Schmid, and Radu Horaud. Matching images with different resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, pages 612–618. IEEE Computer Society Press, Juin 2000.

[15] Yves Dufournaud, Cordelia Schmid, and Radu Horaud. Image matching with scale adjustment. *Computer Vision and Image Understanding*, 93(2):175–194, February 2004.

[16] Il-Kyun Jung. *SLAM in 3D Environments with Stereovision*. PhD thesis, LAAS, Toulouse, 2004.

[17] E. Hygounenc, I-K. Jung, P. Soueres, and S. Lacroix. The autonomous blimp project at laas/cnrs: Achievements in flight control and terrain mapping. *International Journal of Robotics Research*, 33(4/5):473–512, 2004.

[18] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision, 60, 2*, pages 91–110, 2004.

[19] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.

[20] T. Lindeberg and Bart M. ter Haar Romeny. Linear scale-space. In Bart M. ter Haar Romeny, editor, *Geometry-Driven Diffusion*, pages 1–77, Dordrecht, Netherlands, 1994. Kluwer Academic Publishers.

[21] Stephen Se, David G. Lowe, and Jim Little. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. In *International Journal of Robotics Research, 21, 8*, pages 735–758, 2002.

[22] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[23] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, May 2006.

[24] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[25] Joost van de Weijer, Theo Gevers, and Andrew D. Bagdanov. Boosting color saliency in image feature detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):150–156, 2006.

[26] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002.

[27] Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`, 2004.

[28] L. Matthies and S. A. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3:239–248, 1987.

[29] Larry Matthies. Toward stochastic modeling of obstacle detectability in passive stereo range imaginery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 765–768, Champaign, Illinois (USA), 1992.

[30] Robert M. Haralick, Hyonam Joo, Chung-Nan Lee, Xinhua Zhuang, Vinay G. Vaidya, and Man Bae Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446, November 1989.

[31] B. K. P. Horn. Closed–form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America*, 5(7):1127–1135, 1987.

[32] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5):698–700, 1987.

[33] B. K. P. Horn. Closed–form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987.

[34] T. S. Huang, S. D. Blostein, and E. A. Margerum. Least-squares estimation of motion parameters from 3–d point correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, Florida, 1986*, pages 24–26. IEEE Computer Society Press, 1986.

[35] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380, 1991.

[36] Berthold K.P. Horn. Recovering baseline and orientation from essential matrix. http://www.ai.mit.edu/people/bkph/papers/essential.pdf, 1990.

[37] Larry Matthies. *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University, October 1989.

[38] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[39] P. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[40] Hanzi Wang and David Suter. Robust fitting by adaptive-scale residual consensus. In *Proceedings 8th European Conference on Computer Vision*, Prague, 2004.

[41] David Nister. Preemptive ransac for live structure and motion estimation. In *Proceedings of the 9th International Conference on Computer Vision, Nice*, 2003.