A Real-Time System for Epipolar Geometry and Ego-Motion Estimation

Mårten Björkman, Jan-Olof Eklundh
Computational Vision and Active Perception Laboratory (CVAP)
Department of Numerical Analysis and Computing Science
Royal Institute of Technology (KTH), S-100 44 Stockholm, Sweden
Email: celle@nada.kth.se, joe@nada.kth.se

Abstract

A visually guided mobile platform acting in a dynamic environment needs information, derived from cues such as stereo and motion. In this paper a binocular stereo system, capable of dynamic vergence and ego-motion estimation, will be presented. The ability to dynamically verge the cameras is essential for visual tasks such as tracking, recognition and manipulation, since the computation of 3D geometry will be considerably simplified at fixation. We will show that the epipolar geometry of the presented system can be estimated in real-time. The number of degrees of freedom has been minimized, without affecting the flexibility of the system. Reconstructed 3D data is used in order to determine the ego-motion of the platform, at a very low computational cost. Finally, it will be shown how disparity maps and estimated ego-motion can facilitate the localization of independently moving objects.

1 Introduction

An important theme of computer vision concerns the development of seeing systems, such as a mobile robot provided with one or more cameras. Since such systems operate in a dynamic 3D world and also themselves are moving about, it is both a necessity and an opportunity for them to use 3D information, deriving in particular from stereo and motion cues. However, this raises several problems on estimating scene and ego-motion, computing disparities and depth, and on camera calibration. An additional issue is the need of algorithms allowing implementations in real-time.

In this paper we will consider a mobile system provided with a binocular head-eye system, that uses fixation, i.e. binocularly and dynamically directs its gaze towards points of interest in the 3D world. Fixation is not a necessity for computing 3D information; parallel cameras would do as well [16, 10]. However, it gives the advantage that the objects of interest can be kept within the field of view, which is especially important when the observer (robot) performs rotational movements and when objects in the world move very fast. Furthermore, as a result of fixation disparities near the fixation point are kept small, simplifying other processing. Finally, and as we will use in this paper, the computation of 3D geometry will be considerably simplified at fixation. Fast or near real-time computations are therefore feasible in this case. Of course, there is then a need for actually controlling the gaze, so that the fixation assumption holds. We will only partly address that problem here. It is left for future work. However, we and others have in earlier work shown that binocular, dynamic fixation is indeed feasible in real-time [5, 14, 4].

The system presented in this paper is able to continuously estimate the epipolar geometry of a stereo-head system and the ego-motion of the observer. The epipolar geometry estimation process will be described in Section 2, with simulation results presented in Section 2.4. Using triangulated 3D point information, a very simple and efficient method of determining ego-motion will be described in Section 3. Finally, the different components will be integrated into a complete system described in Section 4, showing experiments performed on a Nomad 200 platform in real-time.

2 Epipolar geometry estimation

In order for a binocular system capable of dynamic vergence to work continuously, the epipolar geometry has to be known at every instance of time. A technical solution is to use counters on the motors controlling the stereo-head, but this has limited applicability. As the observer interacts with a dynamic environment, disturbances easily make the counters lose track of the configuration. Sooner or later a recalibration of the whole stereo system will be required. An alternative approach is to perform the internal calibration in advance and then continuously update the external calibration, the orientations and locations of cameras, as the platform moves around in the environment. Always keeping an estimate of the epipolar geometry up to date is a considerably easier task, than calculating the epipolar geometry from two different cameras, without any prior knowledge at all. The most critical outliers can thereby be identified before the actual estimation process starts.

Typically researchers have been using the essential matrix [12] in order to determine the epipolar geometry. This matrix is often estimated from point matches, and due to outliers robust statistics need to be used [15, 18]. The benefit of the essential matrix is its ability to handle large differences in rotations and locations. In reconstruction wide baselines are preferable, because of the improved accuracy of the final reconstruction, but in the case of a typical stereo-head, such baselines are not at hand. In this presentation an alternative approach, originating from the study of structure from motion, will be used instead.

Let us consider the two images from the left and right camera, as being images from the same moving camera from one instance in time to another. The disparities, vertical and horizontal, can then be seen as approximations of the optical flow [11]. If the motions are too large, the approximation will break down, and in our case we are not trying to estimate motion, but rather the epipolar geometry between two separate cameras. However, it will be shown that the relatively short baseline and small vergence angles that are typically being used by a binocular stereo-head, will make the errors stay within the acceptable bounds. The trouble-spot of the essential matrix, almost parallel cameras systems, will turn out to be configurations that this alternative method can cover.

