

A Reliable Optical Flow Algorithm Using 3-D Hermite Polynomials

Hongche Liu

Center for Automation Research
Department of Electrical Engineering
University of Maryland
College Park, MD 20742
and Robot Systems Division

Tsal-Hong Hong

Martin Herman

Sensory Intelligence
Robot Systems Division

and

Rama Chellappa

Center for Automation Research
Department of Electrical Engineering
University of Maryland
College Park, MD 20742

U.S. DEPARTMENT OF COMMERCE

Technology Administration
National Institute of Standards
and Technology
Bldg. 220 Rm. B124
Gaithersburg, MD 20899

A Reliable Optical Flow Algorithm Using 3-D Hermite Polynomials

Hongche Liu

Center for Automation Research
Department of Electrical Engineering
University of Maryland
College Park, MD 20742
and Robot Systems Division

**Tsal-Hong Hong
Martin Herman**

Sensory Intelligence
Robot Systems Division

and

Rama Chellappa

Center for Automation Research
Department of Electrical Engineering
University of Maryland
College Park, MD 20742

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Bldg. 220 Rm. B124
Gaithersburg, MD 20899

December 1993



**U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary**

**TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology**

**NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Arati Prabhakar, Director**

A Reliable Optical Flow Algorithm Using 3-D Hermite Polynomials

Hongche Liu^{†‡}, Tsai-Hong Hong[‡], Martin Herman[‡], and Rama Chellappa[†]

[‡]Robot Systems Division, National Institute of Standards and Technology (NIST)
Bldg. 220, Rm B124, Gaithersburg, MD 20899

[†]Center for Automation Research/Department of Electrical Engineering,
University of Maryland, College Park 20742

Abstract

Algorithms that attempt to accurately compute optical flow must cope with occlusion, brightness changes, irregular motion, and the aperture problem. Most optical flow algorithms have tried to overcome one or more of the problems mentioned above. Yet, claims about the reliability of computed flow are still based on ad hoc evaluation schemes. While a perfect algorithm that is free of all the problems is not yet available, we present a *reliable* algorithm, which despite all the difficulties, generates the output flow field wherever possible, and associates with the output evaluation metrics that reflect the reliability of the output. The evaluation metrics are complete in the sense that they are theoretically related to the physical phenomena that cause the inherent problems noted above. Our approach to computing optical flow expands the spatio-temporal image in terms of Hermite polynomials and then derives multiple Gaussian smoothed gradient constraint equations, which constitute an overdetermined linear system that can be solved for image flow with a least square method. Using the QR decomposition technique, we extract the residual, condition number, and the determinant associated with the linear system. These measures are shown to correspond to the severity of the aforementioned problems. By thresholding the output using these evaluation metrics, we extract different densities of output that can be compared with other algorithms included in the evaluation scheme developed by Barron, Fleet, and Beauchmin. We show that our algorithm performs consistently better over a wide variety of synthetic and real image sequences.

1. Introduction

During the past decade, a great deal of effort has been spent on the computation of optical flow. Three methodologies have been explored: spatio-temporal energy based[1,12,13,23], correlation based[3,20,25], and gradient based methods[6,14,18,19,10]. There is a great deal of biological motivation for spatio-temporal approaches. These techniques are based on the equation

$$\omega_t = \omega_x \cdot u + \omega_y \cdot v \quad (1)$$

where (u,v) is the velocity of a point in the image, and $\omega_x, \omega_y, \omega_t$ are spatial and temporal frequencies at the point. The local image velocity (u,v) can be estimated by determining the plane of nonzero spectrum in the spatio-temporal frequency domain[12]. The correlation-based method typically finds prominent features or small patches of the image and attempts to match them in successive images. Most researchers have used three strategies to match the candidate features or patches: maximizing the correlation response measurement[20], minimizing the sum-of-squared difference of the corresponding values[2], and coarse-to-fine search strategies[25]. The gradient-based techniques extract optical flow from spatial and temporal derivatives of the image intensity. In one approach[14], the first derivatives and the gradient constraint equation are used:

$$I_x u + I_y v + I_t = 0 \quad (2)$$

where I_x, I_y, I_t denote the partial derivatives of I , the image intensity, with respect to x, y, t respectively. The two components (u, v) of the image velocity are constrained by only one linear equation in (2). In another approach[19], second order derivatives are used to constrain image velocity:

$$\begin{aligned} I_{xx}u + I_{yx}v + I_{tx} &= 0 \\ I_{xy}u + I_{yy}v + I_{ty} &= 0 \end{aligned} \quad (3)$$

