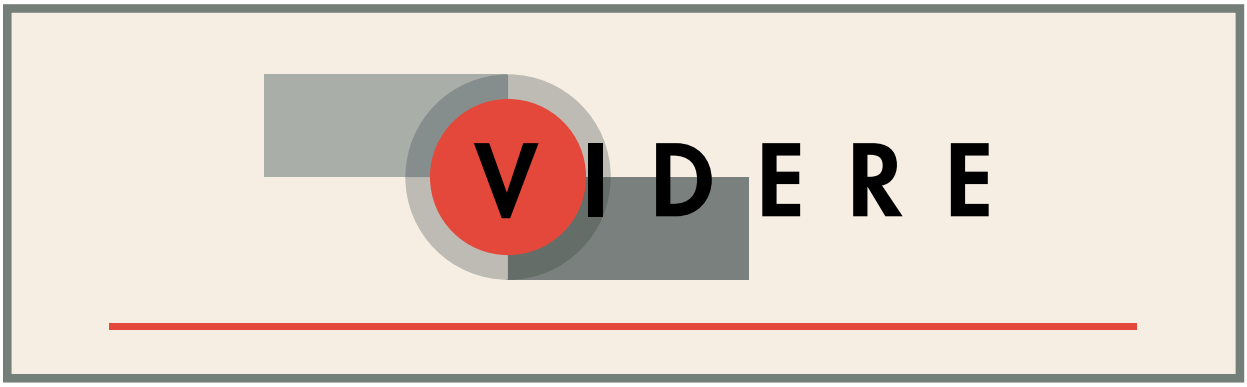


Videre: Journal of Computer Vision Research

Quarterly Journal

Fall 1997, Volume 1, Number 1

The MIT Press

Article 2

A Unified Approach to Image Matching and Segmentation in Stereo, Motion, and Object Recognition via Recovery of Epipolar Geometry

Gang Xu

Videre: Journal of Computer Vision Research (ISSN 1089-2788) is a quarterly journal published electronically on the Internet by The MIT Press, Cambridge, Massachusetts, 02142. Subscriptions and address changes should be addressed to MIT Press Journals, Five Cambridge Center, Cambridge, MA 02142; phone: (617) 253-2889; fax: (617) 577-1545; e-mail: journals-orders@mit.edu. Subscription rates are: Individuals $30.00, Institutions $125.00. Canadians add additional 7% GST. Prices subject to change without notice.

A Unified Approach to Image Matching and Segmentation in Stereo, Motion, and Object Recognition via Recovery of Epipolar Geometry

Gang Xu[1]

In this paper I try to show that through recovering epipolar geometry we can provide a unified approach to the problems of image matching and segmentation in stereo, motion, and object recognition, which have been treated separately so far. Stereo matching has been known as a 1D search problem, while matching in motion and object recognition have been known as 2D search problems. I show that by recovering epipolar geometry underlying the images, the correspondence search problems in motion and object recognition can also be changed to be 1-dimensional. I propose a new approach to recovering epipolar geometry for multiple rigid motions using feature points in two uncalibrated images. Using the recovered epipolar equations, the edge images are then easily matched and segmented. Examples are shown for matching and segmenting motion images with multiple rigid motions, and for matching model view against input view and localizing the 3D object in the input view.

Keywords: clustering, correspondence, epipolar geometry, motion, object recognition, robust estimation, stereo

1. Computer Vision Laboratory, Department of Computer Science, Ritsumeikan University, Kusatsu, Shiga 525, Japan

Introduction

In this section, we review correspondence problems in stereo, motion, and object recognition from the point of view of epipolar geometry and show that once the epipolar geometry is recovered, all can be redefined as a 1D correspondence search problem plus a segmentation problem that can be solved simultaneously.

Stereo

Stereo is one of the earliest problems treated in computer vision [12, 6]. The epipolar constraint has been well known from the beginning, and is used to ease the difficulty of matching. Most algorithms assume that the epipolar lines are given *a priori,* and thus pose the stereo matching problem as a 1D search problem.

The classical technique for obtaining the epipolar geometry is by using an object with known shape and sizes [4, 23, 22, 3]. Often, two cameras are mechanically set to have parallel optical axes, such that the epipolar lines are horizontal in both images [6].

Here we would like to point out that, first, human eyes do not work this way [5]. Rather, they do correspondence using *vergence;* thus the angle between the two eyes is not zero and is constantly changing. Second, even if we try carefully to arrange the imaging geometry that way, there is still error, and usually the corresponding points are not strictly on the same horizontal lines.

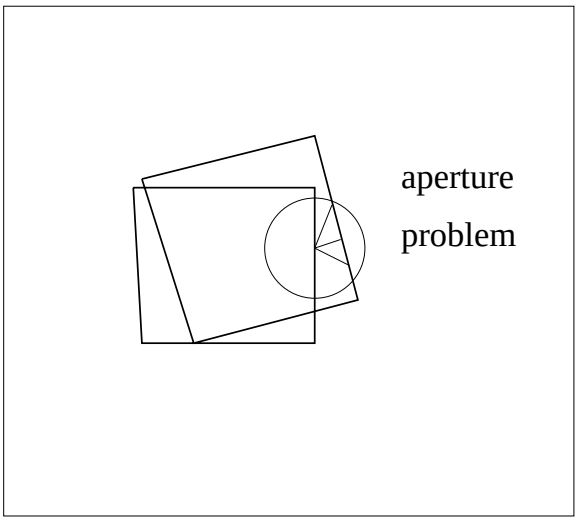
Therefore, in the general case, calibration is necessary to recover the epipolar geometry accurately. As will be seen later, the epipolar geometry can be recovered from two uncalibrated images alone, without the use of any calibration apparatus. Once this is done, we say that the two images are weakly calibrated [15, 31].

Motion

Motion is also one of the oldest problems in computer vision. Ullman's pioneering work on this problem has had great influence on later research [24]. The problem has been divided into two smaller ones. The first is *correspondence*, which decides how points "flow" between consecutive images in the image sequence. The second is called *structure-from-motion*, which determines the 3D structure and motion of the objects from matched points in the images. Most of the past research tries to solve one of the above two problems.

For the correspondence problem, the difficulty is considered to be the *aperture problem,* as shown in Figure 1. That is, in motion, unlike

Figure 1. The aperture problem.



stereo, the displacement is not *a priori* considered constrained to be one-dimensional; rather, it is two-dimensional. The array of displacements is called *optical flow*. Thus, if we see things locally with a small aperture, then there are an indefinite number of possible solutions. Solutions are uniquely determined only for those structures like corners, junctions, spots, etc., which are characterisitic enough to be uniquely localized and distinguished (see Fig.1). Only if we see things with apertures large enough to include these structures can we propagate their flows as constraints, and use them to determine flows for other points, which otherwise would not have unique solutions.

This idea leads to a whole family of algorithms which impose a 2D smoothness constraint on the flow field [8, 7]. For points which happen to have unique solutions, their flows are so determined. For points which do not have unique structures, their flows are determined by minimizing a global sum of derivatives of the flow field.

This family of approaches has three common problems. Firstly, correspondence determined this way is not guaranteed to produce the correct result. For example, in the case of rigid motion, optical flow obtained by imposing a 2D smoothness constraint does not satisfy the rigidity constraint in general. Secondly, the smoothness constraint is not valid along flow discontinuities. Unfortunately, these discontinuities are not known *a priori*. Thus, applying this constraint *blindly* not only gives wrong answers along these discontinuities, but these wrong answers can also propagate to neighboring areas, whose influence can reach very far from those discontinuities. Thirdly, minimizing a global functional of 2D flows is very computationally costly. Even the deepest descent method, which is the least costly algorithm currently available, consumes a lot of computation power. Let alone those algorithms that seek global minimums.

Object Recognition

Object recognition is another of the hardest problems in vision. Actually, it is called the final objective of vision. Marr claims, "Vision is the process of knowing what is where through seeing" [11]. "knowing what" is exactly the problem of object recognition.

The approaches to object recognition have been deeply influenced by Marr's philosophy [11]. Marr claims that to recognize 3D objects, one

must have enough 3D information about the objects to be recognized; thus only when the full 3D shape is recovered from images can the process of recognition start. Under his influence, most early approaches to object recognition assume that 3D object models are given *a priori*, and the task is then to find a particular transformation that projects the model onto a particular part of the image, usually segmented from the background in advance. The problem with this approach is that 3D models are not easily available. It is not easy to determine the shape of objects from input images through stereo, or motion, or other visual cues.

Recently, there have been a number of attempts to avoid 3D object models, but instead to use 2D model views. Ullman proposes to use linear combination of matched model views as object models. [25]. Poggio proposes to use a number of matched views to train an approximation network whose internal representation functions as an object model [14]. Xu proposes to use only one image as an object model [27].

Anyway, if the problem of object recognition is posed as one of matching model views with the input image—which not only includes the target objects but also the background—then correspondence between the model views, and between model view(s) and input view, becomes essential and necessary. We believe that correspondence, localization and recognition are essentially the same process.

Correspondence in Stereo, Motion, and Object Recognition as a 1D Search

Now we redefine the three problems from the perspective of epipolar geometry. The correspondence problem in both motion and object recognition is changed to be a 1D search problem, similar to that in stereo, if the epipolar geometry underlying the images is recovered.

As described above, the stereo correspondence can be solved in a two-step process. In the first step, the epipolar geometry is determined, and in the second step, the correspondence is determined as a 1D search along epipolar lines.

If the scene is stationary, there is no difference between two motion images and a pair of stereo images. What differs between a general motion problem and a general stereo problem is that, in a motion problem, we can have different motions simultaneously. The camera can move, the scene can move in a different way, and more importantly, there can be multiple objects moving independently in the scene. In stereo, only the camera moves. Thus, between general motion images, there can be multiple epipolar geometries, while in stereo there can only be one.

If we can somehow recover all the epipolar geometries underlying two motion images, then the search for correspondence is reduced from 2-dimensional to 1-dimensional. Thus, the aperture problem no longer exists, though ambiguity still exists, or multiple candidates still exist along the epipolar lines, as in stereo matching. The problem of using model views for 3D object recognition can be further divided into two cases. One is of using only one model view, and the other is of using multiple views. Here we stress that, in either case, the correspondence between the model and input views, and between model views is essential. Matching model views is nothing but matching two uncalibrated stereo images. It is also basically the same thing as matching the model view with the input view.

If the object in one model view is the same as the object in the input view, then they should satisfy the epipolar geometry, and the points in model views can be matched against the points in the input view. The problem is again divided into one of finding possible epipolar geometry between a model view and the input view, and one of matching image points based on the recovered epipolar geometry between the model view and the input view. If the input view also includes other objects and background, finding the corresponding points to the model is also localizing the object from other objects and background.

The part of recovering epipolar geometry between a model view and an input view is essentially the same as that in uncalibrated stereo. The difference is that usually there are other objects and background in the input view, which, as described later, brings difficulty to epipolar equation recovery.

Relation to Previous Work

It is noted that recovering epipolar geometry between two uncalibrated images under full perpective projection has been proposed by Zhang et al. [32]. In this paper we are concerned with weak perspective projection. There are only 4 degrees of freedom in the epipolar geometry under weak perspective projection [17], while for full perspective projection there are 7. This reduction in complexity gives us more freedom to treat problems like multiple rigid motions, and object recognition with large portions of background.

It is known to be difficult to robustly recover multiple epipolar geometries from two views without a given correspondence [20]. In this paper we propose a new clustering technique to solve this problem. Also it is novel to see the image matching and correspondence problems in stereo, motion and object recognition as an identical problem and approach it from the viewpoint of epipolar geometry.

For a more comprehensive description of the epipolar geometry under both the full and weak perspective projections, and the epipolar geometry-based approach to stereo, motion and object recognition, please refer to the recent monograph "*Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*"[29].

Epipolar Equation under Weak Perspective Projection

It is well known that under orthographic, weak perspective and paraperspective projections, the epipolar equation is linear about image coordinates [9, 17, 29]. To be self-contained, a derivation of the epipolar equation is given here first, and a geometric interpretation is then given to the equation.

