Real-Time Epipolar Geometry Estimation of Binocular Stereo Heads

Mårten Björkman and 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— **Stereo is an important cue for visually guided robots. While moving around in the world such a robot can use dynamic fixation to overcome limitations in image resolution and field of view. In this paper a binocular stereo system capable of dynamic fixation is presented. The external calibration is performed continuously taking temporal consistency into consideration, greatly simplifying the process. The essential matrix, which is estimated in real-time, is used to describe the epipolar geometry. It will be shown how outliers can be identified and excluded from the calculations. An iterative approach based on a differential model of the optical flow, as commonly used in structure from motion, is also presented and tested towards the essential matrix. The iterative method will show to be superior in terms of both computational speed and robustness, when the vergence angles are less than about** 15°**. For larger angles, the differential model will show to be insufficient and the essential matrix is preferably used instead.**

Keywords— **Epipolar geometry, Active vision, Real-time stereo, Dynamic vergence**

I. Background

Stereo vision can be used for several tasks in the area of robotics. For applications, such as visually guided manipulation, stereo is a valuable cue in order to recognize shapes and pose. Stereopsis has also been used for visual navigation and obstacle detection [11], [18]. Such systems typically consist of two or more small cameras located in parallel with relatively short baselines, since they are expected to work for various distances. Objects located close to the observer might otherwise only be seen by one of cameras, greatly complicating depth estimation and making the system useless for things such as obstacle avoidance. However, for manipulation accuracy is crucial and a wider baseline is desired.

This discussion leads us to believe that a mobile platform would benefit from an active stereo vision system with dynamic vergence. Based on the activity at hand, an active system can control its gaze in such a way that, the visual system is always used at its full potential. Ballard [1] gave a number of reasons how active vision can be used to overcome limitations in resolution and field of view, making the visual computations much less expensive, than that of a passive system. Using fixation many visual operations, such as visual navigation, 3D motion estimation and figure-ground segmentation, can further be simplified. Fermüller and Aloimonos [6] used normal-flow in conjunction with fixation to solve problems such as ego-motion recovery, 3D motion and time-to-impact estimation. Daniilidis [5] decoupled the optical flow and found the translation of an observer through two one-dimensional searches, radically simplifying the ego-motion estimation process.

One of the reasons why dynamic vergence has seldom been used in practical robotics, is the problem of external calibration. The relative orientation of the cameras has to be known, in order for the stereo vision system to compute accurate estimates of depths in metric space. Even if the mechanics of the system is highly advanced and the position of every joint can be measured at any given point in time, it might not be enough. Motor counters may still be used to get an initial estimate of the epipolar geometry, but if you like to calculate a dense disparity map and find image regions of independent motion, the epipolar geometry better be estimated using information available in the images themselves. A system based on the external calibration being performed continuously or at least every now and then, may also be cheaper to manufacture and more likely to handle disturbances the robot might suffer from.

In this work the essential matrix, as described by Longuet-Higgins [14], is used to establish the correspondence between two projections of a single scene. The fact that a stereo head system in fixation constrains the number of possible configurations, the number of degrees of freedom can be kept low. The essential matrix used here has earlier been presented in a review of stereo head configurations by Brooks et al [3]. The essential matrix is found using linear optimization and compared to results that of nonlinear optimization, as well as to a method [20] based on the full set of degrees of freedom. Robust statistics is utilized to make the results less sensitive to outliers in the dataset. Typically some kind of support measure is used to evaluate the sufficiency of estimates [7], [17] , but calculating such a measure can be relatively costly on a constrained system like a fixating stereo head system. Instead a simple analysis of eigenvectors is performed, reducing the computational cost without affecting the robustness of estimates. This will turn out to be at least one order of magnitude faster, than if support has to be calculated, making the speed sufficient for real-time use.

Structure from motion is commonly derived from differential motion, as described by Longuet-Higgins and Prazdny [13]. The use of differential motion in stereo is limited, because it is expected that the motion of the camera from one frame to another is small compared to the distances of objects in the scene. However, in this paper it will be shown that these requirements in fact hold for a stereo head system used under normal conditions. This is due to the fact that typical vergence angles are small and the baseline short compared to distances in 3D space. A simple iterative method for estimation of structure and geometry, similar to that of Heel [10], is presented and evaluated towards approaches based on the essential matrix. It will be shown that not just the speed, but also the robustness, will be superior to methods using the essential matrix. However, the method is based on the assumption that the median distance to objects in 3D space is approximately known. In most real application this is not a serious limitation.

The remainder of this paper is organized as follows. In Section II the essential matrix of a typical binocular stereo-head will be derived and an explanation of its parameters given. The same section also describes how the parameters can be estimated and outliers identified in an efficient manner. An alternative iterative approach based on a differential model is presented in Section III and tested towards methods using the essential matrix. Simulations are performed in Section IV and the performance of the different methods evaluated. The stereo system is then integrated into a complete system in which dense disparity maps are calculated. Finally, the system will be tested through a series of real-time experiments as described in Section V. The article ends with conclusions and a discussion about future research.

II. Epipolar geometry

A typical binocular stereo system consists of two different cameras separated by a fixed baseline, as shown in Figure 1. To determine metric distances to objects in 3D space viewed by the two cameras, the orientation of both cameras has to be known. In this section the essential matrix is used to describe such a relation. In a system where the cameras are in constant fixation, the structure of the essential matrix turns out to be simple.

