Robust Dynamic Motion Estimation Over Time

Michael J. Black and P. Anandan

Research Report YALEU/DCS/RR-835
November 1990

Robust Dynamic Motion Estimation Over Time

Michael J. Black and P. Anandan

Department of Computer Science
Yale University
P.O. Box 2158 Yale Station
New Haven, CT 06520–2158

Abstract

This paper presents a novel approach to incrementally estimating visual motion over a sequence of images. We start by realistically reformulating constraints on image motion to account for the possibility of multiple motions. This is achieved by exploiting the notions of *weak continuity* and *robust statistics* in the formulation of a minimization problem. The resulting objective function is non–convex. Traditional stochastic relaxation techniques for solving the minimization problem prove inappropriate for the task as they require many iterations to converge whereas motion estimation must be dynamic. We present a highly parallel *incremental stochastic minimization* algorithm which has a number of advantages over previous approaches. Between any pair of frames in an image sequence, only simple local computations take place. Robustness and accuracy are then achieved by extending the estimation task over time. The incremental nature of the scheme makes it truly dynamic and permits the detection of occlusion and disocclusion boundaries.

1 Introduction

This paper presents an approach for the incremental estimation of visual motion over time. The task of estimating visual motion involves specifying constraints which relate spatiotemporal intensity variations to image motion and express our assumptions about the spatiotemporal variation of the motion itself. We also need an effective procedure for computing a flow field consistent with the assumptions. Each of these steps is critical. The assumptions must accurately model the expected properties of the scene and the image sequence while the computation must be appropriate for motion processing; that is it must be parallel, incremental and robust. We first consider reformulating three constraints on image motion to more accurately model situations containing multiple motions. We then formulate an incremental scheme for exploiting the constraints.

We will focus on three constraints in particular; of course, this is not an exhaustive set of constraints, but serves to illustrate techniques for coping with multiple motions. The *data conservation* constraint states that the image measurements (e.g., the intensity structure) corresponding to an environmental surface patch change slowly over time. The *spatial coherence* constraint is derived from the observation that surfaces have spatial extent and hence neighboring points on a surface will have similar motion. Finally, the *temporal coherence* constraint is based on the observation that the velocity of an image patch changes gradually over time.

A traditional assumption is that within a small image region only a single motion is present. This leads to a Gaussian noise model for the data conservation and spatial coherence assumptions. The assumption however, ignores the case of motion discontinuities [7, 15] and results in either errors in the motion estimate or over smoothing across discontinuities. This paper formulates more realistic constraints which account for motion discontinuities resulting from multiple motions at a given point by exploiting the notions of *weak continuity* [8, 13] and *robust statistics* [3, 14, 19]. The result is increased robustness and accuracy.

The constraints are formulated as energy terms in an objective function. Estimating the

motion is then the task of finding a flow field with minimum energy. With the removal of the simplifying assumption of Gaussian noise the objective function becomes highly non-convex and hence the minimization problem is made more difficult.

The definition of the constraints in terms of local neighbors in a grid allows the problem to be formalized as a *Markov Random Field (MRF)* with a *Gibbs distribution* [13, 12]. Stochastic methods, like simulated annealing [13, 23, 29], are one approach for minimizing such complex functions with many local minima. While they are highly parallel, these approaches unfortunately converge slowly, typically requiring many hundreds or thousands of iterations. This makes them ill suited to motion estimation which must be dynamic.

We propose a new *incremental stochastic minimization (ISM)* algorithm which has many of the the benefits of simulated annealing without the shortcomings. As opposed to minimizing the objective function for the motion between two frames, the ISM approach is designed to minimize an objective function which is *changing slowly over time.* The assumption of a slowly changing objective function is made possible by exploiting current motion estimates to compensate for the effects of the motion on the objective function. In this sense, the algorithm is truly incremental; estimates are carried over from frame to frame and refined over time. The cost of computing the motion estimate is spread over an entire sequence of images.

The amount of computation performed for any pair of frames is minimal; accurate motion estimates are obtained by exploiting the wealth of information available over a long sequence of images. The result is an algorithm which starts with initial, rough, motion estimates and refines them over time. This is a desirable property for many applications. In particular, a mobile robot should always have a motion estimate available no matter how coarse.

Most work in motion estimation has focused on the formulation of the two frame case without addressing explicitly how the flow computation could be made incremental. Those researchers that have exploited longer image sequences [9] have typically focused on processing a spatiotemporal image sequence directly to achieve more complex inferences about

surfaces under motion. This differs greatly from the *on–line* approach presented here.

Previous work has shown that such an incremental stochastic scheme can be used to reliably compute discrete motion estimates, with motion discontinuities, given densely sampled images [6]. The approach described here extends that incremental scheme to deal with fractional motions by applying techniques from continuous annealing [29], and to large motions through the use of hierarchical processing [1, 5].

While the approach is ostensibly designed for computing optic flow, it has more general applicability. The ability to minimize an objective function over time by compensating for image motion may allow other problems to be formulated and solved in this temporal minimization framework. For example, by formulating intensity based segmentation as an optimization problem [8, 11], it may be possible to perform the segmentation over time.

The next section examines approaches for coping with motion discontinuities and techniques for reformulating the constraints to account for multiple motions. The incremental algorithm for exploiting the constraints is then presented in section 3 along with a discussion of motion discontinuities and how they can be detected. Section 4 then extends the algorithm to handle large motions. Experimental results with the algorithm on real data are presented in section 5. We conclude with a discussion of the the significance of the approach and the avenues of research it opens.

2 Multiple Motions, Robust Statistics and Weak Continuity

The paradigm within which we operate is the standard one of specifying our assumptions about the scene and the images in terms of constraints. Each constraint becomes a term in an overall objective function that is minimized to obtain the motion field. We reformulate the traditional constraints to account for multiple motions. Here, we focus on multiple motions occurring at occlusion and disocclusion boundaries [7] and not on the issue of multiple motions resulting from transparency [4].

The constraints are formalized as energy functions over local neighborhoods, or *cliques*, in a grid. For an image of size $n \times n$ pixels we define a grid of *sites:*

$$S = \{s_1, s_2, \ldots, s_{n^2} \mid \forall w \; 0 \leq i(s_w), j(s_w) \leq n-1\},$$

where $(i(s), j(s))$ denotes the pixel coordinates of site s. The energy associated with the constraints is used to compute a *Gibbs* distribution [13] on possible motions. This then leads to a *Markov Random Field (MRF)* model of image motion [6, 21, 22] where a random vector $\mathbf{u}(s) = (u(s), v(s))$ represents the horizontal and vertical components of the motion at site s.

For the remainder of the paper we focus on three constraints [6]: *data conservation*, *spatial coherence*, and *temporal coherence*. The assumptions (and the corresponding prior models) underlying the constraints are violated in areas containing multiple motions. Various approaches have been presented for relaxing the the spatial coherence assumption; in particular, the notion of *weak continuity constraints* has been popular [8, 13] . Less attention has been paid, however, to relaxing the data conservation assumption. In fact, we observe that the two problems are both special cases of the more general statistical problem of *outlier rejection* encountered in *robust statistics* [3, 14, 19].

The general problem is one of finding the best fit of a model to data where we have some (possibly inaccurate) prior model of the statistics of the errors in the data. The least-squares fit of the sort employed typically in these constraints implies a Gaussian noise model. In the case of multiple motions, our prior Gaussian noise model will be incorrect due to outliers. Our goal then is to find the best fit to the data while ignoring outlying data which could corrupt our solution.

