INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONE: 2240-1
CABLE: CENTRATOM - TELEX 460392 - I

SMR/302-12

COLLEGE ON NEUROPHYSICS:
"DEVELOPMENT AND ORGANIZATION OF THE BRAIN"
7 November - 2 December 1988

"Parallel Optical Flow Using Local Voting"

Heinrich H. BULTHOFF
Center for Biological Information Processing
M.I.T.
Cambridge, MA
USA

ICCV 2-1988

Parallel Optical Flow Using Local Voting

James J. Little
Artificial Intelligence Laboratory
Massachusetts Institute of Technology

Heinrich H. Bülthoff
Center for Biological Information Processing
Massachusetts Institute of Technology

Tomaso Poggio
Artificial Intelligence Laboratory
Massachusetts Institute of Technology

Abstract

We describe a parallel algorithm for computing optical flow from short-range motion. Regularising optical flow computation leads to a formulation which minimises matching error and, at the same time, maximises smoothness of the optical flow. We develop an approximation to the full regularization computation in which corresponding points are found by comparing local patches of the images. Selection among competing matches is performed using a winner-take-all scheme. The algorithm accommodates many different image transformations uniformly, with similar results, from brightness to edges. The optical flow computed from different image transformations, such as edge detection and direct brightness computation, can be simply combined. The algorithm is easily implemented using local operations on a fine-grained computer, and has been implemented on a Connection Machine. Experiments with natural images show that the scheme is effective and robust against noise. The algorithm leads to dense optical flow fields; in addition, information from matching facilitates segmentation.

1 Introduction

Optical flow is generated on the retina of an observer by objects moving relative to the observer. The changing patterns of image brightness on the retina define a vector field, the optical flow. In restricted cases, the motion of scene objects can be deduced directly from the optical flow. Rich qualitative information on the motion of objects and their boundaries is contained in the critical points and discontinuities of the optical flow.

The velocity field of object moving space is a 3-D vector field $W(x, y, z)$. When projected into the retinal (image) coordinate system of an observer, this field generates the projected velocity field of object $W_p(x, y)$. What is observed on the retina is the image brightness change $E_t(x, y)$, $E_{t+\Delta t}(x, y)$. The optical flow, $V(x, y)$, is a time-varying vector field describing image brightness change. In general, V and $W_p(x, y)$ are not the same.

In the Vision Machine project [23], our goal is to devise robust methods for computing early vision modules and to integrate these modules. The flexible, robust behavior of the human visual system is in large part due to integration of many early vision modules [4, 22]. The output of this integration stage is a map of the physical discontinuities in the scene, in the image coordinate system. The optical flow and its discontinuities are important inputs to the integration stage and can provide cues to figure-ground separation, as has been demonstrated in the visual system of the fly[25].

1.1 Assumptions

In our approach to computing the optical flow, we make several assumptions about the imaging conditions. First, the time Δt between images is small, on the order of one video time frame (1/30th second). Differential approaches to motion vision rely on the spatial and temporal derivatives of images to compute optical flow [10, 7]. When the velocity, in the image plane, of image elements is small enough so that the Taylor series expansion holds, temporal derivatives are meaningful. The time derivative at a point depends on the projected velocity and the spatial variation in surface luminance. This is the case of *instantaneous* motion; correspondence between elements in the two images is not a problem. In the situations we describe, surfaces are moving too fast for instantaneous temporal derivatives to be meaningful. Our algorithm assumes that image displacements are small with respect to the image size, within a range $(\pm\delta, \pm\delta)$, but that δ can be larger than 1 or 2 pixels. Unlike long-range motion schemes, as in structure-from-motion [28], the distances are relatively small compared to the size of the objects in the scene. We term the restricted motion situation *short-range* motion. Since correspondence of image elements must be determined, motion computation under these conditions resembles binocular stereo.

Between frames the appearance of a moving object can change due to its own motion, camera motion, light source motion, or all three, among other effects [30]. However, when the local brightness variation in the surface albedo is sufficiently large, the errors introduced by these effects are relatively small.

1.2 Constraints on Motion

We use the following constraints to identify the correct optical flow: *uniqueness*, each image point has a unique velocity, and *continuity*, surfaces are locally smooth.