Deriving the Epipolar Equation

We start from the rigid motion equation

$$\mathbf{X} = \mathrm{R}\mathbf{X}' + \mathbf{t} \tag{1}$$

To eliminate Z and Z' we multiply $\mathbf{v}^T = [\, r_{23} \quad -r_{13} \quad 0\,]$ to both sides of it, yielding

$$\mathbf{v}^T\mathbf{X} = \mathbf{v}^T\mathbf{R}\mathbf{X}' + \mathbf{v}^T\mathbf{t} \tag{2}$$

Thus (2) leads to

$$-r_{23}X + r_{13}Y - r_{32}X' + r_{31}Y' + r_{23}t_X - r_{13}t_Y = 0 \tag{3}$$

Assuming the weak perspective projection, we can rewrite the above equation as

$$-r_{23}\frac{Z_c}{f}(u - u_0) + r_{13}\frac{Z_c}{f}(v - v_0) - r_{32}\frac{Z'_c}{f'}(u' - u'_0) + r_{31}\frac{Z'_c}{f'}(v' - v'_0) + r_{23}t_X - r_{13}t_Y = 0 \tag{4}$$

where (u, v) and (u', v') are the measurable image coordinates. Here we assume that the camera optical axis is perpendicular to the image plane, and the vertical and horizontal sizes of pixels are the same. (In the case of CCD cameras, this model is close to reality.) Equation 4 can be rewritten as

$$pu + qv + su' + tv' + c = 0 \tag{5}$$

where

$$p = -r_{23}\frac{Z_c}{f}$$

$$q = r_{13}\frac{Z_c}{f}$$

$$s = -r_{32}\frac{Z'_c}{f'}$$

$$t = r_{31}\frac{Z'_c}{f'}$$

$$c = r_{23}\frac{Z_c}{f}u_0 - r_{13}\frac{Z_c}{f}v_0 + r_{32}\frac{Z'_c}{f'}u'_0 - r_{31}\frac{Z'_c}{f'}v'_0 + r_{23}t_X - r_{13}t_Y$$

Geometric Interpretation

There are many different ways to define a 3D rotation. One of them is to define an arbitrary 3D rotation by three consecutive rotations around the coordinate axes, that is, a rotation by α around the z-axis first, then a rotation by β around the new y-axis, and finally a rotation by $-\gamma$ around the new z-axis.

$$R = R_z(\alpha)R_y(\beta)R_z(-\gamma) \tag{6}$$

$(\alpha, \beta, -\gamma)$ are the same as the *Euler angles*, which are widely used in kinematics of robot manipulators [13]. Note that $-\gamma$ defined with respect to the first image is equivalent to rotating the second image by γ.

The first and third rotations are actually two rotations within the image planes, while only the second rotation is related to depth. Representing **R** by the three angles α, β and $-\gamma$, we have

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta\cos\gamma - \cos\alpha\sin\gamma & \sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta \\ -\sin\beta\cos\gamma & -\sin\beta\sin\gamma & \cos\beta \end{bmatrix}$$

Substituting the components of R for 4, we have

$$\sin\beta\Big(-\frac{Z_c}{f}u\sin\alpha+\frac{Z_c}{f}v\cos\alpha+\frac{Z'_c}{f'}u'\sin\gamma-\frac{Z'_c}{f'}v'\cos\gamma$$
$$+t_X\sin\alpha-t_Y\cos\alpha+\frac{r_{23}Z_cu_0}{f}-\frac{r_{13}Z_cv_0}{f}$$
$$+\quad\frac{r_{32}Z'_cu'_0}{f'}-\frac{r_{31}Z'_cv'_0}{f'}\Big)=0 \tag{7}$$

If $\beta\neq 0$, the above can be rewritten as

$$-u\sin\alpha+v\cos\alpha-\rho(-u'\sin\gamma+v'\cos\gamma)+\lambda=0 \tag{8}$$

where α, γ, ρ, λ have two sets of values:

$$\begin{aligned}\alpha_1&=\text{atan2}(-p,q)\\ \gamma_1&=\text{atan2}(s,-t)\end{aligned} \tag{9}$$

$$\rho_1=\sqrt{\frac{p^2+q^2}{s^2+t^2}}$$

$$\lambda_1=\frac{c}{\sqrt{p^2+q^2}} \tag{10}$$

or

$$\begin{aligned}\alpha_2&=\text{atan2}(p,-q)\\ \gamma_2&=\text{atan2}(-s,t)\end{aligned} \tag{11}$$

$$\rho_2=\sqrt{\frac{s^2+t^2}{p^2+q^2}}$$

$$\lambda_2=\frac{-c}{\sqrt{p^2+q^2}} \tag{12}$$

where atan2(x,y) is the function for *arctangent* used in C programming. It is easy to get the second set of parameters by multiplying -1 to the two sides of (8). It is noted that

$$\begin{aligned}\alpha_1-\alpha_2&=\pm\pi\\ \gamma_1-\gamma_2&=\pm\pi\\ \rho_1&=\rho_2\\ \lambda_1&=-\lambda_2\end{aligned}$$

We further define

$$\theta=\alpha-\gamma \tag{13}$$

α, γ, θ, ρ and λ are called *motion parameters*. They are the only motion information that can be determined from the epipolar equation.

Here, ρ stands for the scale change between the two images caused by different depths Z_c and Z'_c and possibly different pixel scales, and λ stands for the translation along the direction perpendicular to the epipolar lines.

The first two terms of (8) can be thought of as the new vertical coordinate after a rotation of the first image by α, and the next two terms as the new vertical coordinate after a rotation of the second image by γ. Then the equation can be understood as saying that the new vertical coordinates are the same after a vertical translation of the first image by λ (see Fig. 2.) A similar interpretation was independently developed by

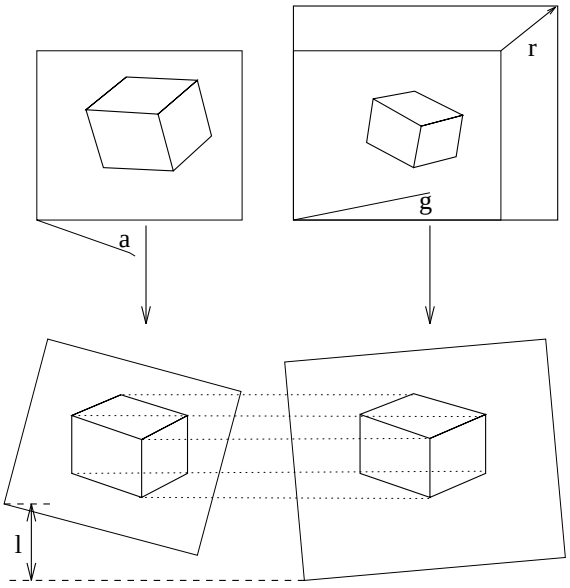


Figure 2. Corresponding points have the same horizontal coordinates after rotation, scaling and translation.

Shapiro *et al*. in [17], but they did not consider λ and the ambiguity in α and γ.

If β is zero, the epipolar equation disappears. The motion can be completely determined as a scaling, a rotation, and a 2D translation, which we call *2D affine motion*. We do not go into details, but only mention that the situation can be distinguished by examining the point coordinates. Details can be found in [29].

Recovery of Epipolar Geometry from Point Matches

In this section, we describe how to determine the epipolar equations, given point matches in pixel coordinates. Neither intrinsic nor extrinsic parameters are assumed to be known. The only assumption is that the points undergo a rigid transformation between the two camera coordinate systems.

Let the point match be $\mathbf{m}_i = [u_i, v_i]^T$ in the first image and $\mathbf{m}'_i = [u'_i, v'_i]^T$ in the second image. They must satisfy the epipolar equation $\tilde{\mathbf{m}}_i^T F_A \tilde{\mathbf{m}}'_i = 0$. It is expanded as

$$f_{13}u_i + f_{23}v_i + f_{31}u'_i + f_{32}v'_i + f_{33} = 0 \tag{14}$$

which is linear in image coordinates. If we define $\mathbf{f} = [f_{13}, f_{23}, f_{31}, f_{32}]^T$ and $\mathbf{u}_i = [u_i, v_i, u'_i, v'_i]^T$, then the above equation can be rewritten as

$$\mathbf{u}_i^T \mathbf{f} + f_{33} = 0 \tag{15}$$

It is easy to see that there are 4 degrees of freedom in the epipolar equation. Thus in general, a minimum of 4 pairs of matched points are required to uniquely determine the affine fundamental matrix.

If more than 4 point matches are available, we can use the least-squares method to determine the epipolar equation more robustly.

We can minimize the sum of squared Euclidean distances to the epipolar lines on which they are supposed to lie. For the first image and second image, they are respectively

$$d_1^2 = \sum_{i=1}^{n} \frac{\epsilon_i^2}{f_{13}^2 + f_{23}^2}$$

$$d_2^2 = \sum_{i=1}^{n} \frac{\epsilon_i^2}{f_{31}^2 + f_{32}^2} \tag{16}$$

Since the scales of the two images are generally different, it is reasonable to allow the two to make different contributions to the final result. Taking into account this scale change ρ, the criterion can then be a weighted sum of the two with the weights representing the scale change:

$$C_1 = \frac{1}{1+\rho^2} d_1^2 + \frac{\rho^2}{1+\rho^2} d_2^2 = 2 \sum_{i=1}^{n} \frac{\epsilon_i^2}{f_{13}^2 + f_{23}^2 + f_{31}^2 + f_{32}^2} = 2 \sum_{i=1}^{n} \frac{(\mathbf{p}_i{}^T \mathbf{f} + f_{33})^2}{\mathbf{f}^T \mathbf{f}} \tag{17}$$

This can also be considered as minimization of summation of squared distances of the 4D points $\mathbf{u}_i$ to the 4D hyperplane $\mathbf{u}^T\mathbf{f} + f_{33} = 0$.

There is a classical solution to the above minimization problem. The solution of f_{33} is

$$f_{33} = -\frac{\sum_{i=1}^{n} \mathbf{u}_i^T \mathbf{f}}{n} = -\mathbf{u}_0^T \mathbf{f} \tag{18}$$

where $\mathbf{u}_0 = [u_0, v_0, u_0', v_0']^T = \frac{1}{n}\sum_{i=1}^{n} \mathbf{u}_i$ is the average of all $\mathbf{u}_i$'s.

Let $\mathbf{v}_i = \mathbf{u}_i - \mathbf{u}_0$ and $W = \sum_{i=1}^{n} \mathbf{v}_i \mathbf{v}_i^T$. As W is symmetric and generally positive semi-definite, it has n real non-negative eigenvalues and n associated eigenvectors. The solution of $\mathbf{f}$ is the eigenvector associated with the smallest eigenvalue of W. For details, see [29, 17].

Note that the rank of W should be 3 if the motion can be modelled by the epipolar equation. If it is 2, then it is because either the motion is only a 2D affine motion or the 3D points are all coplanar.

Recovery of Multiple Epipolar Equations by Clustering

In the last section we described how to estimate epipolar equations, given a set of matched points in two images, that belong to the same rigid object or motion. In this section, we describe the problem of finding multiple epipolar equations as a problem of unsupervised clustering in the parameter space.

Given two images and feature points in them, the task is to segment the feature points into groups that each represent a rigid motion. This can be understood as a combination of segmentation and outlier rejection. The only constraint we use is that if the point matches belong to the same rigid motion, they must satisfy the same epipolar geometry.

We employ the generate-and-clustering strategy. In the weak perspective projection case, for each group of 4 neighboring point matches we can determine one epipolar equation. One epipolar equation, in turn, is projected onto the parameter space as two points. The task then becomes one of finding clusters in that space, where each pair of clusters represents an epipolar equation supported by those groups of corresponding points residing within that cluster.

Note that here we only work with weak perspective projection. The general approach should work as well for the full perspective projection, though with more difficulty as the dimensionality increases [21, 20].

Space of Motion Parameters vs. Space of Equation Coefficients

There are two spaces that we can possibly use for clustering: one is the space specified by the coefficients of the epipolar equation; the other is the space of motion parameters α, θ, ρ and λ, which can be computed from the epipolar equations.

The coefficients have uniform variances, but they are not independent of each other. For instance, the squared sum of f_{31} and f_{32} and that of f_{13} and f_{23} have very high correlation. Also the ratio of f_{31} and f_{32} and that of f_{13} and f_{23} have high correlation.

While the motion parameters computed from those coefficients have different variances, they are generally independent of each other. Based on the above property, we choose to use the space of motion parameters for clustering.