It should be noted that the chosen constraints do not represent an exhaustive set of constraints; they are primarily meant to be illustrative. Additional geometric constraints, for example a rigid body motion assumption, could be added to the current set. In fact, while the chosen constraints are all constraints on optic flow, there is no reason to restrict the paradigm to flow computation. A different choice and formulation of the constraints

might, for example, allow the direct computation of motion and depth [17].

What is critical then is that any formulation of the constraints take into account the possibility of multiple motions at a point. The approaches presented in this section, for coping with multiple motions, are fairly general and may be applicable to other problems as well. Additionally, it is important to recognize that the incremental minimization paradigm presented in the remainder of the paper is also applicable to a wider class of problems.

2.1 The Data Conservation Constraint

The data contribution to the motion estimate may be derived from a gradient based approach (first order [18] or second-order [27]), correlation (sum of squared difference (SSD) minimization [2, 24]) or some other approach. In any case, the data error term embodies the assumption that the intensity of a surface element remains constant over time, although its image location may change. The important point to note is that the error term is quadratic, which is a direct consequence of an additive Gaussian noise model.

We consider the correlation based approach since it is computationally simple and performs well in empirical tests when applied to band-pass filtered images [10]. Let s and t denote image locations, or sites, in S. We define a neighborhood of s, $\eta_D(s)$, for the data conservation constraint as:

$$\eta_D(s) = \{t \mid (i(t), j(t)) = (i(s) + \Delta i, j(s) + \Delta j),\ -c \leq \Delta i, \Delta j \leq c\},$$

which defines a square "window" of size $(-2c + 1) \times (2c + 1)$. Note that according to this definition $s \in \eta_D(s)$.

Given image intensity functions I_n and I_{n+1} between two successive frames (n and $n+1$), the local contribution (at site s) to the data conservation constraint is defined as an energy term E_D over the space of possible displacements (u, v) at site s (note, we will often drop s when it is clear that the function is evaluate at all sites):

$$E_D(u, v, s) = \sum_{t \in \eta_D(s)} (I_n(i(t), j(t)) - I_{n+1}(i(t) + u, j(t) + v))^2. \quad (1)$$

An SSD *surface* is defined over the space of possible displacements (u, v) with the height of the surface corresponding to the data error, $E_D(u, v)$, of that displacement. The minimum of this surface corresponds to the best motion estimate with respect to the data conservation assumption.

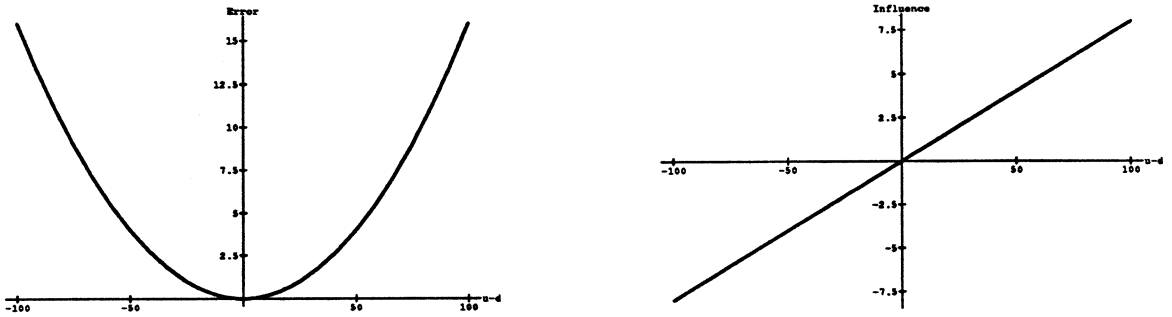
This measure assumes that all the points in the neighborhood $\eta_D(s)$ are translated by the uniform velocity (u, v) and the resulting image is corrupted by additive Gaussian noise. (In practice, even if the velocities vary gradually around (u, v), this measure serves as a good approximation.)

The standard quadratic error measure has the property that as data errors increase, the contribution of the error term increases without bound. As a result, when multiple motions are present within the neighborhood of a site, the correlation computed for one of the motions is corrupted by the data errors corresponding to the other motion.

When multiple motions are present, each motion corresponds to a different surface. We assume that the surfaces have different intensity structures. Then when performing the correlation consistent with one of the motions, there are two distinct statistical populations of errors. For errors measured from the consistent surface we make the standard assumption of Gaussian noise. The errors resulting from the uncorrelated surface can also be modeled by Gaussian noise, but since we assume that the intensity structure of the surfaces is different, the errors will have a larger variance.

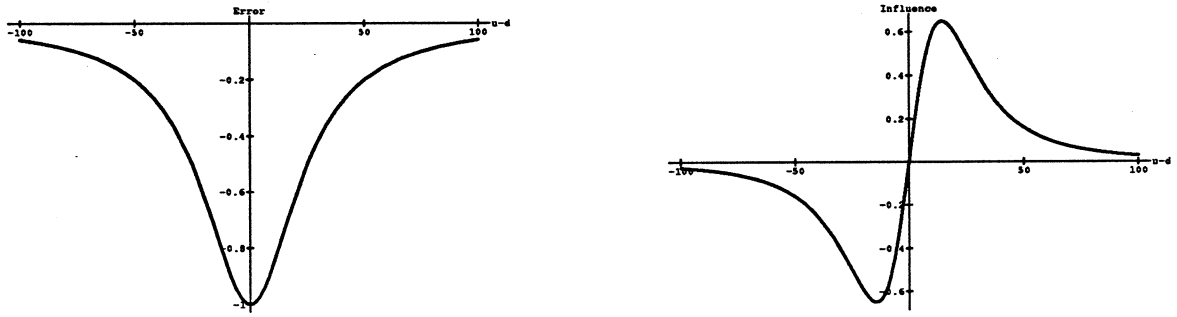
What is needed is a new error measure which takes into account these two statistical populations. Heuristically, we would like such an error measure to behave like the SSD measure when the data errors are small (and hence are more likely to have come from the consistent surface). We also want the influence of large errors (which correspond to the uncorrelated motion) to be reduced; that is we want to treat them as outliers which can be rejected.

One way of characterizing the behavior of an error measure, $\phi(x)$, is by its *influence function*, $\psi(x) = \frac{d}{dx}\phi(x)$, [3, 14]. Qualitatively, the influence function of an error measure



a *b*

Figure 1: *a)* Standard quadratic error measure, *b)* Influence function for the quadratic error measure.



a *b*

Figure 2: *a)* A robust error measure, ϕ_D, *b)* Influence function ψ_D for ϕ_D.

captures the effect of an observation on the solution. For the standard quadratic error measure (figure 1*a*) the influence of errors increases linearly and without bound (figure 1*b*).

What is desired is a new error measure with the following properties. For small errors, consistent with the motion under consideration, the influence should increase approximately linearly. As errors grow, they are less likely to be from the motion under consideration and hence their influence should be reduced.

An error function with these properties (figure 2*a*) is:

$$\phi_D(x) = \frac{-1}{1 + (x/\Delta_D)^2},$$

where Δ_D is a constant scale factor. Examining the influence function of ϕ_D (figure 2*b*) we see it has the desired properties. In particular, the influence of outliers tends to zero. This function ϕ_D is related to the *redescending* estimators used in robust statistics [3, 14, 19].

The data conservation constraint is now redefined as:

$$E_D(u,v,s) = \sum_{t \in \eta_d(s)} \phi_D(I_n(i(t),j(t)) - I_{n+1}(i(t)+u, j(t)+v)). \quad (2)$$

The local "data error" is the sum of the values of the ϕ_D for each of the points within the window. Once again, minimizing this function yields the motion estimate most consistent with the data.