Physical constraints on motion limit the spatial variation of the optical flow field. First, the projected velocity field of a planar patch under arbitrary, rigid 3-D motion is quadratic in (x, y). A constraint that is true under more restrictive conditions is: the projected velocity field of a planar patch, translating parallel to the image plane, is constant. This is true in limited cases but it is often a satisfactory *local* approximation (see [10, 12, 16]). Our algorithm uses the local flow constancy assumption; our experiments with support determined by linear variation have shown relatively little improvement, and only at discontinuities.

2 Regularization Formulation

2.1 Aperture Problem

When a contour is locally a straight line, over an extent large relative to the feature size, the range of possible match points is a one-dimensional set of points lying on the contour. In the formulation of Marr and Ullman [18], only the velocity normal to the straight contour can be recovered; they termed this the *Aperture Problem* because a moving edge, seen through a circular aperture seems to be moving normal to itself, while the transverse component of the velocity is not perceived.

In the optical flow algorithm of Horn and Schunk [10], at each point, there is one equation in two unknowns:

$$u\,dE/dx + v\,dE/dy + dE/dt = 0 \qquad (1)$$

Here, the optical flow computation is always locally underconstrained; an additional constraint must be provided by choosing the smoothest optical flow subject to the single data constraint at each point.

When the image within the aperture is not simply a portion of a line, i.e., when the matrix of second partial derivatives of the image, the image Hessian (E_{ij}), is not zero, then the velocity can be correctly identified [26]. For example, if the aperture contains a corner, the constraints for each edge form a line in velocity space, and the correct velocity lies on their intersection[19]. The aperture problem is an instance of the correspondence problem. When the window used in matching is large, i.e., the features themselves have large spatial extent, then there will likely be some variation in orientation of the included contour and the aperture problem will not occur.

In order to make the optical flow computation well-posed, we regularise[2] the solution, adding constraint to the computation, for

This report describes research done within the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. Support for the A.I. Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010, in part by DARPA under Office of Naval Research contract N00014-85-K-0214, and in part by a grant from Hughes Aircraft Company.

example, by choosing the smoothest optical flow field fitting the data. This leads to formulations of optical flow which apply the regularising constraint to compute the smoothest velocity field which matches the data, both area[10] and contour based[8].

2.2 Regularization for Short-Range Motion

When the projected motion of objects is small relative to the image size, we can restrict the search for corresponding points to small regions in the image. We look for a discrete motion displacement $V(x,y) = (u(x,y), v(x,y)) \in (\pm\delta, \pm\delta)$ to minimize:

$$\int [\phi(E_t(x,y), E_{t+\Delta t}(x+u\Delta t, y+v\Delta t)) + \lambda(du/dx^2 + du/dy^2 + dv/dx^2 + dv/dy^2)]dxdy \quad (2)$$

where ϕ is a comparison function which measures the pointwise match between two images. Even with the small motion assumption, the complexity of this procedure is high; at each point, the number of possible displacements is $(2\delta+1)^2$. The number of possible vector fields for an $N = n \times n$ image is:

$$(2\delta+1)^{2N} \quad (3)$$

Of course, many of these vector fields are far from smooth.

2.3 Approximate Algorithm

We approximate the above equation, using the constraint of piecewise planarity. Recall that the projected motion of a planar patch orthogonal to the viewing direction is constant over the projected area of the patch. Choose a patch diameter ν, dependent on distance to objects in the scene and their expected size in the image. The magnitude of δ depends on the expected velocities of objects in the scene, their distances from the camera, and the time separation Δt between frames.

We construct the optical flow pointwise. For each displacement, each point determines evaluates ϕ at that displacement. Then we sum the match scores over the region P_ν. Each point chooses the displacement which maximises the patch score, over all displacements. This is the operation of "non-maximum suppression", or "winner-take-all" across velocities out of the finite allowed set. We identify a displacement only at those locations (x,y) at which the maximum vote is unique; ties are ambiguous and are eliminated. Fig. 1 shows a schematic diagram of the computational stages, using the L_2 norm as ϕ. The resulting velocity field is the union of these pointwise vector displacements. The number of possible fields is enormously reduced:

$$(2\delta+1)^2 N \quad (4)$$

The approximation, in each overlapping patch $P_\nu(x,y)$ of diameter ν, minimises:

$$\sum_p \phi(E_t(x,y), E_{t+\Delta t}(x+u\Delta t, y+v\Delta t)) \quad (5)$$

ϕ is a comparison function, for example, L_2 norm. Drumheller and Poggio [6] use a similar voting scheme to select correct matches in stereo.

