Real-Time Epipolar Geometry Estimation and Disparity

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

Many visual tasks, such as recognition and visually guided manipulation, benefit from the use of binocular stereo with dynamic vergence. In order to achieve this the epipolar geometry of the two cameras has to be known. In this article we present a system capable of estimating the epipolar geometry of a stereo-head and calculating the binocular disparities in real-time, producing dense disparity maps that, for example, can be used for figure-ground segmentation. The number of degrees of freedom has been minimized, without affecting the flexibility of the stereo-head, speeding up the process considerably.

1. Introduction

A basic feature of active vision systems is their capability of controlling gaze. Ballard [2] gave a number of reasons why this is essential. For instance, it is important in tasks such as visual recognition and visually guided manipulation performed by a mobile observer in a dynamic world, that he can direct his gaze to and hold it on the objects of interest. This requires the use of three-dimensional cues in form of motion and binocular disparities [1], which in addition can be used to separate objects of interest from their background.

We have in our study been concentrating on binocular disparity. The use of disparities for visual guidance of mobile robots is not new [9, 12]. However, due to the high computational cost of calibration, the cameras have typically been precalibrated and dynamic fixation has not been possible. For some robot applications, such as navigation, the lack of dynamic fixation might be acceptable, but in manipulation it is needed. Without it objects of interest may disappear from the binocular field of a stereo vision system or an unacceptable reduction of the resolution in calculated depth may be obtained.

In this article we will investigate the problem of establishing the correspondence between the two cameras of a stereo-head system in real-time on standard hardware, rectifying the images from these cameras to produce images, where the epipolar lines are parallel, so that a conventional algorithm can be used to calculate dense disparity maps. Our intention is to use these maps in a multiple cue framework in order to perform figure-ground segmentation and tracking on a mobile platform.

The problem of finding the epipolar geometry between stereo images has been described and treated numerous times in the literature, see e.g. [11, 13]. In our study we will take the physical properties and constants of the stereo-head system into account and end up with an estimation more robust than a more general approach would have offered. We are also interested in a solution that allows real-time implementation, taking advantage of temporal consistency.

Section 2 describes the epipolar geometry of a stereo-head system and the theory behind the image rectification process. In Sections 3 and 4 we consider the estimation of the epipolar geometry. A series of experiments are also performed in Section 4 and different approaches are evaluated. Section 5 describes how binocular disparities are being calculated and finally the epipolar geometry estimation and the disparity calculation steps are combined into a complete system.

2. Epipolar geometry

A stereo-head system typically consists of two cameras, as seen in Figure 1, where the left camera can be used to define the origin of a coordinate system. We use the baseline between the two cameras to define the x-axis. At fixation the optical axes of the left and right cameras form the epipolar plane. The z-axis is defined as the line on this plane located perpendicular to the x-axis, while the y-axis is parallel to the normal of the plane.

We have three degrees of rotational freedom for each camera in total. However, a rotation around the optical axis of a camera will not change the visual data, just the orientation of the image of that camera. Furthermore, in order to keep the cameras in fixation, there will not be any relative tilt between the cameras and a joint tilt around the baseline, will not change the nature of the problem.

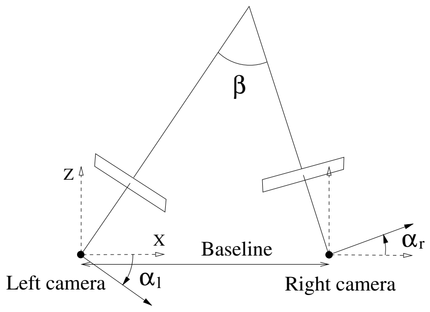

Figure 1. Stereo-head system

Consequently, we only have to consider one degree of freedom for each camera, i.e. a rotation around the y-axis of that camera, represented by the matrices $\mathbf{R}_{\alpha_l}$ and $\mathbf{R}_{\alpha_r}$. These are the matrices we need to know in order to perform the rectification. Using the angles shown in Figure 1, we have

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

and

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

Using homogeneous coordinates in both object and image space the relation between an object point $\mathbf{u} = (x, y, z, w)^T$ and the image points for each of the two cameras, $\mathbf{x_l} = (x_l, y_l, w_l)^T$ and $\mathbf{x_r} = (x_r, y_r, w_r)^T$, can be written as $\mathbf{x_l} = \mathbf{P}_{\alpha_l}\mathbf{u} = \mathbf{R}_{\alpha_l}(\mathbf{I}|\mathbf{0})\mathbf{u}$ and $\mathbf{x_r} = \mathbf{P}_{\alpha_r}\mathbf{u} = \mathbf{R}_{\alpha_r}(\mathbf{I}|-\mathbf{t})\mathbf{u}$, the vector $\mathbf{t} = (k, 0, 0)^T$ being a translation along the baseline.