2.1 The stereo system

The first question we should ask ourselves is whether all degrees of freedoms have to be determined or if some of them can be ignored. If we consider the baseline as a fixed line between the two cameras, there are a total of six degrees of freedom. Each camera has three different rotational degrees of freedom, that is one for each axis. Since a rotation around the optical axis does not add any new information to the system, we keep these rotations fixed. The intention is to keep the cameras in fixation, which means that there is no relative tilt angle between the cameras. A joint tilt of both cameras, will not change the nature of the problem. The only two degrees of freedom left to estimate are the pan rotations. This configuration can be found as one of the special cases in [3]. It should be noted that even if the number of degrees of freedom has been minimized, we have not changed the flexibility of the system.

Assume that we have two sets of corner features, one set for each image, and that these features have been matched, creating a number of image feature pairs that each represent the same point in 3D space. Ignoring those geometry parameters not being used in the current system, the relation between the image feature (x_i, y_i) in the left image and and corresponding point in the right image is given by:

$$\begin{pmatrix} dx_i \\ dy_i \end{pmatrix} = \begin{pmatrix} 1 + x_i^2 \\ x_i y_i \end{pmatrix} \beta + \frac{1}{Z_i} \begin{pmatrix} T_x - x_i T_z \\ -y_i T_z \end{pmatrix}, \quad (1)$$

where horizontal and vertical disparities are given by dx_i and dy_i, and β denotes the vergence angle, that is the difference between the pan angles of the cameras. The translation, here represented by T_x and T_z, is the location of the right camera in the coordinate system defined by the left camera. Since the baseline is measured during the internal calibration, the magnitude of this translation is known. Thus the expression in Equation (1) may be rewritten into a form with the number of unknown parameters reduced by one, using the parameters $t = T_z/T_x$ and $\lambda_i = T_x/Z_i$, denoting the inverted and scaled depths.

Estimating both geometry and depths of a large number of points in a closed expression leads to an explosion of unknown parameters. Solving such a complex system in real-time is difficult. However, in many cases one can assume to have some knowledge about the depths. In some cases this can be derived from some other modality, but often we may at least have a general idea about how far objects in space are located from the observer. In a system like ours the ordering of disparities will always match the ordering of depths. If the median depth to points in 3D space is known, one may find an estimate of all individual depths, matching the median depth to the median horizontal disparity of the feature pairs.

2.2 An iterative algorithm

Our algorithm for estimating these parameters can be summarized as follows:

1. Initialize $\beta^{(0)}$ based on the expected median distance to points in 3D space and the median of horizontal disparities of extracted feature pairs. Let $t^{(0)} = 0$.
2. Update the inverted and scaled depths using the current estimated geometry. The depths are individually found using least squares.

$$\lambda_i^{(n)} = \frac{\mathbf{e_i^T b_i}}{\mathbf{e_i^T e_i}}, \quad \text{where} \quad (2)$$

$$\mathbf{b_i} = \begin{pmatrix} dx_i - (1 + x_i^2)\beta^{(n)} \\ dy_i - x_i y_i \beta^{(n)} \end{pmatrix} \quad \text{and} \quad (3)$$

$$\mathbf{e_i} = \begin{pmatrix} 1 - x_i t^{(n)} \\ -y_i t^{(n)} \end{pmatrix}. \quad (4)$$

3. Estimate the geometry of the stereo-head based on estimated depths, in a least square framework with all valid feature pairs included.

$$\begin{pmatrix} \beta^{(n)} \\ t^{(n)} \end{pmatrix} = (\sum_i \mathbf{A_i^T A_i})^{-1} \sum_i \mathbf{A_i^T d_i}, \quad (5)$$

$$\mathbf{A_i} = \begin{pmatrix} 1 + x_i^2 & -x_i \lambda_i^{(n-1)} \\ x_i y_i & -y_i \lambda_i^{(n-1)} \end{pmatrix} \quad \text{and} \quad (6)$$

$$\mathbf{d_i} = \begin{pmatrix} dx_i - \lambda_i^{(n-1)} \\ dy_i \end{pmatrix}. \quad (7)$$

4. Return to 2. until convergence is achieved.

2.3 Identifying outliers

As mentioned previously the whole process of estimating the epipolar geometry relies on the fact that we have been able to accurately match image features between the left and right images, and that both points of such a feature pair actually originates from the same point in 3D space. Due to specularities, occlusions and repeatable structures in the scene mismatches might occur and the data set will include outliers, that might affect the performance of the estimation. In a system like ours, where optimizations are done using least squares, it is essential that such outliers can be identified and omitted from the calculations.

