Image Matching under Affine Deformations

R. Manmatha *
Computer Science Department,
University of Massachusetts, Amherst, MA-01002.
manmatha@cs.umass.edu

Abstract

Image deformations due to relative motion between an observer and an object may be used to infer 3-D structure. Up to first order these deformations can be written in terms of an affine transform. Here, a novel approach is adopted to measuring affine transforms which correctly handles the problem of corresponding deformed patches. The patches are filtered using gaussians and derivatives of gaussians and the filters deformed according to the affine transform. The problem of finding the affine transform is therefore reduced to that of finding the appropriate deformed filter to use. The gaussian equations are solved both in the special case where the affine transform can be written as a similarity transform and for the general affine transform. The method is local and can handle arbitrarily large affine deformations.

Experiments demonstrate that this technique can find scale changes and optical flow in situations where other methods fail. The experiments also demonstrate the robustness of the technique.

1 Introduction

Changes in the relative orientation of a surface with respect to a camera cause deformations in the image of the surface. Deformations can be used to infer local surface geometry and depth from motion. Since a repeating texture pattern can be thought of as a pattern in motion, shape from texture can also be derived from deformations [6].

Up to first order, there are four kinds of image deformations - a scale change, rotation in the image plane and shear along the x and y axes. The position of the surface's projection in the image also changes with motion. To first order, this image translation together with the deformation can be described using a six parameter affine transformation (**t** , **A**) where

$$r' = t + Ar \tag{1}$$

r' and **r** are the image coordinates related by an affine transform, **t** is a 2 by 1 vector representing the translation and **A** the 2 by 2 affine deformation matrix.

For many applications, it suffices to use affine deformations for the following reason. The image projections of a small planar patch from different viewpoints are well approximated by affine transforms. This result is exactly true under weak perspective projection and a good approximation for perspective projection when the motion is small.

Therefore, the measurement of affine transforms is useful in many applications in vision including registration of images from a moving camera [1], structure from motion and stereo [4] shape from texture, object recognition, texture segmentation of real surfaces (as opposed to Mondrians). Specializations of the affine transform like the similarity transform and optical flow have their own well recognized applications.

The difficulty of measuring the affine transform is illustrated by Figure (1). The image on the right is scaled 1.4 times the image on the left. Accurately measuring the affine transform between the two images first requires that the image patches be correctly matched. Even if the centroids are matched this is difficult to do since the sizes of every portion of the two images differ. This problem arises because traditional matching uses fixed correlation windows or filters. Thus, for example Washington's face in the two pictures cannot be matched by a traditional scheme. The correct way to approach this problem is to deform the correlation window or filter according to the image deformation. For a general affine transform the deformations are even more extreme.

This paper derives a computational scheme where gaussian and derivative of gaussian filters are used and the filters deformed according to the affine transformation. The resulting equations are solved by linearizing with respect to the affine parameters rather than the image coordinates. This allows the linearization point to be moved so that arbitrary affine transforms can be solved unlike traditional methods restricted to small affines. The method is local, applicable to arbitrary dimensions and can measure affine transforms in situations where other algorithms fail. For example, Werkhoven and Koenderink's algorithm [8] when run on the images in Figure (1) returns a scale factor of 1.16 while our algorithm does the matching correctly and therefore returns a scale factor of 1.41.

1.1 Previous Work

Affine transforms between image patches have been recovered in three different ways (for more details see [6]):

---
*Currently visiting IBM Almaden Research Center, San Jose, CA-95120.

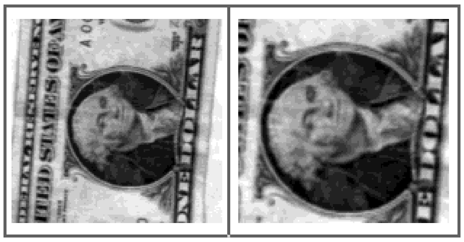

Figure 1: Dollar Bill scaled 1.4 times

1. By linearizing the image intensities or the filter outputs *with respect to the image coordinates.* Linearization limits these algorithms to cases when the affine transforms are small [8, 1, 2].

2. Matching image intensities by searching through the space of all affine parameters. This approach adopts a brute force search strategy which is potentially slow [4].