3 Matching

In computing optical flow for short-range motion, correspondence among image elements must be determined by some form of search, as in stereo. In optical flow, the search for a matching point is two-dimensional, whereas in stereo, the correspondence is restricted to one-dimensional loci, along epipolar lines. However, by assuming that the magnitude of the projected motion is small, we can constrain search to a small region in the image plane.

3.1 Matching Primitives

We formulate matching, using the planar patch assumption, as minimising the difference between a patch of one image containing a feature and a patch of the second image containing the, possibly transformed, feature. Edges are useful descriptors. Edge are localisable, and provide constraint in their normal direction, and record location with sharp variation in image brightness, which are less likely to be disturbed by optical effects (changes of illumination and surface orientation due to rotation). However, other locations not on edges are also useful. Edge maps are just a particular transform of the input brightness. Brightness can be used directly. Generally, matching minimises the comparison function ϕ between a site in the image at time t, E_t, and a site in the image at time $t+\Delta t$, $E_{t+\Delta t}$. The comparison function ϕ may include an edge-detection step or some other image transformation, for example, convolution by $\nabla^2 G$. Transformation by ϕ yields:

$$I_t = \phi(E_t) \quad (6)$$

The general matching criterion, under motion $\underline{v} = (v_x, v_y)$ is to minimise:

$$\phi(I_t(x,y), I_{t+\Delta t}(x+v_x\Delta t, y+v_y\Delta t)) \quad (7)$$

Typically, I is an binary edge map produced by an edge detector such as the zero-crossings of $\nabla^2 G$ or the output of the Canny edge detector [5]. For a binary edge map, ϕ reduces to logical *and*, or logical multiplication, of the binary edge maps, and the optimization becomes maximization. However, this does not treat the two images symmetrically, since it only counts matches; it is better to minimise a matching function ϕ which is the logical difference or *exclusive or* of binary images. Further refinement to the comparison function can be introduced by labeling edges with the direction of the gradient $\nabla G * E$ at the edge. The comparison function then must check that the difference between edge directions is within an allowable range. The comparison function ϕ becomes more complex, but still retains the flavor of a logical comparison. This coupling of comparison function and transformation produces an output only at edge points. An obvious extension to the edge-based method is to use:

$$I = signum(\nabla^2 G * E) \quad (8)$$

letting ϕ again be the *exclusive or*. Nishihara [20] has utilized this comparison function in a stereo system, PRISM. Going one step further, one can use the brightness values directly, letting $I = E$; the comparison function then becomes:

$$\phi(p,q) = |p-q| \quad (9)$$

where the norm is suitably chosen, for example, the L_2 norm. We do not match brightness directly, since the presence of noise makes the process unstable; rather, choosing the pixel whose brightness minimises Eq. 9 *regularizes* the solution of matching [2]. There are theoretical arguments that support the equivalence of cross-correlating the sign bit of the Laplacian filtered image and the Laplacian filtered image itself. The argument is based on the following theorem, which is a slight reformulation of a well-known result.

Theorem

If $f(x,y)$ and $g(x,y)$ are zero mean jointly normal processes, their cross-correlation is determined fully by the correlation of the sign of f and of the sign of g (and determines it). In particular

$$R_{f,g} = \frac{2}{\pi} arcsin(R_{\bar{f},\bar{g}})$$

where $\bar{f} = sign\ f$ and $\bar{g} = sign\ g$

Thus, cross-correlation of the sign bit is exactly equivalent to cross-correlation of the signal itself (for Gaussian processes). Notice that from the point of view of information, the sign bit of the signal is completely equivalent to the zero-crossing of the signal. Nishihara first used patchwise cross-correlation of the sign bit of DOG filtered images[20].