The most critical outliers can be found simply by analyzing the disparities and excluding outliers with disparities that can be considered as highly unlikely. Using predefined values for the maximum possible vergence angle $\max(\beta)$ and the inverted scaled depths $\max(\lambda_i)$, the vertical disparities have to stay within bounds according to

$$|dy_i| \leq |x_i y_i| \max(\beta) + |y_i| \max(\lambda_i) + \sigma_y, \quad (8)$$

where σ_y is a margin related to the noise level of the feature extractor. As initial estimates of the geometry will become available, it will be possible to constrain the disparities even further.

It is also possible to weight the least squares expression in step 3, in order to further suppress points of large errors. Such weights can be defined as

$$w_i = \left\{ \begin{array}{ll} 1 - (\frac{\varepsilon_i}{\sigma_e})^2, & \text{if } |\varepsilon_i| < \sigma_e \\ 0, & \text{otherwise} \end{array} \right\}, \quad (9)$$

where $\varepsilon_i = \mathbf{d_i} - \lambda_i^{(n)} \mathbf{e_i}$ represent the errors from the optimization in step 2. In the study presented here, the combination of these two methods showed to be very efficient. However, as will be seen in the next section, if the initial depth values are too far away from the truth, correct feature matches might be eliminated by mistake, leading to serious errors in convergence.

2.4 Simulations

The method was evaluated performing a series of simulations based on randomly generated clouds of 3D points. The points were evenly spread in image space and in depth, located between 200 cms and 600 cms away from the observer. The baseline was set to 20 cms and the field of view was chosen to be $45°$ for both cameras. The 3D points were projected onto the 360×288 pixel image planes, with noise of 1 pixel in standard deviation added to their positions. Each simulation involved 500 image feature correspondences, out of which 20% were outliers. The outliers were generated in the same fashion, but with an additional noise source of 30 pixels in standard deviation.

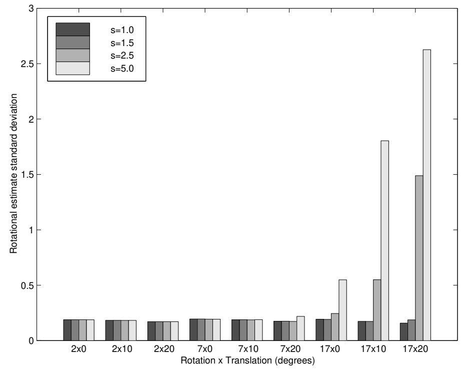

Figure 1. Standard deviation of the rotational component for different combinations of true rotational and translational components and errors in initial depths.

The bars in Figure 1 and 2 show the standard deviations of the results from simulations with different combinations of true rotational and translational components, where each simulation involves 1000 trials each. For every combination, different initial estimates of the median depth values were used, with the factor *s* representing the relative difference between the true median depth and the estimated one used in the simulations.

As seen the convergence breaks done if the initial depth estimate is too far away from its real value and the vergence angle is relatively large. However, for small vergence angles, cases where the essential matrix has significant difficulties, the algorithm shows no problems in converging, even if the initial depth estimate is totally wrong. Compared to the essential matrix, when simplified to the same number of degrees of freedom, the methods show similar performance in the rotational component, but the translational component is significantly easier to determine using the method presented here. The major advantage of the presented method is the speed. Typically around 25 iterations

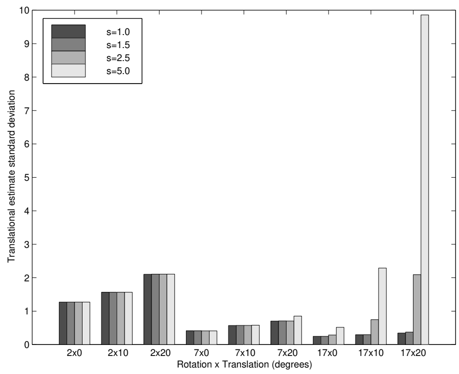

Figure 2. Standard deviation of the translational component for combinations of true rotational and translational components and errors in initial depths.

are necessary to achieve proper convergence, at a cost of about 15 ms on a 450 MHz Pentium III. Further analyses and comparisons to the essential matrix can be found in [2].

