Computation of Optic Flow by Multilevel Relaxation

Frank C. Glazer

COINS Technical Report 87-64

Computer and Information Science Dept.
University of Massachusetts
Amherst, Massachusetts 01003

Abstract

Multilevel relaxation algorithms for the computation of optic flow are developed and experiments show the expected increased convergence rate over single level relaxation, although some experiments present a problem of divergence at coarse levels. A *local mode analysis* of the relaxation equations shows that convergence is at least as fast as simple smoothing, and that, with strong gradients, convergence is accelerated towards the constraint line. The local mode analysis does not account for coarse level divergence. Divergence is then shown to be due to spatial variation in the image data. Fixed up/down cycling schemes are used to overcome the divergence problem.

Research supported in part by:
National Science Foundation grants MCS75-16098 A01 and MCS79-18209
DARPA grant N00014-82-K-0464
University of Massachusetts under a University Fellowship

Contents

1 Introduction

In a dynamic imaging situation, the motion of objects, viewers, or light sources induce the corresponding motions of their projections on the imaging plane. **Optic flow** is the apparent motion of image features in the imaging plane and can be represented as a vector field in that plane. This field specifies the instantaneous velocity of the corresponding image component at points in the image plane. In a vision system, the flow field provides a source of information about the structure and motions of objects in the viewed scene.

One major class of algorithms for computing optic flow is composed of **gradient-based** techniques which use the information contained in spatial and temporal derivatives of the image function $F(x, y, t)$ [Fennema & Thompson 79, Horn & Schunck 81, Glazer 81, Haralick & Lee 83, Nagel 83]. These techniques are most often characterized by a two stage approach: first, the image gradients at individual image locations are used to estimate the component of image motion parallel to the spatial gradient at that point; then, constraints from multiple locations are combined, giving the complete flow vectors.

Horn and Schunck's method of computing of flow from local flow constraints involves the formulation of a variational problem in which the solution optimally satisfies the gradient-based constraints in conjunction with a smoothness constraint [Horn & Schunck 81]. Many other problems in low-level computer vision can be formulated as variational problems and equivalently as partial differential equations [Glazer 82, Terzopolous 84, Poggio *et al.* 85]. These problems are characterized by (1) variational formulations of global measures of constraint satisfaction; (2) discrete formulations of the continuous problem, e.g., finite difference approximations; and (3) relaxation algorithms involving the iterative application of local neighborhood operators. An important feature of such formulations is that they can be solved by discrete algorithms which are uniform, local, and parallel when implemented on 2D grids of simple locally connected processors.

A major limitation of variational methods is the large number of iterations that may be necessary in the relaxation process. Relaxation is used as a local update process that ultimately enforces a particular global optimization. Many iterations may be needed due to slow propagation of global information by the local update processes. Multilevel relaxation techniques, introduced in the general field of numerical analysis, overcome this problem of asymptotically slow convergence [Brandt 77a, Brandt 77b, McCormick & Trottenberg 83]. Operating on a set of grids of varying spatial resolution, these techniques essentially solve for coarse components (low frequencies) of the solution at coarse grid levels and fine components (high frequencies) at fine levels. In general, low-level vision algorithms based on variational formulations with relaxation processes are amenable to multilevel methods [Glazer 82, Terzopolous 84]. In particular, multilevel methods have been applied to the computation

of optic flow in variational schemes [Glazer 82, Enkelmann 86, Terzopolous 86].

Just as single level relaxation algorithms fit naturally and efficiently into 2D cellular architectures, in a corresponding way, multilevel relaxation algorithms are well suited to hierarchcial multigrid processing architectures. Specifically the *processing cone* architecture [Hanson & Riseman 74, Hanson & Riseman 80] provides the interlevel and intralevel interconnections needed.

The computational cost of using relaxation to compute optic flow is very dependent on the number of iterations needed. This number is in turn dependent on the image data, a fact which has not been noted in the cited prior work. We will demonstrate how the lack of fine details in an image necessitates more iterations to compute optic flow. In this paper we show how a multigrid relaxation scheme can be used to reduce this cost and make it relatively independent of the image data.

In Section 2, we review the gradient-based approach to optic flow computation including a basic variational method used to compute optic flow. In Section 3, multilevel relaxation is reviewed. The remainder of this paper is then devoted to the application of multilevel relaxation to the computation of optic flow. The multilevel optic flow relaxation algorithm is presented in Section 4. In Section 5, the single-level and multilevel algorithms are compared. Some multilevel cases are seen to diverge at coarse levels. The next two sections contain a further analysis of the multilevel optic flow equations. In Section 6, a *local mode analysis* is performed to provide insight into the convergence properties of multilevel optic flow relaxation. In Section 7, the relationship between spatial variation in the image data and convergence/divergence of relaxation is shown. Finally in Section 8, a fixed cycling scheme is employed to successfully avoid the problem of divergent relaxation.

2 Gradient-Based Optic Flow Computation

Gradient-based techniques are used to compute optic flow from image sequences using the structural information contained in spatial and temporal derivatives of the image "function" $F(x, y, t)$. Various formulations for this exist, including: (a) looking for "edges" in the three-dimensional xyt space [Glazer 81, Haralick & Lee 83], (b) computing gradients of the image in the three-dimensional space [Fennema & Thompson 79, Horn & Schunck 81], and (c) solving a variational problem in which the disparity field is chosen to minimize some measure of dissimilarity [Nagel 83]. We will use a *first order* gradient-based technique in which only first order derivatives of the image function are used. Various developments of the equations exist (see for example [Fennema & Thompson 79, Horn & Schunck 81] or [Glazer 87, Section III.4.3]) all of which lead to a formulation of the problem in which the optic flow (u, v) at a point in the image plane is related to the first partial derivatives of

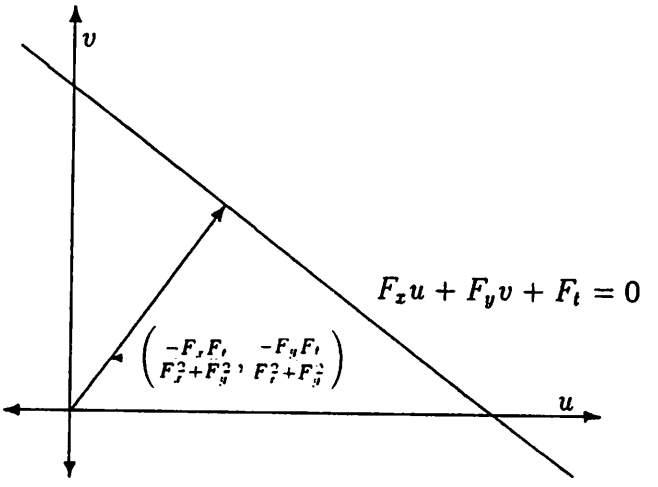

Figure 1: The velocity constraint line

The velocity constraint line is shown in uv "velocity space". It is the locus of velocities (u, v) satisfying the equation $F_x u + F_y v + F_t = 0$. The corresponding edge flow vector is also shown along with the equations for its components.

the image function F as follows:

$$F_x u + F_y v + F_t = \frac{dF}{dt} = 0 \tag{1}$$

Generally speaking, the total derivative dF/dt is defined with respect to some path $(x(t), y(t), t)$ through xyt space. If this path is defined to be successive instances of a moving image feature, then the derivatives $u = dx/dt$ and $v = dy/dt$ represent the optic flow. Setting the total derivative to 0 represents the assumption that the image value at the moving point maintains a constant intensity.

Equation 1 specifies a line in u, v velocity space called the **velocity constraint line** (see Figure 1). This line is perpendicular to the spatial gradient (F_x, F_y). This line can be represented by the point on it which lies closest to the origin (in velocity space):

$$\left(\frac{-F_x F_t}{F_x^2 + F_y^2}, \frac{-F_y F_t}{F_x^2 + F_y^2}\right) \tag{2}$$

This point defines the **edge flow vector**. It is perpendicular to the constraint line and has a length equal to the distance between the line and the origin. Figure 1 shows a velocity constraint line and the corresponding edge flow vector. For any (u, v) on the velocity constraint line, this edge flow is its component parallel to the spatial gradient (perpendicular to an edge).

Velocity constraint lines—or the equivalent edge flow vectors—can be computed at all points in the image that have non-zero gradient. However, due to the local ambiguity of image motion, they do not determine an optic flow field. Other information must be brought to bear to constrain further our choice of a flow field.

Horn and Schunck used a variational principle to accomplish the computation of optic flow from edge flow. They included two constraints: minimum deviation from the constraint line; and smoothness of the flow field. The resulting variational problem is to find the optic flow field $(u, v) = (u(x, y), v(x, y))$ that minimizes the following functional

$$\iint (F_x u + F_y v + F_t)^2 + \alpha^2(|\nabla u|^2 + |\nabla v|^2)\, dxdy \tag{3}$$

The first term in the functional is the square of the rate of change of image brightness as expressed in Equation 1. The second term is a roughness measure of u and v, where ∇ is the gradient operator. α is a relative weighting factor between the two constraints. The equivalent PDE system (Euler's equations, [Courant & Hilbert 53, Section IV.3.4]) is

$$\alpha^2 \Delta u - F_x^2 u - F_x F_y v = F_x F_t \tag{4a}$$

$$\alpha^2 \Delta v - F_x F_y u - F_y^2 v = F_y F_t \tag{4b}$$

These equations can be represented on a discrete x-y grid using finite difference approximations for the partial derivatives. A set of equations results, one pair per grid point, in which the variables are the components (u, v) of the optic flow vector at each grid point. These equations can be solved using relaxation algorithms that are local—only neighboring grid points combine data values; uniform—the computation is the same at each grid point; and parallel—computations can be performed simultaneously at all grid points. Discrete representation and relaxation equations for Equation 4 are given in Section 4.

The derivation of Equation 1 involves the assumption that local image structure can be adequately approximated by a first order equation. There are cases in which this assumption does not hold, namely, when the image contains high frequency detail or when the optic flow has high magnitude. In these cases, the computed optic flow is not correct. A hierarchical method has been developed which overcomes these limitations [Glazer 87, Chapter V] while retaining the simplicity of first order approximations. It involves a coarse-to-fine control strategy in which early approximate estimates of flow computed at coarse grid levels are later subsequently refined at successively finer grid levels. This extension of single level optic flow computation algorithms is independent of the use of multilevel algorithms to improve the efficiency of relaxation described in this paper.

3 Multilevel Relaxation

Multilevel relaxation is an algorithmic extension of iterative relaxation. By representing the spatial domain at multiple levels of resolution these algorithms apply the basic local iterative update to a range of neighborhood sizes. Local updates on coarser grids introduce a more global propagation of information.

The variational approach to various problems in low-level computer vision leads to iterative relaxation algorithms. Unfortunately the number of iterations necessary for convergence can be very high—on the order of $O(d^n)$, where d is the distance (in nodes) that information has to travel and n is the order of the PDE's being solved [Brandt 77b, p.281]. For example, Grimson's surface interpolation algorithm [Grimson 81], requiring second order smoothness, takes thousands of iterations to approach final solutions. (This number is based on our unreported experiments. See also [Terzopolous 82].) In the problem domain of elliptic PDE's this slowness has been overcome by using multilevel relaxation algorithms [Brandt 77a, Brandt 77b]. In this approach, a standard iterative algorithm is applied on grids of different resolution. Each grid covers the entire data domain and they are organized in a hierarchical structure. At each level the problem is solved in a different spatial bandwidth. Thus, the various processing levels cooperate to compute the final result, which is represented at the highest resolution level. The number of iterations required is of order $O(d)$ [Brandt 77b, p.278].

3.1 The Multilevel Method

The slow convergence of relaxation algorithms is due to the fact that solutions which must satisfy a global condition (the variational problem) are arrived at by the local propagation of information. The greater the range over which information must be sent, the slower the smoothing will be.

A longstanding technique used to overcome this problem is to initialize relaxation with a coarsely computed estimate of the solution. Coarse approximations can be projected down to a finer level to serve as the initial estimate for relaxation at the fine level. Such an approach can be applied at multiple levels of resolution in a coarse-to-fine strategy.

While significantly speeding up the early stages of relaxation, ultimate convergence is still slow. Projection of coarse solutions for use as initial fine estimates does provide a "head start" over single-level methods. However, after a few iterations of relaxation at the finer level, error components remain that are attenuated only very slowly in subsequent iterations. Brandt [Brandt 77a, Brandt 77b] significantly extended the multilevel approach by showing how the coarser grid can be used in the process of improving the first approximation. His algorithms attack the problem of error removal by creating coarse

level approximate problems which can be solved more quickly on coarser grids. Brandt combined the techniques of projecting estimates and coarsening "residual equations" to develop iterative schemes that move up and down a set of varying resolution grids.

The extent to which one pass of relaxation computes a new approximation closer to the actual solution is dependent on the spatial frequency content of the error. Sharp (high frequency) changes or errors are quickly smoothed. This is due to the local nature of the smoothing. It is the slow (low frequency) error components that resist elimination. These ideas can be formalized in a *local mode analysis* of the particular update equation [Brandt 77a, Brandt 77b] whereby the error reduction is considered as a function of spatial frequency.

After a few iterations on a given grid, high frequency error is considerably reduced while low frequency error remains. At this point we can approximate the remaining problem on a coarser grid. The remaining error is a higher frequency on the coarse grid, hence further relaxation at this level can reduce it effectively. When convergence is attained at the coarse level, that solution can be interpolated back to the fine level. This interpolation introduces some high frequency error which is easily reduced by a few more iterations.

These processes easily generalize to multilevel cyclic algorithms. Approximate solutions are sent down to finer levels after they have converged. When convergence slows at finer levels they are sent up for coarser processing. The role of relaxation in such a system is not to completely reduce the error, but to smooth it out; that is, to reduce high frequency components of the error. Lower frequency components are reduced on coarser grids. What is essentially happening in such a system is that different grid levels solve the problem in different spatial frequency bands.

The basic operators that are involved in a multilevel relaxation scheme are:

relax : Reduce error components at a given level.

This step is defined by the discrete approximation to the PDE to be solved and the particular iterative update scheme chosen.

project : Send coarse solutions down.

The essential step here is the projection of corrections by interpolation from coarse to fine grids.

reduce : Send problems to be solved up.

Coarse equations whose solution is the correction at a fine level are created by this process.

3.2 Formal Development

3.2.1 Basic Method

In the basic multilevel method the domain of definition $\mathcal{A}$ of the problem is represented discretely by a set of uniform square grids $G^0, \ldots, G^k, \ldots, G^M$ with mesh-sizes $h_0, \ldots, h_k, \ldots, h_M$. Each grid covers the full domain and we assume that the interlevel mesh size ratio is $\rho \triangleq h_i/h_{i+1} = 2$. Brandt advises that this ratio always be used, since it provides close to optimal convergence rates, and the simplest intergrid architecture [Brandt 77a, p.353].

Suppose we have a PDE with suitable boundary conditions which we wish to solve over $\mathcal{A}$, such as $\Delta U(x, y) = 0$. We represent this as:

$$LU(\mathbf{x}) = F(\mathbf{x}) \qquad \text{for all } \mathbf{x} \in \mathcal{A} \tag{5}$$

where L is a general differential operator and U and F are vector valued functions over $\Re^n$. U is the solution we seek. At each level k we can form the finite difference approximation to the PDE's;

$$L^k U^k(\mathbf{x}^k) = F(\mathbf{x}^k) \qquad \text{for all } \mathbf{x}^k \in G^k \tag{6}$$

If L is a linear operator, then Equation 6 is a matrix equation relating values at neighboring grid points.

Following Brandt's development [Brandt 77a], let u^M be an approximate solution of the G^M problem with $L^M u^M = f^M$. The error is $U^M - u^M$ and we define the **residual** to be $r^M \triangleq F^M - f^M$. The residual is the difference between the desired and the actual right-hand-side. Then

$$L^M U^M - L^M u^M = F^M - f^M \triangleq r^M \tag{7}$$

If L is linear and if we let $V^M \triangleq U^M - u^M$, then V^M satisfies the **residual equation**:

$$L^M V^M = r^M \qquad (\triangleq F^M - f^M) \tag{8}$$

and the exact discrete solution is $U^M = u^M + V^M$. If u^M is computed by some relaxation iterations on G^M then r^M has little high frequency content (relative to the grid size h_M). This allows the residual equation to be accurately approximated on a coarser grid. The optimal time to perform this switch to a coarser grid occurs when the residual r^M is smoothed out and convergence has slowed down.