Recently, Uras et al. [29] have reformulated the instantaneous motion problem to use the brightness gradient constancy assumption. Then there are two constraints at each point in the image and the algorithm only suffers from the aperture problem when in fact the image only contains, in the patch (or aperture), one-dimensional information. The gradient ∇E can also be used as a primitive in our algorithm, and the comparison function ϕ is the magnitude of the vector difference of gradients.

3.2 Local Support from Features

Patchwise comparison in the voting method implicitly creates larger features from individual elements. A solitary pixel, of course, is not a suitable primitive for matching [17]. A single edge pixel, by comparison,

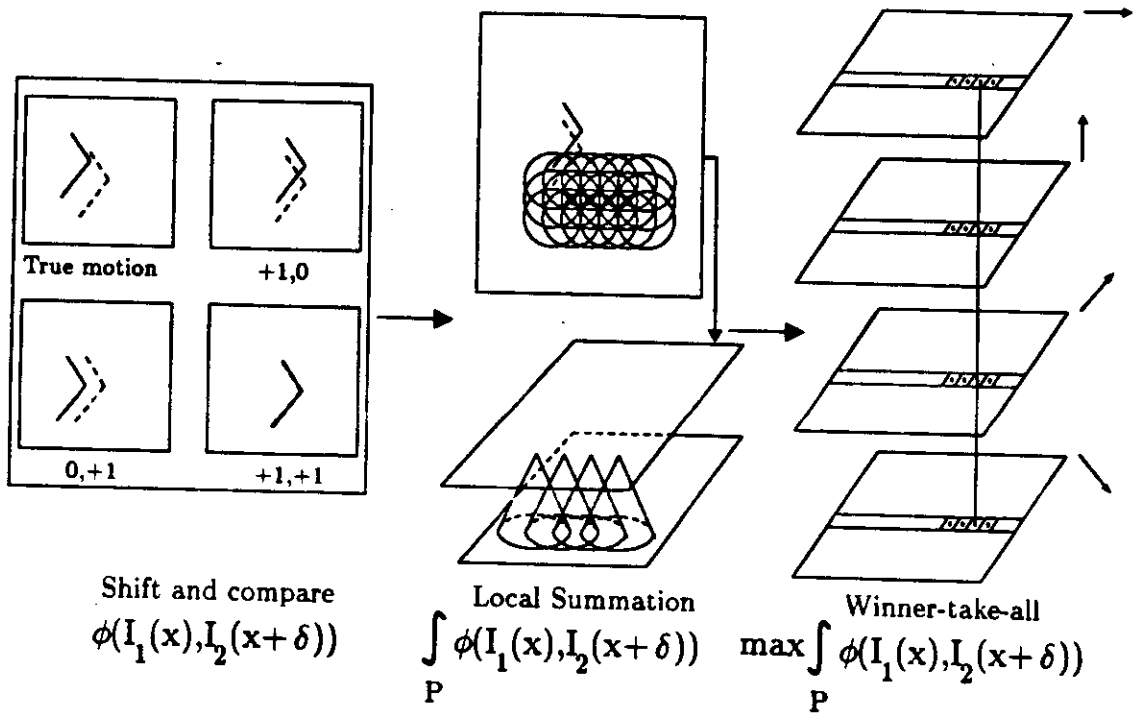

Figure 1: Schematic of the algorithm

is the result of computation which has support over a large area, specifically the area of the filter used in the edge detection step, typically, hundreds of pixels. The edge point is a compact local description of brightness variation at that point. Consider using as a feature a patch P of pixels surrounding each point. Then the comparison function for the L_2 norm becomes

$$\phi(x,y) = \sum_P ((I_t(x,y) - I_{t+\Delta t}(x + v_x\Delta t, y + v_y\Delta t)))^2 \quad (10)$$

and the matching minimises the sum of the squared differences in brightness. Of course, this reduces to maximising

$$\frac{\sum_P (I_t(x,y) \times I_{t+\Delta t}(x + v_x\Delta t, y + v_y\Delta t)}{\sum_P (I_{t+\Delta t}(x + v_x\Delta t, y + v_y\Delta t))^2} \quad (11)$$

which is the normalised cross-correlation.

