CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Robust Computation of Optical Flow in a Multi-Scale Differential Framework

Joseph Weber
Jitendra Malik

UCB-ITS-PWP-93-4

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

July 1993

ISSN 1055-1417

Robust Computation of Optical Flow in a Multi-Scale Differential Framework

Joseph Weber and Jitendra Malik

Department of Electrical Engineering and Computer Science
University of California at Berkeley
Berkeley, CA 94720
email: jweber@eecs.berkeley.edu, malik@eecs.berkeley.edu

July 28, 1993

Abstract

We have developed a new algorithm for computing optical flow in a differential framework. The image sequence is first convolved with a set of linear, separable spatiotemporal filters similar to those that have been used in other early vision problems such as texture and stereopsis. Our analysis of the measurement errors leads us to develop an algorithm based on a robust version of total least squares. Each optical flow vector computed has an associated reliability measure which can be used in subsequent processing. The performance of the algorithm on the data set used by Barron et al. (CVPR 1992) compares favorably with other techniques. In addition to being separable, the filters used are also causal, incorporating only past time frames. The algorithm is fully parallel and has been implemented on a multiple processor machine.

By being fully parallel, the algorithm can be performed by an array of processors in real time. In addition, the differential method is computationally less expensive than matching methods for computing visual motion. The output of the linear filters can also be used in other visual tasks such as stereo and recognition. Thus, this approach to motion detection can be part of a real time vision application system in which linear filters provide a basis for visual tasks such as passive ranging and moving object detection. For vehicle surveillance, the system provides individual vehicle speeds and directions. For autonomous vehicles, the system would provide both stereo correspondence for range information and optical flow for collision avoidance in a single computational framework.

Keywords: optical flow, differential approach, brightness constancy assumption, total least squares, multi-channel filtering, robust statistics, real time vision

1 Introduction

A number of different approaches to recovering optical flow have been proposed: those based on correlation, energy and differential considerations. A recent survey is due to Barron et al. [1, 2] where the different approaches were compared on a series of synthetic and real images. They found that a phase-based approach by Fleet and Jepson [3] performed the best numerically.

We have developed a new algorithm for computing optical flow in the differential framework which performs comparably to the Fleet and Jepson approach but with less computational cost and a higher density of estimates. We start with a multi-channel filtering of the intensity response, thus producing an overconstrained system of equations in the motion parameters. The convolution with a series of filters is a common starting point for a number of early vision tasks such as edge detection, stereo and texture discrimination [4, 5, 6, 7]. In Section 2 we analyze the sources of error in the differential method as falling into 3 categories: (a) stochastic error due to sensor noise, (b) systematic error due to large displacements and (c) model error where the underlying model is violated. This analysis leads to an algorithm based on a robust version of total least squares. This algorithm is outlined in Section 3.

The implementation is described in Section 4. In Section 5 the algorithm is run on the series of synthetic and real sequences used by Barron et al. Thus a direct comparison between our work and others can be made. A high density of estimates was found for all sequences, implying that the "aperture problem" occurs rarely in most images. A confidence measure is available as a byproduct of the total least squares formulation. Through a simple experiment we demonstrate that this measure is related to the estimated accuracy of the motion vector. We also look at a scale pyramid implementation of the filter responses in this section to demonstrate that this more efficient method of computing multiple scale responses does not degrade the performance significantly.

Finally, in Section 6 we describe a parallel implementation on a multiple processor and examine the speedup of the algorithm. This demonstrates that a real-time parallel version of the algorithm may be possible.

2 The Differential Constraint Equation

A single constraint equation for the components of the optical flow is derived from the constancy assumption. The constancy assumption assumes that some function of the brightness is constant between time frames.

$$I(x, y, t) = I(x + u\delta t, y + v\delta t, t + \delta t) \tag{1}$$

where $I(x, y, t)$ is the brightness or some function of the brightness at location (x, y) and time t. The vector field $\vec{v} = (u, v)$ is the optical flow and is a function of image coordinates (x, y). We assume a differentiable brightness function and make a Taylor expansion of the right hand side of equation (1):

$$I_x u\delta t + I_y v\delta t + I_t \delta t = \mathcal{O}(\vec{v}^2 \delta t^2) \tag{2}$$

where I_x, I_y and I_t are partial derivatives with respect to space and time, evaluated at the point (x, y, t). The right hand side represents the remaining terms of the Taylor expansion. This term contains products of higher spatial and temporal derivatives of the brightness function as well as higher powers of the displacements. By assuming a unit temporal delay, δt, this equation can be written in a more compact form:

$$\nabla I \cdot \vec{v} + I_t = \mathcal{O}(\vec{v}^2) \tag{3}$$

The right hand side is usually assumed small and set to zero. The validity of ignoring the right hand side of equation **(2)** is dependent on the spatial frequency content of the intensity pattern and the magnitude of the displacement.

This constraint equation has been used in motion detection **for** some time [8]. It consists of a single equation in the two unknowns which forms a single constraint line in velocity space. Any velocity on this line is valid for this single equation. This was called the "aperture problem'' since for special cases the velocity could not be determined uniquely. Horn and Schunck [9] introduced a smoothness constraint in order to solve uniquely for displacement. A number of other authors [10, 11, 12, 13, 14] produced two or more linear equations in u and v by assuming constancy of partial derivatives and other functions of the intensity. A third approach [15, 16] is to assume the velocity field is locally constant and to combine constraint equations from neighboring pixels. A review of these and other approaches such as correlation and energy models can be found in [1].

In our approach, we first convolve the image sequence with a set of linear spatio-temporal filters, $f_i(x,y,t)$. These are Gaussian derivatives of first **or** second order at a number of orientations and scales. These are the same filters used in some other approaches to early vision, such as in stereo and texture discrimination [5, 6, 7]. Each convolved image, $\mathbf{I}_i = I * f_i$, has its own constraint equation of the form **(2).** This results in an overconstrained system of equations in the unknowns u and v.

$$\begin{pmatrix} I_{1x} & I_{1y} \\ I_{2x} & I_{2y} \\ . & . \\ I_{nx} & I_{ny} \end{pmatrix} \cdot \vec{\mathbf{v}} = \begin{pmatrix} -I_{1t} \\ -I_{2t} \\ . \\ -I_{nt} \end{pmatrix} + \vec{\mathcal{O}}(\vec{\mathbf{v}}^2) \tag{4}$$

The spatio-temporal partial derivatives, I_{ix}, I_{iy}, I_{it}, can be considered to be the result of convolution of the image sequence I with linear, spatio-temporal filters since $I_{ix} = (\mathbf{I} * f_i)_x = I * f_{ix}$. Defining the matrix $\mathbf{A}$ and vector $\vec{\mathbf{I}}_t$ the system of equations can be written as:

$$\mathbf{A} \cdot \vec{\mathbf{v}} = -\vec{\mathbf{I}}_t + \vec{\mathcal{O}}(\vec{\mathbf{v}}^2) \tag{5}$$

The spatial extent of the filters brings in information from neighboring pixels so the aperture problem exists only for degenerate cases.

The fundamental problem now is to solve the overconstrained system of equations **(5)** so as to obtain as accurate an estimate of $\vec{\mathbf{v}}$ as possible. We begin by analyzing the sources of error.