3. Line based methods which match the closed boundaries of corresponding regions [3, 7].

Of these approaches, only the third deals with the problem of corresponding deformed patches. However, it is limited to homogenous regions with closed boundaries.

Patches deformed under similarity transforms may also be matched using the Mellin-Fourier Transform ([5]). Although possible, recovery of the similarity transform has not been demonstrated. The main drawback to these techniques is that they are inherently global and they are not applicable to general affine transforms.

2 Deformation of Filters

The following discussion assumes that the image translation is zero (equivalently known translation). A method for recovering the image translation will be given in section 3.1.2. It is also assumed that shading and illumination effects can be ignored. These can, however, be taken care of multiplying the equations by an additional constant factor. Most of what follows is valid for arbitrary dimensions; the 2-D case is used as a specific example.

Notation Vectors will be represented by lowercase letters in boldface while matrices will be represented by uppercase letters in boldface.

Consider two Riemann-integrable functions F_1 and F_2 related by an affine transform i.e.

$$F_1(r) = F_2(Ar) \tag{2}$$

Define a generalized gaussian as

$$G(r, M) = \frac{1}{(2\pi)^{n/2} det(M)^{1/2}} exp(-\frac{r^T M^{-1} r}{2}) \tag{3}$$

where $\mathbf{M}$ is a symmetric positive semi-definite matrix. Then it may be shown that the output of F_1 filtered with a gaussian is equal to the output of F_2 filtered with a gaussian deformed by the affine transform (see [6] for details) i.e.

$$\int F_1(r) G(r, \sigma^2 I) dr = \int F_2(Ar) G(Ar, R\Sigma R^T) d(Ar) \tag{4}$$

where the integrals are taken from $-\infty$ to ∞. $\mathbf{R}$ is a rotation matrix and Σ a diagonal matrix with entries $(s_1\sigma)^2, (s_2\sigma)^2 ... (s_n\sigma)^2$ $(s_i \geq 0)$ and $R\Sigma R^T = \sigma^2 AA^T$ (this follows from the fact that AA^T is a symmetric, positive semi-definite matrix).

Intuitively, equation (6) expresses the notion that the gaussian weighted average brightnesses must be equal, provided the gaussian is affine-transformed in the same manner as the function. The problem of recovering the affine parameters has been reduced to finding the deformation of a known function, the gaussian, rather than the unknown brightness functions. The equation is exact and is valid for arbitrary dimensions.

The level contours of the generalized gaussian are ellipsoids rather than spheres. The tilt of the ellipsoid is given by the rotation matrix while its eccentricity is given by the matrix Σ, which is actually a function of the scales along each dimension. The equation clearly shows that to recover affine transforms by filtering, one must deform the filter appropriately; a point ignored in previous work [1, 2, 8, 4]. The equation is local because the gaussians rapidly decay.

The integral may be interpreted as the result of convolving the function with a gaussian at the origin. It may also be interpreted as the result of a filtering operation with a gaussian. To emphasize these similarities, it may be written as

$$F_1 * G(r, \sigma^2 I) = F_2 * G(r_1, R\Sigma R^T) \tag{5}$$

where $r_1 = Ar$.

In the special case where the affine transform can be written as $A = sR$ i.e. a scale and a rotation, the above equation reduces to,

$$F_1 * G(r, \sigma^2) = F_2 * G(r_1, (s\sigma)^2) \tag{6}$$

Note that this equation is valid for an arbitrary rotation..

Similar equations may be written using derivative of gaussian filters (for details see [6]).

3 Solving the Gaussian Filtering Equations

It is instructive to first solve for the case where the affine transform $\mathbf{A}$ = s $\mathbf{R}$ (i.e. a scale change and a rotation) The general affine transform will be considered in a later section.