3.3 Local Support from Constant Optical Flow

The assumption of locally constant optical flow leads us also to find the matching which minimises Eq. 10; the size of the patch is determined by the assumed size of the projected planar patches on the object surface. We are assuming a constraint on the local variation of the optical flow.

Local support is, of course, in our formulation, overlapping from pixel to pixel. Each pixel, surrounded by its patch, of diameter ν, independently chooses the optical flow to maximise matching in its patch. Because the patches overlap, the optical flow field has a smoothness property, depending upon ν. When $\nu > 0$, the local support region (assumed to be square) of each point shares $\frac{\nu}{\nu-1}$ of its area with the support regions of its near neighbors in the grid. When ν is zero, the data used by each pixel is entirely independent of all other pixels; as ν increases, the flow field becomes smoother, because the support region of adjoining pixels are essentially identical.

3.4 Choosing between Primitives

Finally, the output of the detection process can be defined either at some subset of the pixels, typically the edge points, or at all pixels in the image. Edges produce reliable results, because edge elements are less affected by optical effects, and their sparseness reduces ambiguity. When output is restricted to edge points, this sparseness is also a liability, requiring subsequent interpolation. We do not distinguish points where there is an edge feature, so that, even if the input is sparse, the output selection of a displacement is defined everywhere.

On the other hand, brightness primitives are dense, and are defined everywhere, but may be subject to ambiguity and be perturbed by optical effects. However, working directly with brightnesses over large regions has several advantages: assuming that noise is independently distributed, the expected sum of brightness differences over large regions should be biased by the noise, but to the same extent everywhere. By avoiding the edge detection step, the effects of noise are limited. This effect is apparent in our experiments. We have found from experimentation that brightness primitives are reliable, and will outline a scheme for combining both brightness-based and edge-based schemes. The reliability of brightness-based methods can be increased by simple preprocessing to remove any offset introduced between frames. In practice, we transform brightness values by subtracting the local average brightness, computed by convolution with a large Gaussian.

4 An Iteration Scheme

The implementation of the local quadratic variation constraint is computationally expensive, even for coarse discretisation of the velocity values. The simple assumption of local constancy is adequate in practice. Though it may be unnecessary in practice to consider quadratic support, it is of interest to develop a scheme that allows for a fast approximate solution based on the constant field assumption that is then refined in terms of higher order assumptions such as linear and quadratic support. It is natural to consider an iteration that first finds the best "constant" solution, then refines it with the best "linear" correction and finally finds the best "quadratic" correction. In general, the best quadratic correction does not provide the best quadratic approximation. Results however about the estimation of polynomial operators (see for instance [21] theorem 4.2) suggest that iterating the procedure should converge to the best quadratic approximation. In this way we can find the best "constant" estimation of the optical flow and then re-

fine it by successive iterations that cycle from the lowest to the highest order and to the lowest again. For the best linear estimate, the field contains six coefficients, the two constant terms, plus the four linear coefficients (two each for x and y). Then the number of possible fields (compare with Eq. 4 depends on the range $\pm\kappa$ over which the linear coefficients vary:

$$(2\delta + 1)(2\kappa + 1)^4 N \tag{12}$$

5 Parallel Implementation

We have implemented the local voting algorithm on a fine-grained parallel machine, the Connection Machine[9], which is particularly well-suited to the form of this algorithm [14]. The initial image transformations $E_i \rightarrow I_i$, smoothing, gradient operations, and edge detection, are all implemented on the Connection Machine. These operations depend only on data found in nearest-neighbors in the image coordinate system. Iterating over the displacement range $(\pm\delta, \pm\delta)$ is accomplished by shifting I_i over $I_{i+\Delta i}$ using local data movement operations, which are relatively fast for any machine, such as the Connection Machine, that embeds a two-dimensional mesh. Evaluation ϕ on the shifted I_i is a purely pointwise computation. Then, the sum over P_ν can be performed in a variety of ways: first, *region summation* [14] computes the sum of any large square region in a mesh on the Connection Machine in constant time, using hypercube wires, and, second, shifting and summing $\phi(I_i, I_{i+\Delta i})$ iteratively over the width ν in x and y will calculate the sum in time proportional to ν. "Non-maximum suppression" or "winner-take-all" is, of course, pointwise.