1. ***Stochastic error.*** In the presence of sensor noise, we expect that the measurements of I_{ix}, I_{iy}, I_{it}, the spatiotemporal derivatives of $I * f_i$, would be corrupted with noise. We will make the standard convenient assumption that sensor noise is independent from pixel to pixel and has a Gaussian distribution. This is analyzed further in subsection **2.1.**

2. ***Systematic error for large displacements.*** The system of equations **(5)** is derived by neglecting second order terms, so we expect systematic errors whenever the local velocity is large. The magnitude of the error is dependent on a number of factors including the scale of the filter f_i being used and the local spatial frequencies present in the image neighborhood. This is analyzed further in subsection **2.2.**

3. ***Errors dole to model failure.*** In subsection **2.3** we group together the errors that arise due to violation of certain key assumptions of the differential approach: (a) Constancy of image brightness which is

not strictly true whenever there is a significant specular component, and (b) that the optical flow field is locally constant over the support of the filters, which is not true if the filter support straddles a depth discontinuity or when there is a significant rotational or dilational component in the flow field.

2.1 Stochastic Error and Total Least Squares

If we knew that the errors were confined to the measurements of I_{ti}, i.e. the right hand side of the system (5), then the correct approach is well known from estimation theory. We find the classical weighted least squares solution which from the Gauss-Markov theorem is the best one can do. [1] The weight matrix can be determined by examining the covariance matrix of the filters f_{ti}.

However, the classical least squares method makes the implicit assumption that the measurements on the left hand side I_{xi}, I_{yi} are error-free and that the errors are confined to the measurements on the right hand side It_i. This assumption is not true, impelling us to use the ***total least squares*** method. Total least squares is also known as *orthogonal* ***regression*** or *errors-in-variables* ***regression*** [17].

The essential difference between classical least squares and total least squares can be made clear by a simple 1-D example. Suppose we wish to fit a line to a group of points, (x_i, y_i). In classical least squares we wish to find the values of the slope and intercept, (m, b) which minimizes the sum squared difference between the y_i and the predicted y.

$$\min_{b,m} \sum (y_i - mx_i - b)^2 \tag{6}$$

This minimizes the vertical distances between the line and the measurements y_i It assumes the variables x_i are error free and all noise is contained in the y_i. Total least squares allows for errors in the x_i variables too. It wishes to minimize the perpendicular distance between the line and the measured points (see Figure 1). This was referred to as ***eigenvector fit*** in [18]. The idea of allowing errors in all variables when fitting data has been around for some time [19, 20]. The concept was extended to multivariate problems about 20 years ago [21]. The connection to the singular value decomposition of the measurement matrix was pointed out by Golub and Van Loan [22] and Van Huffel and Vandewalle [17].

In the total least squares framework, (5) is usually written as

$$\left[\mathbf{A}|\vec{\mathbf{I}}_t\right] \begin{pmatrix} \vec{\mathbf{v}} \\ 1 \end{pmatrix} = \vec{\mathbf{0}} \tag{7}$$

The combined matrix $\left[\mathbf{A}|\vec{\mathbf{I}}_t\right]$ is referred to as the measurement matrix. This form recognizes that each entry in the measurement matrix is subject to noise. In total least squares, an estimate is found by making the smallest, in terms of its Frobenius norm, perturbation to the measurement matrix such that (7) has a solution [17]. This is in contrast to least squares where only the measurement vector $\vec{\mathbf{I}}_t$ is perturbed to find a solution. The estimate using total least squares is

$$\vec{\mathbf{v}} = -(\mathbf{A}^T\mathbf{A} - \sigma_3^2\mathbf{I})^{-1}\mathbf{A}^T\vec{\mathbf{I}}_t \tag{8}$$

where σ_3 is the smallest singular value of the measurement matrix. The Frobenius norm of the perturbation needed to make (7) consistent is simply σ_3. Equation (8) is very similar to the standard least squares solution. The latter is obtained by setting σ_3 to zero.

[1] Of course, we all know that this is just a consequence of assuming that the sensor noise has a Gaussian distribution, an assumption rarely verified in practice. One appeals to the Central limit theorem and hopes for the best.

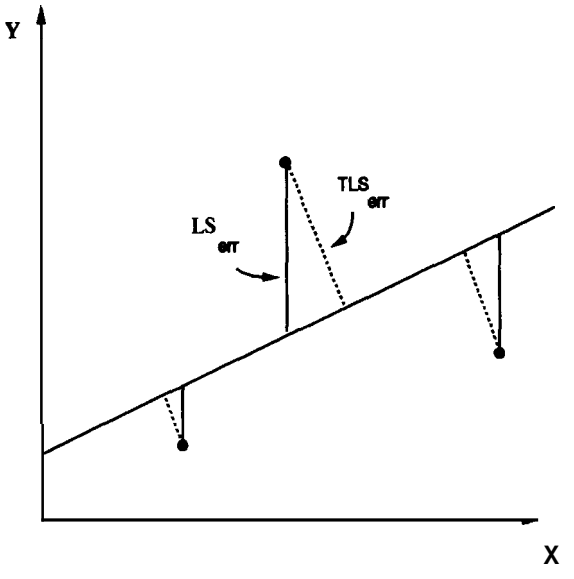


Figure 1: Difference between least squares and total least squares for fitting a line to a collection of points. Least squares assumes the errors are in the y variables and thus minimizes the vertical distance between the line and the points. Total least squares allows for errors in both x and y and thus minimizes the perpendicular distance between the line and the points.

Total least squares assumes the error in each element of the measurement matrix is independent and identically distributed (the error matrix is white). If this is not the case, total least squares can actually perform worse than standard least squares. This is similar to the requirement in standard least squares that the errors in the measurements be normalized. We can use prior estimates of the measurement variances to whiten the measurement matrix.

2.2 Systematic Errors due to large displacements

The systematic error term in the constancy equation (2) is often ignored. However this term can easily be larger than the stochastic error for relatively small displacements. The constraint equations makes a linear approximation to the underlying intensity function and is thus invalid for large displacements. The assumption breaks down quadratically in the displacements. If we operate in a single spatial dimension we can examine the relative magnitude of this term. If the signal is a simple sinusoid, it is obvious that the linear approximation is valid only for a fraction of the wavelength of the sinusoid. (Figure 2). If the wavelength, λ, of the stimulus is known, we can limit the acceptable displacement between time frames to some fraction of the wavelength.

$$|v| < \rho X = 2\pi\beta/\omega \tag{9}$$

A natural limit is $\beta = 1/2$ since displacements greater than half a cycle would introduce aliasing. This limit on displacement as a function of stimulus frequency has a biological basis too. In random dot kinematograms it has been shown that the upper displacement, d_{max}, for coherent motion detection falls off as the inverse of frequency [23, 24]. The exact value of β above is in doubt however. Multiple scales and orientations (as in our model) may confuse the issue. An interesting debate on the exact value of β can be seen in [25] and [26]. From a numerical point of view, Battiti et al. [27] examined the systematic