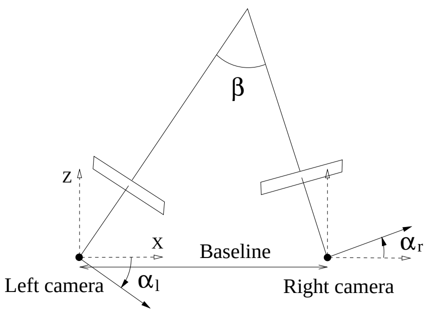

Fig. 1. Stereo-head system

We introduce a coordinate system by using the baseline to define an x-axis, as seen in Figure 1. At fixation the stereo-head is constrained, such that the two optical axes always intersect somewhere in-front of the cameras. This point together with the two camera centers define a plane and we may use the normal of this plane as y-axis, while the z-axis is located on the plane and perpendicular to the baseline. For an unconstrained system one might expect as many as six degrees of freedom, three rotational degrees for each camera. However, the full flexibility of the system might be kept, even if only two degrees are used. We do not need any rotations around the optical axes, because such rotations will not change the visual data, only its orientation. Since the cameras are kept fixated, the optical axes have to lie on the same plane and there is no relative tilt between the cameras. A joint tilt of both cameras around the baseline, will not change the nature of the problem and can be ignored.

Consequently, only two degrees of freedom need to be considered, that is the rotations around the y-axis of each camera, represented by the angles α_l and α_r. These rotations can be described by the corresponding rotation matrices

$$\mathbf{R}_{\alpha_\mathbf{l}} = \begin{pmatrix} \cos(\alpha_l) & 0 & \sin(\alpha_l) \\ 0 & 1 & 0 \\ -\sin(\alpha_l) & 0 & \cos(\alpha_l) \end{pmatrix}, \tag{1}$$

$$\mathbf{R}_{\alpha_\mathbf{r}} = \begin{pmatrix} \cos(\alpha_r) & 0 & -\sin(\alpha_r) \\ 0 & 1 & 0 \\ \sin(\alpha_r) & 0 & \cos(\alpha_r) \end{pmatrix}. \tag{2}$$

Using the rotation matrices we derive the projection matrices $\mathbf{P}_{\alpha_\mathbf{l}} = \mathbf{R}_{\alpha_\mathbf{l}}(\mathbf{I}|\mathbf{0})$ and $\mathbf{P}_{\alpha_\mathbf{r}} = \mathbf{R}_{\alpha_\mathbf{r}}(\mathbf{I}|-\mathbf{t})$. These two matrices describe how an object point $\mathbf{u} = (x, y, z, w)^T$, shown here in homogeneous coordinates, is being projected onto image planes of the left and right cameras. The image point in the left image, $\mathbf{x_l} = (x_l, y_l, w_l)^T$, is a result of the projection $\mathbf{x_l} = \mathbf{P}_{\alpha_\mathbf{l}}\mathbf{u}$. For the right camera the projection also includes a translation along the baseline, $\mathbf{t} = (k, 0, 0)^T$, which means that the right image point $\mathbf{x_r} = (x_r, y_r, w_r)^T$ can be expressed as $\mathbf{x_r} = \mathbf{P}_{\alpha_\mathbf{r}}\mathbf{u}$.

Using the fact that the image points, originating from a certain point in space, and the camera centers have to be located on the same plane, it is possible to derive a constraint known as the epipolar constraint. If an image point is known in one image, the corresponding point in the other image will be constrained to a line as defined by the following equation: $\mathbf{x_l^T}\underline{\mathbf{E}}\mathbf{x_r} = \mathbf{x_l^T}(\mathbf{R}_{\alpha_\mathbf{l}}\mathbf{T}\mathbf{R}_{\alpha_\mathbf{r}}{}^{\mathbf{-1}})\mathbf{x_r} = \mathbf{0}$, where $\mathbf{T}$ is the skew-symmetric matrix associated with the translation $\mathbf{t}$, that is

$$\mathbf{T} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & k \\ 0 & -k & 0 \end{pmatrix}. \tag{3}$$

After performing some simple calculations, we find that the essential matrix describing the epipolar constraint can be expressed by

$$\underline{\mathbf{E}} = k \begin{pmatrix} 0 & -\sin(\alpha_l) & 0 \\ -\sin(\alpha_r) & 0 & \cos(\alpha_r) \\ 0 & -\cos(\alpha_l) & 0 \end{pmatrix}. \tag{4}$$

This constraint has got one major advantage; the stereo problem has efficiently been decoupled from the structure problem. We can find the relation between the two cameras, without knowing the exact depth for each and every point seen by the cameras. The essential matrix can further be decomposed into a product $\mathbf{RS}$ of two separate components, a rotational component and a translational one. The component $\mathbf{R} = \mathbf{R}_{\alpha_l}\mathbf{R}_{\alpha_r}{}^{-1}$, is a rotation matrix that describes the relative orientation between the two cameras. This factor is important in order to relate the disparities to actual distances to points in space. The skew-symmetric translational component $\mathbf{S} = \mathbf{R}_{\alpha_r}\mathbf{T}\mathbf{R}_{\alpha_r}{}^{-1}$ can be considered as the position of the right camera relative to the left.

A. Linear optimization

As we saw earlier the true essential matrix can be found in a two-dimensional space of possible epipolar constraints. This space is spanned by the two angles α_l and α_r, and an optimal estimation of the true matrix $\underline{\mathbf{E}}$ could possibly be found solving a nonlinear optimization problem. However, such a problem typically relies on a number of iterations and an initial guess close to the true solution is critical. An alternative linear approach is finding the matrix $\mathbf{E}$ that minimizes the error $\mid \mathbf{x_l^T E x_r} \mid$, where

$$\mathbf{E} = \begin{pmatrix} 0 & f_1 & 0 \\ f_2 & 0 & f_4 \\ 0 & f_3 & 0 \end{pmatrix}. \tag{5}$$