Sub-pixel Accuracy

The data error term $E_D(u,v)$ as defined is discrete. Sub-pixel motion estimates can be obtained by interpolating the error surface. In the case of the quadratic error measure it is possible to interpolate the surface by fitting a quadratic about the minimum (u_0, v_0) [1, 25].

When the Gaussian noise assumption is violated such interpolation is incorrect. However, with the new robust estimator, a simple quadratic interpolation is inappropriate. Interpolating the new error surface is achieved by using *bi–cubic splines.*

Without loss of generality, assume that the motion is less than a pixel. Then to perform a bi–cubic interpolation requires a 5×5 pixel search centered about zero displacement. First a spline is fit to each row in the error surface, which requires that the first and second derivatives of the surface, along the row, be computed. These values can be stored. Then computing the value of the surface at any sub-pixel displacement involves first computing the interpolated value for each row, then fitting a spline to the new sub-pixel column.

Of coarse, more accurate sub-pixel estimates could be achieved in the presence of multiple motions if a surface segmentation is available. In this case, the points corresponding to each surface could be correlated separately. If no segmentation is available, the robust error measure provides an improvement over the traditional quadratic measure.

2.2 The Spatial Coherence Constraint

Our modification of the standard spatial coherence, or "smoothness," constraint is similar to the use of *weak continuity constraints* in Markov Random Field approaches to image

restoration [8, 13]. This idea has been explored by a number of researchers in the context of surface reconstruction (usually from stereo) and motion field computation including [8, 20, 26, 28]. In particular, we follow the formulation used by Geman and Reynolds in [12].

The neighborhood for the spatial coherence constraint is defined to be the nearest neighbors of a site s at location (i, j) in the grid:

$$\eta_S(s) = \{t \mid (i(t), j(t)) \in \{(i+1, j), (i, j+1), (i-1, j), (i, j-1)\}\}.$$

We formulate the constraint as consisting of a sum of error terms $E_S(\mathbf{u}, s)$ defined locally at site s as:

$$E_S(\mathbf{u}, s) = \sum_{t \in \eta_S(s)} (\mathbf{u}(s) - \mathbf{u}(t))^2, \tag{3}$$

where $\mathbf{u}(s) = (u(s), v(s))$ is the motion vector at site s. This is the standard quadratic smoothness term [2].

Once again, the spatial coherence assumption and its standard (quadratic) formulation are invalid in areas containing multiple motions. To deal with this we use the weak continuity constraint. The spatial coherence constraint can be reformulated as:

$$E_S(\mathbf{u}, s) = \sum_{t \in \eta_S(s)} \alpha(l)(\mathbf{u}(s) - \mathbf{u}(t))^2 + \beta(l),$$

where l is a *continuous line process* variable, $0 \leq l \leq 1$, $\alpha(0) = 0$ and is increasing, and $\beta(0) = 0$ and is decreasing. The value of l can be thought of as indicating the likelihood of a discontinuity and $\beta(l)$ can be thought of as a penalty for introducing a discontinuity. When l is close to 0 the likelihood of a discontinuity is high when it is close to 1 the likelihood is low. This is a generalization of the Blake and Zisserman formulation [8].

The line processe variables can be removed from the smoothness constraint by first minimizing over them [8, 12] resulting in an equivalent minimization problem:

$$E_S(\mathbf{u}, s) = \sum_{t \in \eta_S(s)} \phi_S(\mathbf{u}(s) - \mathbf{u}(t)), \tag{4}$$

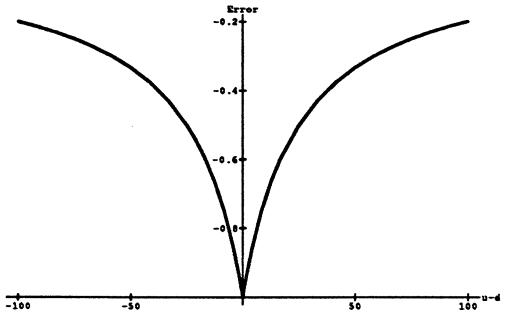


Figure 3: Geman and Reynolds ϕ function.

which is just a function ϕ of the difference in the neighbors' flow. For the appropriate choice of α and β (see [12]) we have

$$\phi_S(x) = \frac{-1}{1 + |x|/\Delta_S},$$

which is shown graphically in figure 3. This error measure, like ϕ_D, saturates as errors increase thus performing outlier rejection. This property of becoming non-committal as the differences increase amounts to a weakening of the spatial coherence assumption.

2.3 The Temporal Coherence Constraint

The temporal coherence constraint is intended to capture the notion that the motion of a particular surface element changes gradually over time. Our current formulation of the constraint embodies the assumption that the image plane acceleration of a patch is constant (over time). This can be regarded as a first approximation to a more accurate model, namely constant 3-D acceleration.

The constant acceleration assumption can be expressed as:

$$\begin{aligned} \Delta \mathbf{u}_t^* &= \text{const} \\ &= \mathbf{u}_t^* - \mathbf{u}_{t-1}^*, \end{aligned}$$

where $\mathbf{u}^*$ denotes the actual image velocity of a given surface patch, $\Delta\mathbf{u}^*$ denotes the true acceleration, and t denotes time. In practice, however, these true estimates are unknown, so this assumption should be applied to estimated quantities.

Let $\mathbf{u}^p$ and $\Delta\mathbf{u}^p$ denote the predicted velocity and acceleration and $\mathbf{u}^e$ and $\Delta\mathbf{u}^e$ the estimated values. We predict the new velocity at time t of a given patch as the estimated motion at the previous time instant plus the predicted acceleration:

$$\mathbf{u}_t^p = \mathbf{u}_{t-1}^e + \Delta\mathbf{u}_t^p. \tag{5}$$

Since the estimated accelerations may be noisy, we predict the new acceleration to be a temporal average of previous estimates. This can be obtained by,

$$\Delta\mathbf{u}_t^p = \alpha\Delta\mathbf{u}_{t-1}^e + (1-\alpha)\Delta\mathbf{u}_{t-1}^p \tag{6}$$

$$\Delta\mathbf{u}_{t-1}^e = \Delta\mathbf{u}_{t-2}^e + (\mathbf{u}_{t-1}^e - \mathbf{u}_{t-2}^e), \tag{7}$$

where $0 \leq \alpha \leq 1$ controls the rate at which new information replaces previous information. The recursive form of this equation implies that this estimate is a weighted sum of previous estimates.

Given a prediction of the new velocity of a patch $\mathbf{u}_t^p = (u_t^p, v_t^p)$, the temporal constraint is formulated as,

$$E_T(u,v,t) = \phi_T(u - u_t^p) + \phi_T(v - v_t^p), \tag{8}$$

where ϕ_T is the same function used in the smoothness error term, with a possibly different Δ_T.

Implementing the constraint requires maintaining a correspondence between sites and moving patches of the environment. The obvious solution is to use the estimated motion field itself to determine the correspondence of points over time. This amounts to tracking points over a sequence of images. The details of a tracking scheme are described later in the paper.

3 Recovering the Flow Field

The constraints of the previous section, together with the weak continuity assumption, embody our assumptions about the world. The three constraints can now be combined to form

an objective function:

$$H(u,v,t) = \beta_D E_D(u,v) + \beta_S E_S(u,v) + \beta_T E_T(u,v,t), \tag{9}$$

where the $\beta_\star$ are constant weights which control the relative importance of the constraints. Based on our assumptions, the best interpretation of the motion, (u,v), is the minimum of this function.