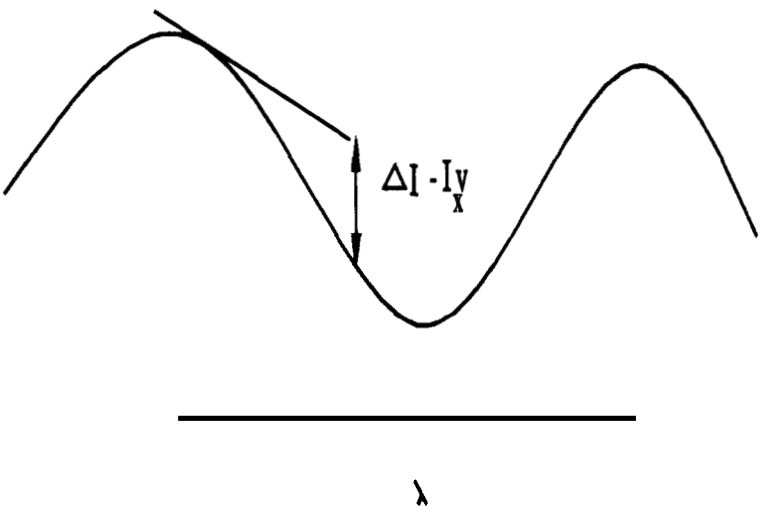


Figure 2: The linear slope approximation for a sinusoid is only valid for a fraction of a wavelength. The range of velocity estimates that a gradient-based approach can reliably detect is limited by the spatial frequencies of the underlying intensity function.

error implicit in gradient techniques which use finite differences to estimate partial derivatives. In the one dimensional case for a translating sinusoid of frequency w, they find that the estimated velocity, $\hat{v}$, as a function of the true velocity, v, is

$$6 = \sin(\omega v)/\sin(\omega) \tag{10}$$

The difference between v and 6 comes from the linear approximation of finite differences. The relative error for this component is

$$\left|\frac{6 - \text{v}}{v}\right| = 1 - \frac{\sin(\omega v)}{\text{v}\sin(\omega)} \tag{11}$$

Writing this as a function of β at the maximum velocity,

$$\left|\frac{6 - \text{v}}{\text{v}}\right| = 1 - \frac{\omega\sin(2\pi\beta)}{2\pi\beta\sin(\omega)} \tag{12}$$

For a reasonable range of frequencies, a value of β about $1/2\pi$ results in a fractional error of less than 10%. Thus we set $\beta = 1/2\pi$. The resulting limit on the displacements allowed is $|v| < 1/\omega$.

Unfortunately we do not know the spectrum of the intensity function. However in our implementation, the intensity function is convolved with a series of Gaussian and Gaussian derivative functions. These are either low **or** band-pass filters. Thus we know the expected frequency response of the stimulus. If we use **a** low pass filter with a cutoff frequency ω_c then the maximum velocity estimate which can be considered valid is $|\vec{\mathbf{v}}| < 1/\omega_c$ from the above analysis. The filters we use consist of Gaussian functions and their derivatives. The n'th derivative of a Gaussian of standard deviation σ has its maximum frequency response at $\omega = \sqrt{n}/\sigma$ [28]. If we use this frequency in limiting the maximum displacement we find that for the response formed by filtering with the n'th derivative of a Gaussian, the maximum displacement we can

accept without knowing the stimulus spectrum is

$$|\vec{\mathbf{v}}| < \sigma/\sqrt{n} \tag{13}$$

2.3 Systematic errors due to model failure

There are situations where the underlying assumptions of the model are violated. Constancy of image brightness (1) is not strictly true whenever there is a significant specular component **or** when occlusion occurs. The model also assumes that the optical flow field is locally constant over the support of the filters, which is not true if the filter support straddles a depth discontinuity **or** when there is a significant rotational or dilational component in the flow field. In these situations, regression is not valid and these measurements should be labeled as outliers.

When calculating the total least squares solution, the singular values of the measurement matrix are available. The smallest singular value, σ_3, is equivalent to the Frobenius norm of the perturbation needed to make the equations consistent. We define a consistency ratio $\frac{\sigma_3}{\sigma_2}$. This is the ratio of the norm of the perturbation to the smallest eigenvector of the resulting measurement matrix. When the assumptions are violated, a large relative perturbation will be needed to make the equations consistent. We discard scale groups which require a perturbation greater than a given threshold, C_t, to become consistent, assuming they are due to model failure. This discards the outliers before combining scales in a second total least squares. The ratio is scale independent and therefore can be used to compare estimates between scales.

3 Model

Based on the above analysis, our model for multi-scale motion analysis consists of two total least squares steps. The image is first convolved with a collection of filters. These filters are separated according to scale. Thus we may have m different filters each of the same scale but differing in orientation and frequency response, and n such groups of these filters. The common scales of these groups form a geometric sequence. If the smallest scale is of size σ_0, then the i'th scale is of size $\sigma_o^{(i-1)}$.

Partial derivatives are computed via finite differences and weighted according to known prior noise variances.

The n scale groups each individually form an estimate for the velocity via the total least squares formula in **(7).** This velocity estimate is deemed valid if its magnitude is less than the maximum allowed for that scale via equation **(13).** The estimate is also rejected if the ratio of the two smallest singular values of the measurement matrix is above the consistency threshold, C_t.

The remaining valid estimates are combined again via a total least squares formulation. The weights in this step have been divided by the consistency estimate of that scale's equations. This was the ratio of the two smallest singular values of the measurement matrix.

This two step method contains two elements which make it robust. First, the ratio threshold prevents scale groups with poor estimates from participating in the second stage. Secondly, those scales which do participate are weighted by their individual residuals. In many iterative robust techniques, the process of finding an estimate, weighting by updated covariances and repeating is common [29].

By weighting the second stage by the singular values ratio, we insure that the scale which most accurately estimates the motion has the strongest influence in the multi-scale fusion. This is in contrast to coarse-to-fine methods which assume larger scales have a correct but coarse estimate.

A consistency ratio for the combined scales is also available. This serves as a useful indicator **of** accurate optical flow. If this ratio is larger than the threshold C_t that pixel location is marked as not having an

estimate. We feel that this may indicate the presence of motion boundaries and our future work will investigate this. Figure 3 outlines the method.

4 Implementation

The input to the model is simply two response frames separated in time. These two inputs are created from a sequence of images via convolution with separable Gaussian functions. First, each frame of the sequence is convolved with spatial functions. The functions used were the standard normalized Gaussian function, edge and bar filters which consist of a derivative of a Gaussian along one spatial dimension and a standard Gaussian along the orthogonal spatial dimension, and the symmetric Laplacian of a Gaussian. These filters have been used to model receptive fields of neurons in the visual cortex [28]. It has been argued that they form a good basis for early vision tasks such as edge detection, stereo and texture discrimination [30, 6, 7]. The size of each Gaussian function was set so that each filter shared a common peak frequency response. Thus they shared a common scale, σ_0. Figure 4 is the spatial responses for the scale group of $\sigma_0 = 16$ pixels for the first 2 Gaussian derivatives and the Laplacian of a Gaussian. This set was the one most often used in the experiments. Notice that each filter has a zero DC response and thus is not influenced by global lighting changes.

The sizes of the scale groups followed the progression $\sigma_0 = 1, \sigma, \sigma^2, \ldots, \sigma^{n-1}$ where σ was 1.8 for the experiments. This is a natural scale space representation as used in pyramid implementations. A pyramid scheme can be used to decrease the computational load for the many convolutions required without a significant loss in performance (see Section 5.3).