The 3x4 projection matrices $\mathbf{P}_{\alpha_l}$ and $\mathbf{P}_{\alpha_r}$ can further be used to derive the matrix [1] $\underline{\mathbf{F}}$ of the epipolar constraint $\mathbf{x_l^T}\underline{\mathbf{F}}\mathbf{x_r} = \mathbf{x_l^T}(\mathbf{R}_{\alpha_l}\mathbf{T}\mathbf{R}_{\alpha_r}{}^{-1})\mathbf{x_r} = \mathbf{0}$, where

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

We see that the epipolar geometry is expressed by

$$\underline{\mathbf{F}} = k \begin{pmatrix} 0 & -sin(\alpha_l) & 0 \\ -sin(\alpha_r) & 0 & cos(\alpha_r) \\ 0 & -cos(\alpha_l) & 0 \end{pmatrix}, \quad (4)$$

[1] Underline here denoting that the matrix is representing the true epipolar constraint and yet not known.

which can further be decomposed into the two matrices $\mathbf{R} = \mathbf{R}_{\alpha_l}\mathbf{R}_{\alpha_r}{}^{-1}$ and $\mathbf{S} = \mathbf{R}_{\alpha_r}\mathbf{T}\mathbf{R}_{\alpha_r}{}^{-1}$. The first matrix $\mathbf{R}$ is the relative rotation between the cameras and the most important part, if we want to end up with an accurate relation between calculated disparities and real distances in object space. The skew-symmetric translational component $\mathbf{S}$ can be considered as the position of the right camera relative to the left camera.

2.1. Rectification

In order to do the rectification, we need to compute the camera matrices $\mathbf{P}_{\alpha_l}$ and $\mathbf{P}_{\alpha_r}$. This can be done by first finding an estimate $\mathbf{F}$ of $\underline{\mathbf{F}}$, that minimizes

$$\mathbf{x_l^T}\mathbf{F}\mathbf{x_r}, \quad \text{where} \quad \mathbf{F} = \begin{pmatrix} 0 & a & 0 \\ b & 0 & d \\ 0 & c & 0 \end{pmatrix}. \quad (5)$$

This estimate further needs the property of allowing a decomposition into **RS**, and we achieve this using a method proposed by Hartley [6]. This approach is based on a singular value decomposition $\mathbf{UDV^T}$ of $\mathbf{F}$, with $\sigma_1 = \sqrt{a^2 + c^2}$, $\sigma_2 = \sqrt{b^2 + d^2}$ and 0 being the singular values of $\mathbf{F}$ and $\mathbf{D} = diag(\sigma_1, \sigma_2, 0)$.

Since an **RS** decomposition of **F**, such as described above, is possible only if $\sigma_1 = \sigma_2$, it has become a common practise to use an approximation of $\mathbf{F}$, $\hat{\mathbf{F}} = \mathbf{U}\hat{\mathbf{D}}\mathbf{V^T}$, where $\hat{\mathbf{D}} = diag(\sigma, \sigma, 0)$ and $\sigma = (\sigma_1 + \sigma_2)/2$. This is the closest approximation of **F**, under the Frobenius norm, that can be factored into a product **RS**. Through a singular value decomposition the orthonormal matrices **U** and **V** can be derived as

$$\mathbf{U} = \frac{1}{\sigma_1}\begin{pmatrix} a & 0 & c \\ 0 & \sigma_1 & 0 \\ c & 0 & -a \end{pmatrix} \quad (6)$$

and

$$\mathbf{V} = \frac{1}{\sigma_2}\begin{pmatrix} 0 & b & -d \\ \sigma_2 & 0 & 0 \\ 0 & d & b \end{pmatrix}. \quad (7)$$

In fact, we easily see that we would have arrived to the same estimate, even if we let $c = -d = -1$, and seek the variables a and b. Keeping c and d constant to these values ensures that all the objects space points will be located in front of both cameras.