The more realistic formulation of the objective function used here means that H has many local minima making the task of finding the (u,v) which minimize the function more difficult. As mentioned earlier, the definition of the constraints in terms of local neighborhoods on a grid allows the problem to be formalized in terms of Markov Random Fields.

Each site in the MRF can be thought of as representing a small environmental surface patch. Associated with each site s is a continuous random vector (u_s, v_s) which represents the current image displacement of the corresponding surface patch. The discrete *state space* $\Lambda_s(t)$ defines the possible values that the random vector can take on at a given time t. The space Λ_s will be defined formally below; for now we simply note that each sight s has its own individual state space; this will prove important for minimizing the continuous problem.

For each site, we construct a probability density function Π defined over the range of possible displacements Λ using a *Gibbs distribution* as follows:

$$\Pi(u,v,t) = Z^{-1} e^{-H(u,v,t)/T(t)}, \text{ where: } Z = \sum_{(u,v)\in\Lambda(t)} e^{-H(u,v,t)/T(t)} \tag{10}$$

where t is the current time instance. The quantity $T(t)$ can be thought of as a *temperature* which serves to sharpen (or flatten) the distribution.

Standard *simulated annealing* techniques (in this case a *Gibbs Sampler* [13, 23]) can be used to find the minimum (u,v) by sampling from Λ according to the distribution Π with logarithmicly decreasing temperatures. As the temperature is lowered, the probability distribution of Π becomes concentrated about the minimum while the stochastic nature of the process prevents it from getting trapped in local minima. The result is that at high temperatures the sampling process freely chooses among the possible displacements, but

as the temperature is lowered, the minimum is chosen with increasing probability. Given an infinite amount of time and a logarithmic cooling schedule this process is guaranteed to converge to the correct solution. In practice, a sufficiently slow linear cooling schedule appears to provide acceptable convergence.

Notice that E_S is the only constraint which is dependent on its neighbors' motion estimates. While updating a site s, the estimates of its neighbors $t \in \eta_S(s)$ must be held fixed. By partitioning the sites using a checkerboard pattern, half the sites can be update at once while the other half remains unchanged. The current algorithm, which is implemented on the Connection Machine with a physical processor for each site, fully exploits this parallelism.

There are two main problems with this simulated annealing approach. First, the Monte Carlo techniques used to sample Π assume a discrete state space. If we want more than a discrete approximation to the image flow, then we need to be able to solve the continuous minimization problem for arbitrary fractional displacements.

The second problem is the computationally intensive nature of the simulated annealing algorithm. For reasonable results, hundreds or thousands of iterations of the annealing algorithm may be necessary to compute the flow between two images. This has a decidedly non-dynamic flavor. Ideally a motion algorithm should involve fast simple computations between a pair of frames, and exploit the fact that tremendous amounts of data are available over time.

The first problem can be solved by using a *continuous* variant of simulated annealing [23, 29]. The solution to the second problem is more radical. By tracking small patches of a scene over an image sequence, we will modify the basic annealing concept to work on changing data over time. The strict convergence results of simulated annealing will be lost, but the result is an incremental algorithm which produces good empirical results and meets many of the requirements of a truly dynamic motion algorithm.

3.1 Continuous Annealing and Sub-Pixel Displacements

To use simulated annealing with a Gibbs distribution we need to have a finite state space at any given time. The idea that allows us to solve continuous problems is that the state space can vary over time depending on the local properties of the function being minimized. At a given time t, we have an estimate of the motion $\mathbf{u}_t$, and consider making small changes $\Delta\mathbf{u}_t$ to the estimate. Vanderbilt and Louie[29] define a method which is *adaptive* in nature in that the state space (defined by the step size, $\Delta\mathbf{u}_t$) automatically adapts to the local shape of the function being minimized.

The basic idea is to use the covariance matrix of a random walk to characterize the shape of the function. We set the state space so that it best explores the function by making the covariance matrix of the state space proportional to the covariance matrix of the random walk. Intuitively, if the variance along a particular search direction is large, then we want to increase the step size in that direction to get a coarse view of the function. When the true minimum has been chosen at a coarse level, the variance will shrink. To explore the minimum more finely, the area covered by the state space should shrink resulting in smaller step sizes.

At a given site and at a given time, the state space Λ is always a discrete 3×3 neighborhood of the current estimate, but the area covered by the neighborhood varies based on the current step size $\Delta\mathbf{u}_t = [\Delta u_t, \Delta v_t]$. Given a current estimate $\mathbf{u}_t = [u_t, v_t]$, at time t the state space Λ is defined as:

$$\Lambda = \{\mathbf{u} + \Delta\mathbf{u} \mid \Delta\mathbf{u} = \mathbf{Q} \cdot \mathbf{l},\ \mathbf{l} = [l_1, l_2]^T,\ l_1, l_2 \in \{-(3/2)^{\frac{1}{2}}, 0, (3/2)^{\frac{1}{2}}\}\}, \tag{11}$$

where $\mathbf{Q}$ is a 2×2 matrix which controls the step size. Elements of the state space are all examined with equal probability, so the choice of trial steps is governed by a uniform probability distribution $\mathbf{g}(\mathbf{l})$ which over $\{-(3/2)^{\frac{1}{2}}, 0, (3/2)^{\frac{1}{2}}\}$ has zero mean and unit variance.

Since the mean of Λ is $\mathbf{u}$, the covariance matrix $\mathbf{s}$, of the state space is simply:

$$s_{ij} = \sum_{\Delta\mathbf{u} \in \Lambda} \Delta\mathbf{u}_i \Delta\mathbf{u}_j \mathbf{g}(\mathbf{l}). \tag{12}$$

Vanderbilt and Louie [29] note that this can be expressed as:

$$\mathbf{s} = \mathbf{Q} \cdot \mathbf{Q}^T. \tag{13}$$

Hence we can generate a state space with any desired covariance matrix $\mathbf{s}$ by solving for $\mathbf{Q}$ using Cholesky decomposition and then using $\mathbf{Q}$ to generate the state space in equation 11.

As mentioned, the covariance matrix of the actual steps that would be taken in a random walk can be used as a measure of the local shape of the function. We want the covariance matrix of the state space to to be proportional this covariance matrix of a random walk. The actual step taken at a time t is determined by the probability distribution $\Pi(\mathbf{u}_t + \Delta\mathbf{u}_t)$ defined over the space of displacements. Using Π we can compute the mean μ at time t (note we drop t when it is constant across all terms):

$$\mu_i = \sum_{\mathbf{u}\in\Lambda} \Pi(\mathbf{u})\mathbf{u}_i. \tag{14}$$

The covariance matrix $\mathbf{S}$ of Π given the current step size is:

$$S_{ij} = \sum_{\mathbf{u}\in\Lambda} (\mathbf{u}_i - \mu_i)(\mathbf{u}_j - \mu_j)\Pi(\mathbf{u}). \tag{15}$$

We make the covariance matrix of the state space at time $t+1$ proportional to $\mathbf{S}^{(t)}$:

$$\mathbf{s}^{(t+1)} = \chi\mathbf{S}^{(t)}, \tag{16}$$

where χ is a scaling factor. Now solving $s^{(t+1)} = \mathbf{Q} \cdot \mathbf{Q}^T$ for $\mathbf{Q}$ gives the $\mathbf{Q}$ for determining the state space at the next time instant.