Clearly, (u, v) can be recovered from (3) wherever they are linearly independent. Because of the sensitivity of numerical differentiation to noise, velocity estimates from second derivatives are not very accurate. In [10], Haralick and Lee present a facet model approach to obtain image ve-

locity based on fitting a function to the intensities in local neighborhoods of spatio-temporal images. However, the approach requires first and second order partial derivatives in space and time. A set of (5x5x5) operators were used for estimating the derivatives. The method is very sensitive to noise.

Our approach is similar to the facet model method in that both estimate a local neighborhood of the spatio-temporal image with a polynomial. The difference is in the use of an orthogonal Hermite polynomial basis to estimate the derivatives in an image neighborhood. Not only is our approach based on more numerically stable estimation techniques but in addition, as the behavior of Hermite polynomials is modeled by Gaussian derivatives, our gradient constraint equations are more robust. Numerous physiological experiments [9,16,26] support the theory that the visual receptive field can be modeled by Gaussian derivatives of various widths. Indeed, Gaussian derivatives are linear, spatial shift and scale invariant, isotropic and insensitive to noise. In addition, they are rotation and dimension invariant.

We categorize our algorithm as a gradient-based method. The major advantages over the previous gradient based methods are numerical stability, characterization of the performance of the algorithm, and simplicity. Numerical stability is achieved by using Hermite polynomials. The performance of the algorithm is evaluated using the residual, condition number, and the determinant obtained from an overextended linear system of gradient constraint equations. Simplicity is due to the fact that the filtering process accomplishes smoothing and differentiation simultaneously. The computations involve only convolutions and solution of linear systems.

As our extensive experiments attest, our algorithm performs consistently well over a large variety of images. Furthermore, the paradigm presented here is justified by its power to unify and predict theoretical relationships between image phenomena and computed flow field. To the best of our knowledge, only Kearney et al.[15] has exploited such a relationship as an integral part of the algorithm.

The organization of this paper is as follows. We lay out the theoretical foundation in Section 2.

Implementation details including the discussion of confidence measures are presented in Section 3 followed by extensive experiments and comparisons in Section 4. Section 5 concludes the paper. Analytical results about the interpretation of confidence measures are aggregated in appendices.

2. Algorithm

In this section, we establish gradient constraint equations similar to (2) and (3) and expand them to higher order. The difference here is that the gradients are computed by Gaussian derivatives. The algorithm estimates optical flow by fitting a three dimensional polynomial to a spatio-temporal image sequence. The model is defined as follows:

$$I(x, y, t) = F(x-ut, y-vt) \quad (4)$$

Equation (4) is based on the assumption that the motion in the image is constant over time and the image brightness pattern does not change over time. Using the Hermite polynomial basis to expand the function I and F , optical flow (u, v) can be estimated. The algorithm consists of two stages: 1, convolution of the image I with a set of derivative masks generated by Hermite polynomials and 2, least square error solution of an overdetermined linear system.

2.1 Hermite Polynomials

The n th Hermite polynomial $H_n(x)$ is a solution of

$$\frac{d^2 H_n}{dx^2} - 2x \frac{dH_n}{dx} + 2nH_n = 0. \quad (5)$$

The $H_n(x)$ are derived by Rodrigues' formula [11]

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}. \quad (6)$$

It is clear from (6) that

$$H_n(x) \in P_n \text{ where } P_n \text{ is the set of polynomials of degree } n.$$

The computation of $H_n(x)$ is especially easy due to the following recursive relations:

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$$

$$H_0(x) = 1$$

$$H_1(x) = 2x$$

The orthogonality of $\{H_n(x)\}$ can be verified from (5) and (6).