Definitions and Assumptions

Based on the above observations, we can model the points in the parameter space as a summation of probability processes representing the clusters, and a random process representing points resulting from wrong hypotheses of wrong correspondences. Since they are random, the probability of their forming clusters is small. Let the cluster centers be $\boldsymbol{\mu}_i, i = 1, \ldots, c$. The density function can be written as

$$p(\mathbf{x}|\mu) = \sum_{i}^{c} P(\boldsymbol{\omega}_i)\, p(\mathbf{x}|\boldsymbol{\omega}_i, \boldsymbol{\mu}_i) + a P(\boldsymbol{\omega}_0) \tag{19}$$

subject to

$$\sum_{i=1}^{c} P(\boldsymbol{\omega}_i) + P(\boldsymbol{\omega}_0) = 1 \tag{20}$$

where $p(\mathbf{x}|\boldsymbol{\omega}_i, \boldsymbol{\mu}_i)$ is the probability distribution for cluster $\boldsymbol{\omega}_i$ with center $\boldsymbol{\mu}_i$, $P(\boldsymbol{\omega}_i)$ is the probability of belonging to i-th cluster, $P(\boldsymbol{\omega}_0)$ is the probability of belonging to none of the clusters, and $aP(\boldsymbol{\omega}_0)$ is the constant density function corresponding to the random process.

$p(\mathbf{x}|\boldsymbol{\omega}_i, \boldsymbol{\mu}_i)$ has to be a function decreasing monotonically with distance from the center $\boldsymbol{\mu}_i$. Usually a Gaussian function is used. This means that, due to errors arising from discretization and calculations, there is a difference between the estimated epipolar equation and the true epipolar equation, and the difference obeys the Gaussian distribution. Suppose that the covariance matrix is identical for all clusters. Then the Gaussian distribution can be written as

$$p(\mathbf{x}|\boldsymbol{\omega}_i, \boldsymbol{\mu}_i) = (2\pi)^{-2} \mid \boldsymbol{\Sigma} \mid^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\} \tag{21}$$

$P(\boldsymbol{\omega}_i)$, $i = 0, \ldots, c$ are not known *a priori*. They can be estimated if the number of points for each cluster and the number of random points are known. Assuming that the total number of points in the space is n, the number of points for the i-th cluster is n_i, and the number of random points is n_0, subject to $\sum_{i=0}^{c} n_i = n$, then

$$P(\omega_i) = \frac{n_i}{n}, \qquad i = 0, \ldots, c \tag{22}$$

Needless to say, the ease of finding the clusters depends on how many random points exist. Thus it is vital to limit the number of random points, that is, $P(\omega_0)$.

Estimating Covariance Matrix

Assuming that the distribution of points in the motion parameter space obeys a Gaussian function, it is necessary to estimate its covariance matrix in order to use it for clustering. Though the covariance matrix varies with different images and different motions, it is still feasible to assume that a few typical covariance matrices represent the range within which any variance matrix may reside. Thus the approach we take is to estimate a few variance matrices from examples, and use them for clustering.

Now suppose we are given n points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_n$. μ is estimated as

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k \tag{23}$$

and the covariance matrix is estimated as

$$\boldsymbol{\Sigma} = \frac{1}{n} \sum_{k=1}^{n} (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T \tag{24}$$

The covariance matrix we use for clustering is

$$\begin{bmatrix} 1231.437134 & -32.256741 & -0.746871 & 440.431793 \\ -32.256741 & 8.477827 & 0.007116 & 22.509176 \\ -0.746871 & 0.007116 & 0.003669 & -0.882438 \\ 440.431793 & 22.509176 & -0.882438 & 539.617126 \end{bmatrix}$$

which is computed using Equation 24 from local groups of known correspondences between two real images.

The Maximal Likelihood Approach

Assuming that the n samples $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ are independent of each other, the joint density is the mixture density,

$$P(S|\boldsymbol{\mu}) = \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\mu}) \tag{25}$$

where $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_c\}$. The maximal likelihood approach tries to seek a $\boldsymbol{\mu}$ that maximizes $P(S|\boldsymbol{\mu})$.

A $\boldsymbol{\mu}$ that maximizes $P(S|\boldsymbol{\mu})$ also maximizes $\log P(S|\boldsymbol{\mu})$. The constraint on $\boldsymbol{\mu}$ is

$$\nabla_{\boldsymbol{\mu}_i} \log P(S|\boldsymbol{\mu}) = \mathbf{0}, \qquad i = 1, \ldots, c \tag{26}$$

From (25), we define $\mathbf{q}$ as

$$\mathbf{q} \equiv \nabla_{\boldsymbol{\mu}_i} \log P(\mathbf{S}|\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}_i} \sum_{k=1}^{n} \log p(\mathbf{x}_k|\boldsymbol{\mu}) = \sum_{k=1}^{n} \nabla_{\boldsymbol{\mu}_i} \log p(\mathbf{x}_k|\boldsymbol{\mu}) \tag{27}$$

$$i = 1, \ldots, c$$

Substituting (19) for the above equation yields

$$\mathbf{q} = \sum_{k=1}^{n} \{ \frac{1}{p(\mathbf{x}_k|\mu)} \nabla_{\boldsymbol{\mu}_i} \{ \sum_{j=1}^{c} p(\mathbf{x}_k|\boldsymbol{\omega}_j, \boldsymbol{\mu}_j) P(\boldsymbol{\omega}_j) + a P(\boldsymbol{\omega}_0) \} \} \tag{28}$$

$$i = 1, \ldots, c$$

Assuming independence of $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j (i \neq j)$, we have

$$\nabla_{\boldsymbol{\mu}_i} p(\mathbf{x}_k|\boldsymbol{\mu}) = \nabla_{\boldsymbol{\mu}_i} \{ p(\mathbf{x}_k|\boldsymbol{\omega}_i, \boldsymbol{\mu}_i) P(\boldsymbol{\omega}_i) \} = P(\boldsymbol{\omega}_i) \nabla_{\boldsymbol{\mu}_i} p(\mathbf{x}_k|\boldsymbol{\omega}_i, \boldsymbol{\mu}_i) \tag{29}$$

$$i = 1, \ldots, c.$$

Thus,

$$\mathbf{q} = \nabla_{\boldsymbol{\mu}_i} \log P(\mathbf{S}|\boldsymbol{\mu}) = \sum_{k=1}^{n} \frac{P(\boldsymbol{\omega}_i)}{p(\mathbf{x}_k|\mu)} \nabla_{\boldsymbol{\mu}_i} p(\mathbf{x}_k|\boldsymbol{\omega}_i, \boldsymbol{\mu}_i) \tag{30}$$

$$i = 1, \ldots, c$$

Using the Bayesian rule, the above equation can be rewritten as

$$\mathbf{q} = \sum_{k=1}^{n} P(\boldsymbol{\omega}_i|\mathbf{x}_k, \mu) \nabla_{\boldsymbol{\mu}_i} \{ \log [p(\mathbf{x}_k|\boldsymbol{\omega}_i, \boldsymbol{\mu}_i)] \}, \qquad i = 1, \ldots, c \tag{31}$$

Substituting (21) for the above equation and setting $\mathbf{q}$ to be zero yields

$$\mathbf{q} = \sum_{k=1}^{n} P(w_i|\mathbf{x}_k, \boldsymbol{\mu}) \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) = \mathbf{0}, \qquad i = 1, \ldots, c \tag{32}$$

That is,

$$\boldsymbol{\mu}_i = \frac{\sum_{k=1}^{n} P(\boldsymbol{\omega}_i|\mathbf{x}_k, \boldsymbol{\mu}) \mathbf{x}_k}{\sum_{k=1}^{n} P(\boldsymbol{\omega}_i|\mathbf{x}_k, \boldsymbol{\mu})}, \qquad i = 1, \ldots, c \tag{33}$$

For samples where $P(\boldsymbol{\omega}_i|\mathbf{x}_k, \boldsymbol{\mu})$ is small, little is contributed to $\boldsymbol{\mu}_i$. This is intuitively appealing and suggests using only those samples which are close to $\boldsymbol{\mu}_i$.

Equation 33 is difficult to apply directly, but it does suggest an iterative procedure. If we can obtain reasonable initial estimates for $\boldsymbol{\mu}_i(0), i = 1, \ldots, c$, they can be updated using

$$\boldsymbol{\mu}_i(t+1) = \frac{\sum_{k=1}^{n} P(\boldsymbol{\omega}_i|\mathbf{x}_k, \boldsymbol{\mu}(t)) \mathbf{x}_k}{\sum_{k=1}^{n} P(\boldsymbol{\omega}_i|\mathbf{x}_k, \boldsymbol{\mu}(t))}, \qquad i = 1, \ldots, c \tag{34}$$

until no significant change is available. This procedure involves updating the class means by readjusting the weights on each sample at each iteration. It provides a theoretical basis for the c-means clustering algorithm [16].

From the Bayesian rule, we get

$$P(\boldsymbol{\omega}_i|\mathbf{x}_k, \boldsymbol{\mu}) =$$

$$\frac{P(\boldsymbol{\omega}_i)(2\pi)^{-\frac{d}{2}} \mid \boldsymbol{\Sigma} \mid^{-\frac{1}{2}} \exp\{-\frac{1}{2}\|\mathbf{x}_k - \boldsymbol{\mu}_i\|^2_{\Sigma^{-1}}\}}{\sum_i P(\boldsymbol{\omega}_i)(2\pi)^{-\frac{d}{2}} \mid \boldsymbol{\Sigma} \mid^{-\frac{1}{2}} \exp\{-\frac{1}{2}\|\mathbf{x}_k - \boldsymbol{\mu}_i\|^2_{\Sigma^{-1}}\} + a P(\omega_0)} \tag{35}$$

where $\|\mathbf{x}_k - \mu_i\|^2_{\boldsymbol{\Sigma}^{-1}} = (\mathbf{x}_k - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu}_i)$. It is clear from the above equation that the probability $P(\boldsymbol{\omega}_i|\mathbf{x}_k, \boldsymbol{\mu})$ is large when $\mathbf{x}_k$ is close to $\boldsymbol{\mu}_i$. This suggests classifying $\mathbf{x}_k$ to class $\boldsymbol{\omega}_i$ when $\|\mathbf{x}_k - \boldsymbol{\mu}_i\|^2_{\boldsymbol{\Sigma}^{-1}}$ is small.

If $\mathbf{x}_k$ is much closer to μ_i than to any other μ_j, then

$$p(\mathbf{x}_k|\mu) \approx P(\omega_i)(2\pi)^{-\frac{d}{2}} \mid \mathbf{\Sigma} \mid^{-\frac{1}{2}} \exp\{-\frac{1}{2}\|\mathbf{x}_k - \mu_i\|^2_{\Sigma^{-1}}\} + a\,P(\omega_0)$$

Further, if the number of random points is also small, i.e., $P(\omega_0)$ is small, then $P(\omega_i|\mathbf{x}_k, \mu) \approx 1$. Substituting this for (34), it becomes a mere average, i.e.,

$$\mu_i(t+1) = \frac{\sum_{\mathbf{x}_k \subset R} \mathbf{x}_k}{\sum_{\mathbf{x}_k \subset R} 1} \tag{36}$$

where R denotes the range of $\mathbf{x}_k$ such that $\|\mathbf{x}_k - \mu_i\|^2_{\mathbf{\Sigma}^{-1}} < T$. Here T is a suitable threshold determining the size of R.

If the number of random points is not small, then $P(\omega_i|\mathbf{x}_k, \mu)$ is not close to 1, which means that a mere average does not give a correct answer.

Whether or not a class can be regarded as a cluster depends on how many points are in that class. For each class, we can count the number of points falling in that class; this count has to be larger than a threshold for any class to be regarded as a cluster representing a true epipolar equation.

Robust Estimation Using the Exponential of a Gaussian Distribution

As analyzed in the last subsection, if there are no random points the center can be determined as the mean vector of all the points falling within the range. However, if there are random points, merely taking the mean of all points is risky, because any random point within the range, but near the boundary of the range; results in a greater deviation from the true value. Thus, it is desirable to allow less contribution from points far from the center and more contribution from points close to the center.