We now need to choose the scale factor for χ. Assume a step size $\Delta\mathbf{u}$ and imagine the case in which Π is uniform so $\mathbf{s}^{(t)} = \mathbf{S}^{(t)}$. No information is being gained with the current step size so we should increase it. If $\chi > 1$ then the step size will be increased by a factor of $\sqrt{\chi}$ on the next iteration. Over time, as the the algorithm settles into the true minimum, the variance will decrease. The result will be decreasing step sizes which allow the minimum to be explored more precisely.

To prevent the state space from growing or shrinking too rapidly, we control the rate at which new information from $\mathbf{S}$ overwrites the previous information:

$$\mathbf{s}^{(t+1)} = \alpha\chi\mathbf{S}^{(t)} + (1-\alpha)\mathbf{s}^{(t)},$$

where α can be viewed as a damping factor.

3.2 Incremental Minimization

The obvious disadvantage of using simulated annealing is that its computational expense is prohibitive. However, since we expect the changes in the images and in the scene to be gradual and predictable, the iterative minimization process can be extended over an image sequence. This will also allow the motion detection algorithm to exploit the wealth of information available over time to achieve greater sensitivity and robustness. Such a process, however, must ensure that the various properties estimated for an image patch are propagated along with the patch. This amounts to warping the grid of sites according to the motion estimate. This section describes our incremental minimization approach which includes the tracking of image patches and the propagation of their properties.

When a new image is acquired, the current motion estimate at a given site (representing a particular surface patch) is used as the starting point for the continuous annealing algorithm and to compute the predicted motion used in the temporal coherence constraint. The current temperature at that site is used as the initial temperature, which is then lowered according to the annealing schedule.

After a fixed (usually small) number of iterations of the annealing process, each site has a new motion estimate and temperature. The various properties of the associated surface are then propagated to the new site where the patch has moved. The propagation algorithm described below also detects occlusion and disocclusion boundaries.

Warping

For now, assume that all motions are less than a pixel (this assumption will be relaxed in the following section). Each site s first determines which of its neighbors moving towards it. This is done by examining its own motion and the motion of its eight immediate neighbors to identify those sites whose new location is estimated to be within a pixel of the site s. Let this set of neighbors be denoted as $\eta(s)$. New estimates of image properties at each site are obtained by a weighted interpolation of the properties stored at the sites belonging to this refined neighborhood. Examples of properties belonging to a site are its motion, temperature, and state space. Additional properties like image intensity or higher level information about surface membership may also be present.

The contribution of each neighboring site $t \in \eta(s)$ is weighted by two factors: the proximity of the new location of that site to the location of s, and the probability of the motion estimate at t. Heuristically, the distance factor serves as a kind of linear interpolation of the properties, while the second factor serves as a type of confidence measure. The contributions to s of all its neighbors are accumulated in this way, and the result is then normalized according to the distance and the confidence of the neighbors.

The following expression more precisely describes the interpolation process. Let ρ be a property of interest. Then the new estimate of $\rho(s)$ is:

$$\rho(s) = \frac{1}{w(s)} \sum_{t \in \eta(s)} p(t) d(s,t) \rho(t), \tag{17}$$

$$w(s) = \sum_{t \in \eta(s)} p(t) d(s,t) \tag{18}$$

where w is a normalizing term, $p(t)$ is the probability of the estimated motion vector ($\mathbf{u}(t)$) at t, and $d(s,t)$ is the distance between the projection of site t (according to its estimate motion) and the location of site s.

Occlusion and Disocclusion

The propagation algorithm outlined above can be made sensitive to the presence of occlusion and disocclusion around each site. To explain how this is done, observe that the normalizing factor w roughly measures the total flow into a site. In the absence of motion discontinuities this should be approximately unity. However, if occlusions are present within the neighborhood of a site, we may expect multiple sites to move towards it, thereby increasing the total in-flow. Similarly, if there is a disocclusion, we may expect the total flow to be less than unity.

The current version of our algorithm includes a simple implementation of the idea described above for occlusion/disocclusion detection. The net flow, which is measured by the quantity w is estimated and compared against two thresholds, one above and one below unity, in order to categorize a site as occlusion, disocclusion, or normal. This is obviously too simple to handle complex situations and may fail even in simple situations. For example, if there is significant divergence (or convergence) present within the neighborhood of a site, net flow will differ from unity, even if there are no motion discontinuities. Even when the motion around the site is a simple translation, the lack of high confidence motion estimates can lead to small estimates of the net flow.

In the current algorithm no special processing is done at occlusion sites, other than to simply indicate them as such. A more sophisticated approach would involve modifying the propagation scheme to take contributions from processors which correspond to the occluding surface. If this information were available from higher level processes as a property of the site, it could easily be incorporated.

On the other hand, a disoccluded site indicates a new patch of the environment which was previously hidden from view. For this new patch, there is no prior motion estimate, hence the annealing process should be initially uncommitted about the motion. This is achieved by initializing the site to have a high temperature. Note that even if false disocclusions are detected due to low confidence motion estimates (as explained above), increasing the

temperatures may still be useful to extend the search space at that site.

It should be clear that unlike standard annealing, our algorithm uses different temperature for the different sites and dynamically modifies the temperature according to the information available at a site. As a patch is tracked, its temperature will decrease over time. Hence, the temperatures of patches that have been tracked over many frames and whose motion is precisely known tend to be lower than those of more recently disoccluded (i.e., new) patches.

Convergence

Unlike simulated annealing, we have no theoretical convergence results for this new incremental minimization scheme in which we attempt to minimize a function which is changing over time but doing so in predictable ways. Empirical results indicate that the approach does in fact converge to the correct sub-pixel motion estimates. Obviously, the degree to which the constraints accurately reflect the physics of the world will affect both the convergence and the accuracy of the algorithm. The current model and the constraints used are first order approximations to the correct physical models, since the various continuity constraints are imposed on the image domain and ignore the three-dimensional structure of the scene. We expect, however, the framework presented here can be extended to incorporate more precise models of the scene and its geometry.

4 Spatio-Temporal Pyramid

The previous section described how small motions can be estimated over time. This section concerns itself with computing large motions. The most obvious way to estimate large motions is to expand the state space to be larger than 3×3 and increase the maximum allowed step size, but this results in a loss of efficiency and communication between distant sites. Additionally, correlating small patches of the world over large distances is unreliable. To achieve efficient and robust computation of large motions we adopt a multi-resolution

strategy.

The multi-resolution scheme developed here combines elements of the coarse-to-fine [2, 10, 21] and spatio-temporal filter [16] approaches. We reject a strict coarse-to-fine approach for two reasons. The first is its sequential nature. We favor the spatio-temporal filter approach which can be viewed as layers of detectors, tuned to certain spatio-temporal frequencies, with all the detectors operating in parallel. The second problem with the coarse-to-fine strategy is that the computation is no longer local. Non-zero displacements at the low spatial frequencies will result large displacements in the high spatial frequencies. Refining the estimates at the high spatial frequencies will involve communication with distant processors. This violates our goal of simple local computation.

We start by constructing a pyramid of spatially filtered and subsampled images so that at the highest level in the pyramid the largest motion is less than a pixel. Each level of the pyramid can be thought of as a Markov Random Field which is responsible for estimating motions of one pixel or less. Since the maximum detectable displacement within each level is one pixel or less, the continuous annealing process described in the previous section can be applied at each level. The annealing process is applied to each level in parallel so that each level estimates its motion simultaneously and independently based on the previous motion field, spatial coherence, and the data error.