3 Ego-motion estimation

As mentioned previously knowledge about ego-motion is essential for an active observer to act in a dynamic environment. Based on ego-motion the gaze can be directed in such as way that objects of interests can be found and tracked as easily as possible. Ego-motion estimation based on a single camera [7, 13] has shown to be hard and usually requires considerably more computational power than what a real-time system would allow. One of the benefits of using a binocular setup is the way calculation of ego-motion can be simplified and also used to segment out independently moving objects in the scene. After the epipolar geometry has been determined and feature points have been reconstructed, the distribution of 3D points can be matched in time. When points have been matched the calculation of ego-motion is considerable less complicated than if it was based on 2D image features.

Typically, the components of the ego-motion are found minimizing the least-squares problem

$$\sum_{i=1}^{N} \|\mathbf{y_i} - (\mathbf{R}\mathbf{x_i} + \mathbf{t})\|^2, \tag{10}$$

where $\mathbf{R}$ is a 3×3 rotation matrix, $\mathbf{t}$ is a translation vector and, $\mathbf{x_i}$ and $\mathbf{y_i}$ represents the *i* th reconstructed feature point of two consecutive time frames. Decoupling of rotation and translation is possible considering the fact that the clouds of 3D points can be expected to have the same shape in both instances of time. The translation will then be given by the difference in position between the centers of the two clouds, that is $\mathbf{t} = \bar{\mathbf{y}} - \bar{\mathbf{x}}$. Subtracting the centers of the clouds from the 3D points, we get two new sets of points $\acute{\mathbf{x}}_\mathbf{i} = \mathbf{x_i} - \bar{\mathbf{x}}$ and $\acute{\mathbf{y}}_\mathbf{i} = \mathbf{y_i} - \bar{\mathbf{y}}$, and an objective function

$$f_1(\mathbf{R}) = \sum_{i=1}^{N} \|\acute{\mathbf{y}}_\mathbf{i} - \mathbf{R}\acute{\mathbf{x}}_\mathbf{i}\|^2. \tag{11}$$

3.1 SVD-based estimation of rotation

Horn [8, 9] gave two alternative ways of solving this problem, the first using quaternions and the second by fitting of orthogonal matrices. However, in this study an elegant approach based on singular value decomposition (SVD) and proposed by Arun et al [1] was used. It is clear that by rewriting the objective function in Equation (11) as

$$f_1(\mathbf{R}) = \sum_{i=1}^{N} (\acute{\mathbf{y}}_\mathbf{i}^\mathbf{T}\acute{\mathbf{y}}_\mathbf{i} + \acute{\mathbf{x}}_\mathbf{i}^\mathbf{T}\acute{\mathbf{x}}_\mathbf{i} - \mathbf{2}\acute{\mathbf{y}}_\mathbf{i}^\mathbf{T}\mathbf{R}\acute{\mathbf{x}}_\mathbf{i}). \tag{12}$$

the problem is equivalent to maximizing

$$f_2(\mathbf{R}) = \sum_{i=1}^{N} \acute{\mathbf{y}}_\mathbf{i}^\mathbf{T}\mathbf{R}\acute{\mathbf{x}}_\mathbf{i} = \mathrm{Trace}(\mathbf{RH}), \tag{13}$$

where

$$\mathbf{H} = \sum_{i=1}^{N} \acute{\mathbf{x}}_\mathbf{i}\acute{\mathbf{y}}_\mathbf{i}^\mathbf{T}. \tag{14}$$

Arun et al showed that the rotational component $\hat{\mathbf{R}}$ maximizing $f_2(\mathbf{R})$ can be found by first performing a SVD of $\mathbf{H}$. The decomposition $\mathbf{H} = \mathbf{UDV^T}$ consists of two orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$, and a diagonal matrix $\mathbf{D}$ of nonnegative elements. The minimizer of Equation (13) will be given by $\hat{\mathbf{R}} = \mathbf{VU^T}$. In order for $\hat{\mathbf{R}}$ to be a rotation, the determinant has to be equal to $+1$ and not -1, which corresponds to a reflection. However, if the points are not coplanar this is not a problem, since the determinant will always be $+1$.

In our study it was seen that most rotational errors were found in the magnitude, rather than in the orientation of the rotational axis. Based on this knowledge, the estimates were improved by a nonlinear optimization performed in the magnitude of rotation only, with features weighted as described below.

3.2 Region of uncertainty

So far it has been assumed that errors have the same distribution for all points and in every direction. It can be easily shown that this is not the case. Since the major source of error in the reconstruction is in the image feature extraction, a triangulated 3D point will then be located in a region of uncertainty as shown in Figure 3. The width W of this region will be proportional to the distance Z from the camera-set to the point in space. There is also a linear dependency between the elongation of the region and the distance, which means that the height H of the region increases quadratically as Z grows.