Next, the filtered responses are convolved in the time dimension with a causal standard Gaussian. This is non-zero only for past time frames. The standard deviation of this Gaussian was set to 3 video frames in order to emulate human response curves which show temporal recruitment up to about 100 milliseconds. Thus only the past 10 frames contribute significantly to any filter response. These numbers could change depending on the frame rate or known motions.

The partial derivatives of the responses are computed through a forward finite difference cube. This is simply the average of 4 adjacent pixel forward finite differences. Since the stimulus was already convolved with Gaussian functions, we felt a more sophisticated scheme for obtaining first partials was not needed. The forward differences actually provide an estimate of the partial derivatives on a lattice which is offset one half pixel in each spatial dimension and one half of a frame in the temporal dimension.

If a given scale group contains adequate texture such that the condition number of the measurement matrix was finite, a velocity estimate for that scale group is computed. It is known that it is important to 'whiten' the measurement matrix such that each element is identically distributed and independent [17]. We assume that the output of each filter is independent. For the zero-DC filters used, the oriented filters within a scale group actually are orthogonal. The Laplacian is not a linear combination of the elongated second order derivatives, but it is not orthogonal. When the different scale groups are combined, the filters are not orthogonal, but the magnitudes of the interaction terms are smaller than the diagonal elements and are ignored. The partial derivative within a single equation however do not have the same noise distribution due to the elongated Gaussian filters used (i.e. $< I_x^2 > \# < I_y^2 >$). The partial derivative along the orientation of the filter has a higher noise response than the one perpendicular to the orientation. Since the oriented filters come in rotated pairs, a simple sum and difference of the two resulting equations produces two equations with partials of equivalent noise variance yet remain independent. The total least

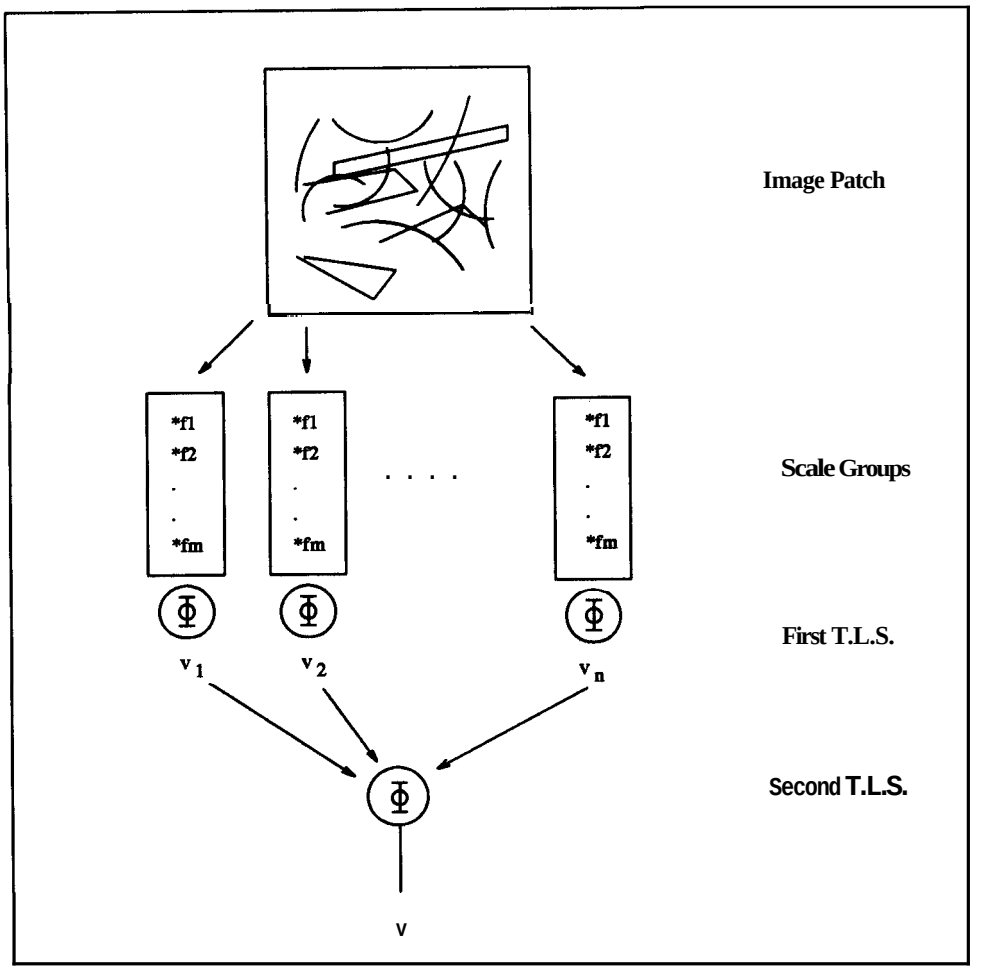


Figure 3: Multi-scale gradient technique for motion. A patch of the image is convolved with groups of linear spatio-temporal filters. Each group is tuned to the same spatial scale. Each group makes its own estimate for the velocity using total least squares. The estimates are combined in a second re-weighted total least-squares formulation. The magnitude of the velocity estimate a group may present is limited by the systematic error.

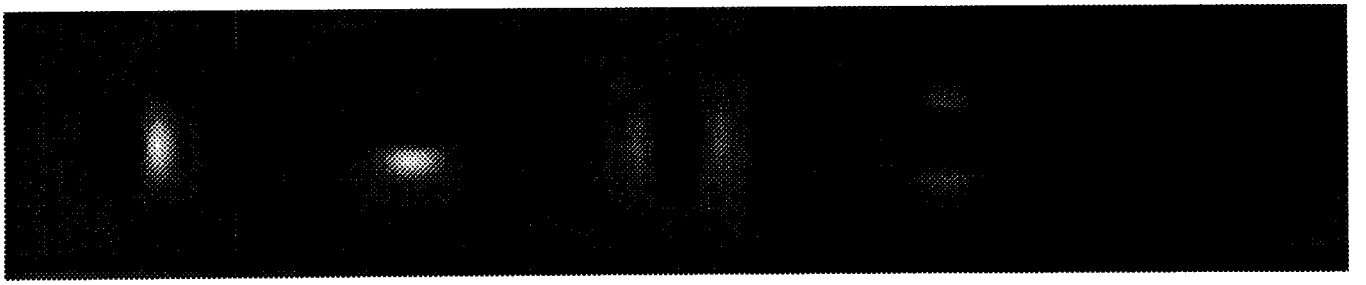


Figure 4: Spatial impulse response of the filter set used for a single spatial scale. The filters are either the Laplacian of a Gaussian or products of Gaussians and a Gaussian derivative. These filters all have a zero DC response. The same set of filters is used in some approaches to early vision.

squares solution (7) is simply

$$\vec{\mathbf{v}} = \left(\begin{bmatrix} \sum I_{xi}^2 \phi_i & \sum I_{xi} I_{yi} \phi_i \\ \sum I_{xi} I_{yi} \phi_i & \sum I_{yi}^2 \phi_i \end{bmatrix} - \sigma_3^2 \mathbf{I} \right)^{-1} \begin{pmatrix} -\sum I_{xi} I_{ti} \phi_i \\ -\sum I_{yi} I_{ti} \phi_i \end{pmatrix} \tag{14}$$

The summations are over the filtered responses within that scale group. The weights ϕ_i are the expected variances of each measurement. This formula is very similar to the one used by many authors [31, **16,** etc.] who attempt to overcome the underconstrained system (2) by making an implicit smoothness assumption. These approaches sum over the responses from surrounding pixels, thus assuming the velocity to be the same within a local region.