The use of homogeneous coordinates implies that $\underline{\mathbf{F}}$ can only be found up to a constant. This means that the actual distance between the two cameras, given by the constant k in Equation 4, can not be retrieved, but fortunately it can normally be regarded as known. In conclusion, after performing the singular value decomposition and forcing the two largest singular values to be equal, we get the following estimate of $\underline{\mathbf{F}}$,

$$\hat{\mathbf{F}} = \begin{pmatrix} 0 & \frac{a}{\sigma_1} & 0 \\ \frac{b}{\sigma_2} & 0 & \frac{d}{\sigma_2} \\ 0 & \frac{c}{\sigma_1} & 0 \end{pmatrix}. \qquad (8)$$

2.2. RS decomposition

As we mentioned earlier the $\mathbf{RS}$ factorization of the epipolar geometry $\hat{\mathbf{F}}$ can be seen as two different components. The factor $\mathbf{R}$ is the rotational component and $\mathbf{S}$ the translational one. Using the approach described in the previous section we can express the rotational component as

$$\mathbf{R} = \begin{pmatrix} c_\beta & 0 & -s_\beta \\ 0 & 1 & 0 \\ s_\beta & 0 & c_\beta \end{pmatrix}, \qquad (9)$$

where $c_\beta = -\frac{ab+cd}{\sigma_1\sigma_2}$ and $s_\beta = \frac{ad-cb}{\sigma_1\sigma_2}$. We see that $c_\beta^2 + s_\beta^2 = 1$, which means that $\mathbf{R}$ really is a rotation. In the stereo-head system, shown in Figure 1, the value c_β can be identified as $cos(\beta)$, where $\beta = \alpha_l + \alpha_r$ is the vergence angle between the two cameras.

Similarly, the translational component can be shown to be

$$\mathbf{S} = \frac{1}{\sigma_2}\begin{pmatrix} 0 & -b & 0 \\ b & 0 & d \\ 0 & -d & 0 \end{pmatrix}, \qquad (10)$$

with d/σ_2 corresponding to $cos(\alpha_r)$. Now we have everything we need to know, in order to perform the rectification at the end of this paper.

3. Robust parameter estimation

The epipolar geometry estimation relies on the assumption that we already have a set of image point correspondences. Each pair represent a single point in object space, as seen by the left and right camera respectively. In order to create such a set of pairs, we use the Harris corner detector [5] to locate corners in each of the two images. For each corner in one of the images, a possible match is searched among the corners in the other image.

We use cross-correlation of 9×9 pixel image intensity patches to estimate matching strengths. The search is restricted to regions defined by a predefined disparity limit, which is based on the possible configurations of the stereo-head. This matching is performed in both directions and for each corner, the corner in the other image with the maximum strength is recorded. Corner matches that exist in both directions are considered as valid matches. The *i*th valid match will be denoted as $(\mathbf{x_l^i}, \mathbf{x_r^i})$ in this text.

Note that we still have no guarantee that these valid matches really represent the images of the same point in object space, i.e. true correspondences. However, in real data we have empirically observed that the absolute majority, 80-90%, of the valid matches are indeed correct and that the valid matches typically represent about 40% of the complete data set.

3.1. Random sampling

In estimating the epipolar geometry we must minimize the influence of possible outliers. This can be done by random sampling. Due to noise, occlusions and repeatable patterns in the images, a considerable fraction of the matches can be expected to be wrong. Using a least squares approach, the errors of these outliers would dominate heavily, increasing the likelihood of a complete failure. Instead of using as much information as possible to get an accurate estimation of $\underline{\mathbf{F}}$, we use a feasible subset of randomly selected matches so as to minimize the probability of the subset including an outlier that would make the result unusable. This can be done by picking a number of subsets of randomly selected valid matches and for each subset $M(j)$ minimize

$$\sum_{i\in M(j)} (ax_l^i y_r^i + by_l^i x_r^i + cw_l^i y_r^i + dy_l^i w_r^i)^2. \qquad (11)$$