6 Combining Several Transformations

In Section 3.4 we argue that the distinction between brightness values and edges should not be sharply drawn. It is, however, advantageous when using brightness values to consider the effects of local variation of illumination, noise, and other optical effects on brightness values. Various analyses have shown that, when brightness values vary rapidly in the image, their position can be tracked well across time. So, we combine edge-like information with brightness computation by suppressing the votes at pixels where the gradient magnitude is low. This allows more reliable data to dominate the less reliable data. An alternative scheme for combining edge and brightness primitives simply sums the votes and proceeds as before.

7 Discontinuities

The optical flow field derived by our algorithm is a 2-dimensional vector field defined in the image plane. It is customary to detect the projected boundaries of objects in the projected velocity field [1]. Motion discontinuities can be found by using an algorithm that was suggested by data on the insect visual system [25]. The idea is to inhibit or veto the value of the optical flow at each point by the average value of the field over a large region centered at that point whenever the motion is of the same type. The scheme suggested by the fly method is the following: at each x, y consider separately the x and the y component of the optical flow, take its value and divide it by the average value, computed over a large region. Fig. 4 shows the output of this operation on two examples. The average may be Gaussian weighted. Interestingly, the basic operation is very similar to a recent proposal by [13]. It is also similar to performing a center-surround operation (such as the Laplacian of a Gaussian, but with much larger surround) on the logarithm of the optical flow.

Alternatively, discontinuities can be detected during the computation of optical flow. At motion discontinuities the assumption of a constant (or linear or quadratic) motion field is obviously wrong. Two surfaces, undergoing different motions, are in the patch. One would expect therefore that the "votes" at a motion discontinuity (say in the case of the "constant" motion algorithm) would fail to support clearly any single velocity. In fact, regions of close "ties", or equivalently of winners with locally minimum votes, often delineate motion discontinuities. Our implementation of this procedure scales the number of votes at a location by the total number of features in the voting neighborhood. Close ties receive values near 0.5. This idea can be developed further by considering more complete statistics of the votes [6, 27]. Because we compute discrete displacements, rotational motion causes optical flow variation which generates discontinuities between neighboring flow vectors. This type of discontinuity can be identified by examining coarser discretisation of the displacement range, or by examining the higher order coefficients in the flow variation (see Section 4).

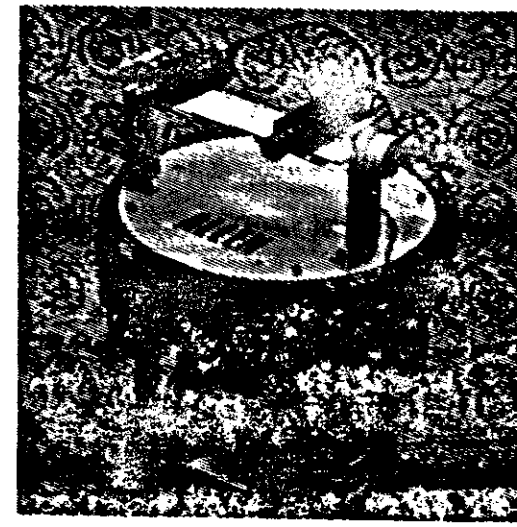

Figure 2: Mobile robot

Figure 3: Sparse output from edge-based algorithm

8 Experiments

The algorithm with various image transformation and comparison schemes has been run on several images acquired under relatively unconstrained conditions. All images are taken in indoor lighting conditions, where illumination is far from constant, even under small displacements in the world.

Fig. 2 shows an image of a mobile robot as it undergoes translational motion in the direction its cameras point. The projected motion is relatively small, under 8 pixels per frame. Fig. 3 shows the optical flow, output at edge elements, where the input is the thresholded zero-crossings of $\nabla^2 G$ and the comparison is a *exclusive or*. Fig. 4 shows the output of the discontinuity detection scheme suggested by studies of the fly.