To derive a global motion estimate, the motion estimates from each level are combined so that the large motions, detected at the low spatial frequencies, dominate. This increases robustness in the presence of noise and may have some biological justification based on experiments in *motion capture*. This is essentially a coarse-to-fine strategy without refinement, in which large motions are determined solely at the lower spatial frequencies. Since they are not refined at higher frequencies, large motions are known with less absolute precision than smaller motions. The relative precision however, will be the same across levels.

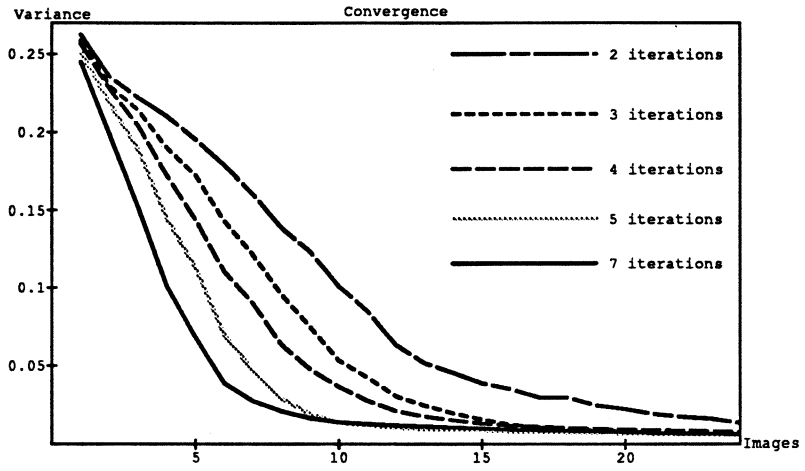


Figure 4: **Convergence Experiments.** Variance as a function of the number of frames in a 25 image sequence. The results are plotted for tests involving varying numbers of iterations of the annealing algorithm per frame (from 2 to 7).

5 Experimental Results

The incremental algorithm has been tested on real and synthetic image sequences. Experiments with controlled synthetic data illustrate the performance of the algorithm. The real image sequences, demonstrate the algorithm's ability to achieve qualitatively good motion estimates in the presence of noise. Without ground truth, no quantitative analysis of the real motion sequences is possible.

5.1 Synthetic Motion Experiments

While no theoretical proof of convergence exists, in practice the ISM algorithm converges to the correct solution even in the presence of noise. To illustrate the convergence properties of the algorithm a synthetic image sequence was generated. The sequence consists of a 64×64 pixel uniform random signal over the range $[0, 255]$ which is undergoing a uniform translation of one half pixel to the right and down per frame.

The initial experiments consider a noiseless signal and examine the convergence of the algorithm over time. Error is computed as the variance of the motion estimate in a region

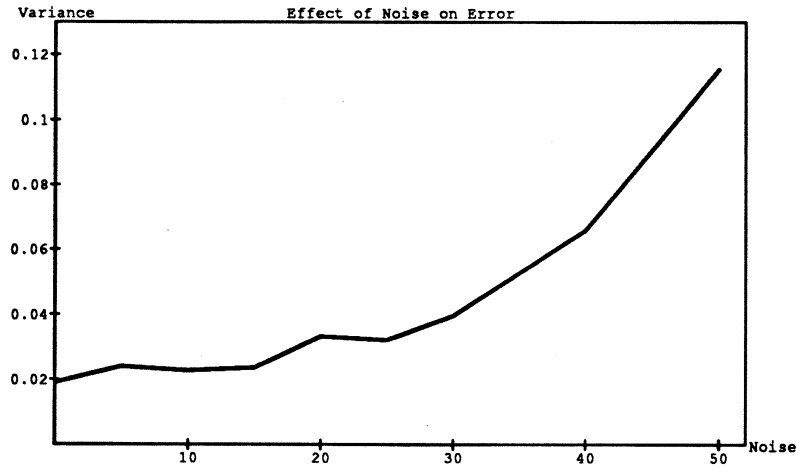


Figure 5: **Noise Experiments.** Variance as a function of noise (from 0% to 50%) is plotted plotted for a 10 frame sequence with 5 iterations per frame.

of the image which is visible for the entire sequence. Error in recently disoccluded regions will be initially higher. Figure 4 plots variance of the motion estimate as a function of the number of images examined in the sequence. The variance is plotted for trials using 2, 3, 4, 5 and 7 iterations per frame. Even with only three iterations per frame the algorithm converges to the correct solution within approximately 25 frames.

The next experiment addresses the effect of noise on the convergence of the algorithm. Uniform random noise over the range $[-\gamma/2, \gamma/2]$ was added to each image in the sequence, where γ is a percentage of the total intensity range. Figure 5 shows the effect of zero to 50 percent noise on the variance of the motion estimate. The experiment is performed with only a 10 frame image sequence with five iterations per frame. The results indicate the the algorithm is tolerant to fairly large amounts of noise (up to about 30%). Above that, longer sequences or more iterations per frame would be required to reach acceptable levels of error.

5.2 Motion Discontinuities

The following experiment involves an image sequence consisting of eight 64×64 square images; the last image in the sequence is shown in figure 6*a*. The images contain a soda can

in the foreground; the motion of which is is slightly less than one pixel to the left between each frame. The can is moving in front of a textured background which is also undergoing a slight motion to the left; there is no vertical motion.

Since all the motion is less than a pixel, this sequence tests the sub-pixel accuracy of the algorithm independently of the multi-resolution strategy. The flow field, computed to sub-pixel accuracy, is shown in figure 6*b*. The actual horizontal and vertical components of the flow field are shown in figures 6*c* and 6*d* respectively. The images can be interpretated roughly as follows: gray areas correspond zero motion, dark areas to leftward or upward motion and bright areas to rightward or downward motion. Notice that over-smoothing does not take place and flow discontinuities are maintained. Also notice that the errors in the vertical motion estimate correspond to areas of low image contrast. A longer image sequence or more iterations per frame would likely reduce the errors.

Occlusion and disocclusion boundary estimates are shown if figure 6*e*. The brighter the area, the more likely it is to be an occlusion boundary. Similarly, dark areas indicate disocclusion. It is important to remember, that while these results show only the final frames in the image sequence, both flow and discontinuity estimates are available at all times.

5.3 Nap-Of-the-Earth Experiment

The final experiment tests the full algorithm, including the multi-resolution strategy. The test sequence consists of 100 images of size 128 × 128 pixels. The images were acquired from a camera mounted on a helicopter in *Nap-Of-the-Earth* (NOE) flight. The sequence is challenging in many respects. First the range of motion in the images is wide; from 0 to approximately 4 pixels. To cope with motions of up to 4 pixels, a three level pyramid was used. Second, there are areas in the images of low contrast where good data estimates are not available. Finally, the motion is complex and changing; there is pitch, yaw and rotation in addition to translation. The actual motion is corrupted by jitter introduced by the camera mounting and turbulence.