The coarsened residual equation is

$$L^k V^k = I_M^k r^M \tag{9}$$

where I_M^k is a reduction operator which computes a coarse (level k) version of r^M. Relaxation on the coarse grid produces an approximation v^k of the correction V^M. An improved level M solution is then obtained by projecting v^k to level M and adding this interpolated correction to u^M:

$$u^M \leftarrow u^M + I_k^M v^k \tag{10}$$

In the basic multilevel relaxation algorithm each v^k (the function defined on the grid G^k; $k = 0, \ldots, M-1$) serves as a correction for the approximate v^{k+1} previously obtained on the next finer grid G^{k+1}. The equation to be (approximately) satisfied by v^k is

$$L^k V^k = r^k \tag{11}$$

where r^k approximates the residual left by v^{k+1}, that is

$$r^k = I_{k+1}^k (r^{k+1} - L^{k+1} v^{k+1}) \tag{12}$$

The equation on G^k is thus defined in terms of the approximate solution on G^{k+1}. On the finest grid, the equation is the original one ($L^M V^M = r^M \leftarrow L^M U^M = F^M$) so we have:

$$r^M = F^M \qquad v^M = u^M \tag{13}$$

When a correction v^k is arrived at, it is projected down to the finer level and added to the current estimate v^{k+1}. The projection is:

$$v^{k+1} \leftarrow v^{k+1} + I_k^{k+1} v^k \tag{14}$$

3.2.2 Full Approximation Storage

In the basic method described in the previous section, an approximate solution only exists on the fine-level grid. All coarser levels deal only in correction surfaces which approximate solutions for changing residual equations. Brandt also developed FAS (Full Approximation Storage) algorithms in which each grid level stores the full current approximation [Brandt 77a, Brandt 77b]. *These functions, at varying levels of resolution, provide a hierarchy of descriptions of the solution.* This makes the FAS-type methods particularly appealing to problems in computer vision where structures of interest in the image can occur at many sizes. For this reason, we have chosen to work with FAS methods.

The approximation u^k at level k is the sum of the correction v^k and its base approximation u^{k+1};

$$u^k = I_{k+1}^k u^{k+1} + v^k \qquad k = 0, 1, \ldots, M-1 \tag{15}$$

We can rearrange the general residual equation (Equation 7), in which L may be nonlinear, to get $L^k U^k = L^k u^k + r^k$. If we set u^k in this equation to an initial estimate computed

from the finer level, i.e. $u^k_{init} = I^k_{k+1} u^{k+1}$ and use the expression for r^k in Equation 12, then substituting in we get

$$\begin{aligned} L^k U^k &= L^k u^k_{init} + r^k \\ &= L^k (I^k_{k+1} u^{k+1}) + I^k_{k+1} (r^{k+1} - L^{k+1} v^{k+1}) \end{aligned} \tag{16}$$

This is a new equation $L^k U^k = R^k$, with the right hand side $R^k = L^k u^k_{init} + r^k$. In this form R^k depends on r^k which in turn is computed from the residual r^{k+1} and correction v^{k+1} from level $k+1$. To complete the transformation to the FAS method, we must represent R^k in terms of R^{k+1} and u^{k+1} alone. This is done as follows. First, note that at level $k+1$ we have $u^{k+1} = u^{k+1}_{init} + v^{k+1}$ and $R^{k+1} = L^{k+1} u^{k+1}_{init} + r^{k+1}$. Apply L^{k+1} to the first equation and then subtract the result from the latter equation. This gives

$$R^{k+1} - L^{k+1} u^{k+1} = r^{k+1} - L^{k+1} v^{k+1} \tag{17}$$

Substituting Equation 17 into Equation 16, the correction equations can be rewritten as

$$L^k U^k = R^k \tag{18}$$

where

$$R^k = L^k (I^k_{k+1} u^{k+1}) + I^k_{k+1} (R^{k+1} - L^{k+1} u^{k+1}) \qquad k = 0, \ldots, M-1 \tag{19}$$

and

$$R^M = F^M \tag{20}$$

When a correction u^k is arrived at, it is projected down to the finer level, updating the current estimate u^{k+1}. The projection of the correction can be thought of as the addition of a correction term, i.e. $u^{k+1} \leftarrow u^{k+1} + I^{k+1}_k v^k$. This is done explicitly in the non-FAS algorithm. For the FAS algorithm v^k is given implicitly as $u^k - I^k_{k+1} u^{k+1}$ (see Equation 15). Thus, the projection is:

$$u^{k+1} \leftarrow u^{k+1} + I^{k+1}_k (u^k - I^k_{k+1} u^{k+1}) \tag{21}$$

$$\leftarrow I^{k+1}_k u^k + (I - I^{k+1}_k I^k_{k+1}) u^{k+1} \tag{22}$$

This is the FAS version of Equation 14. In the second form of the right hand side, the operator $I - I^{k+1}_k I^k_{k+1}$ is a high-pass filter. Thus, we see that *the projection updates u^{k+1} by replacing its low frequency component by a projection of u^k.*

For linear problems equations 11–13 are equivalent to 18–20. A key aspect of the FAS method is that the function stored on a coarse grid G^k approximates the fine grid solution in that $u^k = I^k_M u^M$. These functions at varying levels of resolution provide a hierarchy of descriptions of the solution.

$k \leftarrow M$; start at finest level
$R^k \leftarrow F^M$; initial right-hand-side
$v^k \leftarrow u^M$; initial solution estimate
Until u^M has converged **Do**
  $u^k \leftarrow \text{Relax}[L^k \cdot = R^k]u^k$: a relaxation "sweep"
  If u^k has converged **Then**
    If $k < M$ **Then**
      $k \leftarrow k + 1$; go down one level
      $u^k \leftarrow u^k + I^k_{k-1}(u^{k-1} - I^{k-1}_k u^k)$; add correction
  Else if convergence is slow **Then**
    If $k > 0$ **Then**
      $k \leftarrow k - 1$; go up one level
      $u^k \leftarrow I^k_{k+1} u^{k+1}$: initial estimate
      $R^k \leftarrow I^k_{k+1}(R^{k+1} - L^{k+1}u^{k+1}) + L^k u^k$: new right-hand-side

Figure 2: Full Approximation Storage, multilevel relaxation

The FAS algorithm is shown in pseudocode in Figure 2. In the next few sections, we further discuss the details in this algorithm, namely, projection and reduction of images between grid levels, selection of relaxation operator, and measures of convergence.

3.2.3 Interpolation: Projections and Reductions

The notation I^k_m represents the generic operation that transfers data from grid G^m to grid G^k. We will always transfer data between adjacent levels using suitably defined I^{k+1}_k and I^{k-1}_k. The operator I^{k+1}_k is a projection down one level. The values on the finer grid are interpolated from those on the coarser. The accuracy of a given interpolation algorithm can be measured by the **order of interpolation** which is a measure of how the error of interpolation is reduced as the mesh-size goes to zero ($h \rightarrow 0$). In particular nearest-neighbor interpolation, in which interpolated values are equal to the nearest known value, is of order $O(h^{-1})$. While linear interpolation is of order $O(h^{-2})$. Interpolation of order $O(h^{-n})$ is called **n-th order interpolation**.

The order of interpolation must be greater than or equal to the order of the differential equations [Brandt 77a, p.377]. The *order of a differential equation* is the highest degree of all differential operators in the equation. *The differential equations we are solving are all second order, hence we use bilinear projections.* In this case, the value at a fine pixel is computed from the values of the four nearest coarse pixels by bilinear interpolation.

3.2.4 Relaxation

If we are to approximate the residual equation on a coarse grid then the residual r^k must have little high frequency component. *Relaxation sweeps* are used to smooth the error component. The *Gauss-Seidel* iterative algorithm is a common example. In this algorithm, during one sweep the points $\mathbf{x}^k$ in G^k are scanned one by one in some fixed order. At each point the old value of $u^k(\mathbf{x}^k)$ is replaced by a new value which satisfies Equation 6. Having completed such a sweep, the equations are not yet solved since they are coupled together. Standard (non multilevel) iterative algorithms apply a long sequence of such relaxation sweeps. They can be very slow since relaxation has little effect on smooth errors which can have very small local errors (small residuals) relative to their own magnitude. However, in multilevel methods relaxation sweeps are only used to smooth the error. This can be done in only a few iterations.

The Gauss-Seidel algorithm is not parallel since grid points are updated in a sequential order. New values at a given point will in general depend on new values at some neighboring points. The **Jacobi method** involves similar update equations at each node, but they are applied at all grid points in parallel. This is the method we use because it is makes full use of cellular arrays of processors.

3.2.5 Convergence Measures

Convergence is measured using the **residual norm** — the Euclidean L_2 norm of the residual. It is defined as

$$\sqrt{\frac{1}{N}\sum(r^k - L^k u^k)^2}$$

where the sum is over all N grid points. (Note that it has the form of the *root-mean-square-error*.) The **rate of convergence** is measured as the ratio of consecutive residual norms from one iteration to the next. Convergence is "slow" when this ratio rises above a threshold parameter (0.6 in our experiments and in [Brandt 77a]). Convergence at the finest level is defined by a user supplied tolerance (threshold) below which the residual norm must fall. Convergence at an intermediate level is defined by a dynamic threshold. This coarse level "target" threshold is set, when we pass up a level, to a fraction of the current residual norm at the fine level. This fraction is another parameter in the algorithm (0.3 in our experiments and in [Brandt 77a]). Brandt claims robustness of such algorithms to the extent that variations of these two parameter settings produce little qualitative change in performance.

3.2.6 Fixed Cycling Schemes

The two parametrized decisions that are used in controlling the cycles are global computations, that is, they are computed from the current solution estimate over the entire grid. This is due to the fact that the decisions are based on the residual norm which is an integral (continuous case) or a summation (discrete case) over the entire domain. It will be seen later—as Brandt has pointed out—that for some problems *we can forego the computation of the residual norm and thus attain a purely local and parallel computation.* Moreover, for optic flow computation, such a control scheme will allow us to avoid divergence problems.

3.3 Normalized Coordinate Systems

We will use normalized coordinate systems at each level to make intralevel operators independent of the mesh size for the given level. The **base coordinate system** is determined by the finest level (L). The interpixel spacing at the finest level is 1 unit in the base coordinate system. The interpixel spacing at a coarser level k, when measured in the base system is $\rho^{L-k} = (h_{L-k}/h_{L-k+1}) \cdot \ldots (h_{L-2}/h_{L-1}) \cdot (h_{L-1}/h_L)$. In the **normalized coordinate system** at a given level, the distance between two adjacent pixels is 1 unit. Image operators which involve finite differences can be then be computed without using h_k terms, h_k being the distance between pixels. Intra-level computations and comparisons must take into account the difference in the grid spacings as given by the ratio $\rho \triangleq h_{k-1}/h_k$. This adjustment is described below.

If regular image data (intensity, brightness, etc.) is moved between grid levels, projected or reduced, then no adjustment for the normalized coordinate systems is necessary. However, if image gradient data is moved, then adjustments must be made. Whether such adjustments must be made and what they are is completely determined by the *order* of the quantity we are moving between levels. This is now defined.

Image operators that approximate spatial derivatives with finite differences involve the mesh size h_k. The order of the derivative corresponds to the power of h_k which appears in the denominator of the operator (in the non-normalized form). See, for example, the discrete Laplacian in Section 4.2.2 (Equation 26), in which the term $1/h_k^2$ appears. Now, consider the units we use to measure the quantities being manipulated. Let I represent the unit of image data and L the unit of length. Application of a differential operator results in a change of units. For example, if a first difference is applied to image data, then the resultant units are I/L. If the Laplacian is applied to the original data, then the resultant data is in units of I/L^2. The change in units is directly related to the order of the differential operator.

We define the **order** of a quantity as the negative of the exponent of its length units. We

also define the **order** of a finite difference operator as the order of the differential operator it approximates. As noted above this is equal to the exponent of h_k in the non-normalized form. When a finite difference operator is applied, the order of the result is equal to the sum of the order of the operator and the order of the operand.

When data of order n is reduced, an extra factor of ρ^n must be applied. Let u^k be measured in units of I/L^n and let $\tilde{u}^k$ represent the same quantity in the normalized coordinate system. Then the relationship between these two measures is $\tilde{u}^k = h_k^n u^k$ where h_k is the mesh-size (i.e., the length per pixel-spacing). If A is a local averaging reduction operator, then reduction of u^k would be given by $u^{k-1} \leftarrow Au^k$. If normalized coordinates are used at levels k and $k-1$, then by substitution $\tilde{u}^{k-1}/h_{k-1}^n \leftarrow A(\tilde{u}^k/h_k^n)$ and the reduction becomes $\tilde{u}^{k-1} \leftarrow (h_{k-1}/h_k)^n A\tilde{u}^k = \rho^n A\tilde{u}^k$.

Projections involve an adjustment in the opposite direction. When data of order n is projected, an extra factor of ρ^{-n} must be applied. The demonstration of this is analogous to that for reductions. These results are now summarized.

Reduction and Projection of Order n data

If data u of order n is represented in a pyramid using normalized coordinates at each level, then it is reduced using

$$u^{k-1} \leftarrow \rho^n A u^k \tag{23}$$

and it is projected using

$$u^{k+1} \leftarrow \frac{1}{\rho^n} P u^k \tag{24}$$

where A and P are (non-normalized) reduction and projection operators respectively and we have dropped the tildes ($\tilde{}$).

4 Multilevel Optic Flow Computation

4.1 An Optic Flow PDE

Let us represent the optic flow field as $(u, v) = (u(x, y), v(x, y))$ where u is the x-component of velocity and v is the y-component. Let the dynamic image be given as $F(x, y, t)$. In Section 2, Horn & Schunck's variational principle was described. It includes a first order smoothness constraint on u and v and the velocity line constraint. The resulting variational problem is to find the optic flow field satisfying Equation 3. The equivalent PDE system is given in Equation 4. We can represent this in the form $LU = F$ as follows:

$$\begin{bmatrix} \alpha^2\Delta - F_x^2 & F_x F_y \\ -F_x F_y & \alpha^2\Delta - F_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} F_x F_t \\ F_y F_t \end{bmatrix} \tag{25}$$

This elliptic system of PDE's generalizes Laplace's equation in that (1) it is a vector field equation and the component equations are coupled, (2) 0th order terms appear, and most significantly (3) the coefficients are non-constant (they depend on x and y).

4.2 Discrete Representation and Computation

4.2.1 Normalized Coordinate Systems

Because normalized coordinate systems are used at each grid level, the order of the quantities being manipulated must be noted. If we examine the equations in the previous section we see the following quantities: (1) optic flow estimates (u, v) of order -1 (since it is measured in units of length per time); (2) spatial derivatives F_x and F_y of order 1; and (3) a time derivative F_t of order 0. The L operator is of order 2, as given by each of its elements, namely, the Laplacian and the products of the spatial derivatives. The order of the equations themselves is 1 (L applied to (u, v)) and thus this is the order of the residual. In the course of multilevel processing, the optic flow estimates and the residuals will be transferred between levels. Reductions and projections must take their respective orders into account.

Optic flow vectors in normalized coordinates will be represented as $\tilde{\mathbf{u}} = (\tilde{u}, \tilde{v})$. They are related to the unnormalized form by $h_k\tilde{\mathbf{u}} = (h_k\tilde{u}, h_k\tilde{v}) = (u, v) = \mathbf{u}$. Spatial gradients in normalized coordinates will be represented as $(\tilde{F}_x, \tilde{F}_y)$. They are related to the unnormalized form by $(\tilde{F}_x/h_k, \tilde{F}_y/h_k) = (F_x, F_y)$.

4.2.2 Relaxation

The particular relaxation operator used is determined by the discrete approximation to the Laplacian that is used and by the iterative relaxation scheme. We choose Jacobi relaxation because it is a parallel updating method. The discrete Laplacian is used in all of the following experiments is

$$\Delta^k = \frac{1}{h_k^2}\Delta = \frac{1}{h_k^2}\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{26}$$

The simpler five-point Laplacian (four 1's surrounding a central -4) is not a good choice for use with Jacobi relaxation because elimination of the highest frequency error is not guaranteed. This is shown by *local mode analysis* in [Glazer 87, Appendix F]. In Section 6, local mode analyses are defined and the results in [Glazer 87] are summarized. Those results show that our choice of discrete Laplacian does provide good convergence for multilevel relaxation using the Jacobi scheme.

The discrete version of Equation 25 is

$$L^k \mathbf{u}^k = \mathbf{r}^k \tag{27}$$

where

$$L^k = \begin{bmatrix} \frac{\alpha^2}{h_k^2}\Delta - F_x^2 & -F_x F_y \\ -F_x F_y & \frac{\alpha^2}{h_k^2}\Delta - F_y^2 \end{bmatrix} \tag{28}$$

$\mathbf{u}^k = [u^k \; v^k]^T$ and $\mathbf{r}^k = [r_1^k \; r_2^k]^T = [F_x F_t \; F_y F_t]^T$.