Such a technique assumes the error to be in one coordinate only. Torr [11] proposed another method that turned out to be more successful here, namely the minimization of the orthogonal error. In this approach we fit a hyperplane $\mathbf{f} = (a, b, c)^T$ through a set of points with coordinates $\mathbf{z_i} = (x_l^i y_r^i, y_l^i x_r^i, y_l^i w_r^i - w_l^i y_r^i)^T$. This is accomplished by minimizing

$$\sum_{i\in M(j)} (\mathbf{f^T z_i})^2 = \sum_{i\in M(j)} \mathbf{f^T}(\mathbf{z_i z_i^T})\mathbf{f} = \mathbf{f^T M f} \qquad (12)$$

subject to $\mathbf{f^T f} = 1$, where $\mathbf{M}$ is a 3×3 moment matrix, with eigenvalues λ_1, λ_2 and λ_3 in increasing order and the corresponding eigenvectors $\mathbf{u_1}$, $\mathbf{u_2}$ and $\mathbf{u_3}$. The vector $\mathbf{f}$ that minimizes (12) is given by the eigenvector $\mathbf{u_1}$ corresponding to the smallest eigenvalue λ_1.

3.2. Discarding bad estimates

Due to noise, improperly selected points or outliers within a subset, λ_2 might be close to λ_1, which leads to multiple possible solutions to the system. In this article we find λ_1 and $\mathbf{u_1}$ using an iterative approach, an inverse power method, that converges with an error decreasing at a rate of λ_1/λ_2. If convergence have not been met in a few iterations, there might be multiple possible solutions and we discard the result and pick the next set of points using random sampling. This turned out to be a quick and efficient way of estimating $\mathbf{u_1}$ and eliminating bad estimates at the same time.

4. Estimating the fundamental matrix

For each subset of valid matches, an estimate $\hat{\mathbf{F}}$, (8) is being calculated using the procedures mentioned above, unless the subset is considered as bad and discarded. Each estimate $\hat{\mathbf{F}}$ can be seen as a sample point $\mathbf{p} = (a/\sigma_1, b/\sigma_2)$ in a two-dimensional space Ω_F. The distribution of sample points can be represented by the second moment of the samples. This moment can be visualized by an ellipsoid, with center $\underline{\mathbf{p}}$ describing the true epipolar constraint $\underline{\mathbf{F}}$.

Small vergence angles typically lead to very ill-conditioned systems and the ellipsoid gets elongated and the center harder to identify. If the epipolar constraint is divided into two components, a rotational and a translational one, this problem can be seen as an increased uncertainty of the translational component in particular. Thus in order to rectify the images using an estimated epipolar constraint, the vergence angles should not be too small. We will come back to that later.

Another common way of finding the estimate $\hat{\mathbf{F}}$, that best matches the true epipolar constraint, is what normally is referenced to as the random sampling consensus paradigm (RANSAC) [4]. Each estimate is tested against all valid matches, in order to calculate in what sense the whole data set supports the estimate. An estimate $\hat{\mathbf{F}}$ is said to be supported by a pair of points $(\mathbf{x_l}, \mathbf{x_r})$ if $|\mathbf{x_l^T}\hat{\mathbf{F}}\mathbf{x_r}| < \sigma_F$, where σ_F is a predefined threshold covering the expected noise of the corner extractions. The estimate with the highest number of supporting matches, will be considered as the best estimate of $\underline{\mathbf{F}}$ and will later be used to perform the rectification.

4.1. Vergence angle estimation

The question that now arises is whether the calculated maximum support will give us a better estimate, than by finding the center of the random samples in Ω_F directly. After all, calculating the support among all the valid matches for each $\hat{\mathbf{F}}$ can be a time-consuming and in our case speed is of major concern.

We tested this by generating a number of points, typically around 400, in a 3-dimensional space, projecting these on two camera image planes located with vergence angle β. The cloud of these random points was located 5 meters away from the cameras, with points spread in random directions around a center with a standard deviation of about two meters. A noise level of about a pixel was further added to the locations of the points in the 360×288 pixel image planes, simulating noise of the feature extractor.