3.1 The Case of Similarity Transforms

To solve equation(6), for example, requires finding a gaussian of the appropriate scale $s\sigma$ given σ. A brute force search through the space of scale changes is not desirable. Instead a more elegant solution is to linearize the gaussians with respect to σ. This gives an equation linear in the unknown α

$$\begin{aligned} F_1 * G(., (s\sigma)^2) &\approx F_2 * G(., \sigma^2) + \alpha\sigma F_2 * \frac{\partial G(., \sigma^2)}{\partial \sigma} \\ &= F_2 * G(., \sigma^2) + \alpha\sigma^2 \nabla^2 F_2 * G(., \sigma^2) \end{aligned} \tag{7}$$

where $s = 1 + \alpha$. The last equality follows from the diffusion equation $\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G$. The above equation is linear in s. To find s, three filtering operations need to be performed, two gaussian filtering operations and one laplacian operation. The key notion here is that the *linearization is done with respect to σ and not the image coordinates.*

Equation (7) is not very stable if solved at a single scale. By using gaussians of several different scales σ_i the following overconstrained linear system is obtained:

$$\begin{aligned} F_1 * G(., \sigma_0^2) &\approx F_2 * G(., \sigma_0^2) + \alpha\sigma_0^2 F_2 * \nabla^2 G(., \sigma_0^2) \\ \ldots\ldots &\approx \ldots.. \\ F_1 * G(., \sigma_i^2) &\approx F_2 * G(., \sigma_i^2) + \alpha\sigma_i^2 F_2 * \nabla^2 G(., \sigma_i^2) \\ \ldots\ldots &\approx \ldots.. \\ F_1 * G(., \sigma_l^2) &\approx F_2 * G(., \sigma_l^2) + \alpha\sigma_l^2 F_2 * \nabla^2 G(., \sigma_l^2) \end{aligned} \tag{8}$$

The above system of equations maybe cast into the following linear least squares problem

$$\Sigma_i ||F_1 * G(., \sigma_i^2) - F_2 * G(., \sigma_i^2) + \alpha\sigma_i^2 F_2 * \nabla^2 G(., \sigma_i^2)||^2 \tag{9}$$

and solved using Singular Value Decomposition (SVD).

In practical implementations the following set of 5 different σ_i (1.25,1.7677,2.5,3.5355,5.0) was found to work well. Note that these are all spaced apart by half an octave (a factor of 1.4). The corresponding filter widths were approximately 8 * σ_i (3,5,7,11,15,21,29,41)

3.1.1 Choosing a Different Operating Point

For large scale changes (say scale change ≥ 1.2) the recovered scale tends to be poor. This is because the Taylor series approximation is good only for small values of α. The advantage of linearizing the gaussian equations with respect to σ is that the linearization point can be shifted (i.e. the right-hand side of (6) can be linearized with respect to a σ different from the one on the left-hand side). Let the right-hand side of (7) be linearized around σ_j to give the following equation

$$F_1 * G(., \sigma_i^2) \approx F_2 * G(., \sigma_j^2) + \alpha' \sigma_j^2 F_2 * \nabla^2 G(., \sigma_j^2) \tag{10}$$

where $s = \sigma_j/\sigma_i(1 + \alpha')$. For example, for s = 1.3, $\sigma_j = 1.4$ implies $\alpha' = -0.07$ which is much smaller than $\alpha = s - 1 = 0.3$. Equation (10) is therefore a better approximation than equation (7). The strategy therefore is to pick different values of σ_j and solve (10) (or actually an overconstrained version of it). Each of these σ_j will result in a value of α'. The correct value of α' is that which is most consistent with the equations. For example, let the correct scale be 1.3, and assume that the σ_j values chosen are $(\sigma, 1.4\sigma, 2.0\sigma)$. Let the values of s recovered using these σ_j be (1.25,1.32,1.5) respectively. Then 1.32 is consistent with expanding around 1.4σ in the sense of being closest to its expansion point. By choosing the σ_j appropriately, it can be ensured that no new convolutions are required.

In principle, arbitrary scale changes can be recovered using this technique. In practice, most scale changes in motion and texture are ≤ 2.5 and therefore three operating points $(\sigma, 1.4\sigma, 2.0\sigma)$ should suffice. The ability to measure arbitrary scale changes follows from the linearization with respect to σ. Other methods linearize the function F or the gaussian with respect to r and are therefore constrained to measuring small affine transforms.