The Jacobi update equations are derived as follows. The discrete Laplacians Δu and Δv can be represented as $3(\hat{u} - u)$ and $3(\hat{v} - v)$ where $\hat{u}$ and $\hat{v}$ are local averages given by:

$$\hat{u} = \frac{1}{12}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 1 \end{bmatrix} * u \qquad \hat{v} = \frac{1}{12}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 1 \end{bmatrix} * v \tag{29}$$

The expressions $3(\hat{u} - u)$ and $3(\hat{v} - v)$ are substituted into Equations 27, 28 and the equations solved for u and v. The resultant equations are used as Jacobi update equations by specifying that at the $(s+1)$-st update iteration, a new value (u_{s+1}, v_{s+1}) at the central pixel is computed from the local average $(\hat{u}_s, \hat{v}_s)$ of old values at the neighbors. The following update equation results:

$$u_{s+1} \leftarrow \hat{u}_s - F_x[F_x\hat{u}_s + F_y\hat{v}_s + F_t]/(3\alpha^2 + F_x^2 + F_y^2) \tag{30a}$$

$$v_{s+1} \leftarrow \hat{v}_s - F_y[F_x\hat{u}_s + F_y\hat{v}_s + F_t]/(3\alpha^2 + F_x^2 + F_y^2) \tag{30b}$$

We convert to the normalized form by substituting in the normalized form of the flow vectors and the spatial gradients, namely, $(h_k\tilde{u}, h_k\tilde{v}) = (u, v)$ and $(\tilde{F}_x/h_k, \tilde{F}_y/h_k) = (F_x, F_y)$, and dividing the equations by h_k to get:

$$\tilde{u}_{s+1} \leftarrow \hat{u}_s - \tilde{F}_x[\tilde{F}_x\hat{u}_s + \tilde{F}_y\hat{v}_s + F_t]/(3\alpha^2 + \tilde{F}_x^2 + \tilde{F}_y^2) \tag{31a}$$

$$\tilde{v}_{s+1} \leftarrow \hat{v}_s - \tilde{F}_y[\tilde{F}_x\hat{u}_s + \tilde{F}_y\hat{v}_s + F_t]/(3\alpha^2 + \tilde{F}_x^2 + \tilde{F}_y^2) \tag{31b}$$

where $\hat{u}$ and $\hat{v}$ are changed in meaning to local averages of $\tilde{u}$ and $\tilde{v}$ respectively.

4.2.3 Projection

The interpolation operators used for projection and reduction are bilinear interpolation $I_{k-1}^k = Bil$ and average-of-the-sons $I_k^{k-1} = A_{2\times 2}$. Substituting into Equation 21, corrections are projected using

$$\mathbf{u}^k \leftarrow \mathbf{u}^k + Bil(\mathbf{u}^{k-1} - A_{2\times 2}\mathbf{u}^k)$$

Using flow vectors represented in normalized coordinates, the projection equation is

$$\tilde{\mathbf{u}}^k \leftarrow \tilde{\mathbf{u}}^k + 2Bil(\tilde{\mathbf{u}}^{k-1} - \frac{1}{2}A_{2\times 2}\tilde{\mathbf{u}}^k)$$

The scale factors 2 and $\frac{1}{2}$ for projection and reduction are used because $\mathbf{u}$ is of order -1. For example, a velocity or displacement of 2 pixel positions at a given level, when reduced, is equivalent to 1 pixel position at the coarser level.

4.2.4 Reduction

Coarse solution estimates are passed up using:

$$\tilde{\mathbf{u}}^{k-1} \leftarrow \frac{1}{2}A_{2\times 2}\tilde{\mathbf{u}}^k \tag{32}$$

The coarse residuals are given by:

$$\mathbf{r}^{k-1} \leftarrow A_{2\times 2}(\mathbf{r}^k - L^k\mathbf{u}^k) + L^{k-1}\mathbf{u}^{k-1} \tag{33}$$

where L^k and L^{k-1} are defined by Equation 28. (From Equation 19.) If we multiply through by h_{k-1} we get:

$$h_{k-1}\mathbf{r}^{k-1} \leftarrow \frac{h_{k-1}}{h_k}A_{2\times 2}\left(h_k\mathbf{r}^k - h_kL^k\mathbf{u}^k\right) + h_{k-1}L^{k-1}\mathbf{u}^{k-1}$$

which is finally converted to the level-independent form of the update equation used ($h_{k-1}/h_k = 2$):

$$\tilde{\mathbf{r}}^{k-1} \leftarrow 2A_{2\times 2}(\tilde{\mathbf{r}}^k - \tilde{L}^k\tilde{\mathbf{u}}^k) + \tilde{L}^{k-1}\tilde{\mathbf{u}}^{k-1} \tag{34}$$

$$\leftarrow 2A_{2\times 2}\tilde{\mathbf{r}}^k - 2A_{2\times 2}\tilde{L}^k\tilde{\mathbf{u}}^k + \tilde{L}^{k-1}\tilde{\mathbf{u}}^{k-1} \tag{35}$$

where the normalized right-hand-sides are $\tilde{\mathbf{r}}^{k-1} = h_{k-1}\mathbf{r}^{k-1}$ and $\tilde{\mathbf{r}}^k = h_k\mathbf{r}^k$, and the normalized operator is given by

$$\tilde{L}^k = h_k^2L^k = \begin{bmatrix} \alpha^2\Delta - \tilde{F}_x^2 & -\tilde{F}_x\tilde{F}_y \\ -\tilde{F}_x\tilde{F}_y & \alpha^2\Delta - \tilde{F}_y^2 \end{bmatrix} \tag{36}$$

Equation 35 can be broken into its component equations as follows:

$$\tilde{r}_1^{k-1} \leftarrow 2A_{2\times 2}\tilde{r}_1^k - 2A_{2\times 2}(\alpha^2\Delta u^k - \tilde{F}_x^2u^k - \tilde{F}_x\tilde{F}_yv^k) + (\alpha^2\Delta u^{k-1} - \tilde{F}_x^2u^{k-1} - \tilde{F}_x\tilde{F}_yv^{k-1}) \tag{37a}$$

$$\tilde{r}_2^{k-1} \leftarrow 2A_{2\times 2}\tilde{r}_2^k - 2A_{2\times 2}(\alpha^2\Delta v^k - \tilde{F}_y^2v^k - \tilde{F}_x\tilde{F}_yu^k) + (\alpha^2\Delta v^{k-1} - \tilde{F}_y^2v^{k-1} - \tilde{F}_x\tilde{F}_yu^{k-1}) \tag{37b}$$

Reduction is given by Equations 32 and 37.

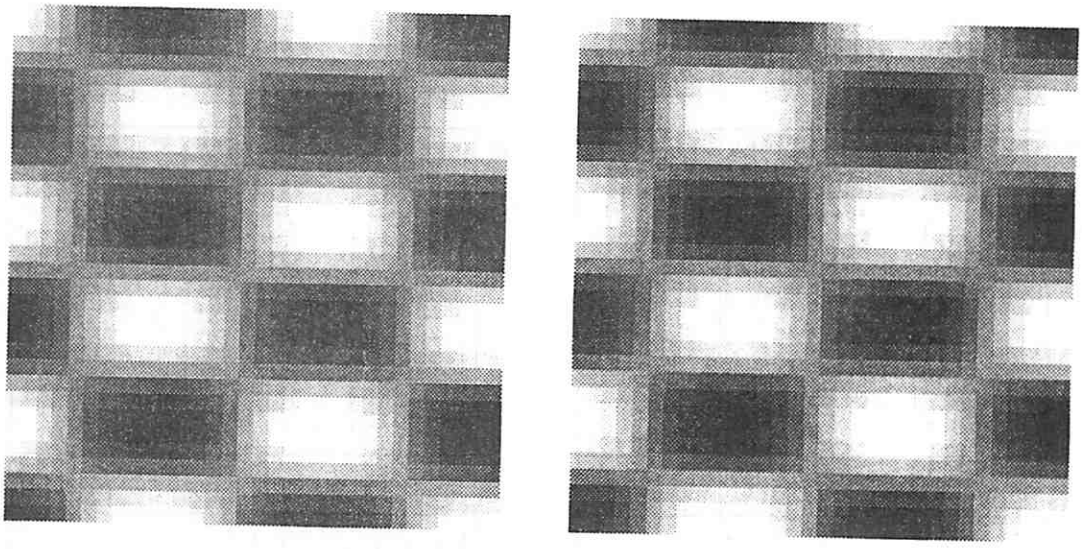

Figure 3: Optic flow test data

(Left) First frame, a cross product of sinusoids with 1% uniform noise added. (Right) Second frame, the first frame displaced 1 pixel up, 1/2 pixel to the right.

5 Experiments

In this section we demonstrate the increased convergence rate obtained by multilevel relaxation over that for single-level relaxation in computing optic flow.

5.1 Single-level Relaxation

The two frames of test data for the first experiment are shown in Figure 3. In the first experiment the motion is translational: 1/2 pixel to the right and 1 pixel up. One percent uniform noise has been added to both test images and is uncorrelated between frames. This test data is designed to be as close as possible to that used in [Horn & Schunck 81]. The single grid Horn & Schunck algorithm is shown in Figures 4 and 5. Figure 4 shows a portion of the image plane at various stages in the iteration. The initial estimate (shown as iteration 0) is given by the edge flow vectors computed from equation 38.

$$\begin{pmatrix} -F_x F_t & -F_y F_t \\ F_x^2 + F_y^2 & F_x^2 + F_y^2 \end{pmatrix} \tag{38}$$

An error graph is plotted in Figure 5. In Figure 5, the residual norm and the mean error magnitude are plotted against iteration number. The latter is defined as the mean of the length of the error vector at each pixel, where the length is given by the Euclidean

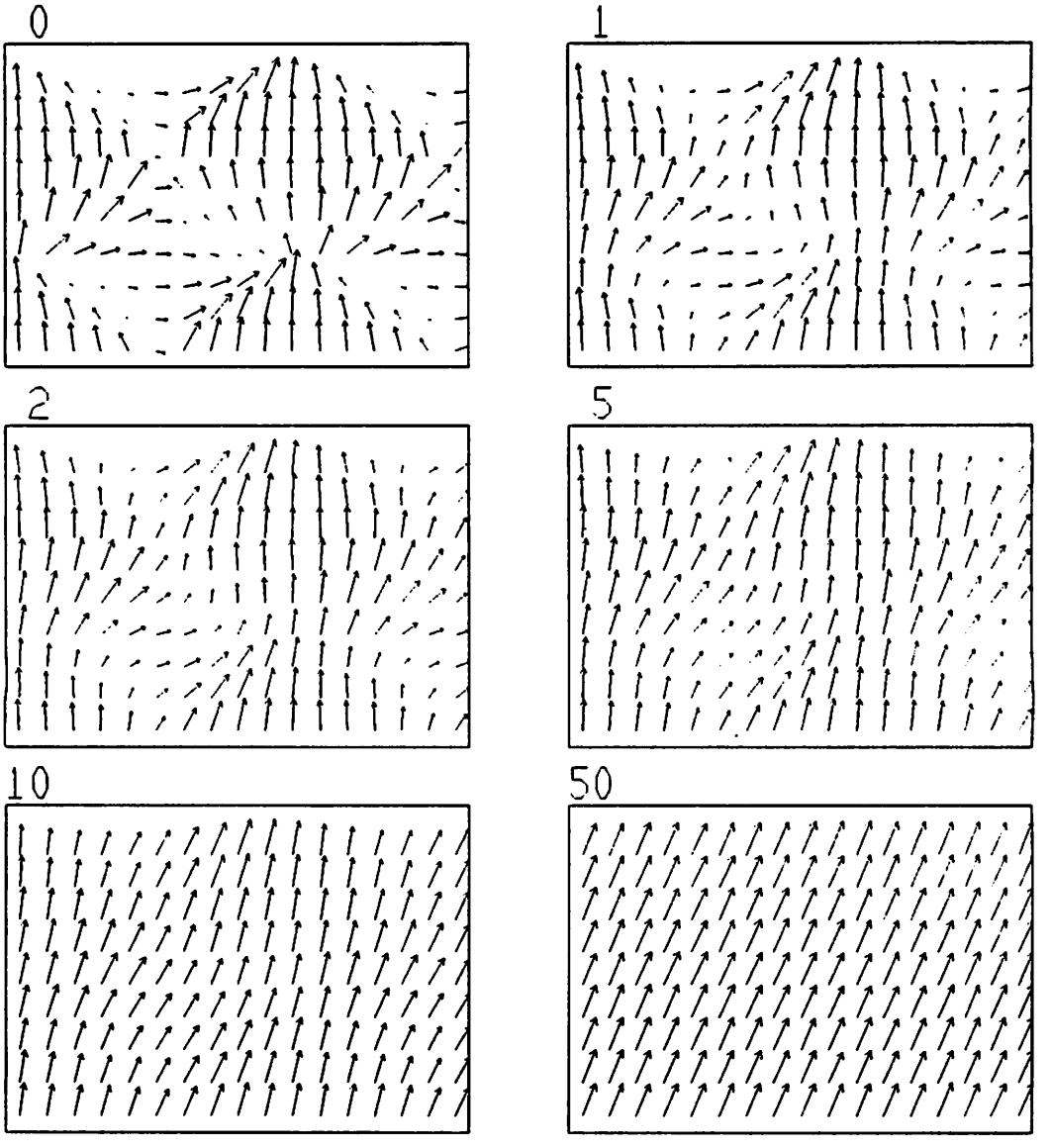

Figure 4: Single-level optic flow computation

Iterations 0,1,2,5,10 and 50. Only a portion of the image plane is shown.

norm, and the error vector is the difference between the computed flow vectors and the ideal flow vector $(D_{row}, D_{col}) = (-1.0, 0.5)$. Note that the ideal flow is NOT equal to the actual solution to the discrete representation of the optic flow equations. There are two reasons for this. First, there is an inherent error involved in a discrete representation due to approximations in the finite difference operators and the finite precision arithmetic. This is known as *truncation error*. Secondly, we have added noise to the data, thereby further corrupting the approximate coefficients derived from the spatial and temporal gradients. In any case, we still see the computed flow vector coming very close to the ideal, with error on the order of .015 pixels. (For further accuracy statistics, see the single-level versus multilevel comparison in Table 1.)

5.2 Multilevel Relaxation

The results of the multilevel algorithm are shown in the next two figures. The FAS hierarchical control strategy is used since it provides a hierarchy of solution fields at the varying levels of resolution. While reductions to levels coarser than level 4 were allowed, that was the highest level that the algorithm chose to go to. In Figure 6 the first 14 iterations are shown for a portion of the full image (the upper left corner). Consecutive iterations at a given level are juxtaposed in the vector plots. At the top of Figure 7 residual norm is plotted versus iteration. By the 22nd iteration, little further reduction in the residual norm is seen. At the bottom of Figure 7 mean error magnitude is plotted versus iteration, showing accuracy at least as good as that obtained by single-level relaxation after many more iterations. The mean error magnitudes in the error graphs are given in the "non-normalized" base coordinate system to aid in comparison

Table 1 compares the actual computed optic flow versus the expected value of $(-1, .5)$ at each pixel. The tables show that after 22 iterations, the multilevel algorithm has reached a solution which, in terms of relative error, is more than twice as good as that reached by the single-level method in 50 iterations.

5.3 Non-Translational Motion

Next we present an experiment using a rotational motion field. Again, this test data is designed to be as close as possible to that used in [Horn & Schunck 81]. The first image frame is the same as the first frame in Figure 3. The second frame is a rotated version of the first frame, where the rotation angle is .05 radians (2.86 degrees) and the center of rotation is the point 1/4 in from the left and bottom sides (i.e., the center of the lower left quadrant). In Figure 8 the edge flow for a rotational motion is shown. This vector field is used as the initial estimate of optic flow.