Notice that we need only invert a 2x2 matrix whose entries are simply weighted sums of filter outputs. Thus only simple operations are required and can be performed in parallel. The filter responses can be accumulated as they are produced and need not be stored in memory. Since the measurement matrix has rank **3,** simple explicit formulas exist for the three singular values, σ_i. They come from solving a cubic equation whose coefficients are combinations of the summations which appear in (14). Since singular values are always real and non-negative, a simpler form of the general solution of the cubic equation can be used. Testing the condition number of the matrix is then just the ratio of the two largest singular values. If the condition number is above **100** the estimate is discarded. This rarely occurred in **our** simulations. Each scale group now evaluates its velocity estimate. The magnitude of this velocity estimate is compared with the maximum magnitude allowed for this scale group and rejected if larger. The ratio of the smallest singular values is compared with the consistency threshold.

The scale estimates that are not discarded are combined in a final, multi-scale total least squares framework. The weighting terms for this combination is modified by scaling the weight from the first matrix by the inverse of the relative error term. Thus the diagonal matrix elements are replaced **by:**

$$\phi_i = \begin{cases} \frac{\phi_i}{\sigma_3/\sigma_2 + \epsilon} & \sigma_3/\sigma_2 < C_t \text{ and } |\vec{\mathbf{v}}| < \mathbf{v}_{max} \\ 0 & \text{else} \end{cases}$$

where ϵ is a small number to prevent division by zero. This weighting gives more credit to scale groups with estimates which best match the constraint equations.

The final estimate computed by combining scales is rejected if the ratio of singular values indicates the equations are deemed inconsistent according to the consistency threshold. One place where this can happen is if different spatial scales are seeing different motions due to some kind **of** motion boundary. Our future work will look at ways of identifying and resolving these situations. For now, they are simply labeled **as** places without estimates (holes).

5 Experimental Results

For the experiments where the true optical flow is known, we use the angular error measure used by Barron et al. [2] to evaluate the results. They measure the error between the true velocity $v = (v_1, v_2)$ and the estimate $8 = (\hat{v}_1, \hat{v}_2)$ as the angle between the unit vectors in 3 space, $\vec{\mathbf{v}} = (|v|^2 + 1)^{-1/2}(v_1, v_2, 1)$.

$$\psi_\epsilon = \arccos(\vec{\mathbf{v}} \cdot \hat{\mathbf{v}}) \tag{15}$$

This is calculated at every pixel value where an estimate is formulated. Also reported is the percentage of pixels without estimates (holes).

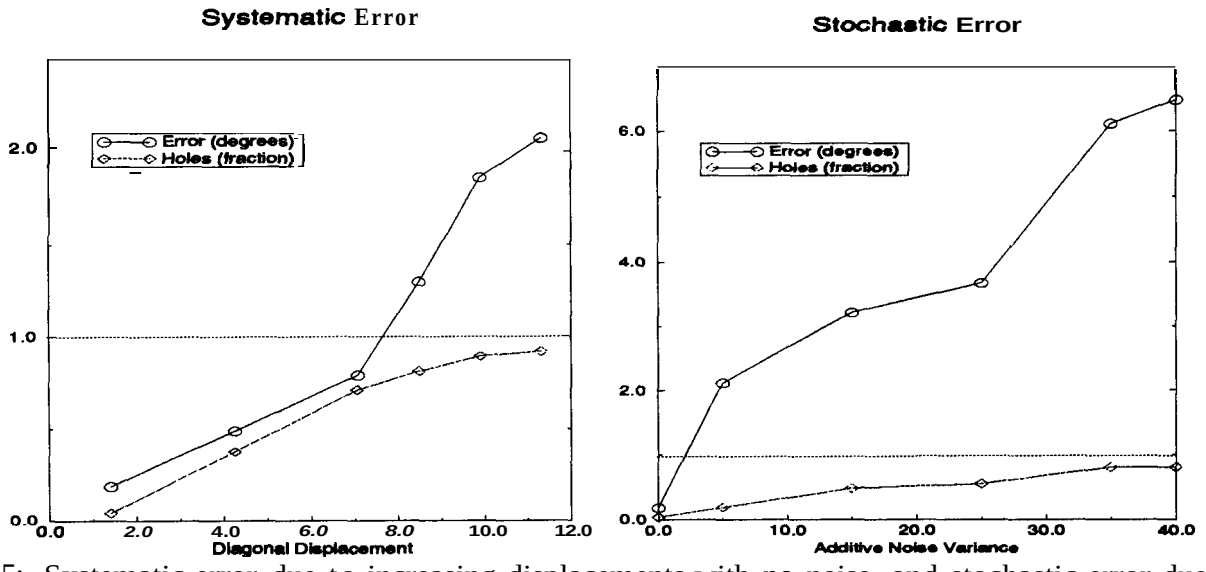


Figure 5: Systematic error due to increasing displacements with no noise, and stochastic error due to additive noise with fixed dispacement. In both cases the stimulus was a Gaussian white noise pattern with variance 1000 units translated diagonally. The consistency threshold was set at 1×10^{-2}. This value keeps the errors to only a few degrees for the full range of allowed displacements.

5.1 Synthetic Data

Systematic Error. A random dot pattern with variance 1000 units is translated diagonally. Each pixel was uncorrelated, thus all frequencies were present. Figure 5 shows the angular error and percentage of holes (no estimate) as a function of diagonal displacement. As the displacement increases the average error increases, but remains less than two degrees. The number of estimates decreases up to the maximum displacement allowed, $1.8^4 \simeq 10.5$ pixels. Thus, up to the maximum allowed displacement, we can keep errors to less that a few degrees. A higher density of estimates can be obtained by lowering the consistency threshold at a cost of increased errors.

Stochastic Error. In order to examine the results of stochastic error we translated the uncorrelated random dot pattern diagonally 1 pixel and added uncorrelated noise to each frame. Figure 5 shows the results as the standard deviation of the added Gaussian noise is increased. Again, the errors remain less than a few degrees as the density decreases.

Consistency Threshold. A random dot pattern of unit variance was translated diagonally 1 pixel and white noise of variance 5 units was added to each frame. Figure 6 displays the average error and percentage of holes as a function of C_t, the equation consistency threshold. The number of estimates decreases and the accuracy of the remaining estimates increases until only a few estimates remain. This demonstrates that the singular values ratio is a good measure of estimate reliability. Such a statistic is often useful in algorithms which use optical flow as input, such as for determining ego-motion or shape from motion.