Simulations were performed using various vergence angles β, and Figure 2 shows the mean error and standard deviation of estimated vergence angles as a function of the real ones, using the two alternative methods. At this point

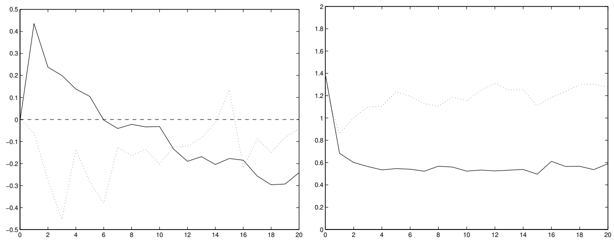

Figure 2. Error and standard deviation of estimated vergence angles as a function of true vergence angles, with estimates based on support (dotted) and center of distribution (solid). The curves represent 200 simulation runs.

we should point out that the valid matches in each subset $M(j)$ were selected such that they are spread all over the image space. We use a bucketing technique, similar to the one described in [13]. If the matches are located too close to each other, the system may otherwise be highly unstable.

The first observation we can make is that the error never seems to exceed one degree. We also see that, as long as the vergence angle is not too small, simply calculating the center of the ellipsoid is enough to get an acceptable estimate of the true vergence angle. Based on the graph showing the standard deviations, we may conclude that the method of using the center of distribution is somewhat more robust, than the approach that is based on support.

4.2. Effect of outliers

Due to outliers among the valid matches some points $\mathbf{p}$ will be located outside the ellipsoid randomly scattered around in Ω_F. In order to test the effect of these outliers, we added an extra 100 erroneous [2] random pairs of points to the data set and ran the same simulations once again. The fraction of 20% may seem as a low figure in a real situation. However, since we have limited our search using the valid matches only, we can be quite sure that most matches that are considered as valid, indeed originate from the same points in object space. Among those points we already have discarded from the original data, the fraction of outliers can be far higher.

Figure 3 shows the fraction of estimates that have been accepted during the process of finding λ_1 and $\mathbf{u_1}$, described in Section 3.2. The dotted lines show that the number of accepted estimates are much fewer, when outliers are added to the data set. We also see that we can expect to run into trouble, if the vergence angle is small and the number of

[2] An error with a standard deviation of 30 pixels was added to the right image point position of an otherwise correct projection.

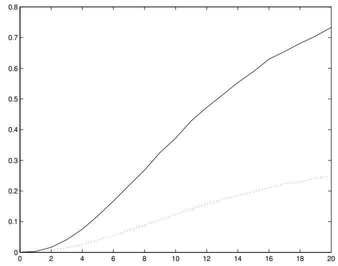

Figure 3. Fraction of accepted estimates. The solid line represent the fractions without outliers and the dotted line show the fractions when 20% outliers have been added to the data set.

accepted estimates gets too low. In fact, a low fraction of accepted estimates during a real run can be used as an indication that the vergence angle is too small and we ought to switch to another model.

The results in Figure 4 show that using the estimate with the maximum support is more sensitive to outliers, than if the center of the distribution was used. However, there seems to be a considerable systematic error in the center calculations, which also can be seen in Figure 2. We have observed that results highly influenced by noise and outliers are scattered around Ω_F and have a mean closer to $(0, 0)$ than $\underline{\mathbf{p}}$, resulting in an underestimation of $\mathbf{p}$.

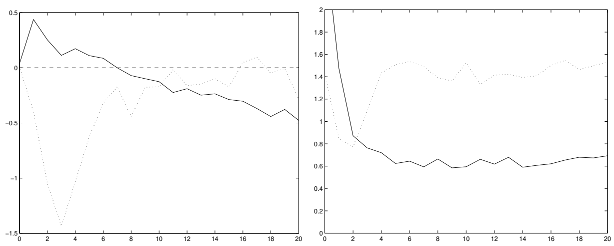

Figure 4. Error and standard deviation of the vergence angle estimation, when outliers have been added to the data set. The estimates are based on support (solid) and center of distribution (dotted). The curves represent 200 simulation runs.

4.3. Improving the estimates

Using an estimation based on the center of the distribution has shown to be an attractive approach, since it is faster than calculating the support for every estimate and the results are at least as good. However, we have seen that we get systematic underestimations when the vergence angle increases and the question is whether this can be improved. In our study we have tested two different approaches.