To limit the influence of random points, we can use a function that decreases faster than the Gaussian with the distance from the center. One such option is the exponential of the Gaussian,

$$p(\mathbf{x}_k|\omega_i, \mu_i) = c_w \exp\{\exp(-\frac{1}{2}\|\mathbf{x}_k - \mu_i\|^2_{\mathbf{\Sigma}^{-1}})\} - c_w \tag{37}$$

where c_w is a constant chosen such that the integral of the distribution over the whole space is equal to 1. Differentiating the function with respect to $\mathbf{x}_k$ yields

$$\exp\{\exp(-\frac{1}{2}\|\mathbf{x}_k - \mu_i\|^2_{\mathbf{\Sigma}^{-1}})\} \exp(-\frac{1}{2}\|\mathbf{x}_k - \mu_i\|^2_{\mathbf{\Sigma}^{-1}})\mathbf{\Sigma}^{-1}(\mathbf{x}_k - \mu_i)$$

Since $\exp\{\exp(-\frac{1}{2}\|\mathbf{x}_k - \mu_i\|^2_{\mathbf{\Sigma}^{-1}})\}$ is always larger than 1, this function decreases faster than the Gaussian function itself. For its integral to be the same as that of the Gaussian, c_w must be larger than $(2\pi)^{-\frac{d}{2}} \mid \mathbf{\Sigma} \mid^{-\frac{1}{2}}$, the coefficient of the Gaussian.

Substituting c_w for equation 31, which does not depend on specific distribution functions, and setting $\mathbf{q}$ to be zero yields

$$\mu_i = \frac{\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \mu) \exp(-\frac{1}{2}\|\mathbf{x}_k - \mu_i\|^2_{\mathbf{\Sigma}^{-1}})\mathbf{x}_k}{\sum_{k=1}^{n} P(\omega_i|\mathbf{x}_k, \mu) \exp(-\frac{1}{2}\|\mathbf{x}_k - \mu_i\|^2_{\mathbf{\Sigma}^{-1}})}$$

$$i = 1, \ldots, c. \tag{38}$$

This means that μ_i is formed as a weighted summation of the $\mathbf{x}_k$, suggesting an iterative procedure.

From the Bayesian rule,

$$P(\omega_i|\mathbf{x}_k, \boldsymbol{\mu}) = \frac{P(\omega_i)c_w \exp\{\exp(-\frac{1}{2}\|\mathbf{x}_k - \boldsymbol{\mu}_i\|^2_{\Sigma^{-1}})\}}{\sum_i P(\omega_i)c_w \exp\{\exp(-\frac{1}{2}\|\mathbf{x}_k - \boldsymbol{\mu}_i\|^2_{\Sigma^{-1}})\} + aP(\omega_0)} \quad (39)$$

Compared with using the Gaussian distribution, for points close to μ_i, $P(\omega_i|\mathbf{x}_k, \boldsymbol{\mu})$ is even closer to 1, because c_w is larger than $(2\pi)^{-\frac{d}{2}} \mid \boldsymbol{\Sigma} \mid^{-\frac{1}{2}}$. This confirms that using the exponential of a Gaussian does produce more robust results than using the Gaussian itself.

We now have an iterative procedure,

$$\boldsymbol{\mu}_i(t+1) = \frac{\sum_{\mathbf{x}_k \subset R} w(\mathbf{x}_k, \boldsymbol{\mu}_i(t))\mathbf{x}_k}{\sum_{\mathbf{x}_k \subset R} w(\mathbf{x}_k, \boldsymbol{\mu}_i(t))} \quad (40)$$

where $w(\mathbf{x}_k, \boldsymbol{\mu}_i(t)) = \exp\{-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_i(t))^T \boldsymbol{\Sigma}(\mathbf{x}_k - \boldsymbol{\mu}_i(t))\}$ and R is defined as the range within which points are classified as belonging to class ω_i. This equation shows that the center is a weighted mean of the points, with the points close to the center having larger weights, while points less close to the center have smaller weights. It agrees with our intuition. It can be proven that Equation 40 is actually an extension to a larger dimension of the 1D Welsch M-estimator [29].

The denominator can actually be used as a measure of the concentration of points. It does not merely count the number of points, but also takes the distribution into account. The more concentrated the points are, the higher the value. Renaming the denominator as C, we have

$$C = \sum_{\mathbf{x}_k \subset R} \exp\{-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)\} \quad (41)$$

Only those classes whose concentrations are higher than a threshold are regarded as indicating true epipolar equations.

Disparity and Spatial Disparity Space

Assuming that the epipolar geometry is recovered, finding correspondence between two motion images, or between a model view and an input view, is similar to that in stereo matching. Here we define disparity as the displacement of a pair of corresponding points along the epipolar lines in the two images.

Defining Disparity

It is easier to define disparity for images taken under the weak perspective projection. For how to define disparity under full perspective projection, see [29]. Given the coefficients of the epipolar equation, we can define the following transformation

$$\begin{bmatrix} \bar{u} \\ \bar{v} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (42)$$

and

$$\begin{bmatrix} \bar{u}' \\ \bar{v}' \end{bmatrix} = \rho \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix} + \begin{bmatrix} 0 \\ -\lambda \end{bmatrix} \quad (43)$$

such that every pair of corresponding points in the two new images have the same vertical coordinates.

$$\bar{v} = \bar{v}' \tag{44}$$

This situation is the same as the standard stereo images in the parallel camera case. It is thus easy to define the disparity. It is simply the difference between the horizontal coordinates of the corresponding points in the two new images,

$$d = \bar{u} - \bar{u}' \tag{45}$$

For a point (u, v), given disparity d, its correspoding point in the other image (u', v') can be directly computed as

$$\begin{aligned} \begin{bmatrix} u' \\ v' \end{bmatrix} &= \frac{1}{\rho} \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\ &+ \frac{1}{\rho} \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} -d \\ \lambda \end{bmatrix} \end{aligned} \tag{46}$$

For the other direction, given a point (u', v') and d, the corresponding point (u, v) is computed by the following equation,

$$\begin{aligned} \begin{bmatrix} u \\ v \end{bmatrix} &= \rho \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix} \\ &+ \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} d \\ -\lambda \end{bmatrix} \end{aligned} \tag{47}$$

Spatial Disparity Space and Smoothness

Looking for a correspondence for each point is the same as determining disparity for each point. This problem can be intuitively represented as a search in a 3D space, with the image as the first and second dimensions and disparity as the third dimension. This space (called *Spatial Disparity Space* or SDS) was originally proposed by Yang et al. for stereo matching [30].

Figure 3 gives an illustration of the space. Let us explain what SDS represents and how the constraints can be rephrased in terms of SDS.

A point (u, v, d) means a pair of correspondences between (u, v) in the first and (u', v') in the second image which has the disparity of d. The uniqueness constraint simply means that in SDS, each column (u, v) can have only one active point, that is, one disparity. If each column has only one active point, then the active points form a surface. The continuity constraint can be understood as a requirement that the neighboring columns should have continuous disparities. If there is a jump in disparity, it means a discontinuity.

In stereo, to find correspondence for each point in the left image is the same as finding a surface in SDS with $u - v$ coordinates identical to the left image. Usually we impose the smoothness constraint on matching, which is the same as requiring the surface to be as smooth as possible (there are several different mathematical representations for quantitizing smoothness [19]).

In motion, when there are different rigid motions, as long as their epipolar equations have been recovered, we can add as many blocks of SDS as there are motions, so that each block represents one motion. The uniqueness constraint applies again. That is, each column can have only one active node along it, no matter how many blocks there are, and no

Figure 3. Spatial Disparity Space for stereo matching.

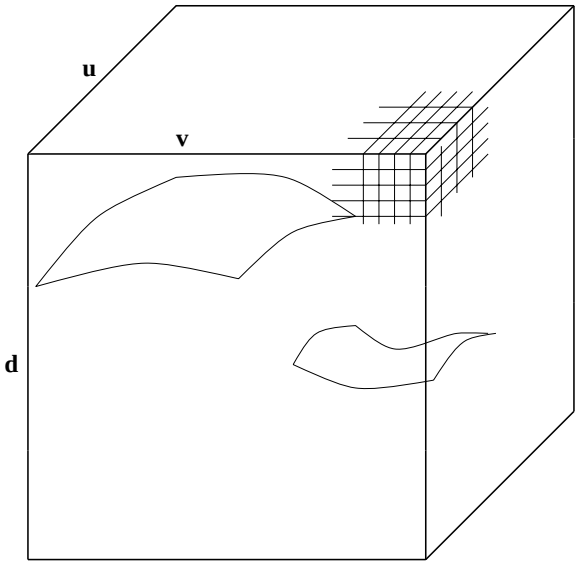
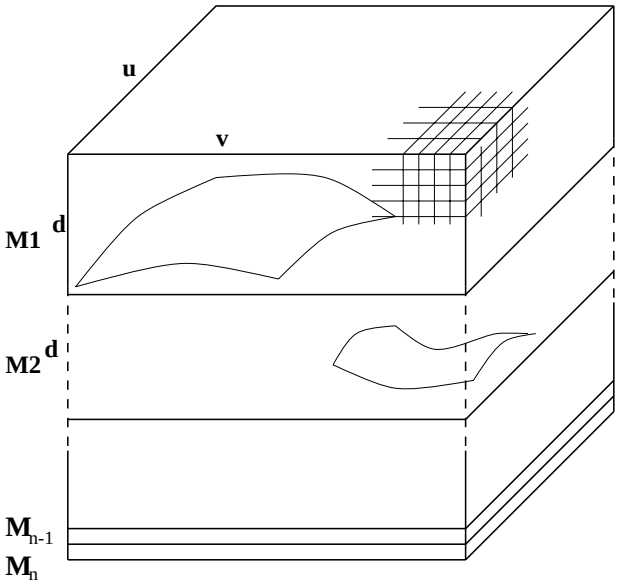


Figure 4. The Extended Spatial Disparity Space for multiple motions.



matter in which block the active node resides. We call this multi-block Spatial Disparity Space *Extended Spatial Disparity Space*, or ESDS.

Figure 4 gives an illustration of ESDS. Note that there are inactive insulator layers between every two neighboring blocks, so that local operations do not cover two different blocks. Also note that for 2D affine motions, each block has only one layer, which does not represent a range of possible disparities, but only represents whether that point belongs to a particular 2D affine motion.

In object recognition, once a possible epipolar equation is recovered, the SDS can be established in the same manner as in stereo. Each node represents a match between a point in the model image and a point in the input image.

If the $u - v$ coordinates of the ESDS are identical with the model image, then the task is to find which node is the true match. In this case, there is only one block.

If the $u - v$ coordinates of the ESDS are identical to the input image, the ESDS can represent the problem of finding identities for all the parts of the input image in the model database. Each part of the input image is matched against a particular model under a particular epipolar equation. The final result is a set of models that match individual parts of the input image. In this case, the problem is similar to that in multiple motions.

As always is the problem in matching, there are multiple candidates for correspondence. In terms of SDS, initially there are more than one active nodes for each column, of which at most one is correct. The problem of matching is to find out which candidate is the correct one.

The constraint commonly used for choosing one from the multiple candidates is the smoothness constraint. That is, the candidates that maximize a particular measure of smoothness over the whole visual field are selected as the matches. This is based on observations that surfaces in our physical world are always smooth except at discontinuites. [19]

Uncalibrated Stereo

Matching different images of a single scene remains a difficult task, despite many years of research. The only geometric constraint between two images is the epipolar constraint. However, in uncalibrated stereo, the epipolar geometry is unknown, because motion between two images can be arbitrary in this case. It is very important to develop a robust technique to match two uncalibrated images, which is the basis not only for stereo, but also for object recognition using model views.

The approach we propose aims at exploiting the only geometric constraint, i.e., the epipolar constraint, to establish robust correspondences between two weak perspective images of a single scene [26]. However, in order to reduce the complexity of the algorithm, we still exploit heuristic techniques to find an initial set of matches. We first extract high curvature points, and then match them using a classical correlation technique, followed by a new clustering technique. More precisely, our algorithm consists of three steps:

- Establish initial correspondences using some classical techniques.
- Estimate robustly the epipolar geometry.
- Establish correspondences using estimated epipolar geometry as in classical stereo matching.

The basic idea is first to estimate robustly the epipolar geometry, and then reduce the general 2D image matching problem to 1D stereo matching.

Note that Zhang et al. have proposed a matching technique for uncalibrated images under full perspective projection [32]. Since we assume weak perspective projection, computation is simpler in our technique, which allows us to simultaneously detect multiple epipolar equations for the case of multiple rigid motions.