After finding an optimum in the four-dimensional space spanned by the parameters f_j, $j = 1 \ldots 4$, a projection to the two-dimensional manifold of possible epipolar constraints is done, using an approach described by Hartley [9]. This method is based on the fact that a matrix can only be factored into $\mathbf{R}$ and $\mathbf{S}$, a characteristic feature of essential matrices as described earlier, if and only if the matrix has two equal non-zero singular values and one singular value equal to 0. Following Hartley, the singular value decomposition of $\mathbf{E}$ can be shown to be $\mathbf{UDV^T}$, where $\mathbf{U}$ and $\mathbf{V}$ are two orthogonal matrices and $\mathbf{D}$ is a diagonal matrix with the singular values $\sigma_1 = \sqrt{f_1^2 + f_3^2}$, $\sigma_2 = \sqrt{f_2^2 + f_4^2}$ and 0. A new essential matrix is found, substituting the epipolar geometry $\mathbf{E}$ with an approximation $\hat{\mathbf{E}} = \mathbf{U}\hat{\mathbf{D}}\mathbf{V^T}$, where $\hat{\mathbf{D}} = diag(\sigma, \sigma, 0)$ and $\sigma = (\sigma_1 + \sigma_2)/2$. This matrix is the one closest to the estimated matrix $\mathbf{E}$ under the Frobenius norm, that satisfies our requirements. We finally end up with an essential matrix that looks like the following:

$$\hat{\mathbf{E}} = \begin{pmatrix} 0 & \frac{f_1}{\sigma_1} & 0 \\ \frac{f_2}{\sigma_2} & 0 & \frac{f_4}{\sigma_2} \\ 0 & \frac{f_3}{\sigma_1} & 0 \end{pmatrix}. \tag{6}$$

B. Parameter estimation

As described in Section II-A the epipolar geometry can be found through a linear optimization in the four-dimensional space spanned by the parameters f_j, $j = 1 \ldots 4$. When estimating these parameters, it is expected that we have a number of extracted corner features. Each point in 3D space will be projected onto the two camera image planes, producing a pair of image features $(\mathbf{x_l^i}, \mathbf{x_r^i})$, as shown above. In our system, we use Harris' corner detector [8] to locate corners in the left and right images. These features have then to grouped into pairs, before the actual epipolar geometry estimation starts. A natural way to recover the epipolar geometry, could be finding the optimal estimate in a least square sense, using all the available correspondences found during the matching process. The more data we use, the better we may suppress the influence of noise. However, since the errors used in the optimization are squared, even single outliers in the data set could make the estimate useless. A common way of reducing the effect of such outliers, is the use of random sampling. Instead of using as many features as possible in the calculations, a minimum set of randomly selected feature pairs is used, so as to minimize the likelihood that the set includes one or more outliers.

We fit a hyper-plane $\mathbf{f} = (f_1, f_2, f_3, f_4)^T$, located in a four-dimensional space, to a set $M(j)$ of points given by $\mathbf{z_i} = (x_l^i y_r^i, y_l^i x_r^i, w_l^i y_r^i, y_l^i w_r^i)^T$ and the sum of squared orthogonal errors from the points to the hyper-plane is minimized, that is minimizing the expression

$$\sum_{i \in M(j)} (\mathbf{f^T z_i})^2 = \sum_{i \in M(j)} \mathbf{f^T}(\mathbf{z_i z_i^T})\mathbf{f} = \mathbf{f^T M f}, \tag{7}$$

subject to $\mathbf{f^T f} = 1$. The second moment matrix $\mathbf{M}$ is a 4×4 matrix with eigenvalues λ_1, λ_2, λ_3 and λ_4 in increasing order, and the corresponding eigenvectors $\mathbf{u_1}$, $\mathbf{u_2}$, $\mathbf{u_3}$ and $\mathbf{u_4}$. The eigenvector $\mathbf{u_1}$ is equivalent to the vector $\mathbf{f}$ that minimizes Equation 7 and λ_1 is the smallest sum of squared errors. The corresponding epipolar geometry, after performing a projection down to the two-dimensional manifold of possible essential matrices, is given by Equation 6. In order to further improve the accuracy of the estimate, Newton's method may be used to perform an additional nonlinear optimization on the manifold. The procedure is repeated numerous times, and for each set the estimated epipolar geometry is evaluated. A commonly used method known as the random sampling consensus paradigm (RANSAC) [7], [17], uses the complete set of correspondences in order to perform such evaluations. The estimate that the highest number of feature pairs can agree on, is selected as the best estimate of the true epipolar geometry. However, since every image feature is considered for each and every evaluation, this process can be time-consuming and might hurt the performance of the system. As will be described later in this paper, we propose just calculating the mean of all estimates that can be considered as reasonable.

Since speed is of major concern in a real-time system, it is important that computations are done as fast as possible. Since $\mathbf{u_1}$ is the only eigenvector we are really interested in, we do not bother to find all the other eigenvectors. Hence we may use an iterative approach such as the Inverse Power method, if we can make sure that convergence will be achieved. We begin calculating the adjoint of the matrix $\mathbf{M}$, that is $\mathbf{M_a} = adj(\mathbf{M}) = det(\mathbf{M}) \cdot \mathbf{M^{-1}}$. Using an arbitrarily chosen initial vector $\mathbf{v_0}$, we iterate a number of times and calculate

$$\mathbf{v_k} = \frac{\mathbf{M_a v_{k-1}}}{\mid \mathbf{M_a v_{k-1}} \mid}. \tag{8}$$

It can be shown that upon convergence $\mathbf{v_k}$ will converge towards $\mathbf{u_1}$, as $k \to \infty$, and the error will decrease at a rate of λ_1/λ_2. Analyzing the eigenvalues of $\mathbf{M}$ shows that there is a strong dependency between the scales of the eigenvalues and different aspects of selected points, such as the noise level of extracted corner features and how the points are distributed in image space and depth. The smallest eigenvalue λ_1 increases quadratically as the level of noise is doubled, but does not change very much as the distribution of points is varying. For the second smallest eigenvalue λ_2 the dependencies are reversed. As the points are spread twice as much over the image space or in depth, the eigenvalue doubles for each dimension. The same eigenvalue is, however, not very sensitive to noise. In conclusion, a low value of λ_1/λ_2 is equivalent to the selected feature points being widely spread in space and suffering from a limited amount of noise. Thus if convergence is achieved using the Inverse Power method, one can assume that a proper set feature points has been selected.