Comparison Sequences. A recent report by Barron et. al. [1] and the corresponding paper in CVPR '92 [2] examined the performance of a number of different optical flow techniques on a series of synthetic and real images. They found that the phase-based approach of Fleet & Jepson [3] was the most accurate. They also felt that Heeger's energy approach [32] and other's based on SSD minimization were computationally prohibitive. We examined the performance of our routine on these same images.

Ten frames of the *Yosemite Sequence* were used as input to the program. The true optical flow is

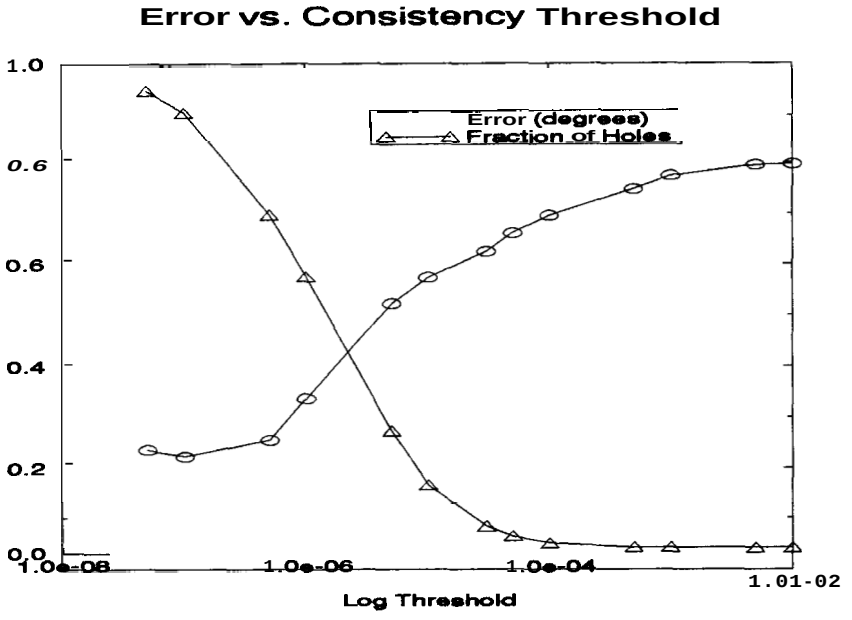


Figure **6:** Average error in degrees and percentage of holes **as** a function of the consistency threshold, C_t, the maximum **of** the ratio σ_3/σ_2 allowed. This ratio relates how much the measurements must be changed in order to fit the velocity model.

known because this is a syntheticly generated sequence. The sequence is of a platform flying over Yosemite valley. Clouds in the image deform as they move. The optical flow ranged in magnitude from zero at the focus of expansion to over **5** pixels per frame. The ***Translating Tree*** sequence consists of a tilted plane with a texture mapped onto it. The motion is perpendicular to the optical axis, but since the plane is tilted the flow ranged in magnitude from **1.8** to **2.3** pixels per frame. The ***Diverging Tree*** consisted **of** the same tilted plane and texture, but the motion is along the optical axis. Velocities ranged from **1.4** pixels per frame on one side to **2.0** on the other. Table **7** lists the average and standard deviation **of** the angular error. The data for the other algorithms was copied from the CVPR paper by **Barron** et al[2]. The performance was comparable, performing better for the Yosemite sequence and worse **on** the translating planes. However, our algorithm uses only **10** frames and **30** linear filters whereas the Fleet & Jepson algorithm used **46** 3-d convolutions and 21 frames, making it computationally more expensive, In addition, our algorithm consistently produced a higher density of vectors. The threshold experiments show that slight improvements can be made by decreasing the error threshold (fixed at 10^{-2} **for** synthetic sequences) at a cost of fewer estimates. Ultimately, the performance must be based on how well the flow field can be used for calculating quantities such as motion and shape parameters.

5.2 Real Sequences

The algorithm was tested on a group of real video images obtained from J.L. **Barron** who received them from the database at Sarnoff Research Centre. We used **10** frames and **25** filters **for** each. Selected frames of the three sequences and the flow produced are shown in Figure **8.** The first sequence is of the camera translating towards the soda can. In the second, the observer translates perpendicular to the line of sight. The tree in the foreground translates more due to perspective. The third sequence is of three independently moving cars. The car on the lower right is obscured by some trees.

Sequence	Algorithm	Avg. Error	Std. Dev.	Density
Translating Tree	Horn & Schunck	33.4	16.46	100
	Heeger	4.79	2.39	13.8
	Anandan	4.54	2.98	100
	Lucas & Kanade (λ_2 >1.0)	1.75	1.43	40.8
	Fleet & Jepson	0.36	0.41	76.0
	Weber & Malik	0.49	0.35	96.8
Diverging Tree	Horn & Schunck	9.85	8.86	100
	Heeger	4.95	3.09	73.8
	Anandan	8.23	6.17	100
	Lucas & Kanade (λ_2 >1.0)	3.05	2.53	49.4
	Fleet & Jepson	1.24	0.72	64.3
	Weber & Malik	3.18	2.50	88.6
Yosemite	Horn & Schunck	22.58	19.73	100
	Heeger	11.74	19.00	44.8
	Anandan	15.54	13.46	100
	Lucas & Kanade (λ_2 >1.0)	5.20	9.45	35.1
	Fleet & Jepson	4.29	11.24	34.1
	Weber & Malik	3.43	5.35	45.2

Figure 7: Comparison of synthetic sequences results with those reported by Barron et. al. [2]. The Weber & Malik algorithm uses only 10 frames and 30 linear filters. The Fleet & Jepson algorithm used 46 3-d convolutions and 21 frames (15 frames for Yosemite).