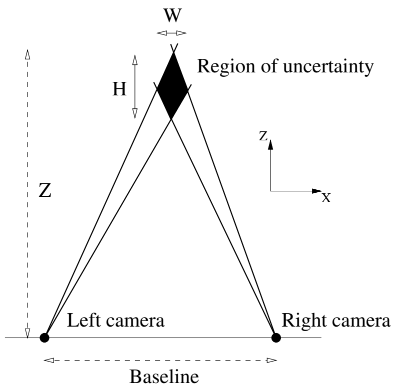

Figure 3. Region of uncertainty

Based on these considerations one might choose to weight the feature points in the objective function of Equation (11). In this study we use different scalar weights for every feature point. Preferably one would use an inverted covariance matrix instead, but this leads to a significantly more complex objective function, which is harder to minimize in real-time. A detailed analysis and treatment of this problem, as well as comparisons to the Cramér-Rao Lower Bound, can be found in [17].

3.3 Estimating translation

The translation can easily be found, calculating the center of the distributions corresponding to the two time frames. This is true at least where no outliers exist in the data set, but it is not at all true in practise. In our system the translation is estimated in two steps. The median translation is separately calculated for each of the three coordinates. The result is then fed as the starting value $\hat{\mathbf{t}}_\mathbf{0}$ of an M-estimator, that iteratively calculates the estimated translations as the weighted mean of the individual points according to

$$\hat{\mathbf{t}}_{\mathbf{n+1}} = \frac{\sum_{i=1}^{N} \sigma_i(\mathbf{y_i} - \hat{\mathbf{R}}\mathbf{x_i})}{\sum_{i=1}^{N} \sigma_i}. \tag{15}$$

The weights σ_i are supposed to reflect the certainty of the calculated translations and has the following definition,

$$\sigma_i = Z_i^{-1} \cdot \left\| \mathbf{y_i} - (\hat{\mathbf{R}}\mathbf{x_i} + \hat{\mathbf{t}}_\mathbf{n}) \right\|^{-1}. \tag{16}$$

The first factor corresponds to the width of the region of uncertainty being proportional to the depth Z_i of point $\mathbf{x_i}$ and the second factor is the error of the same point using the so far estimated rotation $\hat{\mathbf{R}}$ and translation $\hat{\mathbf{t}}_\mathbf{n}$. The process is iterated until convergence, which typically means less than ten iterations.

3.4 Identifying outliers

Outliers are identified and discarded twice in the process. Since the depths of points are invariant under rotation around the observer, it is possible to remove points that have unreasonably different depths from one instance in time to another. This can be done right after matching the image features and before the rotation is estimated. Most of the outliers that correspond to mismatches are identified here.

Outliers that correspond to independently moving objects usually survive this procedure and are being identified in a separate procedure right after the ego-motion has been found. Fortunately, the estimation of the rotational component is relatively insensitive to such outliers and in the case of translation the M-estimator is usually able to suppress outliers that otherwise would hurt the performance. After outliers have been identified, the process of estimating the ego-motion can be repeated, with points tagged as outliers excluded from the calculations. In the study additional iterations did not show to significantly improve the results.

3.5 Simulations

A number of simulations were done in order to evaluate the performance of the ego-motion estimation. The results are presented in Figure 4 below. For each entry 1000 simulations were performed, each including a set of 100 triangulated feature correspondences, out of which 20 are outliers. This corresponds to the typical number of points used in the real experiments presented below. The points are located between 200 and 600 cms away from the observer and randomly spread on the image plane and in depth. The outliers originate from an independently rotating and translating cylinder in the scene. The rotation is around the axis of the cylinder, which is parallel to the y-axis, and at a speed of $10°$, while the translation is about 10 cms in the y-direction.

To simulate errors in the feature extraction, noise with a standard deviation of 1 pixel in each direction was added to the image points, located on the 360×288 pixel image planes. The features were then triangulated back into 3D space, using a baseline of 20 cms. A systematic error of $0.3°$ in the vergence angle was used in this process, reflecting the uncertainty of the stereo system. It should be noted that it is relatively easy to adjust the points for this systematic error, since a small variation in vergence angle simply results in a scaling of reconstructed points. This scaling is easily calculated using the difference in spread of points in space from one frame to another.