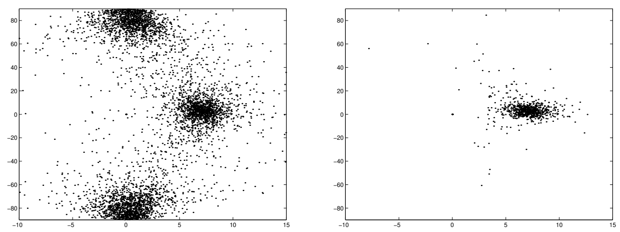

Fig. 2. Distribution of epipolar geometry estimates before and after estimates of low quality have been filtered out. The horizontal axis represents the vergence angle and the vertical one the gaze direction.

C. Quality testing estimates

Unfortunately, convergence is not enough to guarantee that an estimate is actually a good estimate, not too far away from the real epipolar geometry. The left image of Figure 2 shows a number of epipolar geometry estimates, calculated as described above. Further experiments will be presented in greater detail later in Section IV. As seen in the figure, the estimates are clustered around more than one center in the two-dimensional space of epipolar geometries. The cluster center near $(7, 3)$ corresponds to the true epipolar geometry. The existence of the two other clusters can be understood considering the following. In case the selected feature points are being influenced by a high level of noise or if the feature set includes one or more outliers, the two eigenvectors $\mathbf{u_1}$ and $\mathbf{u_2}$ may change order. This problem has been earlier been reported by Ma et al [15]. In case of the eigenvalues changing orders, the iteration in Equation 8 will converge towards the wrong eigenvector, represented by the two other cluster centers in Figure 2.

In order to separate such false samples from real ones, we employ two additional quality tests before we decide whether to accept a sample point as a good estimate of the epipolar geometry or not. The first test ensures that the sum of squared errors is reasonably small, that is the expression in Equation 7 has to be lower than a predefined threshold, which is usually set to a value in a neighbourhood of the variance of the corner detector. The other test involves a check whether every point in $M(j)$ is actually located in front of the cameras, when their three-dimensional positions have been reconstructed. After performing these two quality tests and omitting points that do not pass the tests, we end up with epipolar estimates clustered around one center only, as shown to the right in Figure 2.

III. An iterative approach

In the previous section we tried to estimate the epipolar geometry of a stereo-head system using the essential matrix. As described in Section II, the epipolar constraint can be seen as decoupling the structure from the stereo problem. Without knowing anything about structure, the relative location and orientation of the cameras can be extracted. However, in many applications the depths are in fact approximately known and the question is whether this knowledge can be used to simplify the problem. We might not have an accurate depth value for each individual image point, but at least the distribution of depths may be given. This is often the case for autonomous robots that are typically equipped with range sensors such as lasers and sonars.

Instead of using the essential matrix, one may view the problem as a differential motion problem. The left and right images can be considered as two images taken by the same cameras at two different instances in time, while the camera is moving in 3D space. The vertical and horizontal disparities can be used as an approximation of the optical flow. It can be argued whether such a treatment of the problem is feasible, since such approximations are known to break down if the motion from frame to frame is too large and in our problem we are in fact working with stereo, rather than motion. However, in this paper it will be shown that for a stereo head in fixation, where objects in space are located relatively far away from the platform and the vergence angles are small, one will stay within the bounds of acceptable results.

Considering a stereo-head as shown in Section II and following the derivations by Longuet-Higgins and Prazdny [13], an approximate relation between an image point (x_i, y_i) in the left and right image is given by:

$$\begin{pmatrix} dx_i \\ dy_i \end{pmatrix} = \begin{pmatrix} 1 + x_i^2 \\ x_i y_i \end{pmatrix} \beta + \frac{1}{Z_i} \begin{pmatrix} T_x - x_i T_z \\ -y_i T_z \end{pmatrix} \qquad (9)$$

The values dx_i and dy_i represent the horizontal and vertical disparities respectively, that is the difference in image position between the left and right images. Unlike the essential matrix, the depth Z_i of each point appears in the equation. The translational component, the position of the right camera relative to the left, is given by T_x and T_z, while β represents a rotation around the y-axis, that is the vergence angle, as seen in Figure 1.

One major advantage of the system we are using, is the fact that horizontal disparities of different points in space viewed by the two cameras, always have the same ordering [19], as the cameras are verged. This means that if the distribution of depths is known in advance, we can easily find an estimate of the vergence angle, matching the median depth value to the median horizontal disparity of the image features. The remaining parameters can be estimated by the iterative algorithm presented below. First the depth values Z_i are calculated using the current geometry and then the geometry parameters β, T_x and T_z are updated in a least square framework, with all valid features included. As seen in Equation 9, the depths and translation are related and can only be estimated up to a constant. A similar approach has been presented by Heel [10], even if his system was dealing with the full set of motion parameters and based on Kalman filters in order to track features in space.

A. The algorithm

Letting $t = T_z/T_x$ and denoting the inverted and scaled depth $\lambda_i = T_x/Z_i$, we come up with an iterative algorithm that can be summarized as follows:

1. Initialize the vergence angle $\beta^{(0)}$, matching the expected median distance to points in space to the median horizontal disparity of extracted feature pairs, and let the translational direction $t^{(0)} = 0$.