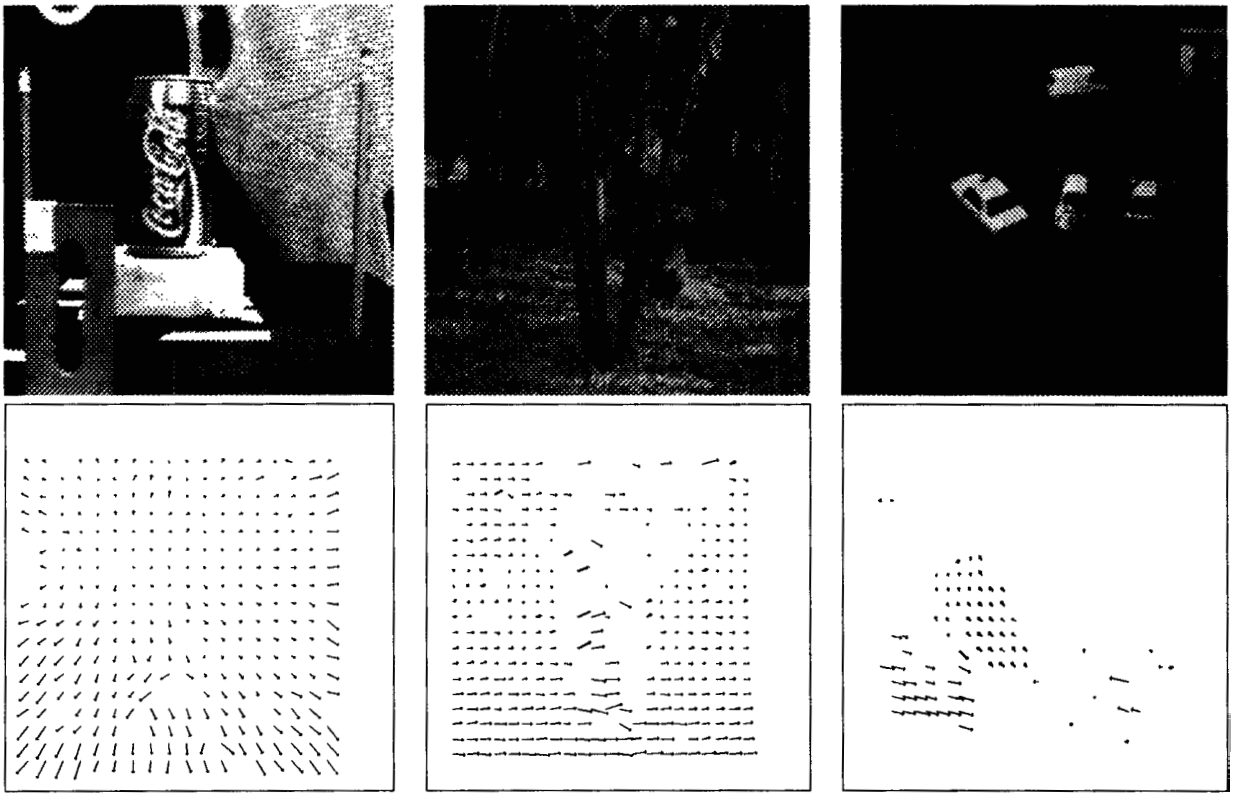


Figure 8: Three frames from real image sequences and the flow recovered.

Sequence	Full Convolutions		Subsampled Filters	
	Avg. Error	Std. Dev.	Avg. Error	Std. Dev.
Translating Tree	0.49	**0.35**	**0.55**	**0.36**
Diverging Tree	**3.18**	**2.50**	**3.30**	**2.72**
Yosemite	**3.43**	**5.35**	**3.77**	**4.83**

Figure **9:** Comparison of the full implementation where each filter output is computed at each pixel and a subsampling scheme where a filter output is calculated at every 2^n pixels.

5.3 Subsampled Filters

The largest scales consist of filters with Gaussian responses many pixels in width. The response of such filters does not change significantly within a range of a few pixels. The full version of the algorithm computes the response of every filter at each pixel. Since the large scale filters do not vary much a more efficient implementation would use a pyramid scheme in which filters of larger scales are calculated at a subset of the full pixel lattice. We created a modified version of the algorithm in which the response of a filter with Gaussian response of size σ is subsampled every n pixels, where $n = \lfloor \sigma \rfloor$. This reduced by about half the number of convolutions for each scale from the previous scale. The results of the subsampled version of the algorithm for the synthetic images are tabulated in Table **9.** The **flow** field became blocky in this scheme. The component values remained almost constant within small sub-blocks of pixels, especially for large displacements, where **only** large scale filters could make valid estimates. Here the estimate must be the same for the entire sub-block of the smallest filter used. Qualitatively the flow was still very good, but consisted of blocks of pixels with the same flow estimate.

6 Multiprocessor Implementation

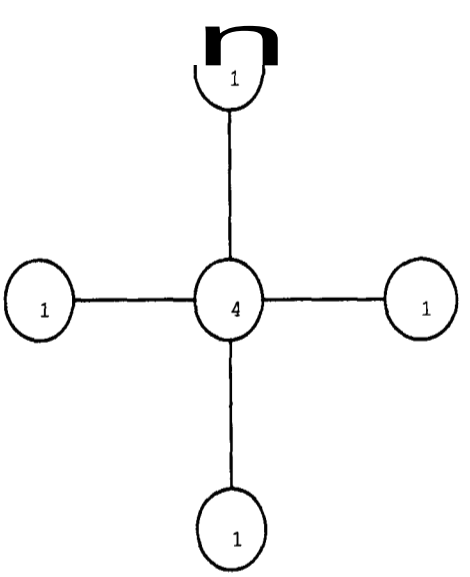
The algorithm described is massively parallel. Each estimate is formed from a small spatio-temporal window of the motion sequence. The previous results were obtained from *a* **SUN** workstation. Processing time was dominated by the convolutions since velocity estimates required **only** a few simple operations on the convolution results per pixel. **A** series of 36, 3-d separable convolutions on a **128** pixel square image took about **4** minutes per frame. To examine the speedup possible with a parallel implementation, a parallel version of the algorithm was created for the **128** processor CM5 from Thinking Machines.

6.1 Parallel Convolution

Instead of the linear convolutions with filter kernels that was used **on** the serial machine, we used the interconnectivity of the processors to perform convolution in a different manner. One way **of** computing a series of convolutions with Gaussian functions of various size is to make use of the Central Limit Theorem. Specifically, if the processor element at each pixel were to perform a simple manipulation of its own data and the data of its neighboring pixel elements, the resulting distribution after a large number **of** iterations **will** be a Gaussian.

For example, suppose we have a one dimensional string of processing nodes, each initially containing a single number. Let each processing node them perform the following operation:

> ***Take two times the value contained at this node and add it to the value held at the node to the immediate right and left. Make this sum the new stored value.***



i=l

1
1 4 1
1

i=2

1
2 6 2
1 6 1 2 6 1
2 6 2
1

Figure **10:** Simple two-dimensional connections between processing elements for computing a Gaussian convolution of the local image. The weights of the weighted sum are inside each processor circle. The result of applying this weighted sum are shown on the right for two iterations. After normalization this sequence converges to a Gaussian distribution.

This operation creates a weighted sum of the form $(1, 2, 1)$ centered at each node. At the next iteration, we have the following weighted sum $(1, 4, 6, 4, 1)$. Placing them in an array we have:

$$\begin{array}{ccccccc} & & 1 & 2 & 1 & & \\ & 1 & 4 & 6 & 4 & 1 & \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 \end{array} \tag{16}$$

The result is the even lines of Pascal's triangle which consists of the even binomial coefficients. It is known that these coefficients approximate the Gaussian distribution for large numbers. Specifically, the binomial coefficients of degree **n** can be approximated by

$$\left(\begin{array}{c} k \\ n \end{array} \right) \simeq C \exp(-2(k - n/2)^2/n) \tag{17}$$

Thus these coefficients are proportional to a Gaussian with standard deviation $\sigma^2 = n/4$. In our algorithm we wish to obtain convolutions with Gaussians of increasing size. We also want derivatives of Gaussian functions. These can be obtained via finite differences. If a processor element takes the difference of the values of its neighbors, it obtains **an** estimate of the derivative at its location. This can be repeated for higher order derivatives, Other filters can be approximated by similar elementary operations [33].

If we normalize the example process such that after the summation the processor divides **by** four, we can preserve the sum total of the data stored at the nodes. Figure **10** shows a two-dimensional extension of the Gaussian kernel. The right side of the figure shows the impulse response of the resulting filter after one and two iterations. After normalization, this series converges to a Gaussian distribution.

The total number of steps required by this processes is $4\sigma_0^{2(s-1)}$ where σ_0 is the base scale and s is the number of different scale groups. The amount of data values which must be passed between processors on each iteration is $4n/\sqrt{N}$ where the image is of size n^2 using N processors.