$\angle\mathbf{t}$	$|\mathbf{t}|$	$\mathbf{R_y}$	$\angle\hat{\mathbf{t}}$	$|\hat{\mathbf{t}}|$	$\hat{\mathbf{R}}_{\mathbf{y}}$
0	0	0		0.20	0.02
0	0	2		0.21	0.01
0	0	4		0.22	0.01
0	3	0	1.41	0.19	0.02
0	3	2	1.58	0.22	0.02
0	3	4	2.00	0.22	0.03
0	6	0	0.75	0.22	0.02
0	6	2	1.22	0.25	0.03
0	6	4	1.57	0.28	0.05
45	3	0	2.71	0.22	0.02
45	3	2	4.75	0.41	0.06
45	3	4	7.64	0.63	0.11
45	6	0	1.45	0.25	0.03
45	6	2	2.09	0.42	0.07
45	6	4	2.76	0.62	0.12
90	3	0	3.15	0.20	0.02
90	3	2	4.57	0.37	0.05
90	3	4	6.15	0.61	0.09
90	6	0	1.75	0.23	0.02
90	6	2	2.29	0.39	0.05
90	6	4	3.34	0.64	0.09

Figure 4. The average mean error of the translational direction and speed, and pan rotational speed shown as functions of their true values.

In the table of Figure 4 the average mean errors of the estimated translational directions $\angle\hat{\mathbf{t}}$ and speeds $|\hat{\mathbf{t}}|$, and pan rotational speeds $\hat{\mathbf{R}}_{\mathbf{y}}$ can be seen as a function of varying true values. In all the cases the tilt angle was kept fixed. The error in its estimated speed was about $0.03°$ for each set of simulations. The translational directions and rotational speeds are shown in units of degrees, while the translational speeds are in units of cms. None of the estimated parameters show any significant bias and hence the actual calculated mean values were omitted from this presentation.

The estimated rotational components are relatively insensitive to translation, but increases almost linearly to the true rotational speeds. The translational speed shows a similar dependency on the motion of the cameras, whereas the directional errors decreases linearly as the observer speed increases. It is primarily due to the low accuracy of the triangulated depths, that makes the direction of small translations parallel to the image plane hard to determine. In the forward direction, the estimated directions will be less reliable if the observer is rotating simultaneously, than if no rotations are involved. In order to further improve the results, we believe that one would benefit from the use of Kalman-filters to track points over multiple frames.

4 System integration

In order for the system to work in practise, the ego-motion estimation procedure described above was integrated with the stereo system presented in Section 2. The complete system was implemented on a Nomad 200 platform with a 450 MHz Pentium III and runs in about 8 Hz. Disparity maps and occupancy grids require an additional 60 ms and are typically used in practical applications, such as obstacle avoidance and navigation. The system consists of the following components:

1. Corner feature extraction
2. Stereo matching of corner features
3. Estimation of epipolar geometry
4. Constrained stereo matching of features
5. Matching of corner features in time
6. Triangulation of 3D points
7. Ego-motion estimation

The corner extraction is performed using Harris [6] corner detector, with the code optimized for MMX, the parallel hardware capabilities of recent Intel and AMD processors, at a cost of approximately 15 ms per image. The corner features are then matched between the left and right images, using normalized cross-correlation. In order to avoid a combinatorial explosion of such correlations, care has been taken to reduce the search areas, using the knowledge of what camera configurations can be considered as possible.

After the epipolar geometry has been estimated the features are rematched, with search areas determined by the camera geometry, in order to maximize the number of matched feature pairs. After a similar matching procedure performed from one instance in time to another, the corresponding 3D points are found using triangulation. These 3D points are finally used in estimating the ego-motion as described in Section 3 above. The number of such points typically represents about one fourth of the original image feature points. The computational cost of determining the ego-motion is as low as 3 ms.

4.1 Experimental evaluations

The performance of the complete system was tested through a number of real-time runs, similar to those presented in Section 3.5, with a person walking around in the scene. Figure 5 shows the estimated vergence angle and gaze direction of a typical run. The values are processed through a moving average filter, with a time window of about 20 frames. The first frame in the figure corresponds to the moment when the whole estimation process starts. After about 80 frames the cameras are verged and at the 140th frame one of the cameras gets occluded by a hand covering the left camera for a few seconds. As seen the system quickly recovers from such a disturbance and is able to resume the estimation process.