Tentative Matching Between Feature Points by Correlation and Rotating Correlation Windows

First, feature points corresponding to high curvature points are extracted from each image. There are several techniques. Here we use the one proposed by Deriche et al. [2].

Since the epipolar geometry is not yet known, the search for a correspondence should be performed, theoretically, on the whole image.

Given a feature point $\mathbf{m}_1$ in image 1, we use a correlation window of size $(2n+1) \times (2m+1)$ centered at this point. We then perform a standard correlation between point $\mathbf{m}_1$ in the first image and all feature points $\mathbf{m}_2$ in the second image.

For a pair of points to be considered a match candidate, the correlation score must be higher than a given threshold. For each point in the first image, we thus have a set of match candidates from the second image (the set is possibly null); and at the same time we have also a set of match candidates from the first image for each point in the second image.

In uncalibrated stereo, the images may have been taken by cameras in different poses, thus large image torsions are possible. To deal with this problem, we have to allow the image windows to rotate. It is sufficient to rotate image windows in only one of the images. Since the two images play a symmetric role, rotating either of them is the same. Let us rotate windows in image 1.

There is no definite answer to the question of resolution of rotation angles. Empirical data show that 16 angles are usually sufficient. Each angle represents 22.5 degrees. Let us call them $\alpha_k, k = 1, \ldots, 16$. Now for each feature point in image 1, we can prepare 16 windows.

If there are M feature points in image 1 and N feature points in image 2, then we have to compute the matching scores for $16 \times \mathrm{M} \times \mathrm{N}$ times.

Since here we assume that the scene is stationary, it is not possible to have feature points being matched with largely different rotation angles. This is especially true for the weak perspective projection, as the image torsions α and γ are everywhere identical in the images.

This property implies that if the image is rotated for the correct angle, then matching scores should be high for all the corresponding feature points with that rotation angle. Under the weak perspective projection, this angle corresponds to θ. If we rotate image 2 by θ, then the correlation will give high matching scores for all corresponding feature points with that rotation angle θ. This suggests a histogram algorithm for evaluation of the rotation angle. As in the last subsection, if the score is higher than a predetermined threshold, that pair of feature points is considered a *match candidate*, with rotation angle θ_k. We can define an indicator $S(\mathbf{m}_{1i}, \mathbf{m}_{2j}, \theta_k)$ for the pair of points $\mathbf{m}_{1i}, \mathbf{m}_{2j}$ with angle θ_k, which is 1 if the score is higher than the threshold, and 0 otherwise. Now we can further define a counter for θ_k,

$$C(\theta_k) = \sum_{i=1,M} \sum_{j=1,N} S(\mathbf{m}_{1i}, \mathbf{m}_{2j}, \theta_k), \quad k = 1, \ldots, 16.$$

Once a single outstanding peak is found, only the match candidates with this particular rotation angle are used for later processes. Usually, ambiguity remains to be cleared.

Unique Correspondence by Robust Estimation of Epipolar Geometry

Using the correlation technique described in the previous subsection a point in the first image is usually paired to more than one point in the second image (which we call *match candidates*), and vice versa. For weak perspective images, there are only 4 degrees of freedom in the epipolar geometry. We can apply the clustering technique to robustly estimate the underlying epipolar geometry. The clustering technique is also good at rejecting outliers. To produce a hypothesis of epipolar geometry, we

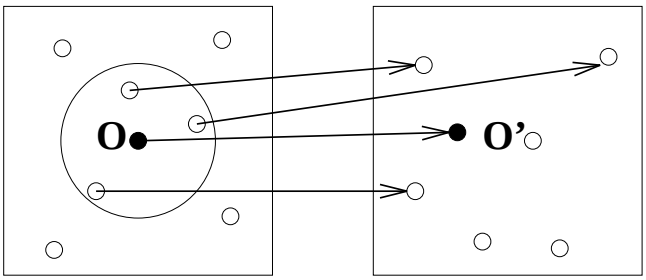


Figure 5. If a particular match candidate is inconsistent with other match candidates, it can be removed by computing the sum of relative distance differences with respect to all other match candidates.

need 4 pairs of point matches, which project onto the parameter space as a pair of points. To limit the number of points that lie outside the clusters, we need to further reduce the ambiguity in the initial matches. The essence here is that relative distances among neighboring points do not change drastically between images.

Suppose that we are given a list of match candidates $\mathbf{u}_k = [u_k, v_k, u'_k, v'_k]^T$, $k = 1, \ldots, n$ between point $[u_k, v_k]^T$ in the first image and $[u'_k, v'_k]^T$ in the second image. We now define the relative distance difference between $\mathbf{u}_k$ and $\mathbf{u}_l$ as

$$r_{kl} = \frac{\sqrt{(u_k - u_l)^2 + (v_k - v_l)^2} - \sqrt{(u'_k - u'_l)^2 + (v'_k - v'_l)^2}}{\sqrt{(u_k - u_l)^2 + (v_k - v_l)^2} + \sqrt{(u'_k - u'_l)^2 + (v'_k - v'_l)^2}} \tag{48}$$

To measure consistency of a particular match candidate $\mathbf{u}_k$ with other match candidates, we compute

$$R_k = \sum_{l=1, l \neq k}^{n} r_{kl} \tag{49}$$

As illustrated in Figure 5, if a particular match candidate is inconsistent with other match candidates, R tends to be very large. If it is larger than a threshold, then it is discarded from the list without further consideration. By doing so, the number of match candidates can be further reduced. In general, however, matches are still not unique.

In our current implementation, the threshold is the average of R for all match candidates.

To generate one hypothesis of epipolar geometry we need 4 point matches. If we generate groups of 4 point matches *randomly*, the number of generated hypotheses can become very large, if the number of feature points is over 10 or 15. The strategy we use to limit the number of hypotheses is to only generate local groups of point matches. For each pair of match candidates, we only generate groups by finding the 3 closest neighbors.

As illustrated in Figure 6, for a pair of points **O** and **O**′, we find the 3 closest neighbors for **O**. As the neighboring points do not necessarily have unique matches, there is usually more than one group of 4 point matches. For each group, we compute the relative distance difference r using Equation 48. If one of the match candidates causes inconsistency, then this group is discarded. If no group remains for the current 3 neighbors, then the trouble-making neighbor is removed, and the next closest neighbor is chosen to form new groups, until at least one group passes the consistency check.

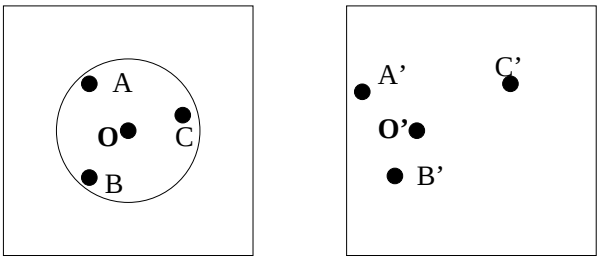


Figure 6. If a particular match candidate in the neighborhood is inconsistent with other match candidates, it is removed and the next closest neighbor is chosen.

For each group that passes the consistency check we compute one epipolar geometry, and project two points corresponding to the same epipolar geometry onto the parameter space. And after this, the clustering algorithm described in Section 4 is applied to find clusters in the parameter space. The corresponding epipolar equations are determined by substituting the motion parameters of the cluster centers for (8).

Once the epipolar equations are determined, we can check which initial match candidates satisfy the epipolar equation while keeping consistent with other matches. By doing this, the individual match that did not have chances to form correct groups can also be found.

Image Matching with the Recovered Epipolar Geometry

Once the fundamental matrix has been determined robustly, we can use the recovered epipolar constraint to determine correspondence for other image features, such as edge points.

Under the weak perspective projection, once the epipolar equations are recovered, disparity can be defined according to (45). Then the matching of edge images can be greatly simplified by making use of this constraint. We can construct the Spatial Disparity Space, in which all possible matches are represented as an active node. We need not search over all the image, once feature points like corners have been matched. The disparities for the matched feature points are computed and used as a rough estimate of the disparity range we need to search over.

Next, the task is to choose only one match for each edge point and delete all others by maximizing the smoothness in the SDS. A stochastic measure of smoothness can be defined as

$$E(u,v,d) = \sum_{u}\sum_{v}\sum_{i=-\Delta u}^{\Delta u}\sum_{j=-\Delta v}^{\Delta v}\frac{(\Delta d_{min}(u+j,v+j))^2}{i^2+j^2} \qquad (50)$$

where $\Delta d_{min}(u+j, v+j)$ is the possible minimal difference between $d(u,v)$ and other active nodes for $(u+j, v+j)$.

There are different algorithms to minimize this energy. We found in our examples that minimizing the energy independently for each edge point gives sufficiently good results, because the search range is quite limited due to knowledge of feature point matching. This algorithm is possibly the simplest among all possible minimization algorithms. It is also used in our approaches to the motion correspondence and segmentation problem, and to the object recognition and localization problem, as described in the following sections.

Figure 7. Two uncalibrated views of a Mac computer.

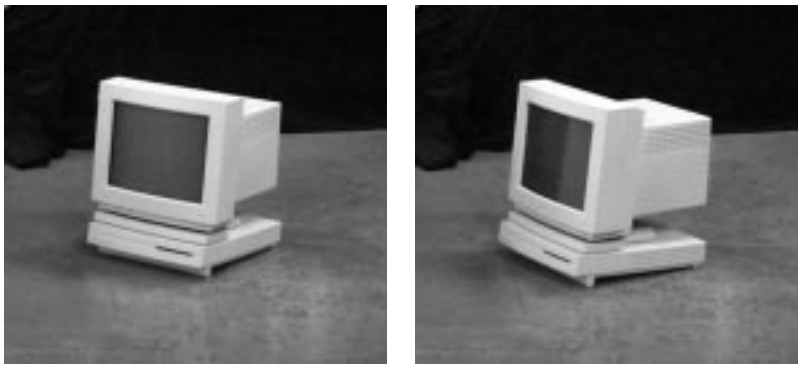
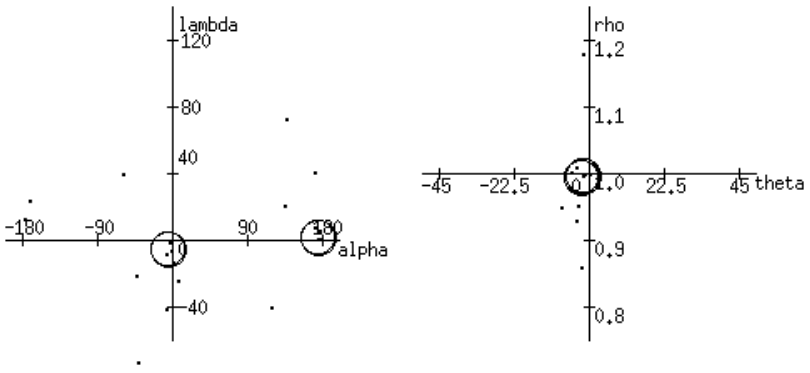


Figure 8. Clusters found for two views of a Mac computer: $\alpha_1 = -6.553879$, $\alpha_2 = 175.980881$, $\theta_1 = -2.845327$, $\theta_2 = -2.244064$, $\rho_1 = 0.996819$, $\rho_2 = 0.999406$, $\lambda_1 = -5.645007$, and $\lambda_2 = 2.335877$.



An Example of Matching Uncalibrated Stereo Images

Using the techniques described so far, we have successfully found the epipolar equation of the following pair of uncalibrated stereo images and matched the two images using the recovered epipolar equation.

Figure 7 shows two uncalibrated images of a Mac computer. Assuming the two images are taken by a weak perspective camera, we first extract feature points from the images, find possible matches by rotated template matching, and then use the inconsistency check to exclude inconsistent matches. From each pair of matches, the closest 3 neighbors in the first image are located, which form groups of 4 pairs of point matches. For each group, if the 4 pairs of matches are consistent with each other, an epipolar geometry is determined. If no group remains after the consistency check, then the trouble-making neighbor is found and removed, and the next closest neighbor is included. This process repeats until we can determine epipolar geometry. The computed epipolar geometries are then projected onto the motion parameter space, and clustering technique is used to find clusters in it. The result is shown in Figure 8. The concentration of the clusters for the true epipolar geometry is several times larger than other "clusters." The feature points that satisfy the recovered epipolar equation are numbered and shown in Figure 9. The epipolar equation is then recomputed using these matched points as

$$0.003u - 0.591v - 0.056u' + 0.576v' + 10.0 = 0$$

The edge images (Figure 10) are then matched using the recovered epipolar equation. The matched edge points in image 2 with respect to

Figure 9. Matched feature points are marked by the same numbers.