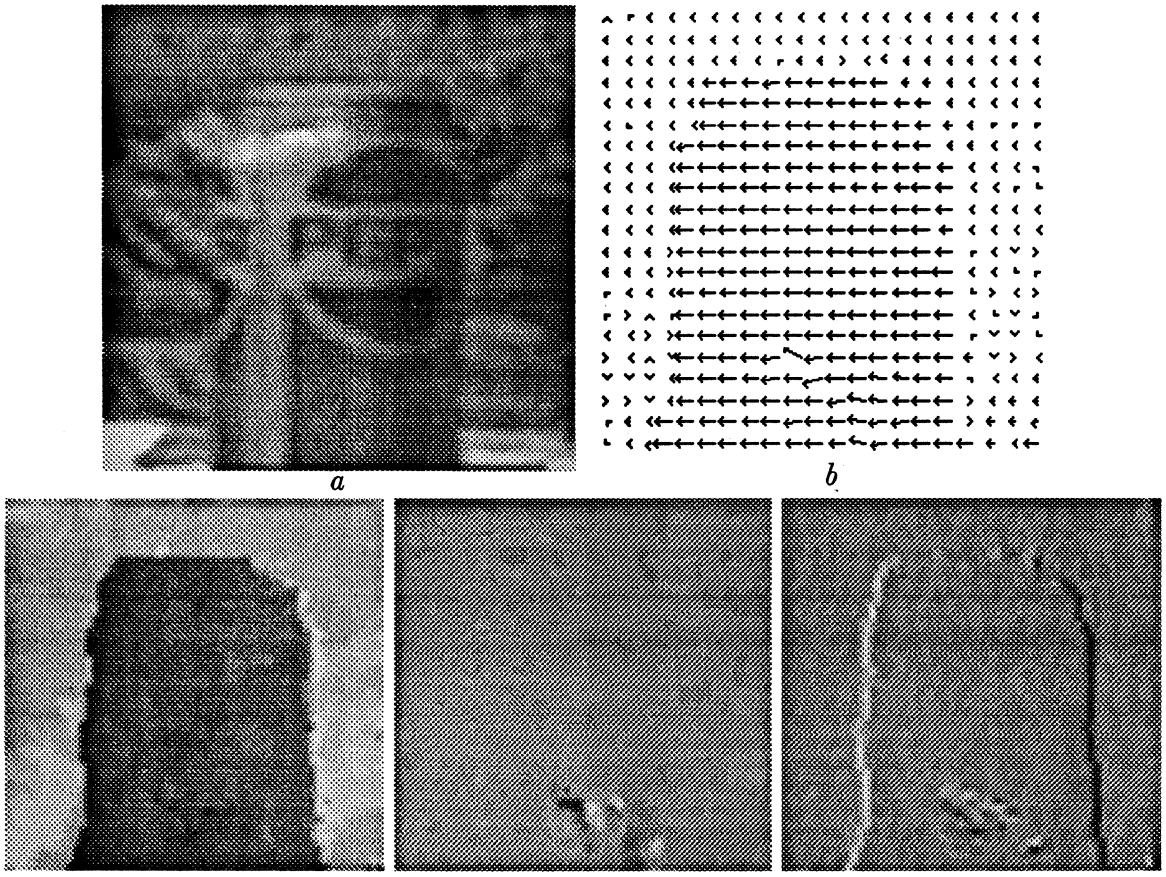


Figure 6: Pepsi can image sequence (results after eight frames): *a)* Intensity image; *b)* Flow field; *c)* Horizontal Displacement; *d)* Vertical Displacement; *e)* Occlusion/Disocclusion Boundaries.

Unfortunately, it is impossible to convey the dynamic behavior of the algorithm over the 100 image sequence in a static format for presentation here. Figure 7 shows six snapshots of the processing after 15, 30, 45, 60, 75 and 90 frames. The data conservation constraint used a 9×9 window with band-pass filtered images. Seven iterations of the annealing algorithm were used per frame with a linear cooling schedule. The various parameters mentioned previously were set as follows: $\Delta_D = 15.0$, $\Delta_S = 0.25$, $\Delta_T = 0.5$, $\beta_D = 2.0$, $\beta_S = 2.5$, $\beta_T = 1.0$.

Even after only 15 frames, noise in the motion estimate is small. In figures $7a,b$ a rotation to the right, in addition to the translation, can be seen. Figures $7c,d$ span a largely translational sequence. Throughout this portion of the sequence however, the aircraft is undergoing significant pitching fore and aft. Despite some additional noise due to the pitching

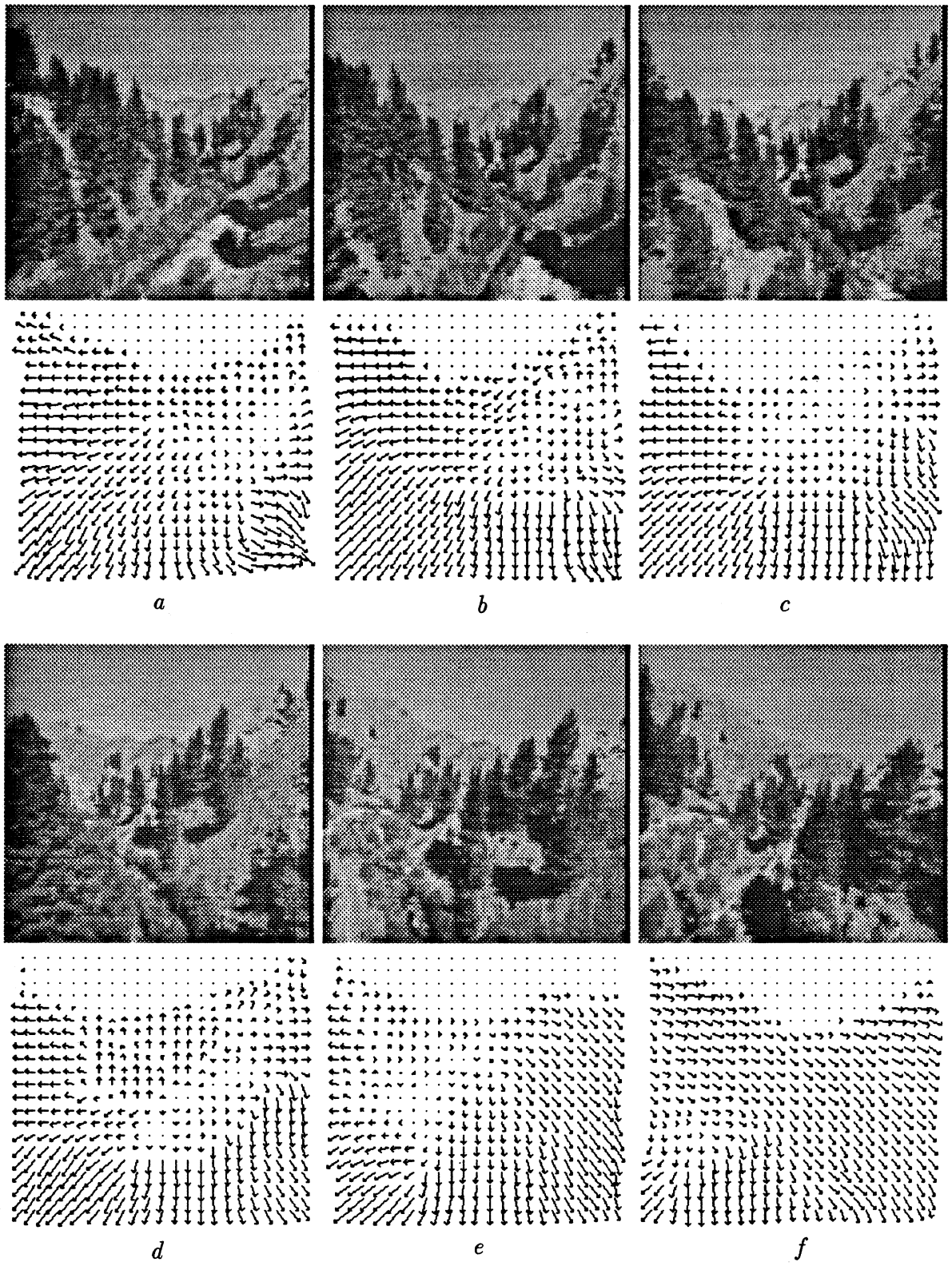


Figure 7: **Nap-Of-the-Earth Helicopter Sequence.** Snapshots of a 100 image sequence are shown with the current image above the motion estimate: *a)* after 15 images, *b)* after 30 images, *c)* after 45 images, *d)* after 60 images, *e)* after 75 images, *e)* after 90 images.

motion, the temporal coherence constraint helps maintain a reasonable motion estimate. In the final portion of the image sequence (*7e,f*) the helicopter is banking while rotating to the left.