2. Update the inverted and scaled depths using the current estimated geometry.

$$\lambda_i^{(n)} = \frac{\mathbf{e_i^T b_i}}{\mathbf{e_i^T e_i}}, \quad \text{where} \tag{10}$$

$$\mathbf{b_i} = \begin{pmatrix} dx_i - (1 + x_i^2)\beta^{(n)} \\ dy_i - x_i y_i \beta^{(n)} \end{pmatrix}, \tag{11}$$

$$\mathbf{e_i} = \begin{pmatrix} 1 - x_i t^{(n)} \\ -y_i t^{(n)} \end{pmatrix}. \tag{12}$$

3. Use the point-wise errors of the previous step, $\varepsilon_i = \mathbf{d_i} - \lambda_i^{(n)} \mathbf{e_i}$ and define the weights

$$w_i = \left\{ \begin{array}{ll} 1 - (\frac{\varepsilon_i}{\sigma_e})^2, & \text{if } |\varepsilon_i| < \sigma_e \\ 0, & \text{otherwise} \end{array} \right\}. \tag{13}$$

4. Calculate the geometry parameters based on estimated depths, using the least square errors of all valid features.

$$\begin{pmatrix} \beta^{(n)} \\ t^{(n)} \end{pmatrix} = \frac{\sum_i w_i \mathbf{A_i^T d_i}}{\sum_i w_i \mathbf{A_i^T A_i}}, \quad \text{where} \tag{14}$$

$$\mathbf{A_i} = \begin{pmatrix} 1 + x_i^2 & -x_i \lambda_i^{(n-1)} \\ x_i y_i & -y_i \lambda_i^{(n-1)} \end{pmatrix}, \tag{15}$$

$$\mathbf{d_i} = \begin{pmatrix} dx_i - \lambda_i^{(n-1)} \\ dy_i \end{pmatrix}. \tag{16}$$

5. Return to 2. until convergence is achieved.

B. Reducing the effect of outliers

As mentioned earlier in Section II-B, we always have to expect that a considerable fraction of the data set suffers from outliers. It is essential for these outliers to be identified as early as possible, since they might otherwise seriously hurt the convergence. In the presented algorithm, the most obvious and dominating outliers can be identified making sure that the expression $|dy_i| \leq (|x_i y_i| \max(\beta) + |y_i| \max(T_z/Z_i) + \sigma_y)$ holds for all accepted feature pairs. The constants $\max(\beta)$, $\max(T_z/Z_i)$ and σ_y are predefined and represent the maximum possible vergence angle, the maximum inverted and scaled depth and the expected noise of the feature extractor respectively. Image correspondences that do not satisfy this condition are omitted from the data set and not used in the iterations.

In order to further suppress the effect of outliers, we use the weights w_i calculated in step 3, in the least square optimization. These weights reflect in what sense the horizontal and vertical disparities can agree on a single depth value. The parameter σ_e is a predefined value covering the expected noise of the feature extractor. It should be noted that the success of this procedure depends on the camera configuration being within bounds previously defined. In cases of unusual configurations or when the initial depths values are far away from the truth, correct feature correspondences may be eliminated by mistake and outliers falsely kept in the data set. The effect of this will be seen in the next section.

IV. Simulations

To evaluate the alternative methods, a number of simulations were performed. Series of 500 randomly generated points each were evenly spread between 10 and 30 baselines in-front of the cameras, for a number of different camera configurations. Each point was projected onto the 360×288 pixel image planes of the left and right cameras. For each projection noise with a standard deviation of about a pixel was added, reflecting the errors of the feature extraction. In order to test the influence of outliers, an additional 30 pixel noise source was added to about 20% of the feature pairs.

The first approach to be tested was based on linear optimization (**linear**) as previously described in Section II-B. Using random sampling a series of 10000 sets, $M(j)$, of six image correspondences each, were generated and for each set the essential matrix was calculated. The quality of the results were then tested as shown earlier, discarding estimates of low quality, i.e. sets that either include outliers or points that are not properly spread, which typically are as many as 90% of all estimates. The mean of the remaining results were finally used as our estimated epipolar constraint. As mentioned earlier the linear optimization procedure does not necessarily lead to an optimum in the two-dimensional domain of possible epipolar constraints. To make sure that a local minimum really is found, one may add an extra nonlinear optimization operation (**nonlinear**). This approach was tested using the same randomly generated sets of points and evaluated towards the linear method.

Instead of just using the mean sample point position as the final estimate of the epipolar constraint, one may also test in what sense a certain estimate is supported by the complete set of image correspondence, similar to what is done in RANSAC. In this approach (**support**), the epipolar estimate that has got the highest number of supporting feature pairs wins and will be used as the final estimate. We say the an estimate $\hat{\mathbf{E}}$ is supported by a pair of points $(\mathbf{x_l}, \mathbf{x_r})$, if $|\mathbf{x_l^T}\hat{\mathbf{E}}\mathbf{x_r}| < \sigma_E$. The constant σ_E is a predefined threshold, covering the expected noise of the corner extractions. There are primarily two reasons for keeping the degrees of freedom of our binocular system as few as possible, the speed and robustness. We tested the robustness, comparing the results to runs made with the full set of six degrees of freedom, using Least Median of Squares (**LMedS**), as proposed by Zhang et al [20]. In this approach random sampling is used as described above and for each set of image correspondences, $M(j)$, an estimate of the epipolar geometry is calculated. For every feature pair in the whole data set the squared error is then obtained and the set with the lowest median of squared errors is considered as the winner. In order to be able to compare the different methods, we decomposed the resulting essential matrix into a rotational and a skew-symmetric one. Their corresponding magnitudes of rotation and translation were then used in the evaluations. The final method that was tested, was the iterative approach presented in Section III (**iterative**). In order to test the robustness to errors in the expected median depth, simulations were performed using a different amount of initial errors.

A. Results

Simulations were performed using the five different methods mentioned above. The resulting vergence angles and gaze directions can be seen in Figure 3 and Figure 4 respectively. The upper graphs show the mean values and standard deviations of the first four methods, whereas the lower graphs represent the results based on **iterative**. Different groups of bars represent the results from different combinations of true vergence angles and gaze directions. Vergence angles of 2°, 7° and 17° correspond to the minimum, typical and maximum angles of a stereo-head system under normal working conditions. The thin horizontal lines in the left graphs represent the true vergence angles, which are shown as references.