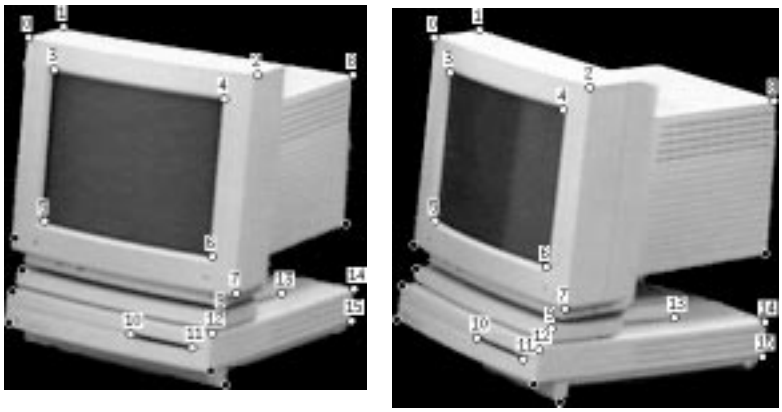


Figure 10. Edges in the two images.

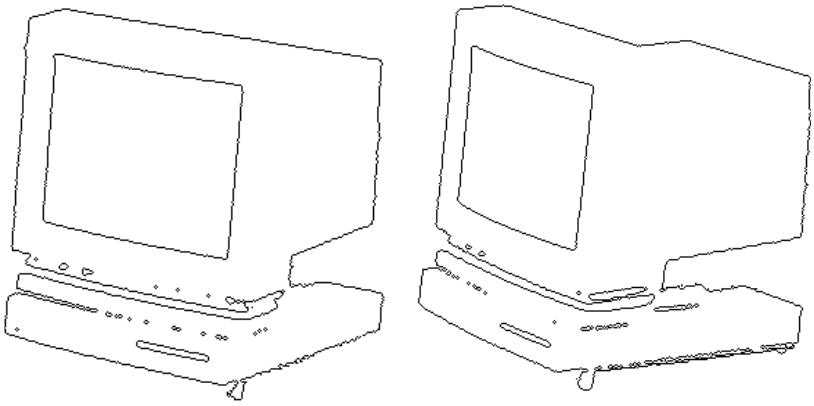


Figure 11. Edge points matched in image 1 with respect to image 2.

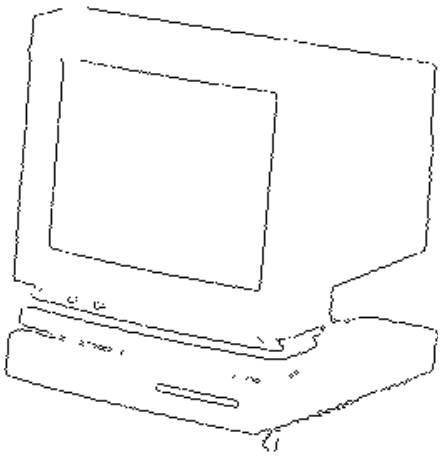


image 1 are shown in Figure 11. It can be seen that most edge points are correctly matched. The unmatched points near the upper left and bottom right corners are due to the mispositioning of epipolar lines for a few pixels along the perpendicular direction.

The matched edge images can be used as model views for recognizing and localizing the Mac computer in other images. See Section 8.

Multiple Rigid Motions: Correspondence and Segmentation

Motion correspondence and segmentation has been one of the main research topics in computer vision.

We start with matching feature points like corners and junctions detected by a corner detector and match them by template correlation. They are then grouped by the epipolar constraint. The obtained epipolar equations are then used for correspondence, as the epipolar constraint reduces the search space from 2 dimensions to 1 dimension, thus eliminating the "aperture" problem. It looks like a problem of stereo matching, but it differs in that there is more than one set of epipolar lines in the images and the boundary is not given.

Segmenting Multiple Motions by Matching Feature Points

Now given two images, we first match feature points by template correlation techniques. Secondly, to find the epipolar equations underlying the two images, we use the generate-and-test strategy, that is, we generate hypotheses of groups of matched points, compute epipolar equations for them, and then see which are shared by many points. The clustering algorithm is discussed in detail in Section 4.

The procedure is:

1. Find corners, junctions, and other feature points in the two images.
2. Establish correspondence between the two sets of feature points by computing correlation between local image patterns around feature points.
3. For each matched point in image 1, find the $k (k \geq 3)$ (in the current implementation, k is 6) closest neighbors, and form a group for each combination of 3 neighbors.
4. For each group, compute the epipolar equation and the corresponding motion parameters; if the points undergo a 2D affine motion (if the third eigenvalue of **W** is smaller than 1.0 in this implementation), then determine the 2D affine motion equations and motion parameters; if the computed motion parameters are too large, discard the group.
5. Find clusters in the motion parameter space.
6. For each cluster, compute the epipolar equation.
7. Merge individual matches that satisfy the epipolar motion equations.

In the current implementation, k is 6. Due to the limited number of feature points in each image, the computation is not as terrible as one might imagine, even though we have to determine the equations for each combination of local groups. Moreover, much of the computation can be done in parallel. The 2D affine motions, if any, can be found in the same way.

We have implemented the algorithm to recover the multiple epipolar equations from feature matching in motion images. Two image sequences are used. One is taken by a moving camera of a scene including a static background and a moving soccer ball. Since the camera is moving, every point in the image is apparently moving. The second image sequence is taken by a static camera of a scene in which two soccer balls

Figure 12. Motion image 1 (left) and motion image 2 (right).

Figure 13. The optical flow of feature points.

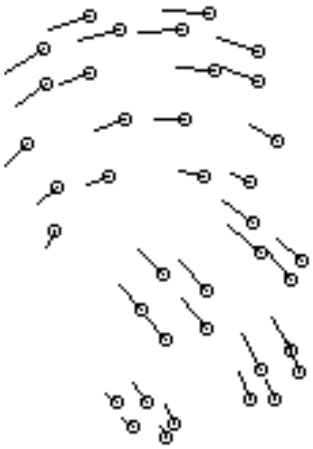
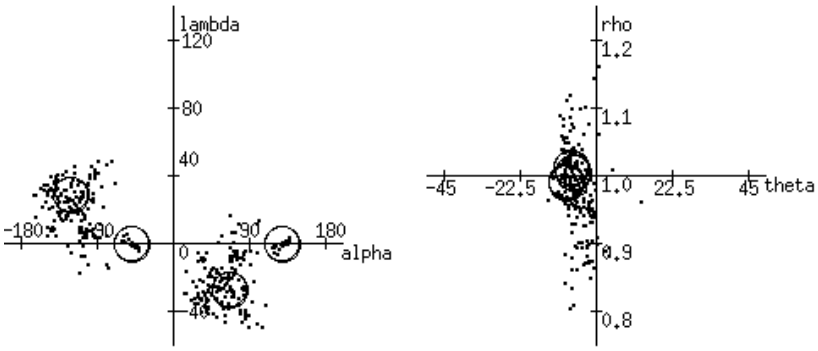


Figure 14. The clusters for the 2 balls in motion parameter space. There are too few points from the background to form a cluster. For ball 1, $\alpha_1 = 66.773567$, $\alpha_2 = -122.562492$, $\theta_1 = -7.010643$, $\theta_2 = -7.681648$, $\rho_1 = 1.007331$, $\rho_2 = 1.013413$, $\lambda_1 = -27.145382$, and $\lambda_2 = 29.840086$. For ball 2, $\alpha_1 = 129.210052$, $\alpha_2 = -50.059856$, $\theta_1 = -9.476761$, $\theta_2 = -9.445133$, $\rho_1 = 0.992618$, $\rho_2 = 0.990300$, $\lambda_1 = -0.544330$, and $\lambda_2 = -0.357600$.



move independently in front of a static background. Thus there are three motions: the two balls undergoing rigid motions represented by epipolar equations, and the background represented by a 2D affine motions. We only have space to show the results for the second sequence.

First, feature points are detected by a modified feature detection operator, see [1, 10]. They are matched through the image sequence (as long as possible). Since the motion between successive images may not be sufficient for clusters to emerge in the motion parameter space, the decision is prolonged until the motion flows are long enough to determine reliable epipolar equations.

Figure 12 shows an example where there are two balls moving independently; Figure 13 shows the flow of the feature points. The clusters for the two balls are located in the motion parameter space shown in Figure 14. The results of segmentation of the flows are shown in Figure 15.

Figure 15. The flows of ball 1 (left) and ball 2 (right).

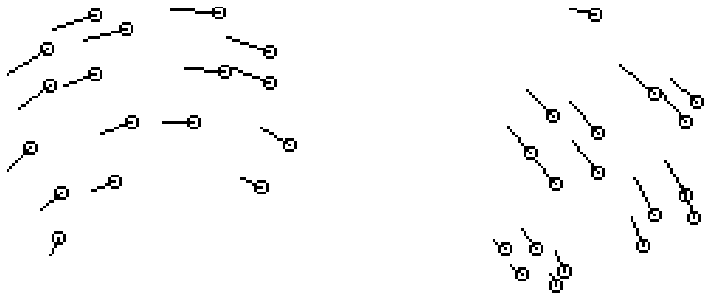
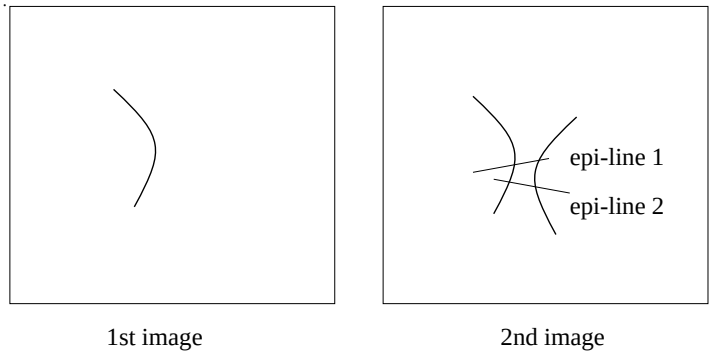


Figure 16. Disparities defined along multiple epipolar lines.



Note that the clustering is not necessary for every pair of consecutive images. Actually, once the feature points are classified into different motions and objects, the groupings can be kept through the image sequence as long as they are visible.

Matching and Segmenting Edge Images with Known Epipolar Equations

Once the epipolar equation is known, by rotating the two images so that the corresponding points always lie on the same horizontal lines, we can define disparity by $d = \bar{u} - \bar{u}'$. To find correspondence for each point is to find which epipolar geometry it belongs to, and to determine its motion disparity associated with that epipolar geometry.

The problem is best illustrated in *Extended Spatial Disparity Space*. As shown in Fig. 41, the Extended Spatial Disparity Space (ESDS) has m layers. F_i represents an epipolar motion. For SDS, there is only one such layer. In ESDS, each node represents a possible correspondence between I_1 and I_2, thus a disparity under one of the epipolar geometry. The height of each layer corresponds to the disparity range, which may be different from motion to motion. If we allow the disparity changes from $-D$ to D, then the disparity range will be $2D + 1$.

For the 2D affine motions, the thickness of each layer is only 1 pixel, because the motion is completely determined.

For each edge point in the first frame, we find out the possible corresponding edge points in the second frame, for which the corresponding disparity, or corresponding nodes in the ESDS, are marked *active*. The operation is illustrated in Figure 16.

For each column, we usually have multiple active nodes (in Figure 16 there are two epipolar lines and 4 active nodes), thus making it necessary to use the smoothness constraint to choose only one from among

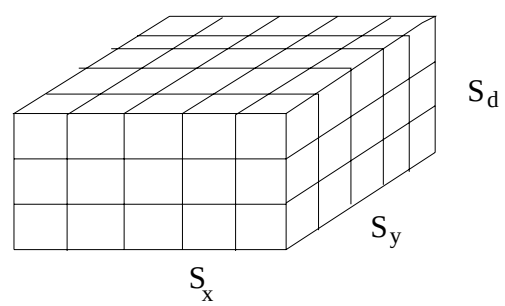


Figure 17. The box range for counting support from neighboring edge points.

the 4 candidates. It can be easily seen from Figure 16 that correct matching does have smoother disparity changes.