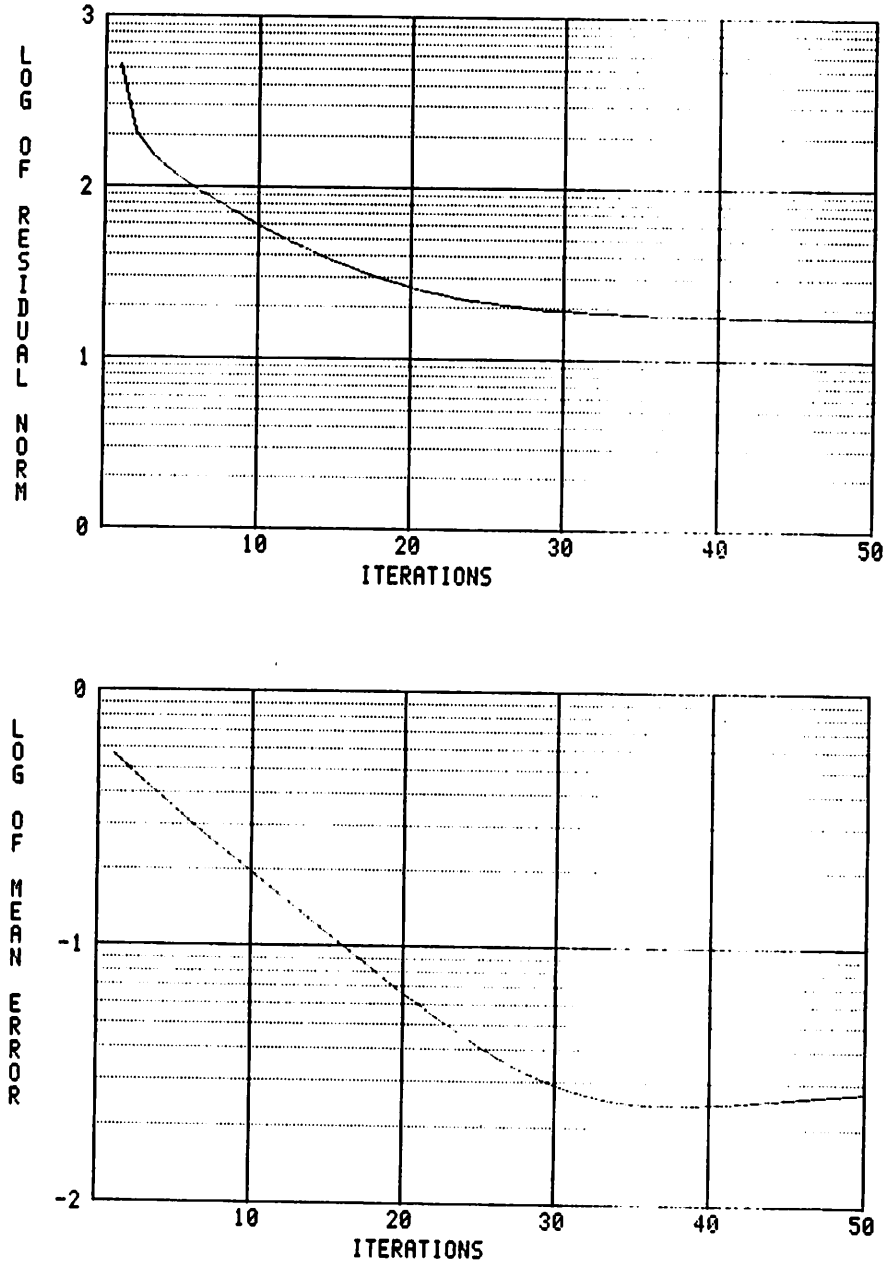

Figure 5: Single-level optic flow computation: error graphs

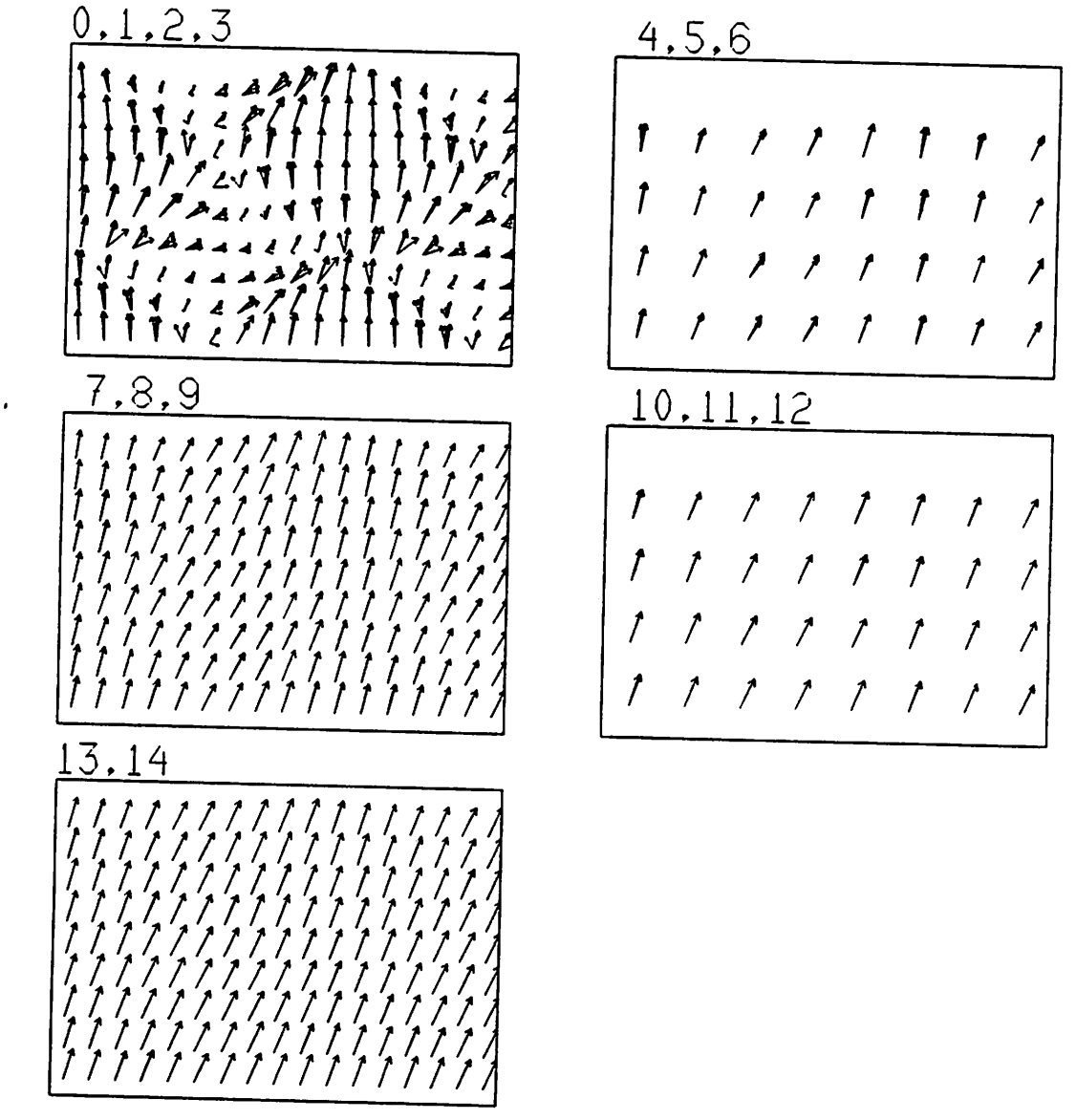

Figure 6: Multilevel optic flow computation

iterations 0,1,2,3 at level 5
iterations 4,5,6 at level 4
iterations 7,8,9 at level 5
iterations 10,11,12 at level 4
iterations 13,14 at level 5

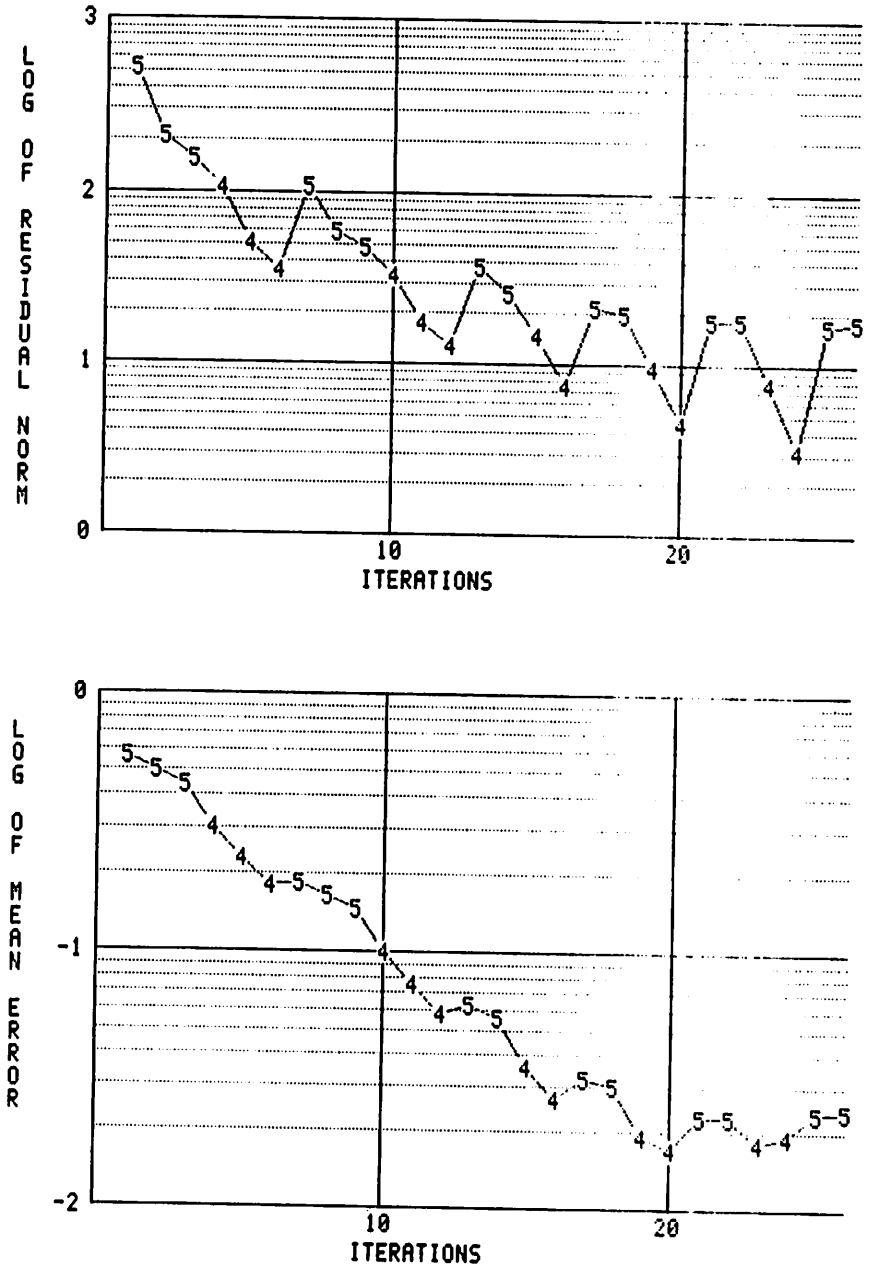

Figure 7: Multilevel optic flow computation: error graphs

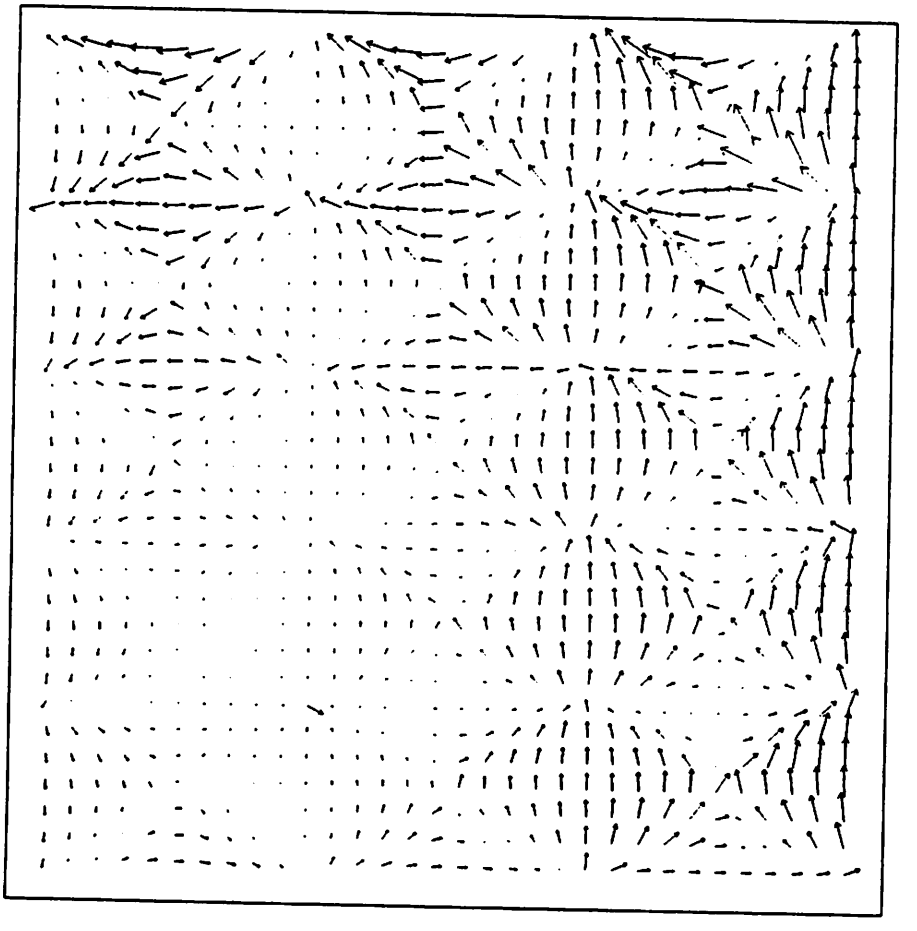

Figure 8: Rotational edge flow

The edge flow field for the rotational motion data as given by Equation 38. This vector field is used as the initial estimate of optic flow.

Single-level, iteration 50

Actual	D_{row}	D_{col}
mean	−1.007	.5205
SD	.01478	.01667
minimum	−1.056	.4595
maximum	−0.9584	.5715
Expected	−1.0	.5

Multilevel, iteration 22

Actual	D_{row}	D_{col}
mean	−1.003	.5082
SD	.01554	.01967
minimum	−1.066	.4541
maximum	−0.9582	.5694
Expected	−1.0	.5

Table 1: Optic flow field statistics

These tables compare the actual computed optic flow against the expected values. The table on the left shows the results after 50 iterations of the single level relaxation algorithm. The relative errors of the mean actual value in the two components are .7% and 4.1%. The table on the right shows the results after 22 iterations of the multilevel optic flow algorithm. The relative errors of the mean actual value in the two components are .3% and 1.6%.

Figure 9 shows later stages in the multilevel relaxation, and Figure 10 shows the error graphs for this experiment. The computed flow fields shown look good to the eye and the fine-level field at iteration 13 has reached a mean error magnitude of less than a tenth of a pixel. However, at level 3 the algorithm has diverged and does not return to level 5 where, as the downward trending data suggests, we would expect an even more accurate fine-level field to have been obtained compared to the final time the algorithm was at level 5. We will consider this problem further.

In this experiment, the highest level allowed was level 3. In a similar experiment not shown, when control was allowed to pass to level 2, divergence occurred and was more extreme. If coarsening is only allowed up to level 4, then divergence does not occur and more accurate level 5 fields are obtained. However, in that experiment convergence at level 4 eventually slowed to the point where the target residual norm was not reached after tens of iterations, leaving control "stuck" at level 4. This behavior will be examined later in Section 7.2.