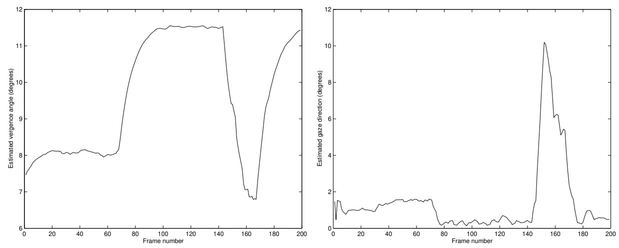

Figure 5. Estimated vergence angle and gaze direction during a period of 200 frames.

In Figure 6 the stereo correspondences between the two images, left and right, can be seen. The lines show the left image features that were possible to match with corresponding points in the right image. Features not being used in the epipolar estimation are simply shown as dots. As seen the assumption that there should not be any relative tilt between the cameras is in fact violated. In the system presented here the median vertical disparity is searched in a region along the x-axis of the images, and since this value is supposed to be zero, it is relatively easy to adjust for this problem.

Figure 6. Stereo correspondences

The performance of the ego-motion was also tested for each run, calculating the standard deviation of the estimated translational direction and speed, as well as of rotation. The estimated rotation turned out to be about as robust as expected, with a standard deviation seldom exceeding $0.1°$, and then only in cases when the rotation was more than $5°$ per update. Due to the difficulty of determining ground truth in case of both translation and rotation, the translational results were harder to evaluate. When moving at a speed of 3.3 cms per frame, the highest standard deviations of the translation, about $4.0°$ in direction and 0.7 cms in speed, were found in the forward direction.

There are a number of reasons why the real results look better than the simulated ones. First of all, corner feature noise with a standard deviation of 1 pixel was shown to be somewhat exaggerated. Especially the direction of translation when moving parallel to the image plane improves as the noise level is reduced. This is primarily due to the fact that depth estimates are so dependent on noise. Secondly, corner features of man-made structures tend to be more likely to survive the matching process in both stereo and time, whereas features of people moving around in the scene are harder to match. Thus the number of outliers due to independent motion turned out to be lower than expected.

4.2 Regions of independent motion

There are different ways of determining areas of independent motion in a sequence of images, and there is probably not a single one that works well under all conditions. One possible approach is analyzing the motion of all individual point matches that were rejected as outliers. Because of the uncertainty in especially the z-component of translations, this is not an easy task. In the system presented here an alternative approach based on the estimated ego-motion and calculated disparity map was used instead.

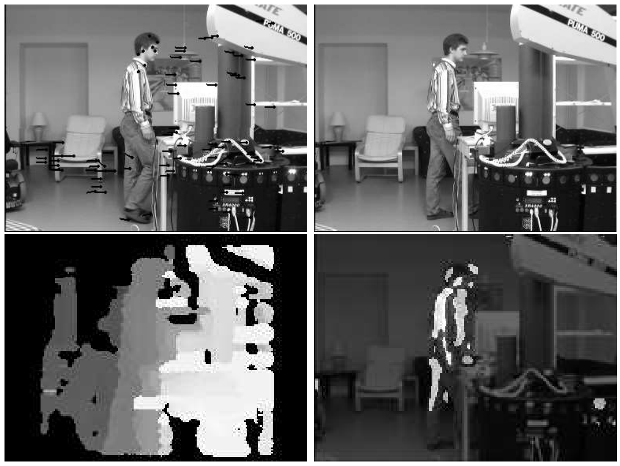

Figure 7. The upper row shows the left camera images from two consecutive frames, and features used in calculating ego-motion. The disparity map and regions of independent motion are shown in the bottom row.

Since the ego-motion can be used to determine the direction of optical flow and disparities corresponds to the magnitude of this flow, an image can easily be warped into an image, that ought to match the new image from the same camera in the next time frame. In the system described here, the warped image and the new image are then low-pass filtered and subtracted. The difference image is finally thresholded. Regions in the image above the threshold corresponds to regions where the optical flow does not match the epipolar geometry and estimated depth, e.g. regions of independent motion.

Figure 7 shows an example from an experiment performed using the procedure described above. The disparity map was generated using a method based on correlations and dynamic programming, followed by median filtering to suppress sparse erroneous data. Further details about the implementation can be found in [2]. In the bottom right image, the highlighted areas show the regions of independent motion that could be found during the experiment.

5 Discussion

In this paper we have shown how the calculation of epipolar geometry of a typical stereo-head system can be simplified. Without hurting the flexibility of the system, the number of degrees of freedom can be kept to a minimum. The epipolar geometry is estimated using an approximation of disparities, similar to what is typically done in structure from motion. The results show that such an approximation is feasible. Compared to methods based on the essential matrix, this method shows better robustness in case of near parallel camera systems. We also described a method, based on the reconstructed 3D data, for estimation of ego-motion at a very low cost. At the end of the paper it was shown how disparity maps and estimated ego-motion can be used to localize regions of independent motion.