The criterion for selection is the smoothness constraint. That is, we select the disparity that is least different from the disparity values of its neighbors. For each disparity, we define a box centered at that node, with the size to be S_x, S_y and S_d (Figure 7). Each active node within the box is counted (since these represent possible disparities), according to a point system, which gives 2 points if there is no disparity difference, 1 point if the disparity difference is equal to 1 pixel, and 0 points otherwise. This is a kind of support for the disparity from neighbors. The higher the total number of points is, the more neighboring points have similar disparities. This, as a smoothness measure, is different from the derivatives defined for a continuous field. Here, for discrete edge points, we can only define a kind of stochastic measure for smoothness.

The total score for a node $C(u, v, d)$ is defined as

$$C(u, v, d) = \sum_{y \in S_y} \sum_{x \in S_x} \max_{d \in S_d} P(x, y, d) \tag{51}$$

where $P(x, y, d)$ is the score that the node (x, y, d') contributes to $C(u, v, d)$:

$$P(x, y, d) = \begin{cases} 2 & d - d' = 0 \\ 1 & d - d' = \pm 1 \\ 0 & \text{otherwise} \end{cases}$$

Along each column in the ESDS, or for each image point (u, v), the disparity node which has the largest count is selected as the true match, and the motion that this disparity represents is the motion that this edge point belongs to.

As the disparity value cannot change very much between successive images, this simple measure gives a good estimate of the disparity variation. Experimental results show astonishingly good correspondences and correct segmentation between edge images.

We have tested the above algorithm to a number of motion images.

Figure 18 shows a pair of edge images taken out from the second image sequence. The epipolar equations for the two moving balls and the motion equations for the background are computed and used for matching the two edge images. The results are shown in Figure 19 for ball 1 and ball 2, and in Figure 20 for the background.

It can be seen from these figures that most of the edge points are correctly segmented. This is encouraging, because we used only very simple local operations. Some of the misclassified edge points are due to poor performance of the edge detectors, which fail to detect some edge points at important locations. For more about this approach, refer to [28].

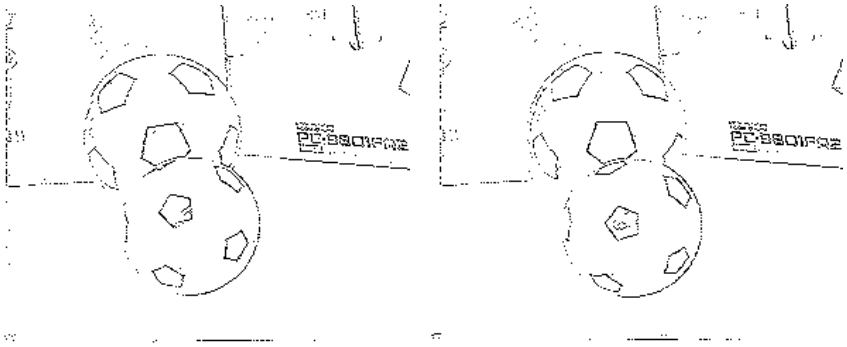


Figure 18. Motion image 1 (left) and motion image 2 (right).

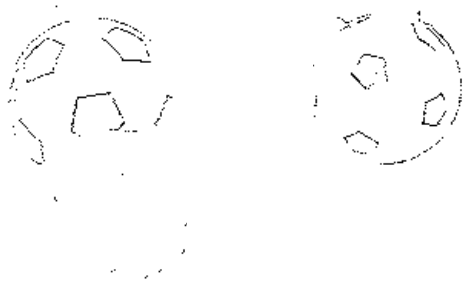


Figure 19. Segmented ball 1 (left) and 2 (right).

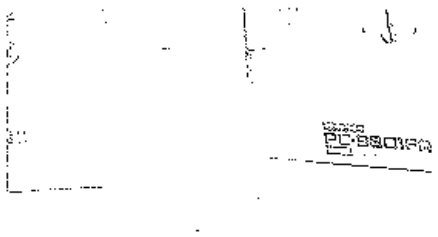


Figure 20. The matched edge points for the background.

3D Object Recognition and Localization

The conventional approaches to 3D object recognition are mostly based on 3D object models. Unfortunately, however, 3D data are not always available for every object. Therefore, it is a natural choice to avoid 3D models and to use 2D model views instead. Then the problem becomes one of matching between model images, and matching between a model image and an input image. Again the epipolar constraint can be applied.

Recognition and Localization with a Single Model View

Using only one model view for 3D object identification is essentially an underconstrained problem in two senses [27]. First, there is no guarantee that we will be able to find a set of unique correspondences. Second, even if we can find a unique solution, there is no guarantee that the two views are of the same object. One extreme example is that some objects may appear completely the same in an image but differ in 3D shape.

On the other hand, we believe that many things can be done with just one model view. Though the case of same-appearance-different-shape is possible in theory, this is a rare case in practice. It is excluded from consideration in computer vision from the standpoint of general viewpoint. Our hope is that our approach can at least reduce the possible matches in a large object database to a manageable number, sometimes to only one. In the case of multiple solutions, we can then use a second model view to remove the ambiguity. This is described later.

The problem of matching a model view with an input view is similar to that of matching uncalibrated stereo images. However, there are two major differences between them. First, since in general there are also other objects than the target object, there is inherently more ambiguity in the data. Secondly, while in object recognition the lighting conditions may change significantly, in stereo we can assume that the two images are taken at the same time, thus the lighting conditions do not change very drastically between them. Therefore, the problem of matching model and input views is generally more difficult.

Basically, we use the same procedure as that used for matching uncalibrated stereo images.

- Find corners, junctions and other high curvature points as feature points.
- Find match candidates between the two sets of feature points by correlation technique, while allowing identical image torsions.
- Form groups of neighboring 4 pairs of matches whose spatial relations are preserved.
- Estimate an epipolar equation for each such group, and find clusters in the motion parameter space.
- Match edge points using an estimated epipolar equation.

For details of each step, see Section 6.

Since there are feature points from other objects in the input image, there is a higher possibility that the true match is not found or not included in match candidates. And this possibility increases as the number of feature points from other objects increases. Since we use the local image pattern correlation to find match candidates, this possibility, of course, also depends on how the image patterns of target object and other objects resemble each other. If they look different, then the possibility does not increase. However, if they look similar, then the possibility increases.

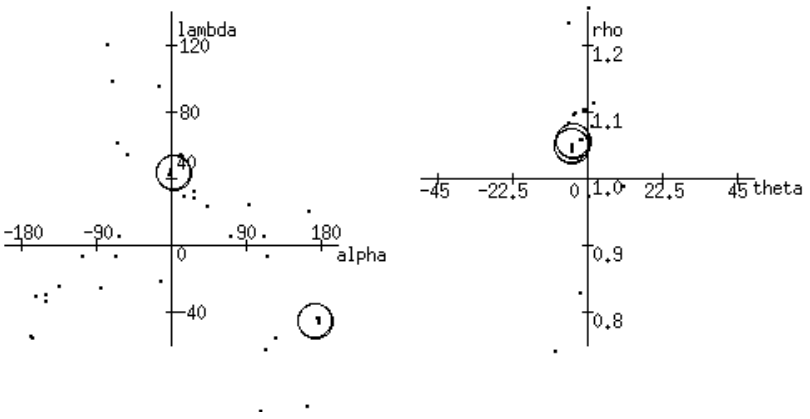
The above algorithm was tested with the example of an office scene. The Mac computer was placed in an office, and an input image (Figure 21) was taken with the background included. The two images used in "An Example of Matched Uncalibrated Stereo Images" are separately used here as model views to locate the Mac computer in the input image. The feature points in all these images are first extracted. Using the procedure described above, the feature points in Mac image 1 and in the input image are matched. The clusters found are shown in Figure 22, and the matched points are numbered and shown in Figure 23.

The edges in the input image, and the edge points corresponding to those in the model image found in the input image using the recovered epipolar equation, are shown in Figure 24.

The same things can be done between the second Mac image and the input image. The clusters found are shown in Figure 25, and the matched points are numbered and shown in Figure 26.

Figure 21. Input view with background.

Figure 22. Cluster found between the first model view and the input view of the Mac computer: $\alpha_1 = 3.400936$, $\alpha_2 = 173.576248$, $\theta_1 = -4.769201$, $\theta_2 = -5.203446$, $\rho_1 = 1.059555$, $\rho_2 = 1.051757$, $\lambda_1 = 44.560696$, and $\lambda_2 = -45.441090$.



Using the recovered epipolar equation, edge images can also be matched. The edge points corresponding to those in the second model image are found in the input image and shown in Figure 27, together with the original input edge image.

To see how the epipolar equation affects the matching of edge images, we matched the model edge image and the input edge image by the same matching algorithm using a wrong epipolar equation. The matched edge points in the input edge image are shown in Figure 28 together with the input edge image. As expected, the edge points are not smooth.

To compare with the result using the correct epipolar equation, we compute the energy defined in (50) for the matching result using correct epipolar equation and that using wrong epipolar equation.

Figure 23. The matched points are marked by numbers in model image 1 (left) and in the input image (right).

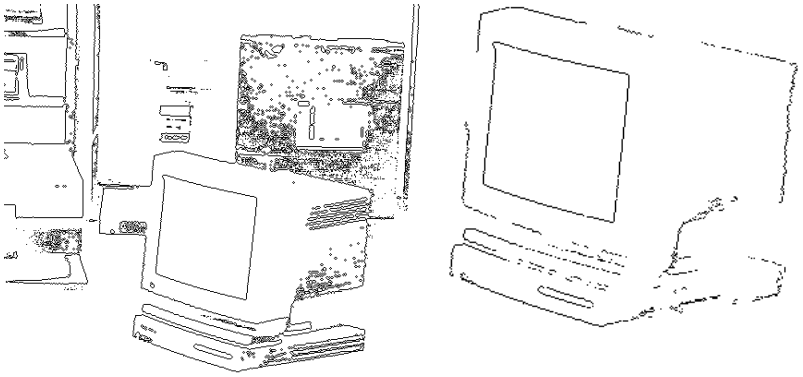


Figure 24. Edges in the input image (left). Edges in the input image matched with respect to model image 1 (right).

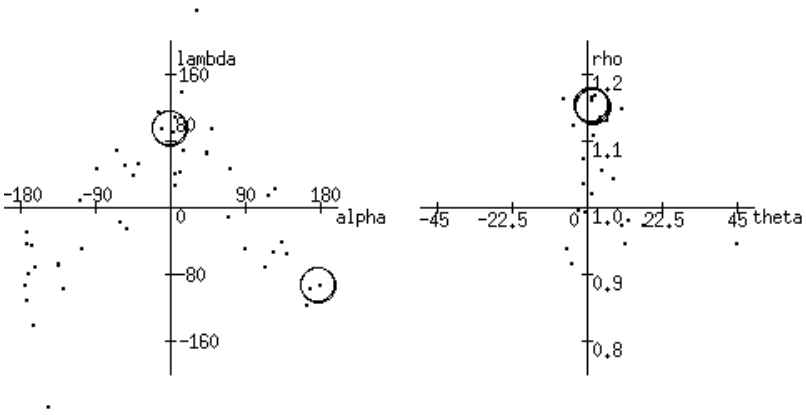


Figure 25. Clusters found between the second model view and the input view of the Mac computer: $\alpha 1 = -2.713166$, $\alpha 2 = 177.041748$, $\theta 1 = 1.943668$, $\theta 2 = 1.194336$, $\rho 11.154302 =$, $\rho 2 = 1.156942$, $\lambda 1 = 96.984818$, and $\lambda 2 = -93.706825$.

Recognition and Localization with Multiple Model Views

While recognizing a 3D object with one model view is inherently underconstrained, using two model views (which must be different) is theoretically sufficient. This conclusion can be obtained from the linear combination theorem by Ullman and Basri [25], and from trifocal tensor theory [18]. In the following, we first show that the intersection of epipolar lines is not an appropriate representation, then we derive the linear combination expression by representing the image coordinates as basis vectors, from which we can easily determine how to choose basis vectors.

Figure 26. The matched points are marked by numbers in model image 2 (left) and in the input image (right).