We see that the estimated vergence of all methods come close to that of the true configuration, since the mean error rarely exceeds one degree and the standard deviations are reasonably small, whereas gaze direction is much harder to estimate. However, when reconstructing depths of image features, the vergence angle is the most critical component. In general the support based methods, **support** and **LMedS**, are not as robust as the other methods. It is worth noting that, contrary to what the higher number of degrees of freedom might suggest, **LMedS** is rather good at estimating the vergence angle, unless the cameras are close to parallel. Using the mean of estimated epipolar geometries, such as in **linear** and **nonlinear**, seems to be a more robust approach. A standard deviation of about three degrees for the gaze direction, which is the worst results of both methods, is probably acceptable in most applications. We may also note a slight improvement of the bias, when **nonlinear** optimization has been added to the ordinary **linear** one. However, it is questionable if these improvements justify the increased complexity of the method.

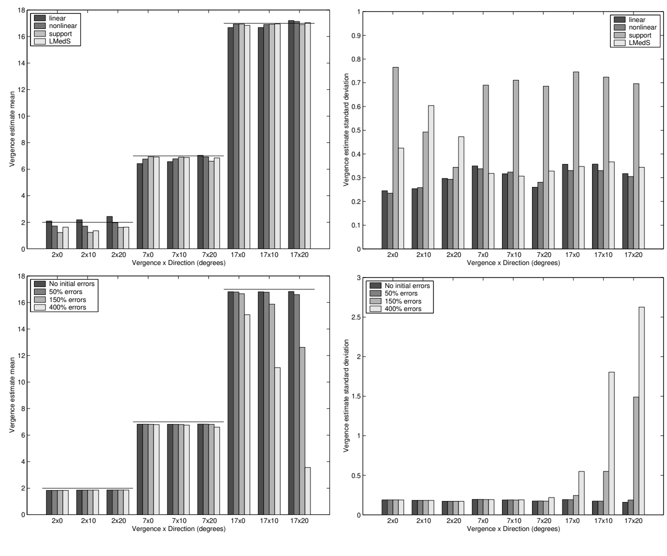

Fig. 3. Mean and standard deviation of the estimated vergence angle, for various true configurations and algorithms. The results of **iterative** are shown in the lower graphs for different initial median depth errors.

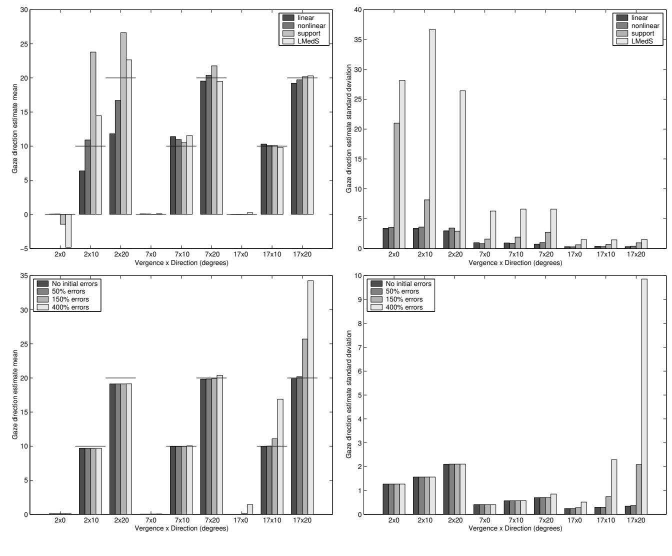

Fig. 4. Mean and standard deviation of the estimated gaze direction, for various true configurations and algorithms. The results of **iterative** are shown in the lower graphs for different initial median depth errors.

The performance of **iterative** is in most cases superior

Method	Cost on R10K
linear	58 ms (26 ms)
nonlinear	81 ms (36 ms)
support	480 ms
LMedS	850 ms
iterative	27 ms (12 ms)

Fig. 5. Computational costs of the different methods performed on a 195MHz MIPS R10K processor.

to that of the methods based on the essential matrix, in terms of both vergence and gaze direction. Even if small vergence angles are still critical, the standard deviations of the gaze direction stay within about 2°. However, if the expected median depth is far from the true one, the algorithm might collapse, especially for combinations of large vergence angles and asymmetric camera settings. Unlike the other methods, the robustness is slightly dependent on the true gaze direction, which is a result of that component being initially set to 0. The iterations converge at a rate remarkably independent from the camera configuration. Typically 20 to 30 iterations are needed for the relative errors of the depth values to decrease to 10% of the original errors. The rate is somewhat higher for near symmetric camera configurations and small vergence angles. In situations where outliers have not be properly eliminated the convergence rate can be much worse, even if the procedure never seems to diverge.

B. Computational cost

The table in Figure 5 shows the execution times of the methods presented above, performed on a 195 MHz MIPS R10K processor. Time required for feature extraction has not been included here. The data clearly shows that the methods that do not rely on any support calculations run considerably faster, than if a support measure has to be obtained for each and every sample set. It is worth noting that **LMedS** is not much slower than **support**, even if the number of degrees of freedom is higher, which indicates that the actual support calculations, which are similar in both methods, constitutes most of the computational cost.

As seen in the table, the **linear** and **nonlinear** are much faster than the methods based on support. However, it is hard tell which one of the two methods to prefer. In applications where the bias of the **linear** method is not critical, one might benefit from a somewhat lower computational cost. The speed of the two methods can be further improved, by setting a limit on the number of feature pairs one wishes to accept. Instead of generating as many as 10000 feature pairs, one might be satisfied when a certain number of accepted pairs has been found (we use 200) and then end the random sampling procedure. The data in parentheses show the execution times when such a limit has been used. It should be noted that the accuracy of this approach is not notably affected.

The **iterative** approach turns out to be considerably faster, than all the other methods. This is basically because we use the fact the we have an idea about the median distance to objects we are observing and use this knowledge to improve the estimation process. In cases where the external calibration is performed continuously and the cameras have not moved too much since the last update, one might benefit from using the previous camera parameters, when performing a new estimation. Instead of using as many as 30 iterations, the method often converge after much fewer iterations. Testing the difference between two consecutive iterations, we end the iterative procedure if convergence has been met earlier, speeding up the process as indicated by the average execution time shown in parentheses.