In these last experiments on rotational motion fields, the basic FAS algorithm fails to work. Divergence at coarse levels prevents good fine-level solutions from being attained. In the next sections, we further analyze multilevel optic flow relaxation equations to show how this divergence is caused by local variations in the image data, where "local" is measured relative to the grid resolution. This being the case, hierarchical control must avoid too

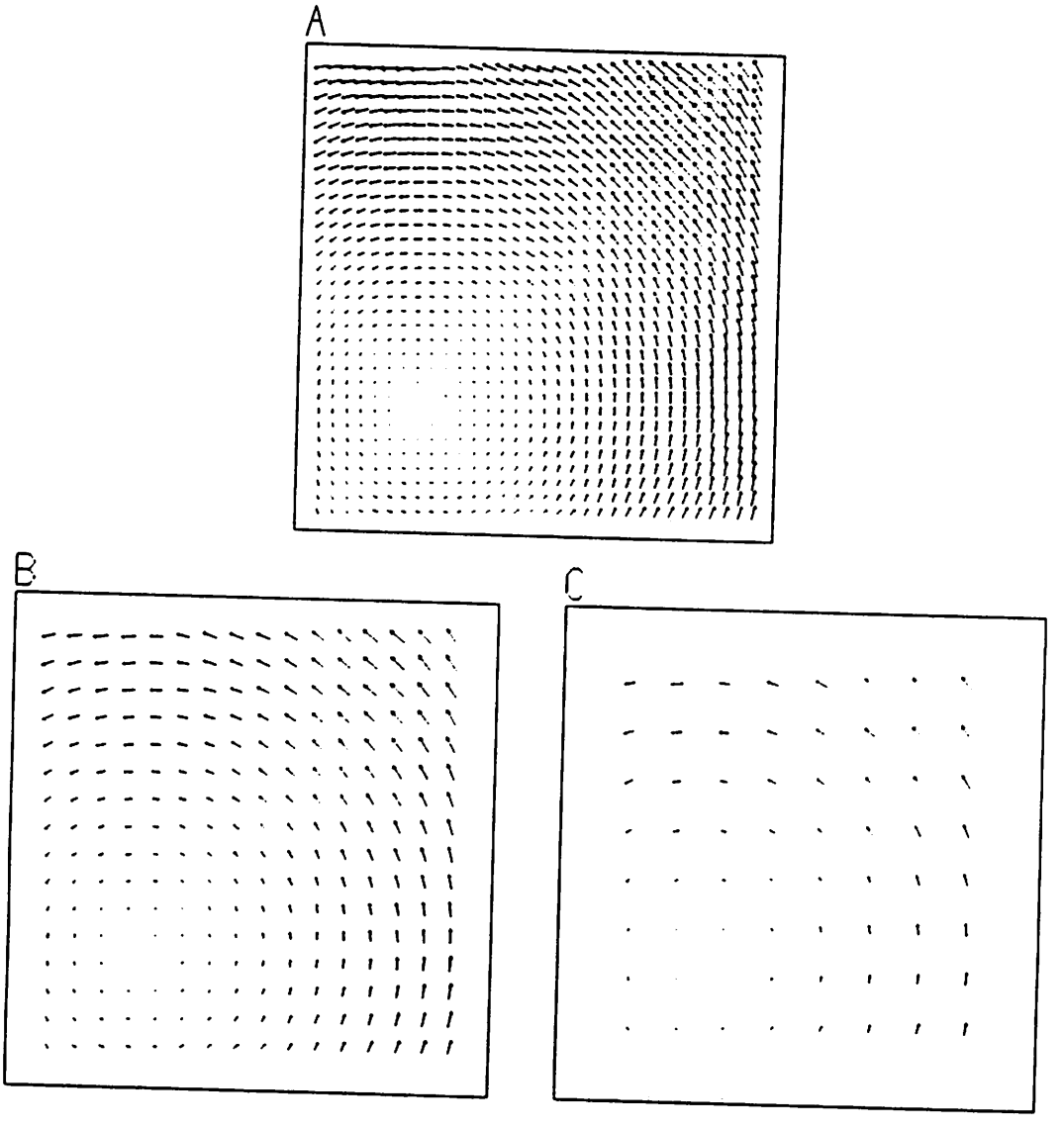

Figure 9: Multilevel flow computation: rotational motion

(a) iteration 13 at level 5

(b) iteration 16 at level 4

(c) iteration 20 at level 3

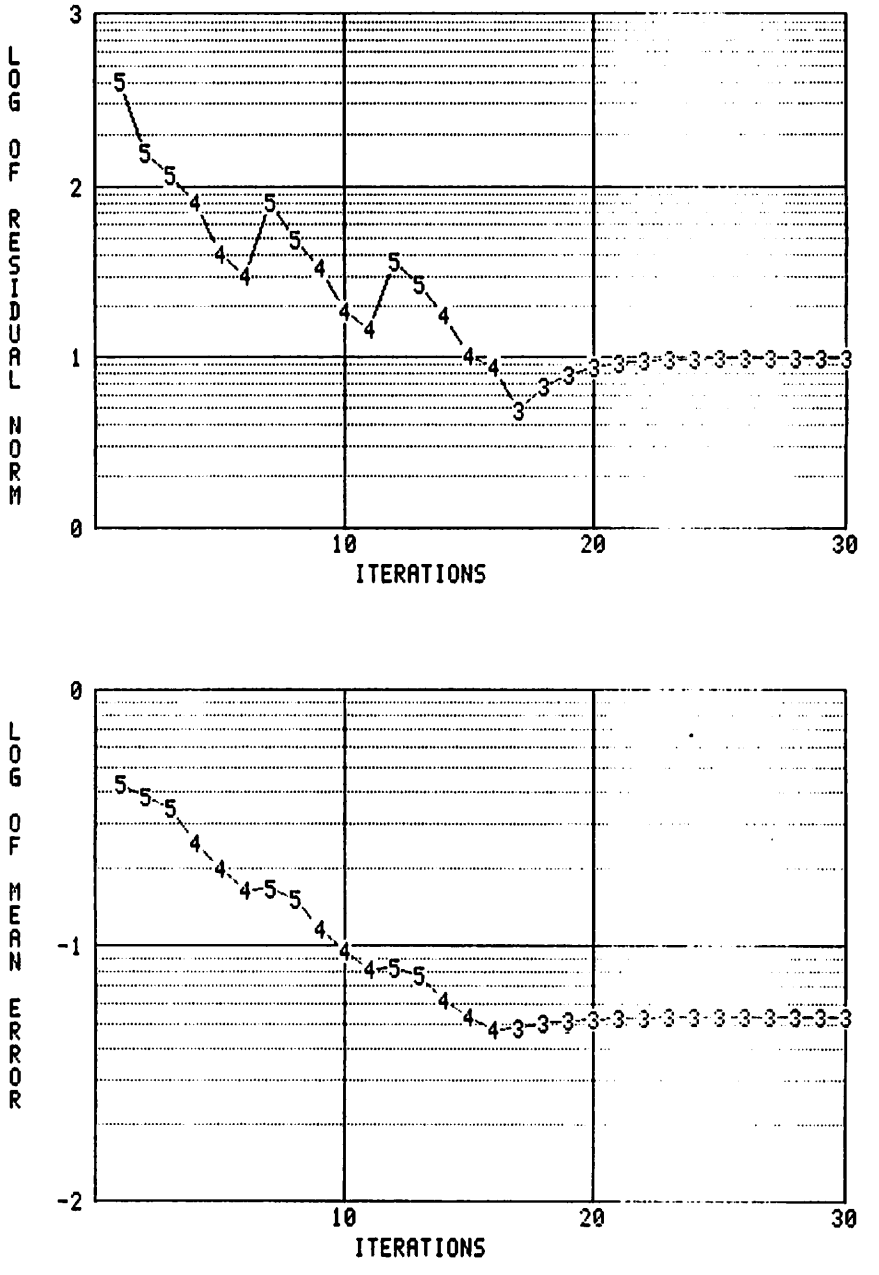

Figure 10: Multilevel flow computation: rotational motion, error graphs

coarse a level. We will see that instituting a fixed cycling scheme, that is, a fixed pattern of reduction and projection, solves the divergence problem.

6 Local Mode Analysis

In this section, we describe Brandt's notion of *local mode analysis* and we consider the local mode analysis of optic flow relaxation. This further analysis of multilevel optic flow relaxation serves to justify our formulation and to provide added insight into the nature of its convergence properties. Brandt's purpose was to provide a good approximate measure of the convergence rate of multilevel algorithms by considering the error reduction performed at each level in a hierarchical algorithm. In our analysis of optic flow relaxation, besides obtaining the approximate convergence rates, we show how the error reduction is related to both the direction and the magnitude of the gradient of the image data. It will be shown that when the gradients are weak (i.e. low magnitude) convergence behavior is equivalent to simple smoothing; and that when gradients are strong convergence will be accelerated towards the constraint line.

Multilevel relaxation algorithms use relaxation to smooth out error components. At a given level, relaxation is effective in smoothing out those components of the error which are high frequencies at that level. The overall rate of convergence for a multilevel algorithm is directly related to the rate at which high frequency error terms are reduced at each level, and it is independent of the rate of reduction of low frequency terms, which as we know can be very slow. Brandt introduced *local mode analysis* to approximate the rate at which high frequency error terms are reduced in the approximate solution [Brandt 77a]. This involves defining a *local convergence rate* to measure error reduction of the high frequency error terms.

Local mode analysis involves a Fourier analysis of the error terms. In this analysis, the convergence rate for individual error frequencies (as defined by the Fourier Transform of the error function) is computed. The local convergence rate is then defined as the maximum individual convergence rate over the higher frequencies. These higher frequencies are selected to be those in the baseband at a given level which are not also in the baseband of the next coarser level. That is, the frequencies which are representable at the given level but not at a coarser level.

The details of local mode analysis can be found in [Glazer 87, Section F.1] where a thorough analysis is performed for the simple case of Laplace's equation. The main result appears in [Glazer 87, Section F.2], where we perform a local mode analysis of optic flow relaxation. We briefly summarize the results of the former analysis before examining the optic flow analysis in detail.

6.1 Local Mode Analysis of Laplace's Equation

Laplace's equation $\Delta U(x,y) = 0$ is the Euler equation for the variational problem in which the solution $U(x,y)$ minimizes the integral $\iint_A \| \nabla U \|^2 \, dxdy$ under a suitable boundary condition. Relaxation algorithms for solving Laplace's equation on discrete grids interatively update (replace) the value of U at given grid locations with a weighted average of the values at neighboring pixels. The form of the weighted average depends on the particular discrete Laplacian chosen.

After a given relaxation iteration, the error δ is equal to the difference between the exact solution to the discrete system of equations U and the current approximation u, i.e. $\delta(m,n) \triangleq U(m,n) - u(m,n)$. Individual frequency components of the error are given by the Fourier transform of the error function $\delta(m,n)$

$$\delta(m,n) = \int_{|\theta| \leq \pi} A_\theta e^{i(\theta_1 m + \theta_2 n)} \, d\theta$$

where $\theta = (\theta_1, \theta_2)$ and $|\theta| = \max(\theta_1, \theta_2)$. The convergence rate for the individual θ component is then defined as

$$\mu(\theta) \triangleq \left| \frac{\overline{A}_\theta}{A_\theta} \right|$$

where A_θ is the Fourier coefficient before a given iteration of relaxation and $\overline{A}_\theta$ is the Fourier coefficient after the iteration. The values of $\mu(\theta)$ range from 0 to 1, with low values indicating fast convergence and high values slow convergence. The **local convergence rate** is defined as the maximum individual convergence rate over the high frequencies, that is:

$$\bar{\mu} \triangleq \max_{\pi/2 \leq |\theta| \leq \pi} \mu(\theta)$$

The results of local mode analysis depends on the choice of finite difference approximations, and on the choice of iterative update scheme. With the Δ in Equation 26 used to approximate the discrete Laplacian and with the Jacobi relaxation scheme used, the individual convergence rate is found to be

$$\mu(\theta) \triangleq \left| \frac{\overline{A}_\theta}{A_\theta} \right| = \frac{1}{3} |\cos\theta_1 + \cos\theta_2 + \cos\theta_1 \cos\theta_2| \tag{39}$$

and hence the local convergence rate is given by

$$\bar{\mu} \triangleq \max_{\pi/2 \leq |\theta| \leq \pi} \mu(\theta) = \mu(\pi, \pi) = \frac{1}{3}$$

Note that at low frequencies (θ near $(0,0)$), individual convergence rates can go to 1, indicating no reduction of the error term. This is of no consequence in a multilevel algorithm since low frequency error is handled by coarser levels.

6.2 Local Mode Analysis of Optic Flow Relaxation

In the local mode analysis of Laplace's equation, the error component at frequency θ before and after relaxation were represented by the Fourier coefficients A_θ and $\overline{A}_\theta$ respectively. In the case of optic flow, the error component is a vector field, and there are two Fourier coefficients (A_θ, B_θ) before and two coefficients $(\overline{A}_\theta, \overline{B}_\theta)$ after relaxation, one for each of the vector components.

The general relationship between these coefficients is given by the *convergence matrix* — a 2×2 matrix relating error components at a given spatial frequency before and after relaxation. This generalizes the single valued individual convergence rate seen in the case of Laplace's equation. The local convergence rates can then be defined in terms of the eigenvalues of the convergence matrix.

First we state the results of the local mode analysis of optic flow relaxation. (The proof is found in [Glazer 87, Appendix F].)

Local Mode Analysis 1 *Suppose we use the discrete Laplacian Δ (Equation 26) in a Jacobi relaxation scheme to do multilevel optic flow relaxation. If (A_θ, B_θ) are the Fourier coefficients of the θ component of the error before relaxation, and (A_θ, B_θ) are the coefficients after relaxation, then these coefficients are related by the* **convergence matrix** *M as follows:*

$$\begin{bmatrix} A_\theta \\ \overline{B}_\theta \end{bmatrix} = M \begin{bmatrix} A_\theta \\ B_\theta \end{bmatrix}$$

$$= \frac{\mu(\theta)}{3\alpha^2 + F_x^2 + F_y^2} \begin{bmatrix} 3\alpha^2 + F_y^2 & -F_x F_y \\ -F_x F_y & 3\alpha^2 + F_x^2 \end{bmatrix} \begin{bmatrix} A_\theta \\ B_\theta \end{bmatrix}$$

where $\mu(\theta) = \frac{1}{3}(\cos\theta_1 + \cos\theta_2 + \cos\theta_1 \cos\theta_2)$. The eigenvalues of M are

$$\lambda_{min} = \mu(\theta) \frac{3\alpha^2}{3\alpha^2 + F_x^2 + F_y^2} \qquad \lambda_{max} = \mu(\theta)$$

The eigenvector associated with λ_{min} is $\nabla F \triangleq (F_x, F_y)$, the gradient of F. The eigenvector associated with λ_{max} is $\nabla_\perp F \triangleq (-F_y, F_x)$, a vector perpendicular to the gradient of F.

The convergence matrix generalizes on the convergence factor $\mu(\theta) \triangleq |\overline{A}_\theta / A_\theta|$ defined in the case of Laplace's Equation. (In a non-rigorous sense it is a measure of $[A_\theta \; B_\theta]^T / [A_\theta \; B_\theta]^T$.)

Various remarks are now in order:

1. When the convergence matrix M operates on error components, λ_{max} provides an upper bound on how much that error component is reduced in magnitude. λ_{max} is,

by definition, the *spectral radius* of M; and, since M is symmetric, it is equal to the *spectral norm* of M; and thus for any vector x we know that $\|M\mathbf{x}\| \leq \lambda_{max} \|\mathbf{x}\|$ where $\|\cdot\|$ is the Euclidean norm [Varga 62].

2. The convergence matrix tells us how error is reduced at different spatial frequencies. The action of M on a component (A_θ, B_θ) of the error vector δ is given by: (1) break (A_θ, B_θ) into component vectors parallel to ∇F and $\nabla_\perp F$; (2) multiply those components by λ_{min} and λ_{max} respectively; (3) add the results of (2) together. More formally, if $(A_\theta, B_\theta) = x_1 \nabla F + x_2 \nabla_\perp F$, then $M(A_\theta, B_\theta) = (\lambda_{min} x_1)\nabla F + (\lambda_{max} x_2)\nabla_\perp F$ and, iterating n times, $M^n(A_\theta, B_\theta) = (\lambda_{min}^n x_1)\nabla F + (\lambda_{max}^n x_2)\nabla_\perp F$

3. The term $\mu = \frac{1}{3}(\cos\theta_1 + \cos\theta_2 + \cos\theta_1 \cos\theta_2)$ is determined by the particular discrete Laplacian Δ that is used. Note that it is equivalent to the individual convergence rate derived in the local mode analysis of Laplace's equation.

4. If the gradient is zero, then $\lambda_{min} = \lambda_{max} = \mu(\theta)$, the convergence matrix is equal to $\mu(\theta)$ times the identity matrix, and relaxation serves to multiply the error vectors by $\mu(\theta)$. Convergence in the two components of the optic flow is identical to that seen for simple smoothing by Laplace's equation. This is consistent with the fact that the optic flow update equation reduces to simple smoothing (as expressed by Laplace's Equation) when the gradient vanishes.

5. If the gradient is not zero, then the convergence rate perpendicular to the gradient is still $\mu(\theta)$, but the convergence rate parallel to the gradient is faster in proportion to the gradient magnitude (squared). Thus, error components perpendicular to the gradient still converge at the simple smoothing rate, while error components parallel to the gradient converge at a faster rate, where the decrease is proportional to the magnitude of the gradient.

If we measure overall convergence by the maximum eigenvalue $\lambda_{max} = \mu(\theta)$, we must conclude that the local convergence rate is the same as that for simple smoothing. The minimum eigenvalue λ_{min} is associated with convergence towards the constraint line, since the direction parallel to the gradient is perpendicular to the constraint line. For strong gradients (large magnitude), λ_{min} is small implying fast convergence. This is a quantitative measure of the enforcement of the velocity line constraint at a point.

There is a second aspect to convergence of optic flow relaxation which is not captured by the local mode analysis. The local mode analysis only depends on the gradient at the central pixel. Any effect that locally varying gradient directions have on convergence are not reflected directly in the convergence matrix. Local mode analysis does not give us any

measure of convergence based on the accumulation of information from multiple constraint lines with varying directions. In particular, it does not lend any insight into the divergence we have seen at coarse levels. In the next section, we study the relationship between coarse divergence and (1) variations in the flow field and (2) variations in the image data. The latter is shown to directly influence both overall speed of convergence and the occurrence of divergence.

7 Analysis of Coarse Approximations

In this section we consider the divergence of the multilevel optic flow relaxation algorithm at coarse levels. This divergence may be due to variations of the disparity field or in the image data itself. We first show that the former is not the case and then go on to analyze the relationship between local structure in the image data and convergence/divergence at varying resolutions.