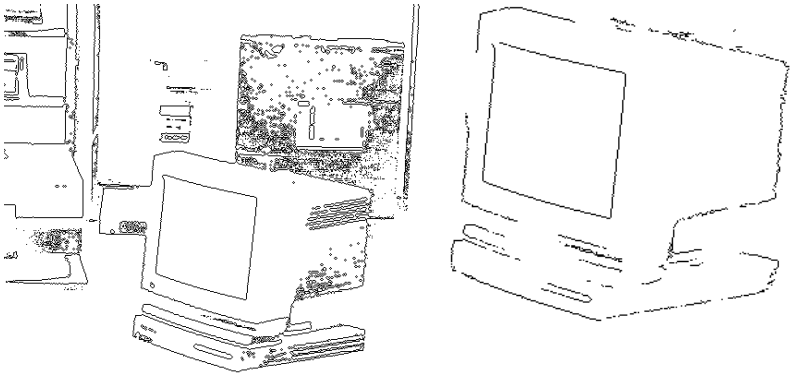


Figure 27. Edges in the input image (left). Edges in the input image matched with respect to model image 2 (right).

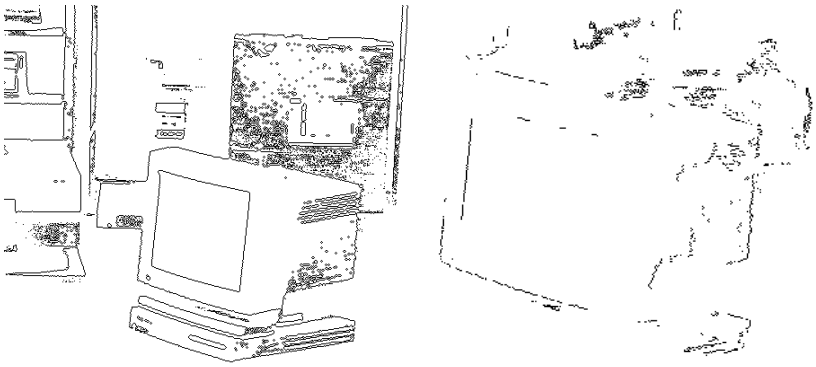


Figure 28. Edges in the input image (left). Edges in the input view matched with respect to model image 2 using a wrong epipolar equation (right).

Intuitively, with two model views, we can draw two epipolar lines in the input image, and the corresponding point should coincide with their intersection.

Suppose we have two model views. We denote points in the first and second model views by (u', v') and (u'', v''), respectively, and a point in the input view as (u, v). If the corresponding points in the three views have been identified, then we have three epipolar equations

$$P_1u + Q_1v + S_1u' + T_1v' + C_1 = 0 \tag{52}$$

$$P_2u + Q_2v + S_2u'' + T_2v'' + C_2 = 0 \tag{53}$$

$$P_3u' + Q_3v' + S_3u'' + T_3v'' + C_3 = 0 \tag{54}$$

only two of which are independent. To solve for the coefficients, we need at least 4 triples of points. Once the coefficients are determined, we can

express a point in the input view by its corresponding points in the two model views,

$$\begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} P_1 & Q_1 \\ P_2 & Q_2 \end{bmatrix}^{-1} \begin{bmatrix} S_1 u' + T_1 v' + C_1 \\ S_2 u'' + T_2 v'' + C_2 \end{bmatrix} \tag{55}$$

Visually, point (u, v) is the intersection of the two epipolar lines. It is thus natural, that if the two lines are parallel, or if the two equations (52) and (53) are not linear-independent, then increasing the number of model views does not solve the problem. Visually, this means that if the epipolar lines overlap each other, then there is an infinite number of solutions for (u, v).

There is another way of determining the coefficients of the linear combination, by using the concept of basis vectors. This does not require the epipolar lines to be parallel [29].

From the two model views of the Mac computer, we have tried to use the matched edge points in the model views to synthesize the Mac image so that it superimposes with that part in the input view.

Using the techniques described above, we can first match the feature points in the two model views and the input view, and then match the edge points using the recovered epipolar equations.

Let us denote a point in the input view by (u, v), a point in model view 1 by (u', v'), and a point in model view 2 by (u'', v''). Then the three recovered epipolar equations for $(u, v) - (u', v')$, for $(u, v) - (u'', v'')$ and for $(u', v') - (u'', v'')$ are respectively

$$0.059751u' - 0.677430v' - 0.107834u + 0.725183v - 38.169512 = 0$$

$$0.038198u'' + 0.673137v'' - 0.064981u - 0.735666v + 56.323223 = 0$$

$$0.012576u' + 0.712034v' + 0.054180u'' - 0.699938v'' - 11.399557 = 0$$

For the two model views, α and γ are $-1.011837°$ and $3.414448°$, respectively. If we choose 3 basis vectors, then $\mathbf{u}'$, $\mathbf{u}''$ and $\mathbf{v}''$ should be chosen such that the basis vectors are as separate as possible (see [29]).

From the epipolar equations, we can determine the equations for linear combination as

$$\begin{aligned} u &= 0.417222u' + 0.518943u'' - 0.013828v'' + 37.847492 \\ v &= -0.036853u' + 0.006085u'' + 0.916225v'' + 73.217812 \end{aligned} \tag{56}$$

Note that we can also use 4 basis vectors instead of 3.

Using (56), an edge image is synthesized and superimposed with the input view (Figure 29). The synthesized edges are shown in black, and the edges in the original input view in gray. It is evident that the synthesized data is very close to the data in the input view.

Conclusion

In this paper I have shown a unified approach to the correspondence and segmentation problems in stereo, motion, and object recognition by recovering the epipolar geometry underlying the images. Algorithms are proposed to solve the specific problems of recovering multiple epipolar geometries from the images, modeling the difference between the images as one-dimensional disparities in the Spatial Disparity Space, and matching the edge images by smoothness. The experimental results show that this new approach to motion and object recognition is simpler and more effective when compared with the conventional ones. There is

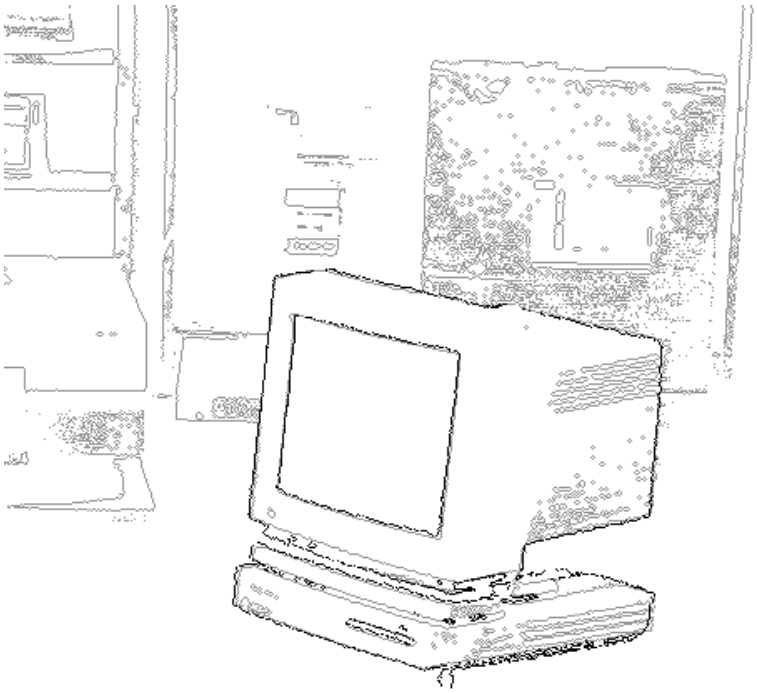


Figure 29. Synthesized edge image superimposed with the original input view.

a question of the performance of the epipolar geometry recovery algorithm when the number of feature points becomes large. Further study is being conducted on this.

Acknowledgments

I thank Saburo Tsuji for many useful discussions. This research was supported by research funds from Inamori Foundation, Nissan Science Foundation, and the Japanese Ministry of Education.

References

[1] R. Deriche & G. Giraudon. Accurate corner detection: An analytical study. In *Proc. Third Int'l Conf. Comput. Vision*, pages 66–70, Osaka, 1990.

[2] R. Deriche & G. Giraudon. A computational approach for corner and vertex detection. *Int'l J. Comput. Vision*, 10(2):101–124, 1993.

[3] O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.

[4] Olivier Faugeras & Giorgio Toscani. The calibration problem for stereo. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, IEEE Publication 86CH2290-5, pages 15–20, Miami Beach, FL, June 1986. IEEE.

[5] M.A. Fischler & O. Firschein. *Intelligence: The Eye, the Brain, and the Computer*. Addison-Wesley Publishing Company, 1987.

[6] W. Eric L. Grimson. *From Images to Surfaces*. The MIT Press, Cambridge, MA, 1981.

[7] E.C. Hildreth. Computations underlying the measurement of visual motion. *Artif. Intell.*, 23:309–354, 1984.

[8] B.K.P. Horn & B.G. Schunk. Determining optical flow. *Artif. Intell.*, 20:199–228, 1981.

[9] T.S. Huang & C.H. Lee. Motion and structure from orthographic projections. *IEEE Trans. PAMI*, 11:536–540, 1989.

[10] K. Kobayashi & G. Xu. An improved det operator. Tech rep, Department of Systems Enineering, Osaka University, Japan, 1993.

[11] D. Marr. *Vision*. W.H. Freeman, San Francisco, CA, 1982.

[12] D. Marr & T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[13] R. P. Paul, editor. *Robot Manipulators: Mathematics, Programming, and Control*. The MIT Press, 1981.

[14] T. Poggio & S. Edelman. A network that learns to recognize 3d objects. *Nature*, 343:263–266, 1990.

[15] L. Robert & O.D. Faugeras. Relative 3d positioning and 3d convex hull computation from a weakly calibrated stereo pair. In *Proceedings of the 4th Proc. International Conference on Computer Vision*, pages 540–544, Berlin, Germany, May 1993. IEEE Computer Society Press. also INRIA Technical Report 2349.

[16] R.J. Schalkoff. *Pattern Recognition: Statistical, Structural and New Approaches*. Wiley, 1992.

[17] L.S. Shapiro, A. Zisserman, & M. Brady. Motion from point matches using affine epipolar geometry. In *Proc. Third European Conf. Comput. Vision*, 1994.

[18] A. Shashua. Algebraic functions for recognition. *IEEE Trans. PAMI*, 17(8):779–789, August 1995.

[19] D. Terzopoulos. The computation of visible-surface representation. *IEEE Trans. PAMI*, 10(4):417–438, 1988.

[20] Philip Torr. *Motion Segmentation and Outlier Detection*. PhD thesis, Department of Engineering Science, University of Oxford, 1995.

[21] P.H.S. Torr & D.W. Murray. Outlier detection and motion segmentation. In S, editor, *Sensor Fusion VI, SPIE Vol. 2059*, pages 432–443, Boston, 1993.

[22] R. Tsai. Synopsis of recent progress on camera calibration for 3D machine vision. In Oussama Khatib, John J. Craig, and Tomás Lozano-Pérez, editors, *The Robotics Review*, pages 147–159. MIT Press, 1989.

[23] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.

[24] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, Cambridge, MA, 1979.

[25] S. Ullman & R. Basri. Recognition by linear combinations of models. *IEEE Trans. PAMI*, 13(10):992–1106, 1991.

[26] G. Xu. Unifying stereo, motion and object recognition via epipolar geometry. In *Proc. Second Asian Conf. Comput. Vision*, 1995. Invited paper.

[27] G. Xu & S. Tsuji. Is a single view sufficient for 3d object recognition? Tech rep, Dept. Control Eng, Osaka Univ, 1992.

[28] G. Xu & S. Tsuji. Correspondence and segmentation of multiple rigid motions via epipolar geometry. In *Proc. 13th Int'l Conf. Pattern Recog.*, pages A213–A217, 1996.

[29] G. Xu & Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*. Kluwer Academic Publishers, September, 1996.

[30] Y. Yang, A. Yuille, & J. Liu. Local, global, multilevel stereo matching. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.* IEEE, 1993.

[31] Z. Zhang. Motion of a stereo rig: Strong, weak and self calibration. In *Proc. Second Asian Conf. Comput. Vision*, pages I274–281, December 1995.

[32] Z. Zhang, R. Deriche, O. Faugeras, & Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, October 1995.