6 Conclusion

This paper has presented a novel approach to incrementally computing motion estimates over a sequence of images. The starting point for the approach is the realistic formulation of constraints on image motion. This means taking into account the possibility of multiple motions using the notions of weak continuity and robust statistics. The resulting minimization problem is difficult to solve and traditional stochastic techniques are inappropriate for motion processing. To ameliorate these problems an incremental annealing algorithm is developed.

The approach has a number of advantages over previous approaches. The incremental and adaptive nature of the scheme makes it appropriate for dynamic motion processing. In particular, the local nature of the computations makes it possible to exploit the high degree of parallelism inherent in the problem using a simple array structured architecture. Additionally, the warping process allows the detection of occlusion and disocclusion boundaries.

Our current research is extending this scheme in a number of directions. First we are exploring new formulations of the temporal minimization problem. In particular we are examining new ways of formulating the temporal coherence constraint which would allow the warping process to be formalized in terms of the objective function. In conjunction with this, we are exploring new ways of performing adaptive stochastic minimization which are more appropriate for cases of non–uniform motion for which a strict cooling schedule is not appropriate.

Additionally, we are considering other possible constraints, for example rigid body motion, which could be brought to bear on the problem. In the context of formulating constraints, there is a great avenue of exploration in the use of robust statistics for dealing with

multiple motions.

Finally, it should be noted that the usefulness of the model extends beyond motion estimation. A model used to compute motion incrementally may also be exploited to incrementally compute traditionally non–motion related image properties. The framework for tracking surface patches over time may permit the extension of traditional two frame algorithms to a sequence of frames.

7 Acknowledgements

This work was paritally supported by the Defense Advanced Research Products Agency, contract number DAAA15–87–K–0001, administered by the Ballistic Research Laboratory. Portions of this work were performed while the first author was at the NASA Ames Research Center, Aerospace Human Factors Research Division, with the support of NASA RTOP 506-47.

References

[1] Anandan, P., "Measuring visual motion from image sequences," *Ph.D. dissertation*, COINS TR 87-21, University of Massachusetts, Amherst, MA, 1987.

[2] Anandan, P., "A computational framework and an algorithm for the measurement of visual motion," *Int. Journal of Computer Vision*, 2, 1989, pp. 283–310.

[3] Besl, P.J., Birch, J.B., Watson, L.T., "Robust window operators," *Proc. Int. Conf. on Comp. Vision*, ICCV-88, 1988, pp. 591–600.

[4] Bergen, J. R., Burt, P. J., Hingorani, R., Peleg, S., "Multiple component image motion: Motion estimation," David Sarnoff Research Center, unpublished report, Jan. 1990.

[5] Black, M.J., "A model for the incremental estimation of discontinuous flow fields," NASA Ames Research Center, Technical Memorandum, Moffett Field, CA, August 1990.

[6] Black, M.J., and Anandan, P., "A model for the detection of motion over time," to appear, *Proc. Int. Conf. on Comp. Vision*, ICCV-90, Osaka, Japan, Dec. 1990.

[7] Black, M.J., and Anandan, P., "Constraints for the early detection of discontinuity from motion," *Proceedings AAAI-90*, Boston, MA, 1990, pp. 1060–1066.

[8] Blake, A. and Zisserman, A., *Visual Reconstruction,* The MIT Press, Cambridge, Massachusetts, 1987.

[9] Bolles, R.C., Baker, H.H., and Marimont, D.H., "Epipolar-plane image analysis: An approach to determining structure from motion," *Internation Journal of Computer Vision*, Vol. 1, No. 1, 1987, pp. 7–57.

[10] Burt P. J., Yen C. and Xu X., "Local correlation measures for motion analysis: A comparative study," *IEEE Proc. PRIP*, 269-274, 1982.

[11] Geman, D., Geman, S., Graffigne, C., and Dong, P., "Boundary detection by constrained optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 12, No. 7, July 1990, pp. 609–628.

[12] Geman, D. and Reynolds, G., "Constrained restoration and the recovery of discontinuities," unpublished manuscript.

[13] Geman, S. and Geman, D., "Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-6, No. 6, November 1984.

[14] Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.,A, *Robust Statistics: The Approach Based on Influence Functions,* John Wiley and Sons, New York, 1986.

[15] Harris, J.G., Koch, C., Staats, E., and Luo, J., "Analog hardware for detecting discontinuities in early vision," *Int. Journal of Comp. Vision,* IJCV, Vol. 4, No. 3, June 1990, pp. 211-223.

[16] Heeger, D. J., "Model for the extraction of image flow," *J. Opt. Soc. Am,* vol. 4, no. 8, August 1987, pp. 1455–1471.

[17] Horn, B.K.P., and Weldon, E.J., "Direct methods for recovering motion," *Int. Journal of Computer Vision,* Vol. 2, No. 1, June, 1988, pp. 51–76.

[18] Horn, B.K.P., and Schunck B.G., "Determining optical flow," *Artificial Intelligence*, vol. 17, 1981, pp. 185-203.

[19] Huber, P. J., *Robust Statistics,* John Wiley and Sons, New York, 1981.

[20] Koch, C., Luo, J., Mead, C., "Computing motion using analog and binary resistive networks," *IEEE, Computer*, March, 1988, pp. 52–63.

[21] Konrad, J., "Bayesian estimation of motion fields from image sequences," *Ph.D. Dissertation,* McGill University, Montreal, Canadan, June 1989.

[22] Konrad, J., and Dubois, E., "Miltigrid Bayesian estimation of image motion fields using stochastic relaxation," *Int. Conf. on Computer Vision,* ICCV-88, pp. 354–362, 1988.

[23] Laarhoven, P. J. M., and Aarts, E. H. L., *Simulated Annealing: Theory and Applications,* D. Reidel Pub. Co., Dordrecht, Holland, 1988.

[24] Lucas B. D., and Kanade T., "An iterative image registration technique with an application to stereo vision," *Proc. 7th IJCAI*, Vancouver, B. C., Canada, pp. 674-679, 1981.

[25] Matthias,L, Szeliski, R., Kanade, T., "Kalman filter-based algorithms for estimating depth from image sequences," Carnegie Mellon University, CMU-CS-87-185, 1987.

[26] Morraquin J., Mitter S., and Poggio T., "Probabilistic solution for ill-posed problems in computational vision," *Proc. DARPA IU Workshop*, Miami Beach, Fl., pp. 293-309, 1986.

[27] Nagel H. H., "Displacement vectors derived from second order intensity variations in image sequences," *CVGIP*, vol. 21, pp. 85-117, 1983.

[28] Terzopoulos, D., "Regularization of inverse visual problems involving discontinuities," *IEEE PAMI*, Vol. PAMI-8, No. 4, July 1986, pp. 413–424.

[29] Vanderbilt D., and Louie S. G., "A monte carlo simulated annealing approach to optimization over continuous variables," *J. of Comp. Physics,* **56**, pp. 259–271, 1984.