The first approach is based on finding a point in Ω_F that locally maximizes a support measure described below. A search is performed in a neighbourhood of $\mathbf{p}$, that is the point corresponding to the estimate based on the center of the distribution described earlier. For each point in the neighbourhood, the corresponding $\hat{\mathbf{F}}$ is found and for each valid match $(\mathbf{x_l^i}, \mathbf{x_r^i})$ we calculate the error $\sigma^i = \mathbf{x_l^i}^{\mathbf{T}} \hat{\mathbf{F}} \mathbf{x_r^i}$ and try to maximize the support function $S(\hat{\mathbf{F}}) = \sum_{\mathbf{i}} \gamma^{\mathbf{i}}$, where

$$\gamma^i = \left\{ \begin{array}{ll} 1 - (\frac{\sigma^i}{\sigma_F})^2 & \text{if } |\sigma^i| < \sigma_F \\ 0 & \text{otherwise} \end{array} \right\}. \qquad (13)$$

The constant σ_F is a predefined threshold covering the expected noise in the feature extraction. The actual search can be implemented in a number of different ways, but due to the lack of space we will not describe them further here.

An alternative method is minimizing the least square expression in Equation 12, using all valid matches that support the initial estimate $\hat{\mathbf{F}}$, i.e. those matches $(\mathbf{x_l^i}, \mathbf{x_r^i})$ that satisfy $|\mathbf{x_l^i}^{\mathbf{T}} \hat{\mathbf{F}} \mathbf{x_r^i}| < \sigma_F$. Unlike the operations during the random sampling, our primary goal now is not eliminating the effect of outliers, but rather decreasing the sensitivity to corner extraction noise. That is why we use as much input data as possible in the minimization. The new estimate can be further improved performing the same procedure an additional number of times.

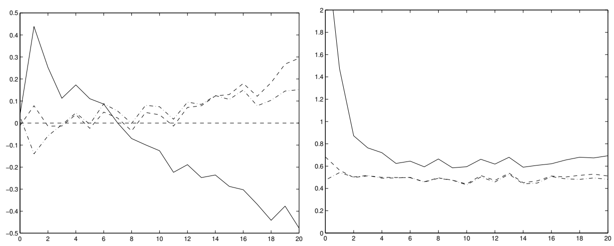

Figure 5. Error and standard deviation before and after optimization. The solid lines represent errors without optimization, the dash-dotted with local search and the dashed using least square.

As seen in Figure 5 the two methods give similar results, even if the method using a least square error minimization is faster than the one based on searching in the neighbourhood of an initial estimate, because calculating the support measure in Equation 13 is rather costly. However, when there are many outliers the least square approach is not as robust as the other one, but the speed makes it attractive anyway and in our application only valid matches are used and the outliers can be expected to be few.

4.4. The translational component

So far we have only covered the rotational component of the epipolar geometry and for some application this is enough. This may be the case if you have a moving platform carrying the stereo-head system, and the primary goal is to calculate the distance to an obstacle and the direction to the obstacle is known from a map or another sensor. As mentioned in Section 2 there is another component, the translational component, and in order to do a proper rectification, we need this component as well. We saw earlier that this component can be seen as the orientation of the baseline relative to the left camera.

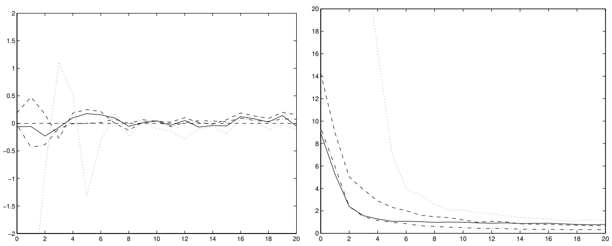

Figure 6. Error and standard deviation of the translational component. Estimates are based on support (dotted) and center of distribution (solid), with optimization using least square (dashed) and local search (dash-dotted). Note the scale of the y-axis.

In the Figure 6 we see the error of the estimated translational components shown as a function of varying vergence angles. It is clear that estimations based on support calculations break down completely, when the vergence angle is less than about five degrees, whereas the center of the distribution method can be used for somewhat smaller angles. No method is really good below two degrees.

The reason we get these errors can be easier understood by considering the following. A small translation along the optical axis does not affect the images more than a slight scaling, when the points in object space are located relatively far away from the cameras. However, since the length of the baseline is relatively small, the effect of this translation on the estimated orientation of the baseline can be rather dramatic, especially when the cameras are close to parallel.

It was concluded earlier that the two optimization techniques resulted in quite similar performances and there was no real difference, favouring any of them. However, if the translational component is of importance, we may consider the fact that the standard deviation of the local search method is about half, than that of the least square approach, for all angles.