The complete system, including frame grabbing, feature extraction, matching, epipolar geometry and ego-motion estimation runs in about 8 Hz on a 450 MHz Pentium III. It is intended to be used on a mobile robot, for various visually based tasks, such as navigation, obstacle avoidance, tracking and manipulation. Further studies have to be made testing the behaviour of the system on real image sequences. It is important for the system to properly handle cases when the number of available corner features is low. Non-retinal cues might be used to overcome such problems. Tracking of moving objects should also be added to the framework, so that their corresponding corner features can be excluded from the ego-motion estimation, further enhancing the performance. The signaling of errors also has to be improved and alternative models need to be tested.

Acknowledgement

The support from the Centre for Autonomous Systems (CAS) and the Strategic Research Foundation (Stiftelsen för strategisk forskning) is gratefully acknowledged.

References

[1] K.S. Arun, T.S. Huang and S.D. Blostein, Least-Squares Fitting of Two 3-D Point Sets, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(5), pp. 698-700, Sep 1987.

[2] M. Björkman and J-O. Eklundh, Real-Time Epipolar Geometry Estimation and Disparity, Tech. Report ISRN KTH/NA/P-00/09-SE, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Mar 2000.

[3] M.J. Brooks, L. de Agapito, D.Q. Huynh and L. Baumela, Direct Methods for Self-Calibration of a Moving Stereo Head, In *Proc. of the 4th European Conf. on Computer Vision*, pp. 415-426, 1996.

[4] K.J. Bradshaw, P.F. McLauchlan, I.D. Reid and D.W. Murray, Saccade and pursuit on an active head/eye platform, *Image and Vision Computing*, 12(3), pp. 155-163, Apr 1994.

[5] D. Coombs and C.M. Brown, Real-Time Binocular Smooth-Pursuit, *Int. Journal of Computer Vision*, 11(2), pp. 147-165, Oct 1993.

[6] C. Harris and M. Stephens, A Combined Corner and Edge Detector, In *Proc. of the 4th Alvey Conf.*, pp. 189-192, 1988.

[7] D.J. Heeger and A.D. Jepson, Subspace Methods for Recovering Rigid Motion I: Algorithm and Implementation, *Int. Journal of Computer Vision*, 7(2), pp. 95-117, 1992.

[8] B.K.P. Horn, Closed Form Solutions of Absolute Orientation Using Unit Quaternions, *Journal of the Optical Soc. of America A*, 4(4), pp. 629-642, 1987.

[9] B.K.P. Horn, Closed Form Solutions of Absolute Orientation Using Orthonormal Matrices, *Journal of the Optical Soc. of America A*, 5(7), pp. 1127-1135, 1988.

[10] K. Konolige, Small Vision Systems: Hardware and Implementation, *8th Int. Symposium on Robotics Research*, Hayama, Japan, 1997.

[11] H.C. Longuet-Higgins and K. Prazdny, The Interpretation of a Moving Retinal Image, *Proc. of Royal Society of London*, B-208, pp. 385-397, 1980.

[12] H. Longuet-Higgins, A Computer Algorithm for Reconstructing a Scene From Two Projections, *Nature*, 293, pp. 133-135, 1981.

[13] J. Oliensis, Computing the Camera Heading from Multiple Frames In *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 203-210, 1998.

[14] K. Pahlavan, T. Uhlin and J-O. Eklundh, Dynamic Fixation and Active Perception, *Int. Journal of Computer Vision*, 17(2), pp. 113-135, 1996.

[15] P.H.S. Torr and D.W. Murray, The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix, *Int. Journal of Computer Vision*, 24(3), pp. 271-300, 1997.

[16] V. Tucakov, M. Sahota, D. Murray, A. Mackworth, J. Little, S. Kingdon, C. Jennings and R. Barman, Spinoza: A Stereoscopic Visually Guided Mobile Robot, In *Proc. of the 13th Annual Hawaii Int. Conf. on System Sciences*, pp. 188-197, Jan 1997.

[17] J. Weng, P. Cohen and N. Rebibo, Motion and Structure Estimation from Stereo Image Sequences, *IEEE Trans. on Robotics and Automation*, 8(3), pp. 362-382, Jun 1992.

[18] Z. Zhang, R. Deriche, O. Faugeras and Q.T. Luong, A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry, *Artificial Intelligence*, 78, pp. 87-119, Oct 1995.