7.1 Variation of the Disparity Field

The multilevel algorithm does not exhibit divergence at coarse levels in the case of pure translational motion. This suggests the possibility that coarse divergence may be related to variation in the disparity field to be computed. Clearly, if a disparity field with large enough variation was chosen, the local smoothness property on which the optic flow relaxation method is based would be violated, and we would not expect a relaxation technique to convergence to a reasonable solution. However, this does NOT explain the occurrence of divergence at coarse levels only as seen in Figure 10. As we show next, the variation of rotational disparity fields is the same at all resolution levels. Thus, with high enough variation, we would have to see divergence at all levels, not just at coarse levels.

Consider, in the continuous case, the rotational disparity field $(u, v) = (u(x, y), v(x, y))$ centered at (x_c, y_c) with angle β. It is given by

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\beta - 1 & \sin\beta \\ -\sin\beta & \cos\beta - 1 \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix}$$

The variation of the field (u, v) is given by the gradients of u and v. Considering their magnitudes, we see that $|\nabla u|^2 = |\nabla v|^2 = 2(1 - \cos\beta)$ — a constant value. Now consider a discrete rotational field formed by sampling the continuous field $(u(x, y), v(x, y))$ at grid points. If finite differences are used to measure the gradient, the result is exact (up to the finite arithmetic precision) because the individual components u and v are linear functions of x and y. Thus, the local variation of a rotational disparity field as given by the finite difference gradient magnitude of u and v is a constant value independent of grid level.

7.2 Variation of the Image Data

We are now left with variations in the image data to explain coarse level divergence. Intuitively, the reason seems to be that the optic flow PDE is not accurately represented at coarse levels because of the spatially varying coefficients. This idea is taken up more fully in the next section. For now, we point out that the coefficients are directly computed from the image data, being products of first differences. Spatial variation in the coefficients is due to spatial variation in the underlying image data. In this section, the relationship between spatial variations in the image data and convergence/divergence of optic flow relaxation is shown by experiment.

Consider the spatial variation of the image data in the previous experiments. The test image in Figure 3 is a product of sinusoids in the x (column) and y (row) directions with wavelengths of 12.8 and 21.33 respectively at the finest level. If a low-pass pyramid is built from this data, the corresponding wavelengths at coarser levels would be 10.67, 5.33, 2.67 at levels $k = 4, 3, 2$ respectively for λ_y and 6.4, 3.2, 1.6 at levels $k = 4, 3, 2$ respectively for λ_x. The level 2 x-wavelength of 1.6 is not actually representable since the maximum wavelength representable at any level is $\lambda = 2$. These numbers suggest that the coefficients for the optic flow PDE cannot be adequately represented at level 2 and that even at level 3 the variation may be too high.

If divergence at some coarse level is related to the degree of local variation in the image data, where "local" is measured relative to the pixel spacing at the coarse level, then we expect that when the test data has less variation (i.e. lower spatial frequencies) convergence will be obtained at coarser levels. This is in fact the case as experiment will show. Before demonstrating this for rotational motion fields, we revisit the case of translational motion for both single-level and multilevel experiments, this time with data of less spatial variation. These experiments on translational motion show the relationship between relative frequency content and convergence/divergence independent of any variation in the motion field. We start with a single-level relaxation experiment that shows how convergence is slower for lower frequency data.

The test data for these experiments is equivalent to that used in the prior experiments (Figure 3) with one important exception: the wavelengths of the sinusoids are doubled to 25.6 and 42.67 in the x and y directions respectively. First, we again consider the case of purely translational motion, using the same translational field of $(-1.0, 0.5)$. In Figure 11 the error graphs are shown for single-level optic flow relaxation. This should be compared to the comparable single-level experiment shown in Figure 5. Convergence is clearly much slower for the low frequency data. This must be attributed to reduced local variation in the direction of the image gradients.

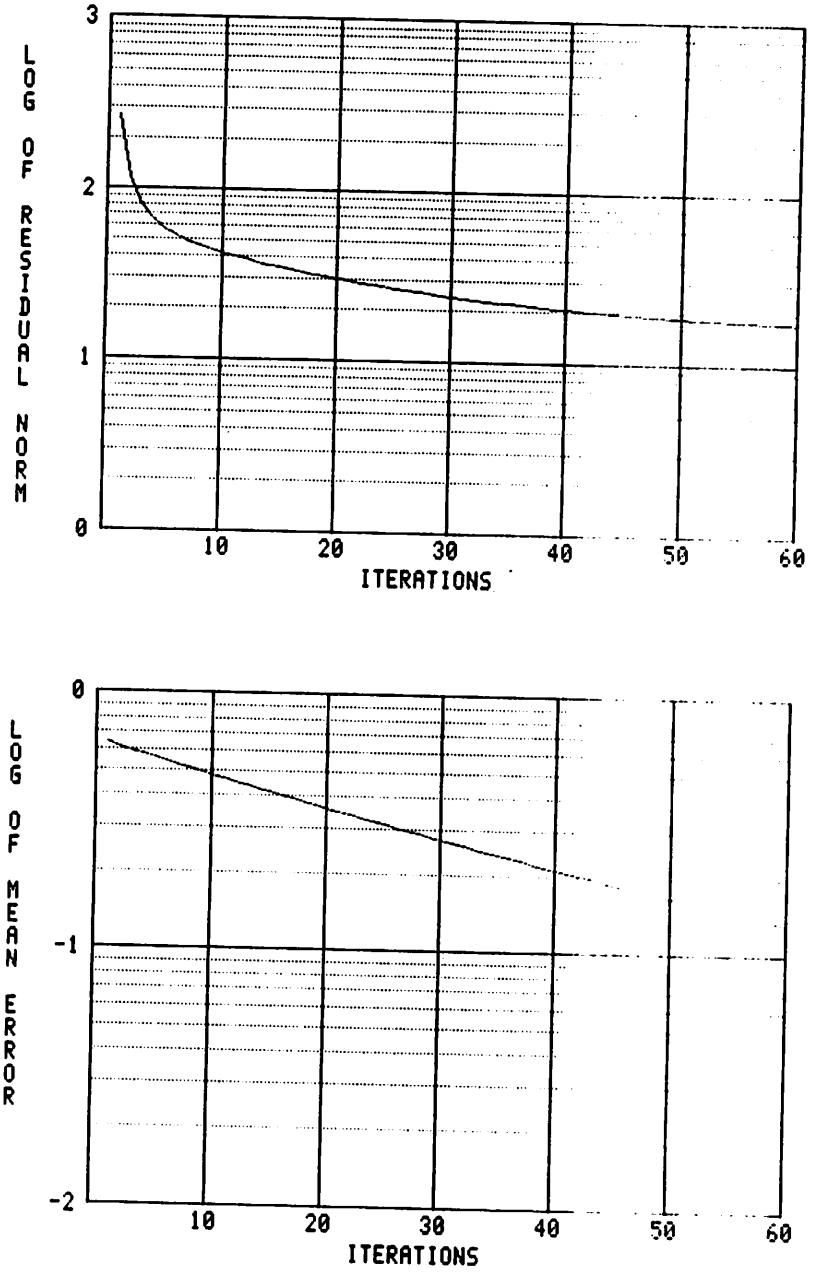

Figure 11: Single-level relaxation: low frequency data

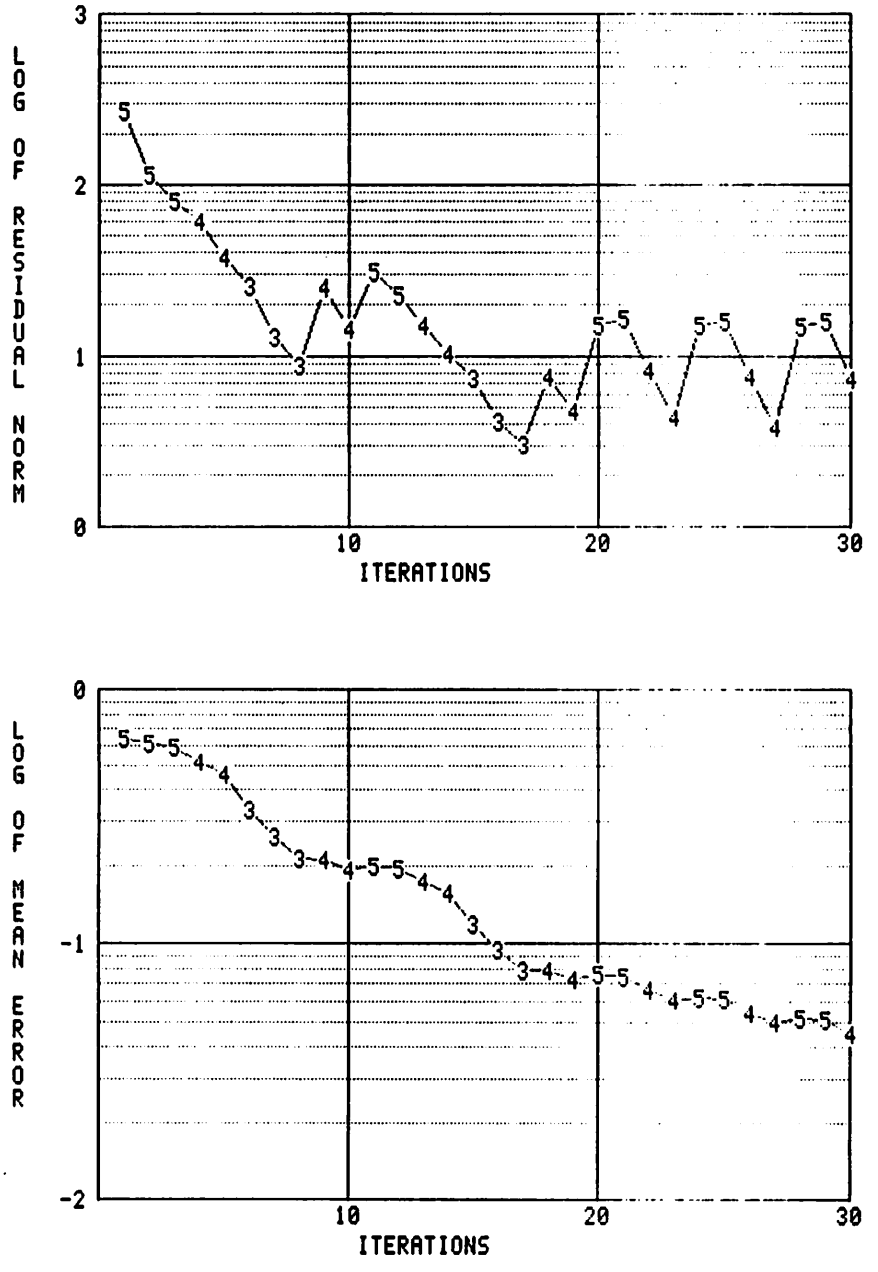

Figure 12: Multilevel relaxation: low frequency data

The next experiment shows how multilevel relaxation fares on the low frequency data. Figure 12 contains the error graphs for this experiment. Compare them to the corresponding experiment for higher frequency data shown in Figure 7. (Note the different ranges of the iteration count axis in these two experiments.) Two points are easily made. First, for the case of lower frequency data, the hierarchical control strategy found it necessary to go to level 3 twice, while for the earlier experiment level 4 proved sufficient. (In both of these experiments, the hierarchical control parameters would have allowed reduction up to level 2.) Secondly, in both cases, convergence (in terms of the residual norm) was reached after about 20 iterations. It is interesting to note that while the single-level relaxation takes longer to converge on the lower frequency data, the multilevel algorithm made use of a yet coarser level to achieve comparable convergence in the same number of iterations used for multilevel solution of the high frequency case.

Now, we come back to the case of rotational motion, this time with the lower frequency data. The error graphs are shown in Figure 13 and should be compared to the comparable higher frequency data experiment shown in Figure 10. As expected, for the lower frequency data, we no longer see divergence at level 3. We do have the remaining problem that at later visits to level 3 convergence eventually slows to where the target residual norm (for that visit) may not be reached. This also occurred in an experiment with high frequency data, but that time at level 4. In both cases, a simple limit on consecutive iterations at one level would prevent getting stuck. This will be implemented in the fixed cycle scheme in Section 8.

8 Further Experiments

In the last section, we established the need to avoid coarse level relaxation when fast local variations in the image data at those coarse levels causes divergence. In this section, we show how coarse level divergence is avoided by changing the hierarchical control strategy to a fixed cycling scheme, i.e. the decisions to move up or down (reduce or project) are made not by a dynamic tracking of the residual norm, but rather by a fixed pattern of up/down moves.

The fixed pattern hierarchical control strategy is important to us for two reasons. The first relates to the divergence problem and the related problem of slowed convergence at coarse levels. By limiting the number of iterations during any one visit to a given level, hierarchical control cannot get stuck at coarse levels when convergence slows at later stages of the overall relaxation process. Moreover, if in fact any divergence at coarse level is seen, then limiting iterations also limits the extent to which this divergence corrupts finer-level solutions.

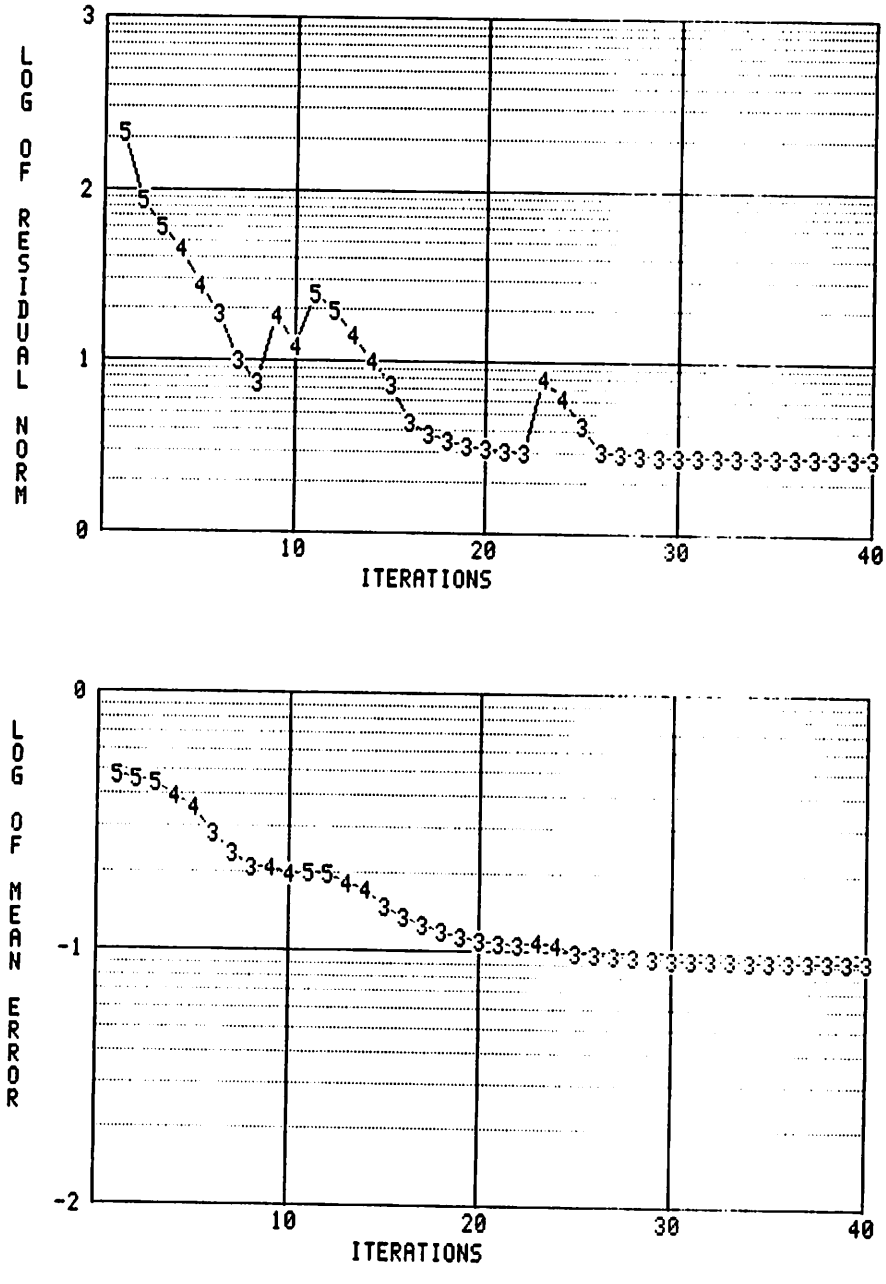

Figure 13: Multilevel relaxation: low frequency data, rotational field

The second benefit of a fixed cycling scheme is that it is no longer necessary to measure the rate of convergence of the residual norm. This is very important for multilevel algorithms run in a parallel architecture. The measurement of residual norm is a global computation, since it is defined as a summation over an entire level of residuals at individual pixels. A dynamic control strategy which makes reduce/project decisions based on the residual norm must at each iteration collect individual residuals together somewhere. In a locally connected architecture, such a global operation introduces high communication costs.