6.2 Implementation

The above algorithm was implemented on a **CM-5,128** processor machine. The images are divided into **128** regions and sent to the processors. The two dimensional Gaussian operation outlined above was performed. At each step the processors performed the operation on the interior of their own patch. The borders **of** each patch needed to be shared with the neighboring processors in order to complete the step. For an image of dimension $n \times n$, this required $4 \times n/8$ elements to be passed per step per processor.

To obtain the image convolved with a geometric series **of** Gaussians, (i.e. with Gaussians of standard deviations $\sigma = 1, 2, 4, 8, 16$ pixels), the simple procedure was performed $4, 16, 64, 256,$ and **1024** times. After the smoothing iterations, finite differences were performed in order to obtain approximations to the result of convolving the original image with the derivatives of Gaussians of various sizes. The final result is a series of numbers for each pixels. These numbers represent a series of new images which are the outcome of convolving the original image with a series of linear filters.

The **CM-5** is not an ideal machine to perform the convolution method outlined above because it requires much communication between processors. The parallel C we used on the CM-5 had a high overhead for communication between some processors, taking many cycles per byte **of** information passed. Our purpose however was to demonstrate the parallizabibilty **of** the algorithm and the speed up experienced when operated in parallel. Convolution **of** a **128** pixel squared image took about **10** seconds, including I/O. We believe that with specialized hardware real-time implementations are possible.

Acknowledgements

This research was partially supported by Texas Instruments, NSF Presidential Young Investigator Grant IRI-8957274 to J.M., Xerox and the PATH project **MOU 83. NSF** Infrastructure Grant number CDA-**8722788** supported the use of the CM-5. We wish to thank **D.** Fleet and **J.** Barron for providing the sequences and true flow, and **A.** Verri for helpful discussions.

References

[1] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," Tech. Rep. **299,** University **of** Western Ontario, **1992.**

[2] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt, "Performance of optical flow techniques," in ***Proceedings of the IEEE CVPR,*** (Champaign, **IL.),** pp. **236–242, 1992.**

[3] D. Fleet and A. Jepson, "Computation of component image velocity from local phase information," ***International Journal of Computer Vision,*** vol. 5, pp. **77–104, 1990.**

[4] J. Canny, "A computational approach to edge detection," ***IEEE tmns. on PAMI,*** vol. 8, pp. **679–698, 1986.**

[5] D. Jones and J. Malik, "A computational framework for determining stereo correspondence from a set of linear spatial filters," in *Procceeding of the Second European Confemnce on Computer Vision,* **1992.**

[6] D. Jones and J. Malik, "Determining three-dimensional shape from orientation and spatial frequency disparities," in *Procceeding of the Second European Conference on Computer Vision,* **1992.**

[7] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *Journal of the Optical Society of America A,* vol. **7, no. 5,** pp. **923–932, 1990.**

[8] C. Fennema and W. Thompson, "Velocity determination in scenes containing several moving objects," *Computer Graphics and Image Processing,* vol. **9,** pp. **301–315, 1979.**

[9] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence,* **no. 17,** pp. **185–203,1981.**

[10] O. Tretiak and L. Pastor, "Velocity estimation from image sequences with second order differential operators," in *Proceedings of the International Conference on Pattern Recognition,* (Montreal), **1984.**

[11] H.-H. Nagel, "On the estimation of optical flow: relations between different approaches and some new results," *Artificial Intelligence,* no. **33, 1987.**

[12] S. Uras, F. Girosi, A. Verri, and V. Torre, "A computational approach to motion perception," *Biological Cybernetics,* vol. **60,** pp. **79–87, 1988.**

[13] A. Verri, F. Girosi, and V. Torre, "Differential techniques for optical flow," *Journal of the Optical Society of America A,* vol. **5,** pp. **912–922,1990.**

[14] M. Srinivasan, "Generalized gradient schemes for the measurement of two-dimensional image motion," *Biological Cybernetics,* vol. **63,** pp. **421–431, 1990.**

[15] B. Lucas and T. Kanade, "An iterative image restoration technique with an application to stereo vision," in *Procceeding of the DARPA IU Workshop,* pp. **121–130,1981.**

[16] M. Campani and A. Verri, "Computing optical flow from an overconstrained system of linear algebraic equations," in *Proceedings of the 3rd International Conference on Computer Vision,* pp. **22–26, 1990.**

[17] S. VanHuffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis.* Frontiers in Applied Mathematics, Philadelphia: SIAM, **1991.**

[18] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* New York, Chichester, Brisbane, Toronto, Singapore: John Wiley & Sons, **1973.**

[19] K. Pearson, "On lines and planes of closest fit to points in space," *Philos. Mag.,* vol. **2,** pp. **559–572, 1901.**

[20] A. Madansky, "The fitting of straight lines when both variables are subject to error," *J. Amer. Statist. Assoc.,* vol. **54,** pp. **173–205,1959.**

[21] P. Sprent, *Models in Regression and Related Topics.* London: Methuen, **1969.**

[22] G. Golub and C. VanLoan, "An analysis of the total least squares problem," *SIAM Journal Numer. Anal.,* vol. **17,** pp. **883–893, 1980.**

[23] R. Cleary and O. Braddick, "Directional discrimination for band-pass filtered random dot kinematograms," *Vision Research,* vol. **30,** pp. **303–316,1990.**

[24] J. Chang and B. Julesz, "Cooperative and non-cooperative processes of apparent movement of rnadom-dot cinematograms," *Spatial Vision,* vol. **1,** pp. **39–45, 1985.**

[25] O. Braddick and R. Cleary, **"Is** there a half-cycle displacement limit for directional motion detection?," *Vision Research,* vol. **31,** no. **4,** pp. **761–762, 1990.**

[26] **V. D. Lollo** and **W.** Bischof, "Yes, there is a half-cycle displacement limit for directional motion detection," *Vision Research,* vol. **31,** no. **4,** pp. **763–765, 1990.**

[27] **R.** Battiti, E. Amaldi, and **C.** Koch, "Computing optical **flow across** multiple scales: An adaptive coarse-to-fine strategy," *International Journal of Computer Vision,* vol. **6,** no. **2,** pp. **133–145, 1991.**

[28] R. Young, "The gaussian derivative theory of spatial vision: Analysis **of** cortical cell receptive field line-weighting profiles," Technical Report **GMR-4920,** General Motors Research, **1985.**

[29] P. J. Huber, ***Robust Statistics.*** Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, **1981.**

[30] J. Koenderink, "Operational significance of receptive field assemblies," ***Biological Cybernetics,*** vol. **58,** pp. **163–171,1988.**

[31] **E.** Simoncelli, E. Adelson, and **D.** Heeger, "Probability distributions of optical **flow,"** ***International Journal of Computer Vision,*** vol. **6,** no. **2,** pp. **133–145, 1991.**

[32] D. Heeger, "Optical **flow** from spatiotemporal filters," in ***Proceedings of the First International Conference on Computer Vision,*** pp. **181–190, 1987.**

[33] **J.** Ben-Arie, "Muti-dimensional linear lattice for fourier and gabor transforms, multiple-scale gaussian filtering, and edge detection," in ***Neural Networks for Perception,*** vol. **1,** ch. **11.2,** Academic Press, **1992.**