3.1.2 Finding Image Translation

Before scale can be recovered, the two patches must be aligned by finding the image translation. Essentially this amounts to finding the optical flow. Assume F_1 is shifted with respect to F_2. Then under a similarity transform,

$$F_1(r + t_0) * G(r, \sigma) = F_2(sRr) * G(sRr, s\sigma) \tag{11}$$

Assume now that an estimate t_0 is available. Then the above equation may now be rewritten as

$$F_1(r + t_0) * G(r - \delta t, \sigma) = F_2(sRr) * G(sRr, s\sigma) \tag{12}$$

The gaussian on the left hand side may now be linearized with respect to $\mathbf{r}$ to give,

$$\begin{aligned} F_1(r+t_0) * G(r, \sigma) - \delta t^T F_1(r + t_0) * G(r, \sigma) \approx \\ F_2(sRr) * G(sRr, s\sigma) \end{aligned} \tag{13}$$

which is again linear in the residual translation δt. The scale parameter can also be linearized as before to give

$$\begin{aligned} F_1(r+t_0) * G(r, \sigma) - \delta t^T F_1(r + t_0) * G(r, \sigma) \approx \\ F_2 * G(., \sigma) + \alpha\sigma^2 F_2 * \nabla^2 G(., \sigma) \end{aligned} \tag{14}$$

which is again linear in both the scale and the unknown translation. As before an overconstrained version of this equation using multiple scales is obtained and solved for the unknown parameters. Large scales are handled as before.

δt is a good estimate of the residual translation only if it is small compared to the σ of the smallest gaussian

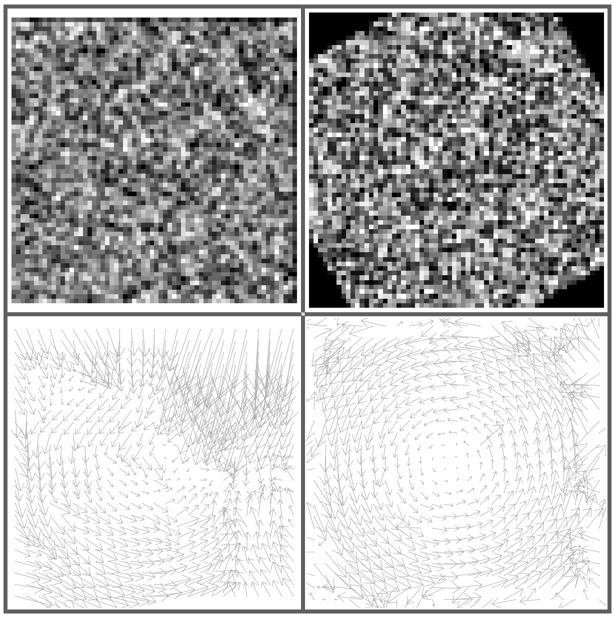

Figure 2: Random Dot Sequence

used. This can be ensured either by using a pyramid type alogrithm or the following strategy. For various values of t_0, the above equation is solved. First, those values of t_0 are marked for which δt is sub-pixel. Amongst these let $t_0 = b_0$ be the value such that the corresponding system of equations (14) has the smallest SVD residual. Then the actual translation is $b_0 + \delta t$. The basic idea is that the translation is localized by doing a local search (similar to a correlation scheme) and then the sub-pixel value recovered using the equations.

The main advantage of the scheme is that scale and translation are recovered simultaneously. Further, since the gaussians are rotation invariant, the translation can be recoverd for arbitrary rotations about an axis perpendicular to the image. No other scheme is able to do this.

The main disadvantage of this scheme is that the local search is expensive. This can be eliminated by using for example a pyramid scheme so that the residual translation at any level is always less than one pixel. The estimate t_0 can then be obtained from the coarser level.

3.2 Experimental Results

Experiments on synthetic images show that the affine transform can be recovered to within a few percent (see [6]).

Figure (2) illustrates the power of this algorithm. A random dot image is scaled by a factor of 1.1 and rotated around an axis perpendicular to the image by 30 deg. On the left is the flow produced by an SSD based pyramid scheme. Note that the algorithm fails quite dramatically because of the large rotation. This occurs because for correct matching the template also needs to be rotated by the same angle. For small angles, the template rotation can be ignored but this cannot be done for large rotations. On the other hand the results of running the algorithm described here are shown on the right-hand side. The flow shown is clearly rotational. *Note that the flow has been computed at every point without fitting a global model.* To the best of our knowledge no other existing algorithm can compute the flow correctly in this situation A histogram of the of the recovered scale values peaks at 1.1 which is the correct value.

