Abstract
We describe a parallel algorithm for computing optical flow from short-range motion. Regularizing optical flow computation leads to a formulation which minimizes matching error and, at the same time, maximizes smoothness of 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 detector-from-motion [26], the distances are relatively small compared to the size of the object in the scene. We use the estimated motion to compute discrete motions. How correspondences of image elements may be determined, motion computation under these conditions resembles interval states. Between frames the agreement of a moving object can change due to its own motion, camera motion, light source motion, and all three, among other effects [20]. However, when the local brightness variation in the surface is sufficiently large, the errors introduced by these effects are relatively small.

1 Introduction
Optical flow is generated on the retina of an observer by objects moving relative to the observer. The changing pattern of image brightness on the retina defines a vector field, the optical flow. In restricted cases, the motion of some objects can be described directly from the optical flow. Rich qualitative information on the motion of objects and their boundaries is contained in the optical flow and discontinuities of the optical flow.

The velocity field of moving objects in a 2-D vector field \( \mathbf{V}(x, y) \). When projected onto the optical flow coordinate system of an observer, this field generates the projected velocity field of objects \( \mathbf{V}(x, y) \). What is observed on the retina is the image brightness change \( \Delta E_r(x, y) \). The optical flow \( \mathbf{V}(x, y) \), is a time-varying vector field describing image brightness changes. In general, \( \mathbf{V} \) and \( \mathbf{V}(x, y) \) are not the same.

In the Vision Machine project [26], our goal is to devise robust methods for computing early visual modules and to integrate there modules. The flexible, robust behavior of the human visual system is largely due to integration of many early motion modules. In [26], the output of the integration stage is a field of the physical discontinuities in the scene that is the optical flow and its discontinuities are important inputs to the integration stage and can provide cues to figure-ground separation, such as demonstrated in [26].

1.1 Assumptions
In our approach to computing the optical flow, we make several assumptions about the input images. First, the size of the features in the images is small, on the order of one retinal image frame. Second, differential approaches to motion analysis rely on the spatial and temporal derivations of images to compute optical flow [26, 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 texture. This is the case of certain natural motions, correspondence between elements in the two images is an issue. Problems with this approach, however, are that the features themselves have large spatial extent, and that the surface texture lines are often discontinuous. 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, there will likely be many variations in orientation of the retinal Jacobian, and the aperture problem will not occur.
is the result of convolutions which has support over a large area, specifically 10x10 area of the filter used in the edge detection stage, typically, hundreds of pixels. The edge point is in a compact local description of brightness variation at that point. Considering any size 3x3 patch \( P \) of pixels surrounding each point. They are compared against the \( L \) point becomes

\[
\phi(x, y) = \sum L_i(x, y) - L_{i+m}(x, y, x+M, y+N) \tag{10}
\]

and the matching is equivalent to the sum of squared differences in brightness. Of course, this reduces to maximizing

\[
\sum L_i(x, y) - L_{i+m}(x, y, x+M, y+N) \tag{11}
\]

which is the normalized cross-correlation.

3.3 Local Support from Constant Optical Flow

The assumption of locally constant optical flow leads us to use the histogram method which minimizes Eq. 10, the sum of the paths that is determined by the assumed use of the proposed phase gradient at the object surface. We are measuring a quantity on the local variation of the optical flow.

Local support in, of course, its formulation, overlapping from patch to patch. Each patch, determined by the patch, of diameter \( M \), independently chooses the optical flow to maximize matching at its patch. Because the patches overlap, the optical flow field has a discontinuous piecewise constant flow with support region \( S \), of \( 2M \) by \( 2M \), the local support region (assumed to be square) of each point squares \( 2M \) by \( 2M \), with the support region of its eight neighbors in the grid. When \( M \) is zero, the data used by each pixel is entirely independent of all other pixels as in interior, the flow field becomes discontinuous, because the support region of adjacent pixels are essentially identical.

3.4 Choosing between Primitives

Finally, the output of the detection process can be defined with or overlap patches of the path; typically the edge points or all pixels in the image. Edges produce reliable results, because edge points are less affected by optical effects, and that brightness samples ambiguity. When output is evaluated for edge points, the appearance is also a limitation, requiring subpixel interpolation. We do not distinguish points why there is an edge feature, so that, even if the output 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 independent and normally 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, and this effect is apparent in our experiments. We have found from experimentation that brightness primitives are reliable, and we will utilize a scheme for combining both brightness-based and edge-based schemes. The reliability of brightness-based methods can be enhanced by simple preprocessing to remove any offset introduced between frames. In practice, we transform brightness values by subtracting the local average brightness, replaced by convolution with a large Gaussian.

4 An Iteration Scheme

The implementation of the local quadratic variation constraint is computationally expensive, even for center discontinuities of the velocity values. The simple assumption of linear constancy is adequate in practice. Though it may be impossible in practice to consider quadratic support, it is often to derive a scheme that allows for a fast approximate solution based on the continuous field assumption that is then refined in work of higher-order assumptions with linear support and quadratic support. It is natural to consider an algorithm that results into the best “consistent” solution that refines it with the best “linear” connection and finally finds the best “quadratic” connection. In general, the best quadratic connection does not provide the best quadratic approximation. Results however about the estimation of polynomial approximations (see for instance [21], Section 4.2) suggest that imposing this constraint should converge to the best quadratic approximation. In this way, we can find the best “consistent” estimation of the optical flow and thus re-
Figure 4: Output of the “Fly Method”

The next series of figures show the algorithm operating on images taken at time intervals 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. Even though human beings are sufficiently large, the projected displacements on the image remain less than a pixel per frame. Fig. 6 shows the output (at edges) of a matching scheme using Canny edges and logical and comparison, above, and the error 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 true minimal region, while below is a discrete skeleton of the same minimal regions. Note how these skeletons correctly identify the figure-ground boundaries associated with the moving figures. Above in these boundaries appear when the optical flow fields are ambiguous—i.e., object boundaries where motion is tangent to the surface tangent. Further simulation experiments with the algorithm may be found in [15].

9 Discussion

The algorithm will not find, in general, the 2-D projection of the over 3-D velocity field. This will happen only when the features used for matching correspond to markings on the 3-D surface and when either the features are sparse (no ambiguity) or the data is strong (coherence). Even when the result is not the true motion field, the algorithm will usually preserve some important qualitative properties [15]. 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 [25], including the barbershop illusion and motion capture [26]. Unlike other optical flow algorithms [13, 14], our scheme is one step, non-iterative, and does not require later smoothing steps to regularize the computation. Further, the mechanism directly provides elegant cues for segmentation. The 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 gradient-enhanced filters, should improve the results. Adaptive choice of neighborhood size, near discontinuities, either during the feed-forward step or during a feedback step.