5. System integration

So far we have mainly covered only one component of a complete system, that is the estimation of the epipolar geometry. As we mentioned earlier the epipolar geometry is essential in order to establish the correspondence between images from the two cameras of the stereo-head system and be able to wrap the images, so that a disparity map can be calculated. A complete system might include the following steps:

1. Corner feature extraction
2. Matching of corner features
3. Estimation of epipolar geometry
4. Rectification of images
5. Calculation of dense disparity map

In our current implementation the corner extraction is performed using the Harris corner detector as mentioned earlier. In order to calculate the epipolar geometry we decided to use the method based on the center of the distribution of estimates, followed by an additional optimization step. The optimized estimates are then passed through a simple moving average filter to the rectification phase, where images are wrapped using the filtered angles and bilinear interpolation to minimize aliasing.

5.1. Calculating the disparities

As mentioned in the introduction, the goal of our study was to calculate dense disparity maps and keep the benefits of dynamic fixation. The goal is to compute disparity maps dense enough for segmentation of images into separate objects in space to be performed. From the epipolar geometry estimation, described in Sections 3 and 4, we get some valuable pieces of information. Firstly, the epipolar geometry itself describes how images can be rectified, making the epipolar lines parallel and disparity calculations easier. Secondly, during the corner matching process, we get an idea of the distribution of disparities in the scene and we may limit our search to a mimimal interval of disparities.

In our current system, disparities are calculated using a combination of correlations and dynamic programming. Numerous groups have already successfully been using correlation as a basis for real-time stereo, but the implementations all look slightly different [8, 7]. Our system relies on correlations of 7×7 pixel windows, using normalized cross-correlation as matching criterion. Sub-pixel accuracy can be achieved by fitting the correlation values around a maximum to a second-degree polynomial. The variance of intensities in each window is computed during the correlation process and used as a confidence measure to eliminate regions that lack sufficient structure.

Dynamic programming can further be used in a maximum likelihood framework to improve the density of calculated disparities [10, 3]. In our implementation, a cost function is used to find the most probable combination of disparities along a scan-line, taking the ordering and uniqueness constraints into consideration. The complexity of finding a unique minimum is greatly reduced by dynamic programming. The strength of the method is its ability to cover areas of limited texturing creating denser disparity maps.

5.2. Implementation

The implementation was done on a SGI Octane machine powered by two 195 MHz MIPS R10000 processors and capable of capturing two digital video images simultaneously. The corner extraction and epipolar geometry estimation steps were partitioned to one of the processors, while the second processor was carrying out the image rectification and disparity calculation. In order to keep the portability of the code, the hardware acceleration capabilities of the SGI system were not used for the implementation.

The disparity maps were calculated with half the resolution of the original images and thanks to an almost perfect load balance between the processors, the frame-rate of the complete system was approximately 6 Hz, when both processors were used and about half the speed otherwise. The most time-consuming parts of the system, the corner feature extraction and disparity map calculation, required about 120 ms and 80 ms respectively. However, the nature of these operations make them suitable for parallel computation on dedicated hardware, such as the MMX capabilities of recent Intel processors.

Epipolar estimation using the center of distribution had a computational cost of approximately 30 ms, when 150 valid matches were found and 10000 random sampling iterations used. If estimates were calculated using support, the cost would have been around three times as much. It should also be noted that the execution time of the first approach is less sensitive to the number of valid matches, while the cost of support calculations increases proportionally to the number of valid matches.

5.3. Performance

During the real-time operations of the system, it was seen that the two optimization methods resulted in similar performance, but the one based on least squares was slightly more stable. On the other hand, the translational component turned out to be far better calculated using the local search variant. However, because of the low computational cost, there is no real reason not to perform both optimization methods simultaneously and use the results with the lowest standard deviation during the warping procedure.

We also tested the performance of the system, calculating the standard deviation of the estimated angles from frame to frame. In Figure 7 the estimated vergence angles and standard deviations during the first frames of a typical run are shown. Note that the cameras are verged at about 20 seconds, but the standard deviation quickly returns to a level well below the level previously predicted. This indicates that the fraction of outliers during real-time operations is lower than 20% or else the feature extraction is more accurate than what was earlier assumed. Figure 9 show the disparity maps produced during the same bootstrap run, when the cameras were verged such as shown in Figure 8.