Fig (1) shows a dollar bill scaled by 1.4. The algorithm correctly recovers the scale as 1.41. Other experiments with scaled and rotated versions of the dollar bill consistently show good recovery of scale within a few percent.

A pair of real images was taken by a camera mounted on a robot arm. The arm was translated towards a wooden block. On the top left in Figure (3) is a portion of the wooden block on which the algorithm was run. The experimental results are also shown in the same figure. The recovered flow clearly radiates from the focus of expansion as it should. A surface plot of the scale again shows a planar surface and a histogram of the recovered scale values peaks at 0.945. The correct value of the scale = 0.95. Note that there are no features on this block which a point or line algorithm could use.

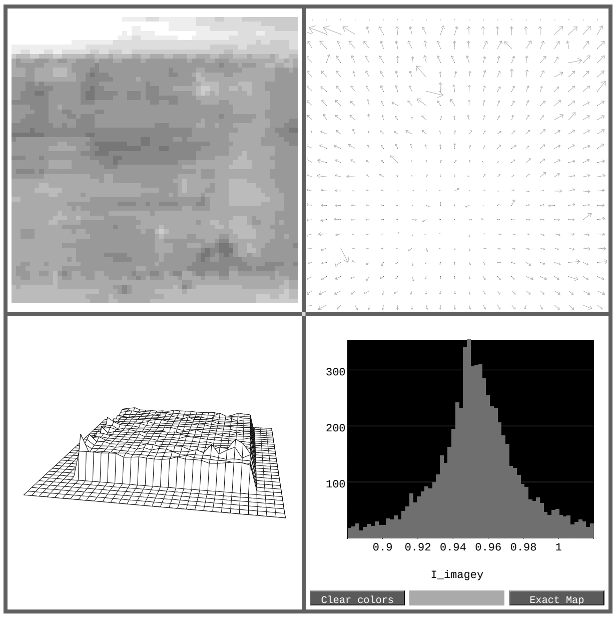

Figure 3: Flow, surface plot of scale and histogram of scale on wood block

4 Solving for the General Affine

The general affine can also be solved by linearization. In this case, the 3 parameters of an elliptical gaussian, the orientation and the scale changes along the major and minor axes have to be solved. The strategy adopted will be to first sample the space of scales and orientations to derive a finite set of filters. The gaussian equation is then linearized with respect to the scales and orientation about the elliptical filter closest to the right orientation and scales.

Recall that the gaussian weighted brightnesses are equal if

$$F_1 * G(r, \sigma^2 I) = F_2 * G(r_1, R(\theta+\phi)\Sigma' R^T(\theta+\phi)) \quad (15)$$

where F_2 is filtered with an elliptical gaussian of orientation $\theta + \phi$ and standard deviations $s_1\sigma_1$ and $s_2\sigma_2$. Linearizing the gaussian on the right with respect to σ_1, σ_2 and θ gives,

$$\begin{aligned} F_1 * G(r, \sigma^2 I) &= F_2 * G(r_1, R(\theta)\Sigma R^T(\theta)) \\ &+ (s_1 - 1){\sigma_1}^2 F_2 * G_{x'x'}(., R(\theta)\Sigma R^T(\theta)) \\ &+ (s_2 - 1){\sigma_2}^2 F_2 * G_{y'y'}(., R(\theta)\Sigma R^T(\theta)) \\ &+ \phi[{\sigma_2}^2 - {\sigma_1}^2] F_2 * G_{x'y'}(., R(\theta)\Sigma R^T(\theta)) \quad (16) \end{aligned}$$

where $G(., R(\theta)\Sigma R^T(\theta))$ is a member of the sample set with sigma's σ_1, σ_2 and $(x', y')^T = R(\theta)(x, y)^T$ i.e. (x',y') are the coordinate axes defined by the major and minor axes of the sample ellipse. Since θ is known, computing (x',y') and hence $G_{x'x'}$, $G_{x'y'}$ and $G_{y'y'}$ is straightforward. This is a good approximation if $(s_1 - 1)$, $(s_2 - 1)$ and ϕ are small.