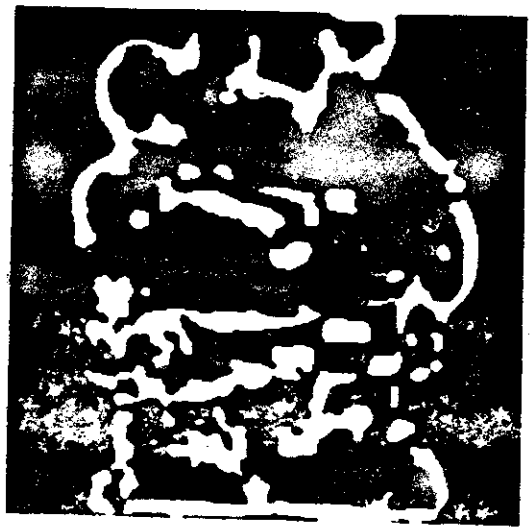

Figure 4: Output of the "Fly Method"

Figure 5: Rotating and translating persons

The next series of figures show the algorithm operating on images taken at time interval approximately 0.1 seconds: in Fig. 5, the person on the right is translating upward, while the person on the left is rotating toward the camera. The time between frames is relatively large, but the projected displacements on the image remain less than 8 pixels per frame. Fig. 6 shows the output (at edges) of a matching scheme using Canny edges and logical *and* comparison, above, and the dense output from brightness-based comparison below. In Fig. 7, discontinuities detected by local minima in the voting function are displayed – above is a continuous skeleton of the local minima regions, while below is a discrete skeleton of the same local minima. Note how these skeletons correctly identify the figure-ground boundaries associated with the moving figures. Gaps in these boundaries occur when the optical flow fields are ambiguous – at object boundaries where motion is tangent to the surface normal. Further details on experiments with the algorithm may be found in [15].

Figure 6: Top: Canny edge-based flow; bottom: brightness-based flow

9 Discussion

The algorithm will not find, in general, the 2-D projection of the true 3-D velocity field. This will happen only when the features used for matching correspond to markings on the 3-D surfaces and when either the features are sparse (no ambiguity) or the disambiguation step (the voting and non-maximum suppression stage) finds the true correspondence (i.e., the underlying assumptions are satisfied). Even when the result is not the true motion field, the algorithm will usually preserve its most important qualitative properties[30]. Most importantly, discontinuities in the derived optical flow will, very often, correspond to object boundaries.

The optical flow algorithm we describe exhibits the same behavior as the human visual system as seen in psychophysical experiments [3] including the barberpole illusion and motion capture[24]. Unlike other optical flow algorithms [11, 31], our scheme is one-step, non-iterative and does not require later smoothing steps to regularise the computation. Further, the mechanism directly provides excellent cues for segmentation. Our future work includes investigation of multiple scale methods for larger motions. Currently, the algorithm operates on single frames, but multi-frame analysis, with the use of spatio-temporal filters, should improve the results. Adaptive choice of neighborhood size, near discontinuities, either during the feed-forward step or during a feedback step,

Figure 7: Continuous skeleton of low relative vote regions

will significantly improve performance near boundaries. In addition, the size of the patch (the neighborhood size) should be reduced when our main assumption is not satisfied. i.e., when the projected velocity field is not locally constant. Fortunately, it is relatively inexpensive to evaluate multiple neighborhood sizes on the Connection Machine, during the comparison stage. Ambiguity arises during matching when the image patch does not contain significant detail. Where the selectivity is low, our algorithm can expand the patch size (or choose one of the larger, pre-selected sizes), to reduce the occurrence of ambiguity.

10 Conclusion

There are several advantages of our method over differential approaches to motion detection. First, it handles larger motions (we have used it for displacements up to 20 pixels), without sacrificing discrimination of small motions. Second, noise is reduced by summation over large support regions and not using derivatives. Our method uniformly integrates many image transformations, and leads to dense optical flow fields. Dense flow fields remove the necessity of interpolating or smoothing the output field. Further, segmentation is improved since it is not combined with interpolation. Additional information for segmentation is provided by the statistics of the voting step.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(4):384–401, 1985.

[2] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. A.I. Memo No. 924, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1987. Also Proc. IEEE, in press.

[3] H. H. Bülthoff, J. J. Little, and T. Poggio. Parallel motion algorithm explains barber pole and motion capture illusion without "tricks". *J. Opt. Soc. Am.*, 4:34, 1987.