In the first two experiments, we use the "high frequency, rotational field" image, that is, the first frame is that shown in Figure 3, and the second frame is a .05 radian rotation of the first frame image. Recall that both images have one percent uniform noise added in, uncorrelated between the two frames. The experiment shown in Figure 10 used this data.

The fixed pattern of hierarchical control that we have used in all of the next experiments is a "three up, two down" pattern, that is, after a reduction to a coarser level three iterations are performed and after a projection to a finer level two iterations are performed. For the first experiment, reduction is only allowed up to level 4 and hierarchical control simplifies to an alternating pattern of two iterations at level 5 (after an initial three) and three iterations at level 4. The error graphs for this experiment are shown in Figure 14.

For the second experiment, reduction is allowed up to level 3 and the "three up/two down" pattern of hierarchical control is seen. The error graphs for this experiment are shown in Figure 15. For both of these experiments, we see a fine-level rotation field extracted with high accuracy at the finest level — level 5. In the first case, at iteration 33, the mean error magnitude goes to .031 pixel-lengths at level 5. In the second case, at iteration 43, the mean error magnitude goes to .037 pixel-lengths at level 5.

In both cases, the resultant vector fields at all levels are clearly seen to be very good approximations to rotational fields. This is shown for the second experiment in Figure 16. For comparison, refer back to the initial solution estimate (iteration 0 at level 5) for this experiment shown in Figure 8. Solution estimates at later stages in the relaxation are shown in Figure 16. Iteration numbers correspond to those shown in the graphs in Figure 15. The rotation field is seen clearly at all levels.

In the third experiment, the .05 radian rotation is applied to the lower frequency data. This is the same image data as used in the experiment shown in Figure 13. In that prior experiment, convergence flattened at level 3 and control did not pass back to the finer levels. In the experiment shown in Figure 17, return to the finest level is assured by the fixed pattern of hierarchical control. Convergence to less than .1 pixel-lengths at the finest level is attained by the 32nd iteration during the fourth visit to level 5.

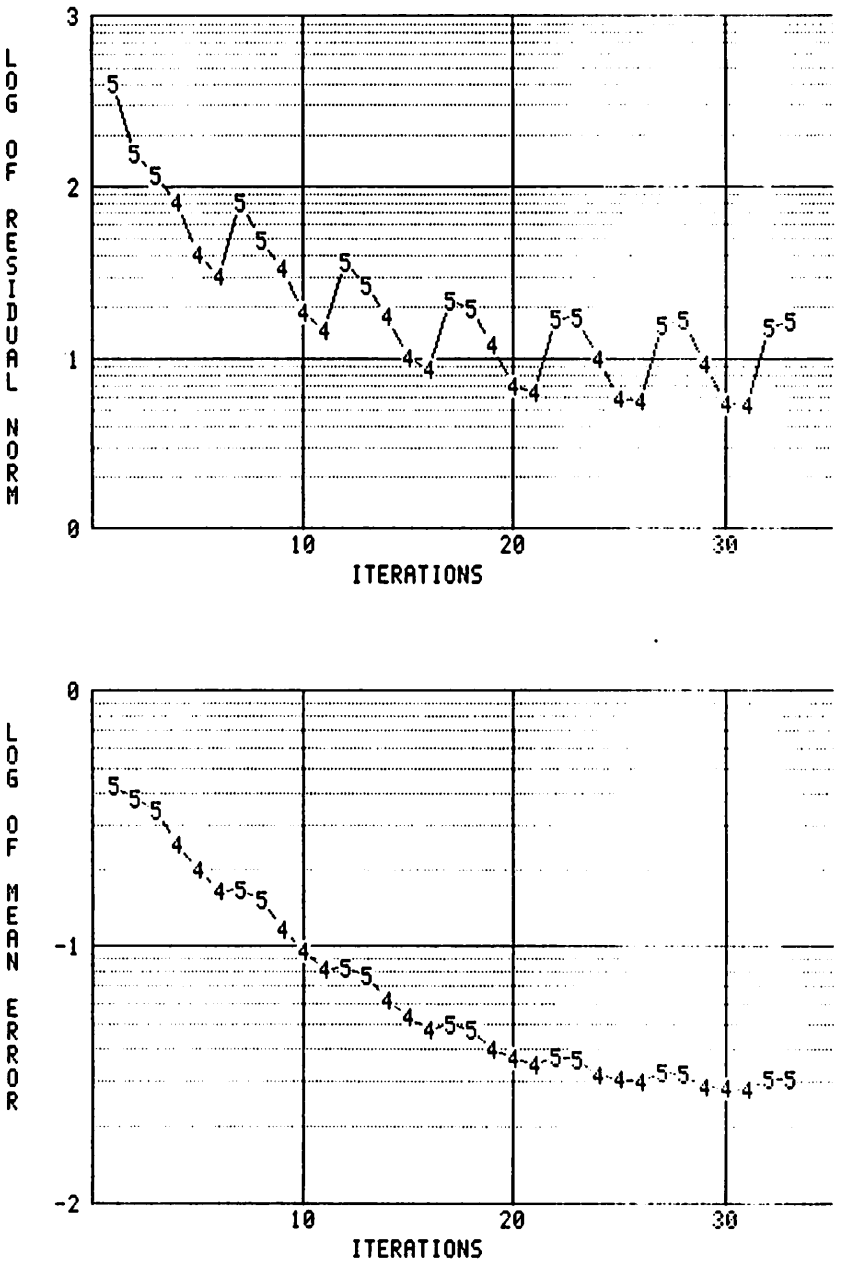

Figure 14: Fixed up/down hierarchical control pattern

The higher frequency data shown in Figure 3 is used with the second frame a 0.05 radian rotation of the first frame data. Reduction was allowed only up to level 4.

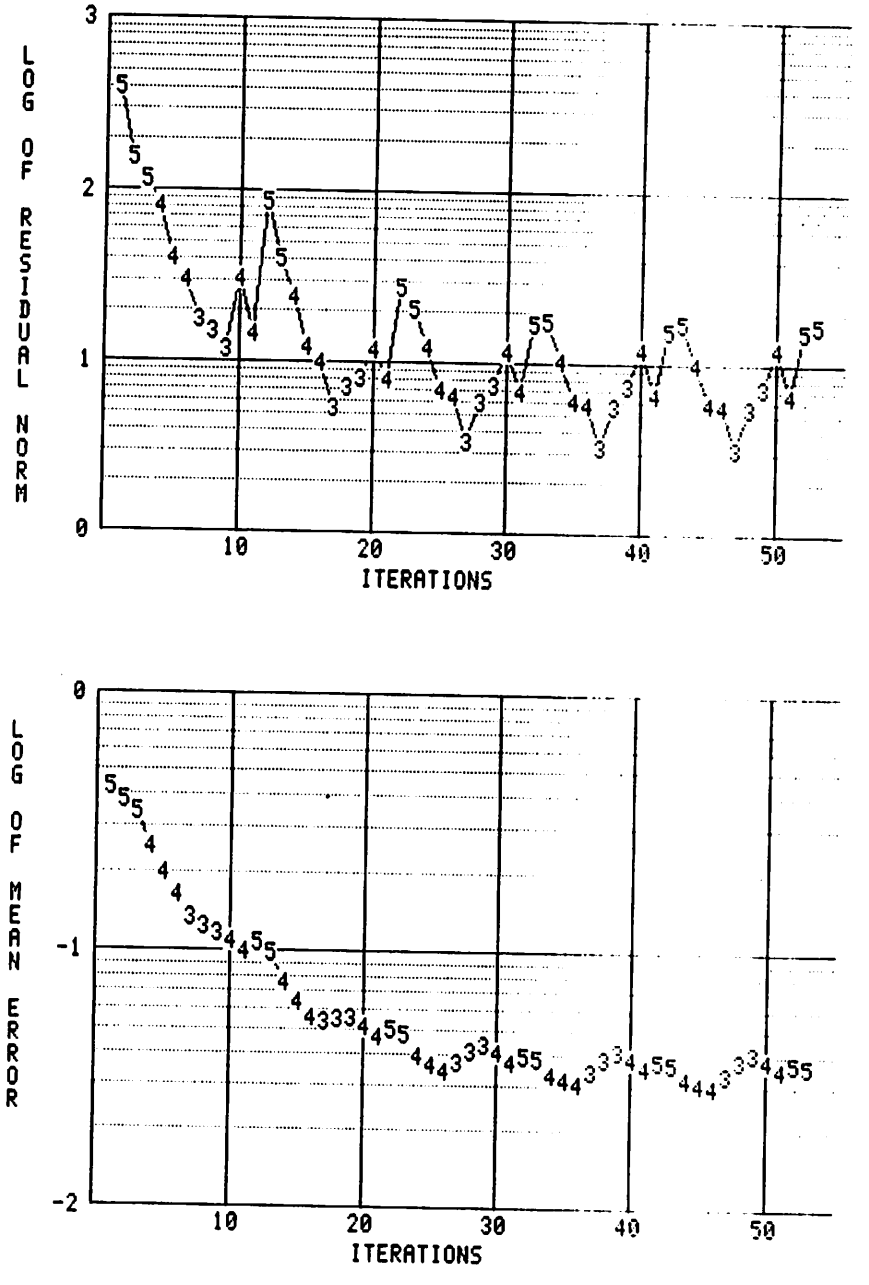

Figure 15: Fixed up/down control, to divergent levels

The image data for this experiment is identical to that used for Figure 14. Reduction was allowed up to level 3.

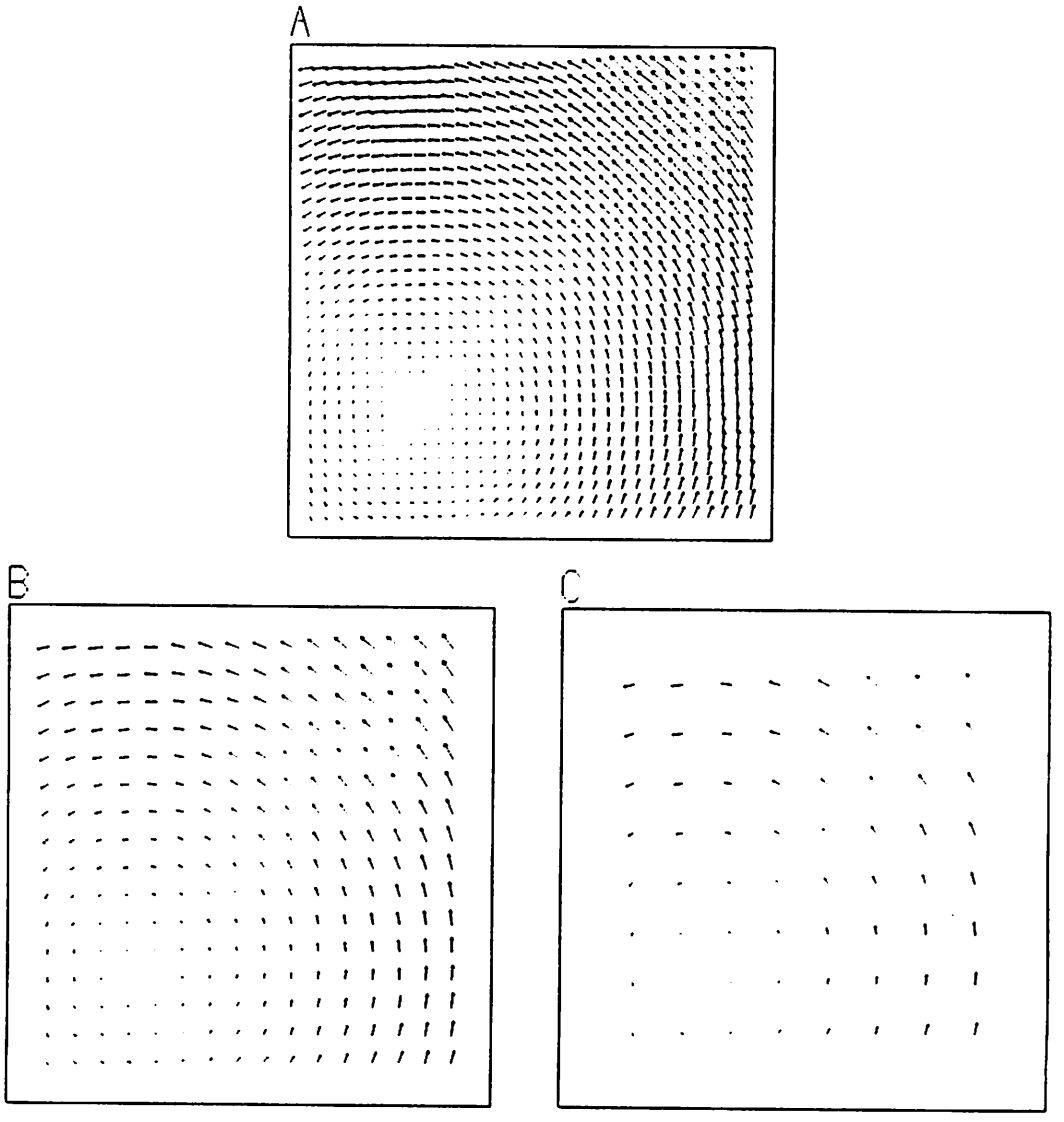

Figure 16: Rotational solution fields

These vector fields are solution estimates for the experiment with error graphs shown in Figure 15.

(a) The fine-level (level 5) solution at iteration 23. The mean error magnitude at this iteration (Figure 15b) is .048 pixel-lengths.

(b) The level 4 solution at iteration 21. The mean error magnitude at this iteration is .047 level-5-pixel-lengths.

(c) The level 3 solution at iteration 19. The mean error magnitude at this iteration is .056 level-5-pixel-lengths.

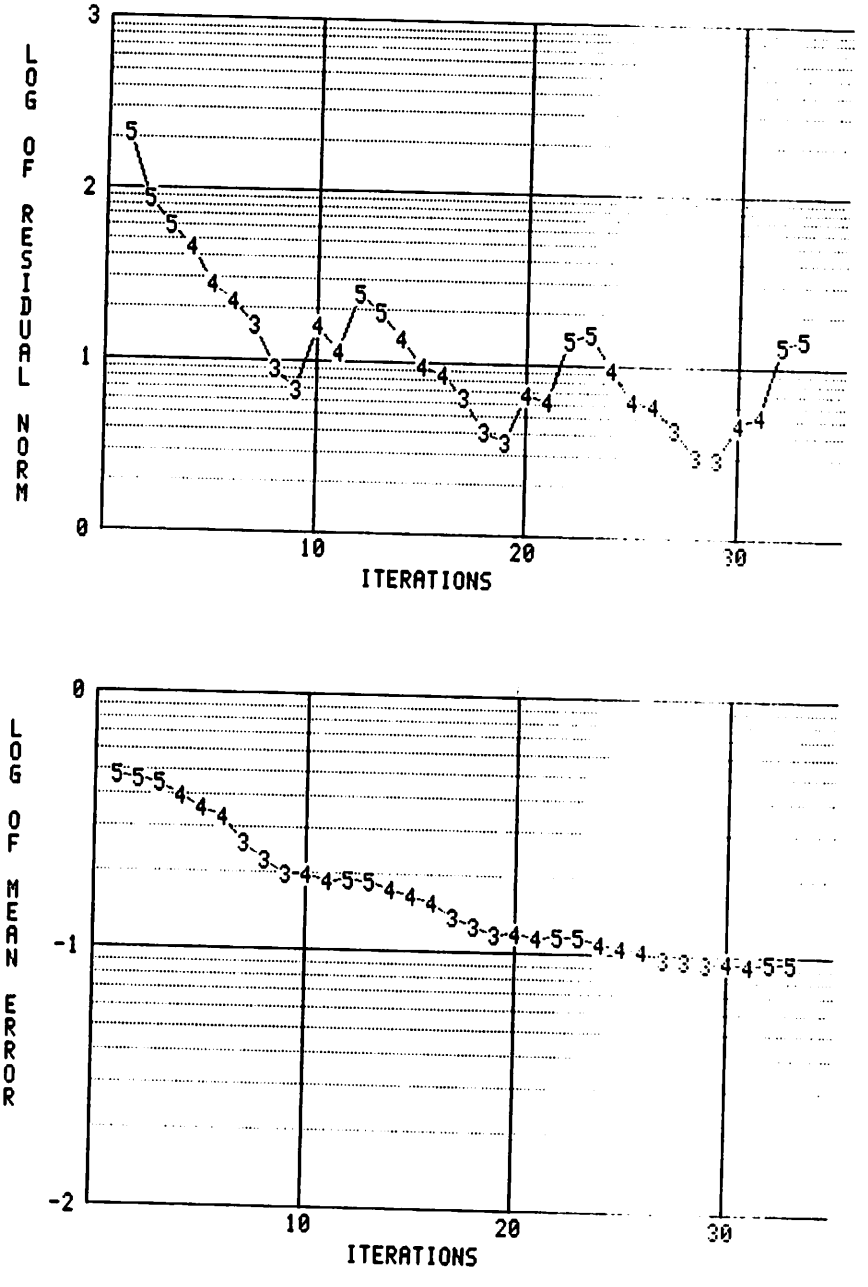

Figure 17: Fixed up/down control, low frequency data

The image data for this experiment is the low frequency data with a rotational flow field, the same data used in the experiment shown in Figure 13. Reduction was allowed up to level 3.

