Measuring the Affine Transform Using Gaussian Filters

R. Manmatha *

Computer Science Department, University of Massachusetts, Amherst, MA-01003.
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. The problem of finding the affine transform is reduced to that of finding the appropriate deformed filter to use. 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.

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 [5].

To first order, this deformation together with the image translation 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. The affine transform is useful because the image projections of a small planar patch from different viewpoints are well approximated by it [5].

In Figure (1) the image on the right is scaled 1.4 times the image on the left. Even if the centroids of the two image patches are matched accurately, measuring the affine transform is difficult since the sizes of every portion of the two images differ. This problem arises because traditional matching uses fixed correlation windows or filters. The correct way to approach this problem is to deform the correlation window or filter according to the image deformation.

* This research was supported by NSF grants CDA-8922572 and IRI-9113690. The author also thanks IBM Almaden Research Center which hosted him for 6 months.

Lecture Notes in Computer Science, Vol. 801
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94

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 [6] 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. For a review of related work see [5].

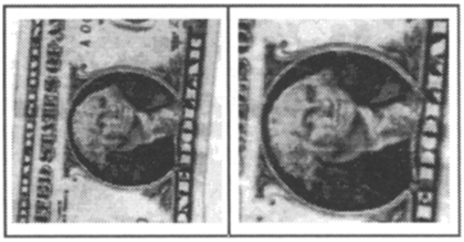

Fig. 1. Dollar Bill scaled 1.4 times

2 Deformation of Filters

The initial discussion will assume zero image translation; translation can be recovered as suggested in section 3. It is also assumed that shading and illumination effects can be ignored.

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 **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 [5] 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 ∞. **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, (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 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, 6, 3]. 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 and will 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 case of similarity transforms, $A = sR$ i.e. a scale change and a rotation, this 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 [5]).

3 Solution for the Case of Similarity Transforms

To solve (6) 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

$$F_1 * G(., (s\sigma)^2) \approx F_2 * G(., \sigma^2) + \alpha\sigma^2\nabla^2 F_2 * G(., \sigma^2) \tag{7}$$

where $s = 1+\alpha$. 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 linear least squares problem is obtained:

$$\Sigma_i ||F_1 * G(., \sigma_i^2) - F_2 * G(., \sigma_i^2) + \alpha\sigma_i^2 F_2 * \nabla^2 G(., \sigma_i^2)||^2 \tag{8}$$

and solved using Singular Value Decomposition (SVD).

The following σ_i (1.25,1.7677,2.5,3.5355,5.0) - spaced apart by half an octave - were found to work well. The corresponding filter widths were approximately $8 * \sigma_i$ (3,5,7,11,15,21,29,41)

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 (other methods linearize the function F or the gaussian with respect to r and are therefore constrained to measuring small affine transforms). Let the right-hand side of (7) be linearized around σ_j to give the following equation

$$F_1 * G(., \sigma_i^2) \approx F_2 * G(., \sigma_j^2) + \alpha' \sigma_j^2 F_2 * \nabla^2 G(., \sigma_j^2) \tag{9}$$

where $s = \sigma_j / \sigma_i (1 + \alpha')$. The strategy therefore is to pick different values of σ_j and solve (9) (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. 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.

Finding Image Translation: Image translation, i.e. optic flow can be recovered in the following manner. Let F_1 and F_2 be similarity transformed versions of each other (i.e. they differ by a scale change, a rotation and a translation). Assume that an estimate of the translation t_0 is available. Linearizing with respect to r and σ gives

$$F_1(r+t_0) * G(r, \sigma^2) - \delta t^T F_1(r+t_0) * G(r, \sigma^2) \approx F_2 * G(., \sigma^2) + \alpha \sigma^2 F_2 * \nabla^2 G(., \sigma^2) \tag{10}$$

which is again linear in both the scale and the residual translation δt. 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.

t_0 is obtained either by a local search or from a coarser level in a pyramid scheme, while δt is estimated from the equation (see [4] for details).

Note that since the gaussians are rotation invariant, the translation can be recovered for arbitrary rotations about an axis perpendicular to the image. No other scheme is able to do this.

3.1 Experimental Results

Experiments on synthetic images show that the affine transform can be recovered to within a few percent (see [5]).

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.

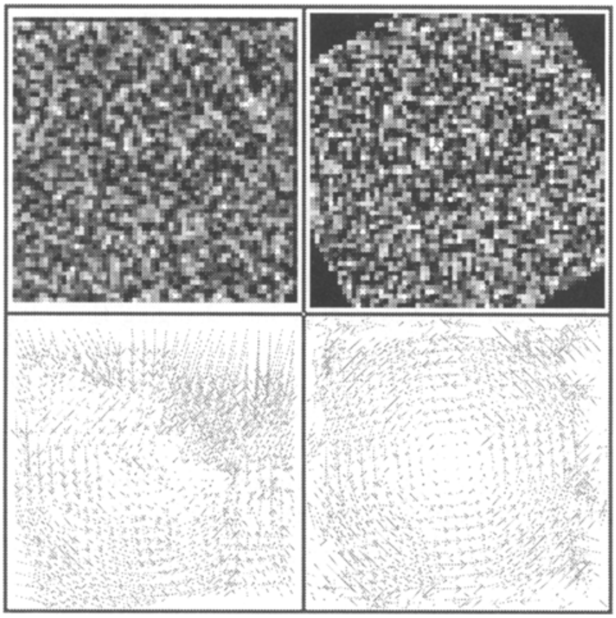

Fig. 2. Random Dot Sequence

Figure (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.

For other examples see [4].

4 Solving for the General Affine

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

$$F_1 * G(r, \sigma^2 I) = F_2 * G(r_1, R(\theta+\phi)\Sigma' R^T(\theta+\phi)) \tag{11}$$

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,

$$\begin{aligned} F_1 * G(r, \sigma^2 I) = & F_2 * G(r_1, R\Sigma R^T) + (s_1 - 1){\sigma_1}^2 F_2 * G_{x'x'}(., R\Sigma R^T) \\ & + (s_2 - 1){\sigma_2}^2 F_2 * G_{y'y'}(., R\Sigma R^T) \\ & + \phi[1/{\sigma_2}^2 - 1/{\sigma_1}^2] F_2 * G_{x'y'}(., R\Sigma R^T) \end{aligned} \tag{12}$$

where $G(., R\Sigma R^T)$ is a member of the sample set with sigma's σ_1, σ_2, $R = R(\theta)$ 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.

In the case where σ_1/σ_2 this approximation may be rewritten so that elliptical gaussians are not needed and circular gaussians suffice.

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 (5), where $(A^T A)^{-1} = {R_2}^T \Sigma^{-1} R_2$ is the SVDto solve 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. 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.
4. R. Manmatha. Image matching under affine deformations. In *Invited Paper, Proc. of the 27nd Asilomar IEEE Conf. on Signals, Systems and Computers,*, 1993.
5. R. Manmatha and J. Oliensis. Measuring the affine transform – i: Scale and rotation. Technical Report Technical Report CMPSCI TR 92–74,University of Massachusetts at Amherst,MA,1992. Also in Proc. of the Darpa Image Understanding Workshop 1993.
6. P. Werkhoven and J. J. Koenderinck. Extraction of motion parallax structure in the visual system 1. *Biological Cybernetics*, 1990.