V. Real-time experiments

We performed a series of experiments, out of which one will be presented here in greater detail. During an interval of about 30 seconds the complete system was running, including the epipolar estimation process based on the **iterative** approach, as well as rectification and disparity calculation. The disparity maps were obtained using a method based on correlations and dynamic programming [16], [12], [4]. The position of the stereo head was kept static, but with the ability for the cameras to verge. It should be pointed out that the data was not processed off-line, but in real-time on a SGI Octane machine with an update frequency of about 8 Hz. A small number of images from the sequence are presented in Figure 6, the upper row showing images from the left camera and the bottom row from the right. The relative tilt between the cameras, that is noticeable in the images, was estimated and adjusted for by a separate process. The three columns constitute image pairs from the 50th, 100th and 150th frames. Between the first and second columns the cameras have verged slightly, from about 8° to around 11.5°. In order to test the ability of the system to reassume convergence after a failure, we let someone block the left camera with a hand for a few seconds. This can be seen in the images from the 150th frame. For the left images the locations of extracted corner features, as well as lines corresponding to valid matches, can be seen.

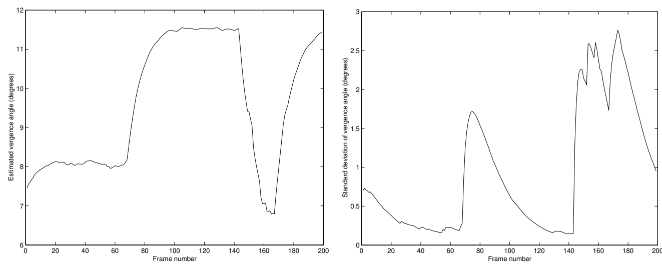

Fig. 7. Estimated vergence angle and its standard deviation

Figure 7 show the estimated vergence angles and their standard deviations. The values are processed through a moving average filter, with a time window of about 20 frames. The first frame in the figures is the very moment when the system starts running. Through the experiments performed in our lab, the actual bootstrap process never seems to fail, if the distribution of features have otherwise

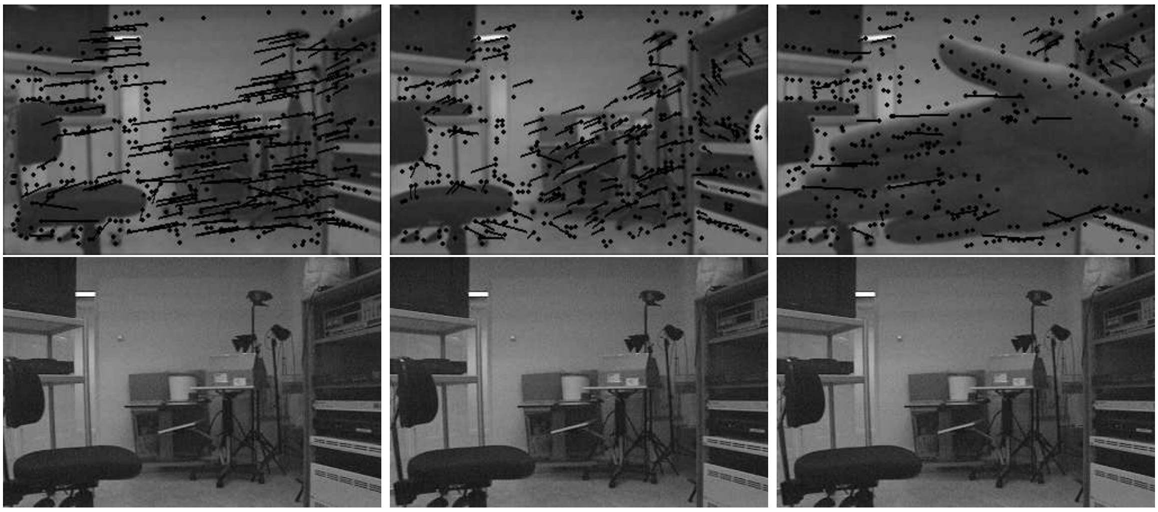

Fig. 6. On the upper row the images from the left camera are shown, whereas the lower row are the corresponding right images. The three columns represent image pairs from the 50th, 100th and 150th frame. The black points represent the extracted corner features, while the lines show valid matches between the left and right images.

been good enough. However, if the lighting-conditions radically change or the cameras get partially occluded as in the 150th frame, one might get serious errors in the geometry estimates, primarily due to a dramatic increase in the number of outliers. In Figure 8 the corresponding graphs for the cyclopean gaze direction are shown. The standard deviation stabilizes at about 1°, which is in the neighbourhood of the errors shown in Figure 4. The same thing is true for the vergence angles, where the standard deviation reaches about 0.2° upon convergence.

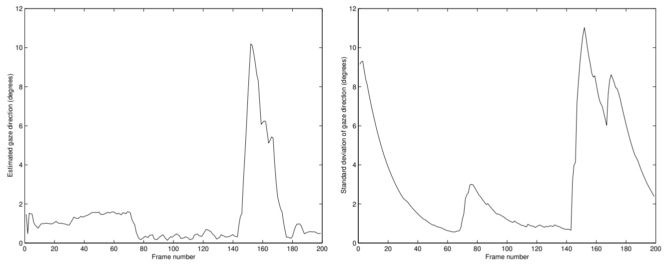

Fig. 8. Estimated gaze direction and its standard deviation

Finally, in Figure 9 the resulting disparity maps can be seen for a number of selected frames. The black parts represent areas where no confident data could be calculated, either because they are too close to the borders or simply because there is hardly any structure in the images. The remaining shades are based on the range of available disparities for each individual disparity map. The lighter shades always mean areas located closer to the cameras, but what it means in metric values is different for each image. A small median filter has also been used in order to eliminate some erroneous results, such as the typical streaking effect of dynamic programming based stereo routines. As seen in the figure the disparity maps are relatively stable, when the cameras are not currently changing vergence angle and both cameras are free from disturbances, such as the hand blocking the view of the left camera in the 150th frame.