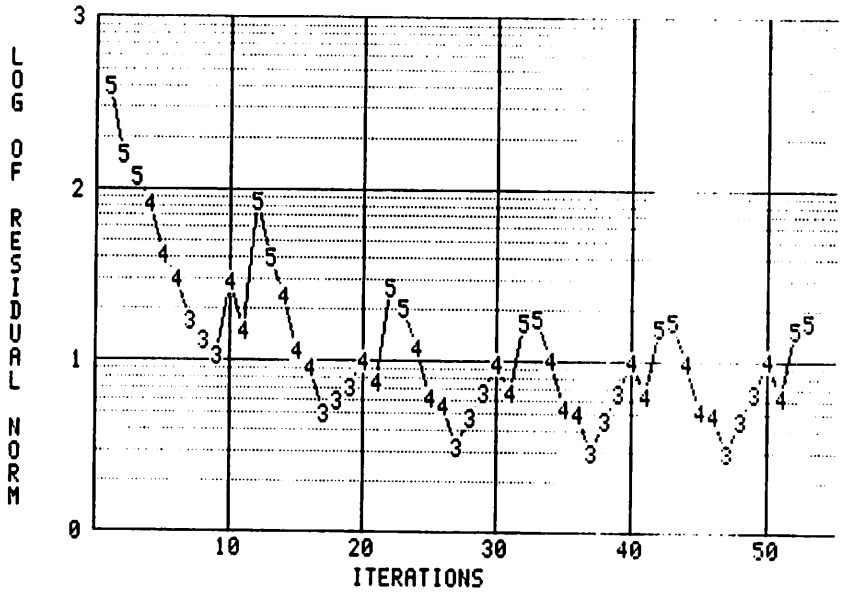

Figure 18: Multilevel relaxation: motion in depth

The higher frequency data shown in Figure 3 is used with the second frame, a 5% upscaling of the first frame data. Reduction was allowed only up to level 4. Residual norm is plotted vs. iteration number.

In the final experiment, multilevel relaxation with a fixed hierarchical control strategy is applied to another type of non-translational motion field. In this case, the second frame was computed to be a scaling up of the first frame by 5% about the point 1/4 in from the left and bottom sides. (Such a scaling corresponds to depth motion, i.e. translation perpendicular to the image plane.) Figure 18 shows results comparable to those for the rotational case shown in Figure 15. The edge flow used as the initial optic flow estimate is shown in Figure 19.

Solution estimates at later stages in the relaxation are shown in Figure 20. The resultant vector fields at all levels are clearly seen to be very good approximations to scaling fields. Iteration numbers correspond to those shown in the graphs in Figure 18. The scaling field is seen clearly at all levels.

9 Summary

In this paper, we have shown how multilevel relaxation can be used to compute optic flow, using significantly fewer iterations than single-level relaxation. After a relatively straightforward application of Brandt's ideas, we arrived at a multilevel optic flow relax-

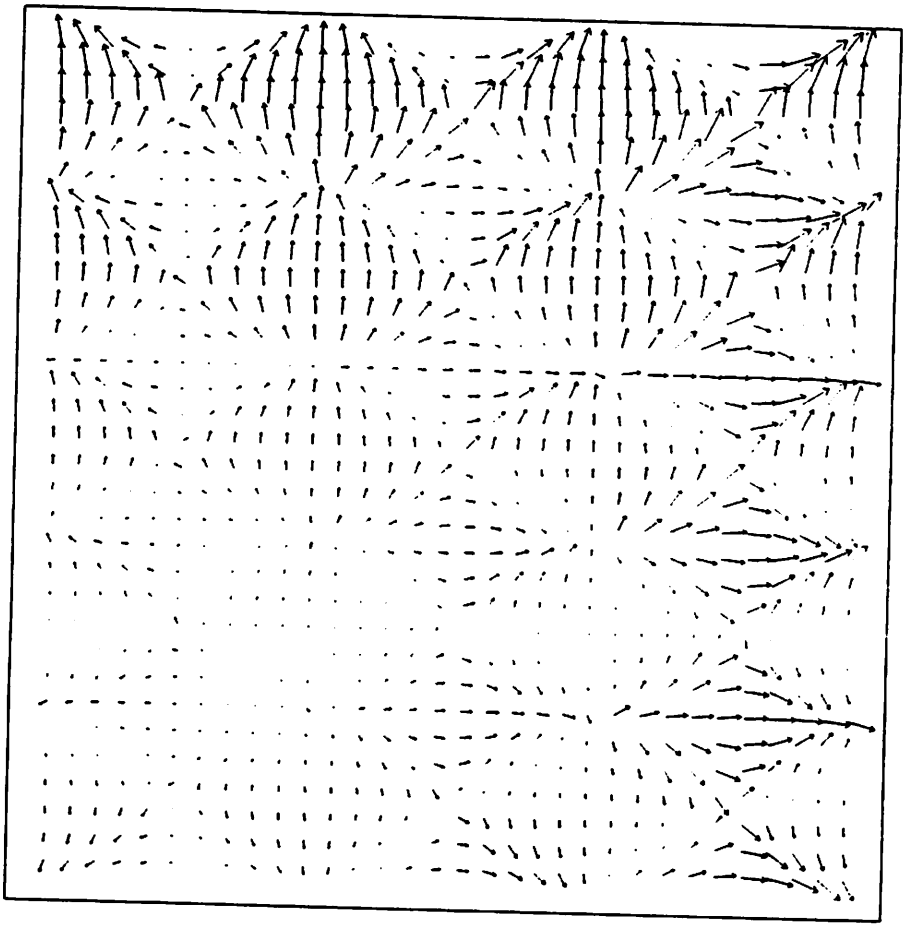

Figure 19: Scaling edge flow

The edge flow field for the depth motion data as given by Equation 38. This vector field is used as the initial estimate of optic flow.

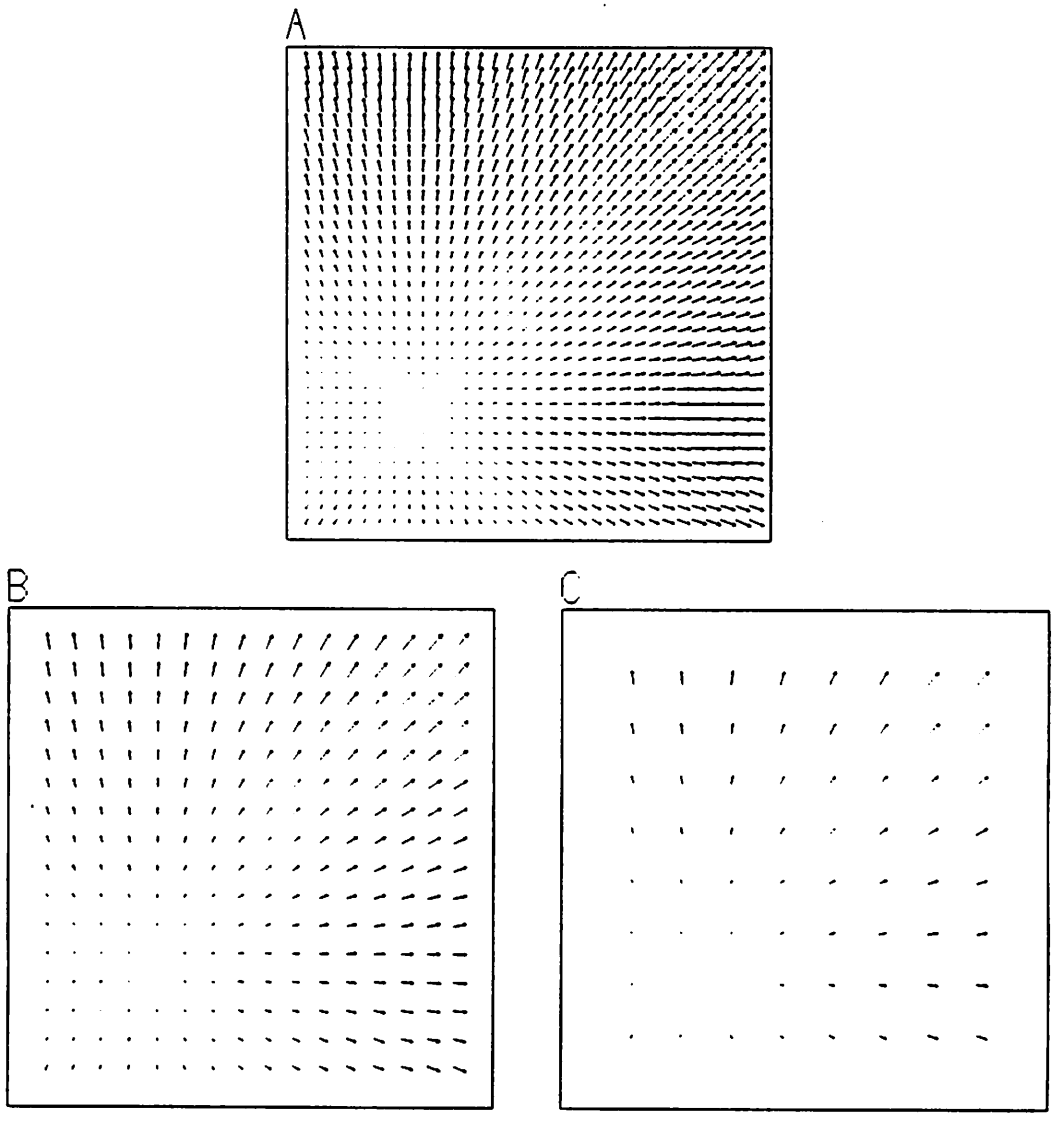

Figure 20: Scaling solution fields

These vector fields are solution estimates for the experiment with error graphs shown in Figure 18.

(a) The fine-level (level 5) solution at iteration 23.

(b) The level 4 solution at iteration 21.

(c) The level 3 solution at iteration 19.

ation algorithm. While exhibiting the expected increased convergence rate over single-level relaxation, some experiments presented a problem of divergence at coarse levels. A local mode analysis of the update equations showed how (1) convergence is at least as fast as simple smoothing; and (2) when the gradient is strong, convergence is accelerated towards the constraint line. However, the local mode analysis did not account for the coarse-level divergence seen in experiments. This divergence was then shown to be due to spatial variation in the image data. That is, when significant variations were on a scale comparable to the pixel spacing at a given coarse level, then divergence was seen. Finally, fixed cycling schemes were introduced as hierarchical control strategies to overcome the divergence problem. Further experiments showed the success of such a method.

References

[Birkhoff & Lynch 84] Birkhoff, G., and Lynch, R.E., *Numerical Solution of Elliptic Problems*, SIAM, Philadelphia, 1984.

[Brandt 77a] Brandt, A., Multi-Level Adaptive Solutions to Boundary-Value Problems, *Mathematics of Computation* **31**(138):333–390, 1977.

[Brandt 77b] Brandt, A., Multi-Level Adaptive Techniques (MLAT) for Partial Differential Equations: Ideas and Software, in *Mathematical Software III*, J.R. Rice (Ed.), Academic Press, 1977.

[Brandt 80a] Brandt, A., Multi-Level Adaptive Finite-Element Methods: I. Variational Problems, In *Special Topics of Applied Mathematics*, Freshe, J., Pallaschke, D., and Trottenberg, U. (Eds.), North Holland, New York, 1980.

[Brandt 80b] Brandt, A., Multigrid Solvers on Parallel Computers, ICASE Report No. 80-23, NASA Langley Research Center, Hampton, Virginia, 1980. Also in *Elliptic Problem Solvers*, M.H. Schultz (Ed.), Academic Press, New York, 1981.

[Courant & Hilbert 53] Courant, R. and Hilbert, D., *Methods of Mathematical Physics, Volume I*, Interscience Publishers, Inc., New York, 1953.

[Enkelmann 86] Enkelmann,W., Investigations of Multigrid Algorithms for the Estimation of Optical Flow Fields in Image Sequences. *Proc. Motion Workshop* pp. 81-87, Kiawah Island, S.C., May 1986.

[Fennema & Thompson 79] Fennema, C.L. and Thompson, W.B., Velocity Determination in Scenes Containing Several Moving Objects, *Computer Graphics and Image Processing* **9**(4):301–315, 1979.

[Glazer 81] Glazer, F., Computing Optic Flow, *Proc. 7th. Int. Joint Conf. on Artificial Intelligence*, pp. 644–647, Vancouver, BC, 1981.

[Glazer 82] Glazer, F., Multilevel Relaxation in Low Level Computer Vision, COINS Tech. Report 82–30, U.Massachusetts, Amherst, 1982. Also in: *Multiresolution Image Processing and Analysis*, A. Rosenfeld (Ed.), Springer-Verlag, 1983.

[Glazer 87] Glazer, F., Hierarchical Motion Detection, Ph.D. Thesis and COINS Tech. Report 87-2, U.Massachusetts, Amherst, 1987.

[Grimson 81] Grimson, W.E.L., A Computational Theory of Visual Surface Interpolation, A.I. Memo 613, MIT AI Lab, Cambridge, MA, June, 1981.

[Hackbusch 85] Hackbusch, W., *Multi-Grid Methods and Applications*, Springer-Verlag, New York, 1985.

[Hageman & Young 81] Hageman, L.A. and Young, D.M., *Applied Iterative Methods*, Academic Press, New York, 1981.

[Hanson & Riseman 74] Hanson, A. and Riseman, E.M., Preprocessing Cones: A Computational Structure for Scene Analysis, COINS Tech. Report 74C-7, U.Massachusetts, Amherst, 1974.

[Hanson & Riseman 80] Hanson, A. and Riseman, E.M., Processing Cones: A Computational Structure for Image Analysis. In: *Structured Computer Vision*, Tanimoto, S. and Klinger, A. (Eds.), Academic Press, New York, 1980.

[Haralick & Lee 83] Haralick, R.M., and Lee, J.S., The Facet Approach to Optic Flow, *Proc. Image Understanding Workshop (DARPA)*, pp. 84-93, June, 1983.

[Horn & Schunck 81] Horn, B.K.P. and Schunck, B.G., Determining Optical Flow, *Artificial Intelligence* **17**(1-3):185-204, 1981. Also in *Proc. Image Understanding Workshop (DARPA)*, April, 1981. Also A.I. Memo 572, MIT AI Lab, Cambridge, MA, April, 1980.

[McCormick & Trottenberg 83] McCormick, S. and Trottenberg, U. (ed.), Multigrid Methods, special issue of *Applied Mathematics and Computation*, **13**(3 & 4), November, 1983.

[Nagel 83] Nagel, H.-H., Displacement Vectors Derived from Second Order Intensity Variations in Image Sequences, *Computer Vision, Graphics and Image Processing* **21**:85-117, 1983.

[Narayanan *et al.* 82] Narayanan, K.A., D.P. O'Leary and A. Rosenfeld, Image Smoothing and Segmentation by Cost Minimization, *IEEE Trans. Systems, Man, and Cybernetics* **12**(1):91-96, 1982.

[Poggio *et al.* 85] Poggio, T., Torre, V., Koch,C., Computational Vision and Regularization Theory, Nature **317**(26):314-319, 1985.

[Rice 83] Rice, J.R., *Numerical Methods, Software, and Analysis*, McGraw-Hill, NewYork, 1983.

[Rosenfeld 83] Rosenfeld, A. (Ed.), *Multiresolution Image Processing and Analysis*, Springer-Verlag, 1983.

[Smith 78] Smith, G.D., *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, Oxford, 1978.

[Terzopolous 82] Terzopolous, D., Multi-Level Reconstruction of Visual Surfaces: Variational Principles and Finite Element Representations, A.I. Memo 671, MIT AI Lab, Cambridge, MA, 1982. Also in: *Multiresolution Image Processing and Analysis*, A. Rosenfeld (Ed.), Springer-Verlag, 1983.

[Terzopolous 84] Terzopolous, D., Multiresolution Computation of Visual-Surface Representations, Ph.D. Thesis, MIT, Dept. of EE & CS, 1984.

[Terzopolous 86] Terzopolous, D., Image Analysis Using Multigrid Relaxation Methods, *IEEE Trans. Pattern Analysis and Machine Intelligence* 8(2):129–139, 1986.

[Varga 62] Varga, R.S., *Matrix Iterative Analysis*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1962.