4.1 Case σ_1/σ_2 is small ($s_1\sigma_1/s_2\sigma_2 < 1.2$).

In this case, elliptical gaussians are not required. Only circular gaussians are used. This is done by setting $\sigma_1 = \sigma_2 = \sigma$. Equation (16) therefore becomes

$$\begin{aligned} F_1 * G(., \sigma) = F_2 * G(., \sigma) + (s_1 - 1)\sigma^2 F_2 * G_{x'x'}(., \sigma) \\ + (s_2 - 1)\sigma^2 F_2 * G_{y'y'}(., \sigma) \quad (17) \end{aligned}$$

Since the gaussians are circular, x', y' can no longer be distinguished and must therefore be solved for. Expanding x',y' in terms of x,y and θ gives

$$\begin{aligned} & F_1 * G(., \sigma) = F_2 * G(., \sigma) \\ & + (s_1 - 1)\sigma^2 F_2 * \\ & \quad [G_{xx}(., \sigma)cos^2\theta + G_{yy}(., \sigma)sin^2\theta - G_{xy}(., \sigma)sin2\theta] \\ & + (s_2 - 1)\sigma^2 F_2 * \\ & \quad [G_{xx}(., \sigma)sin^2\theta + G_{yy}(., \sigma)cos^2\theta + G_{xy}(., \sigma)cos2\theta] \end{aligned}$$

The equation is now slightly non-linear (the unknowns being s_1, s_2 and θ). An overconstrained system is obtained as before by using different values of σ. This system may then be solved by sampling the space of θ's (when it becomes linear in the unknowns $(s_1 - 1), (s_2 - 1)$) or by using the Gauss-Newton technique.

Note that this case is much more general than the small affine case used in the literature. The angle θ may be arbitrary. Further, $s_1\sigma_1$ and $s_2\sigma_2$ may be individually large provided the ratio $s_1\sigma_1/s_2\sigma_2$ is small (When $s_1\sigma_1, s_2\sigma_2$ are large the operating point is shifted in the same manner as discussed for similarity transforms).

For the case $\theta = 0$, tests on synthetic images gave satisfactory results.

Due to lack of space the other case is not discussed here.

4.2 Computing the matrix A

Now $AA^T = R\Sigma R^T$ is the SVD of AA^T. Also, note that $(A^T A)^{-1}$ can be recovered by interchanging the roles of F_1 and F_2 in equation (5). i.e. instead of convolving F_1 with a circular gaussian and F_2 with an elliptical gaussian, F_2 is convolved with a circular gaussian and F_1 with an elliptical gaussian. This gives:

$$F_2 * G(., \sigma^2 I) = F_1 * G(., R_2^T \Sigma^{-1} R_2) \quad (18)$$

where $(A^T A)^{-1} = {R_2}^T \Sigma^{-1} R_2$ is the SVD of $(A^T A)^{-1}$. Therefore $A = R\Sigma^{1/2} R_2$ (again using SVD). All the quantities on the right can be measured and hence **A** can be computed.

Acknowledgements I wish to thank Harpreet Sawhney, John Oliensis and Al Hanson for comments and fruitful discussions.

References

[1] J.R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. 2nd European Conference on Computer Vision*, pages 237–252, 1992.

[2] M. Campani and A. Verri. Motion analysis from optical flow. *Computer Vision Graphics and Image Processing:Image Understanding*, 56(12):90–107, 1992.

[3] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In *Proc. 2nd European Conference on Computer Vision*, pages 187–202, 1992.

[4] D. G. Jones and J. Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Proc. 2nd European Conference on Computer Vision*, pages 395–410, 1992.

[5] J. Rubinstein J.Segman and Y.Y. Zeevi. The canonical coordinates method for pattern deformation: Theoretical and computational considerations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1171–1183, 1992.

[6] R. Manmatha and J. Oliensis. Measuring the Affine Transform – I: Scale and Rotation. Technical Report CMPSCI TR 92–74,University of Massachusetts at Amherst,MA,1992. Also in Proc. of the Darpa Image Understanding Workshop 1993.

[7] H. S. Sawhney and A. R. Hanson. Identification and 3D description of 'shallow' environmental structure in a sequence of images. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 179–186, 1991.

[8] P. Werkhoven and J. J. Koenderinck. Extraction of motion parallax structure in the visual system 1. *Biological Cybernetics*, 1990.