VI. Discussion

For autonomous robots stereo is an important cue for a variety of visually guided behaviours, such as navigation, obstacle avoidance and manipulation. However, different behaviours have different requirements on working ranges and because of the limitations in field of view and image resolution, dynamic vergence often becomes a necessity. Dynamic fixation may also be used to simplify problems like ego-motion estimation and figure-ground segmentation. In this paper a real-time framework for epipolar geometry estimation has been presented. Instead of using counters on the motors of the stereo-head, the external calibration is performed continuously at a low computational cost. Two models have been explored, with the number of unknown parameters minimized without constraining the system.

The first model based on the essential matrix showed to be somewhat sensitive to near parallel camera systems, but for vergence angles larger than about 5° the errors stay within about 1° for vergence angle as well as gaze direction. An alternative approach, based on a differential formulation of optical flow, was also presented and tested towards methods based on the essential matrix. Using an initial estimate of the median distance to objects in 3D space, the geometry parameters and point depths were sought in an iterative fashion. Since the model only works for limited rotations and translations, it could be expected not to be sufficient for modeling the disparities of a stereo system. However, this study showed that it could in fact be used for stereo-head under normal working conditions. The method was faster and also more robust, especially for small vergence angles, than methods based on the essential matrix.

The binocular stereo system in this paper will be implemented on the various mobile platforms of our lab, and used for behaviours, such as obstacle avoidance, navigation and visually based manipulation. A number of stereo based processes has already been added to the current sys-

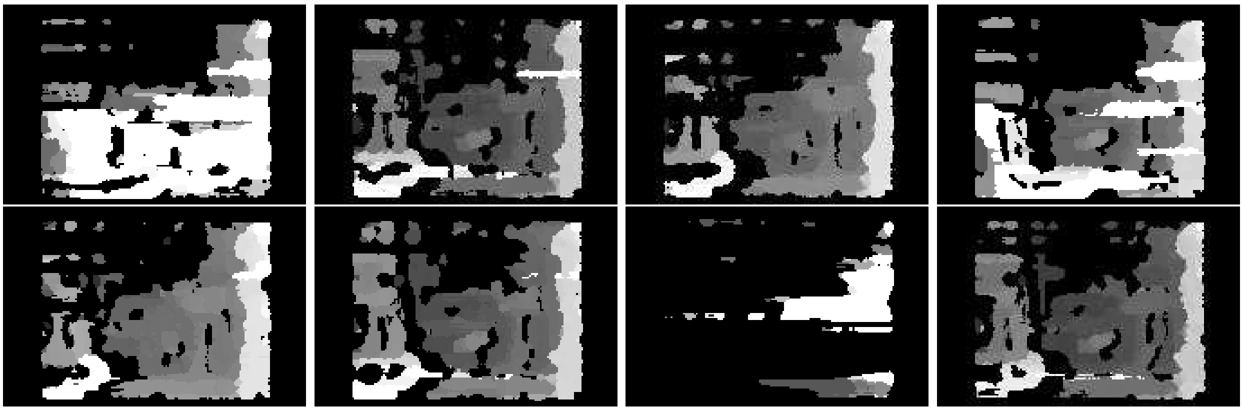

Fig. 9. Images of the resulting disparity maps of the 10th, 25th, 50th, 70th, 90th, 125th, 170th and 195th frames.

tem. Ego-motion estimation has been greatly simplified using the available stereo data. The system also includes a process for identification of independently moving objects [2]. Through forward prediction of an image from one instance in time to another, using estimated ego-motion and disparities, regions of independent motion can be identified as areas where the predicted image does not match the true new image.

References

[1] D.H. Ballard, Animate Vision, *Artificial Intelligence*, 48, pp. 57-86, 1991.

[2] M. Björkman and J-O. Eklundh, A Real-Time System for Epipolar Geometry and Ego-Motion Estimation, In *Proc. of IEEE Computer Vision and Pattern Recognition* pp. 506–513, Jun 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] I.J. Cox, S.L. Hingorani, S.B. Rao and B.M. Maggs, A Maximum Likelihood Stereo Algorithm, *Computer Vision and Image Understanding*, 63(3), pp. 542-567, 1996.

[5] K. Daniilidis, Fixation Simplifies 3D Motion Estimation, *Computer Vision and Image Understanding*, 68(2), pp. 158-169, November 1997.

[6] C. Fermüller and Y. Aloimonos, The Role of Fixation in Visual Motion Analysis, *Int'l J. of Computer Vision*, 11(2), pp. 165-186, 1993.

[7] M.A. Fischler and R.C. Bolles, Random Sampling Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *DARPA Image Understanding Workshop*, pp. 71-88, 1980.

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

[9] R.I. Hartley, Estimation of Relative Camera Positions for Uncalibrated Cameras, In *Proc. of the 2nd European Conf. on Computer Vision*, pp. 579-587, 1992.

[10] J. Heel, Dynamic Motion Vision, *DARPA Image Understanding Workshop*, pp. 702-713, 1989.

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

[12] T. Kanade, M. Okutomi and N. Nakahara, A Multiple-Baseline Stereo Method, *DARPA Image Understanding Workshop*, pp. 409-426, 1992.

[13] 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.

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

[15] Y. Ma, J. Kosecka and S. Sastry, Optimization Criteria and Geometric Algorithms for Motion and Structure Estimation, UC Berkeley Memorandom No. UCB/ERL M99/33, June 1999.

[16] Y. Ohta and T. Kanade, Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(2), pp. 139-154, 1985.

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

[18] 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. the 13th Annual Hawaii Int'l Conf. on System Sciences*, pp. 188-197, January 1997.

[19] W.Y. Yau and H Wang, Fast Relative Depth Computation for an Active Stereo Vision System, *Real-Time Imaging*, 5, pp. 189-202, 1999.

[20] 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, October 1995.