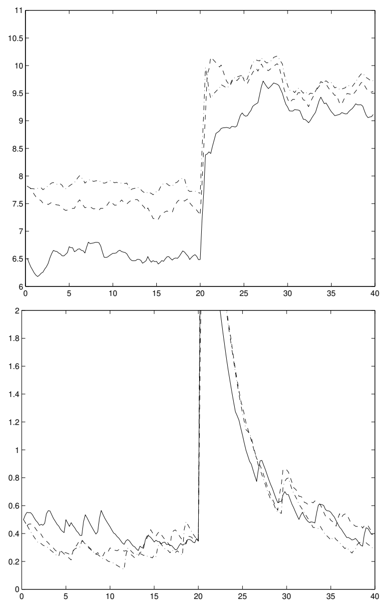

Figure 7. Estimated vergence angles and standard deviations during the first 40 seconds of a real-time run, without optimization (solid), with local search (dashed) and least squares (dash-dotted). Note that the cameras are verged after about 20 seconds.

However, it was also noticed that you easily get serious errors in the estimates, if the lighting-conditions radically change or the cameras get partially occluded, since the number of outliers increases dramatically. Since the cameras are used with automatic white balancing, an intensive light-source only seen in one cameras might complicate the matching procedures, degrading the final estimates. We introduced two separate ways of dealing with these problems. The first included a gate that simply discarded inputs to the moving average filter, in case the values did not match the current expected angles at all. The other more dramatical change involved to use of the previously calculated epipolar geometry to guide the early matching process of extracted feature points.

6. Conclusion

In this article we have presented a system capable of performing real-time estimation of the epipolar geometry of a stereo-head system with dynamic vergence. Based on the physical properties of such a stereo-head, only two degrees of freedom had to be sought, speeding up the process considerably. The reduced number of estimated parameters also had a significant impact on the robustness and practical use of the system. Different estimation techniques were tested and evaluated based on robustness and speed. Using the estimated geometry, images were transformed so that the epipolar lines are located parallel to the scan-lines, allowing the use of standard methods to produce dense disparity maps. In our future work we intend to use these disparity maps, in a multiple cue frame-work, to assist higher-level processes such as recognition and manipulation on mobile robots.

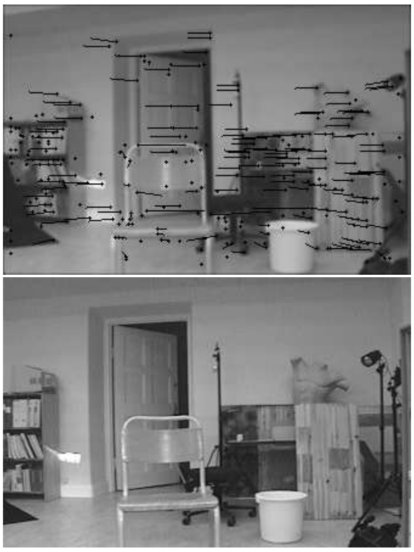

Figure 8. Images from the left and right camera of the stereo-head. The left image includes extracted corners and the valid matches during the matching process.

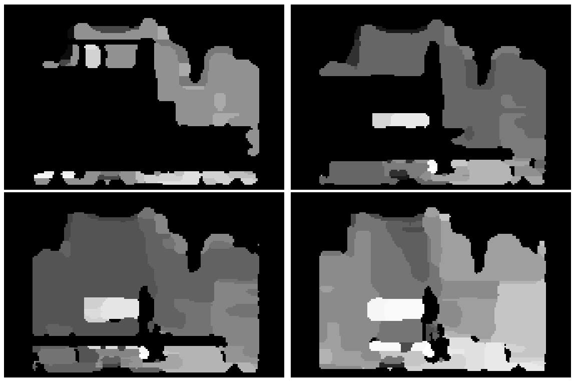

Figure 9. Disparity maps from the 1st, 5th, 10th and 20th frame of a typical operation. The darker areas represent regions that either do not have enough of structure or where occlusions have been identified.

References

[1] P.K. Allen, B. Yoshimi and A. Timcenko, Real-Time Visual Servoing, In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 851-856, 1991.

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

[3] 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.

[4] 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.

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

[6] 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.

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

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

[9] H. Moravec, The Stanford Cart and the CMU Rover, In *Proc. of the IEEE*, 71(7), pp. 872-884, 1983.

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

[11] 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.

[12] 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, January 1997.

[13] 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.