[4] H. H. Bülthoff and H. A. Mallot. Interaction of different modules in depth perception. In *Proceedings of the International Conference on Computer Vision*, pages 295–305, London, England, June 1987. IEEE, Washington, DC.

[5] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[6] M. Drumheller and T. Poggio. On parallel stereo. In *Proceedings of IEEE Conference on Robotics and Automation*, pages 1439–1448, Washington, DC, 1986. IEEE.

[7] C. Fennema and W. Thompson. Velocity determination in scenes containing several moving objects. *Computer Graphics and Image Processing*, 9(4):301–315, Apr. 1979.

[8] E. C. Hildreth. *The Measurement of Visual Motion*. MIT Press, Cambridge, Mass., 1984.

[9] W. D. Hillis. *The Connection Machine*. MIT Press, Cambridge, Mass., 1985.

[10] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[11] J. Hutchinson, C. Koch, J. Luo, and C. Mead. Computing motion using analog and binary resistive networks. *IEEE Computer Magazine*, 21:52–64, March 1988.

[12] J. K. Kearney, W. B. Thompson, and D. L. Boley. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(2):229–244, Mar. 1987.

[13] E. H. Land. An alternative technique for the computation of the designator in the retinex theory of color vision. *Proceedings of the National Academy of Science*, 83:3078–3080, 1986.

[14] J. J. Little, G. E. Blelloch, and T. Cass. Algorithmic techniques for vision on a fine-grained parallel machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence, in press*, 1989.

[15] J. J. Little and H. H. Bülthoff. Parallel computation of optical flow. Technical Report AIM-929, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1988.

[16] J. J. Little and A. Verri. Parallel computation of optical flow. Technical Report AIM-1066, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1988.

[17] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Company, San Francisco, 1982.

[18] D. Marr and S. Ullman. Directional selectivity and its use in early visual processing. *Proceedings of the Royal Society of London B*, 211:151–180, 1981.

[19] H.-H. Nagel. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics, and Image Processing*, 21(1):85–117, Jan. 1983.

[20] H. Nishihara. Practical real-time imaging stereo matcher. *Optical Engineering*, 23(5):536–545, 1984.

[21] T. Poggio. On optimal nonlinear associative recall. *Biological Cybernetics*, 19:201–209, 1975.

[22] T. Poggio, E. Gamble, and J. J. Little. Parallel integration of vision modules. *Science, in press*, 1988.

[23] T. Poggio, J. Little, E. Gamble, W. Gillett, D. Geiger, D. Weinshall, M. Villalba, N. Larson, T. Cass, H. Bülthoff, M. Drumheller, P. Oppenheimer, W. Yang, and A. Hurlbert. The MIT Vision Machine. In *Proceedings Image Understanding Workshop*, Cambridge, MA, April 1988. Morgan Kaufmann, San Mateo, CA.

[24] V. Ramachandran and V. Inada. Spatial phase and frequency in motion capture of random-dot patterns. *Spatial Vision*, 1(1):57–67, 1985.

[25] W. Reichardt, T. Poggio, and K. Hausen. Figure–ground discrimination by relative movement in the visual system of the fly. part II: Towards the neural circuitry. *Biological Cybernetics*, 46:1–30, 1983.

[26] W. Reichardt, R. Schlögl, and M. Egelhaaf. Movement detector of the correlation type provide sufficient information for local computation of 2-D velocity field. *Die Naturwissenschaften, in press*, 1988.

[27] A. Spoerri and S. Ullman. The early detection of motion boundaries. In *Proceedings of the International Conference on Computer Vision*, pages 209–218, London, England, June 1987. IEEE, Washington, DC.

[28] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, Cambridge and London, 1979.

[29] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics, in press*, 1988.

[30] A. Verri and T. Poggio. Against quantitative optical flow. In *Proceedings of the International Conference on Computer Vision*, pages 171–180, London, England, June 1987. IEEE, Washington, DC.

[31] A. L. Yuille and N. M. Grzywacs. A computational theory for the perception of coherent visual motion. *Nature*, 333:71–74, May 1988.