By substituting $G(x)$ for e^{-x^2} in (6), we generalize to Hermite polynomials with respect to the Gaussian function. Let these Hermite polynomials be denoted by $\bar{H}_n(x)$

$$\bar{H}_n(x) = (-1)^n G^{-1}(x) \frac{d^n}{dx^n} (G(x)) \quad (7)$$

Note that $\bar{H}_n(x)$ differs from $H_n(x)$ by a scaling product:

$$\bar{H}_n(x) = \left(\frac{1}{2^{1/2} \sigma} \right)^n H_n \left(\frac{x}{2^{1/2} \sigma} \right) \quad (8)$$

where σ is the standard deviation of $G(x)$.

The scalar product of two functions and the L_2 -norm of a function with $G(x)$ as a weight function are defined as follows:

$$\langle a, b \rangle \equiv \int_{-\infty}^{\infty} G(x) a(x) b(x) dx$$

$$\|a\| \equiv \langle a, a \rangle^{1/2}$$

The orthogonality of $\{\bar{H}_n(x)\}$ can be expressed in the following way:

$$\langle \bar{H}_m, \bar{H}_n \rangle = \sigma^{-2n} n! \delta_{mn}, \quad (9)$$

where
$$\begin{cases} \delta_{mn} = 1 \Leftrightarrow m = n \\ \delta_{mn} = 0 \Leftrightarrow m \neq n \end{cases}.$$

The 3D case of Hermite polynomials is especially simple because they are separable with respect to dimension. Thus the polynomial with order $n = i + j + k$ is:

$$\bar{H}_{ijk}(x, y, t) = \bar{H}_i(x) \cdot \bar{H}_j(y) \cdot \bar{H}_k(t) \quad (10)$$

2.2 Derivation of Gradient Constraint Equations

One of the most important properties of Hermite polynomials is the property of Gaussian derivatives. It is with this property that we are able to establish gradient constraint equations. This property is manifested in the following theorem.

Theorem 1: A one dimensional signal $I(x)$ can be expanded in terms of Hermite polynomials as

$$I(x) = \sum_{k=0}^{\infty} I_k \frac{\bar{H}_k(x)}{\|\bar{H}_k\|^2}$$

$$\text{then } I_k = \langle I, \bar{H}_k \rangle = \langle I^{(k)}, \bar{H}_0 \rangle \text{ where } \bar{H}_0(x) = 1 \text{ and } I^{(k)} = \frac{d^k I}{dx^k} \quad (11)$$

The proof is given in Appendix A.

From Theorem 1, we know that $\{I_k\}$ can be interpreted in two ways: the expansion coefficients and Gaussian derivatives of the image sequence.

Consider our motion model:

$$I(x, y, t) = F(x-ut, y-vt)$$

Now expand both sides with Hermite polynomials,

$$\sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} I_{ijk} \frac{\bar{H}_{ijk}}{\|\bar{H}_{ijk}\|^2} = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} F_{ijk} \frac{\bar{H}_{ijk}}{\|\bar{H}_{ijk}\|^2} \text{ where } \begin{cases} I_{ijk} = \langle I, \bar{H}_{ijk} \rangle \\ F_{ijk} = \langle F, \bar{H}_{ijk} \rangle \end{cases} \quad (12)$$

Suppose the scene is smooth and the motion is steady, which means that the higher order terms in the expansion vanish. We can then use a limited basis to represent the function. If only $\bar{H}_{000}, \bar{H}_{100}, \bar{H}_{010}, \bar{H}_{001}$ are used, we have the following:

$$\begin{aligned} I_{000} &= F_{000} \\ I_{100} &= F_{100} \\ I_{010} &= F_{010} \end{aligned} \quad (13)$$

$$\begin{aligned}
I_{001} &= F_{001} = \langle F, \bar{H}_{001} \rangle = \langle \frac{\partial F}{\partial t}, \bar{H}_{000} \rangle = \langle (-u) \frac{\partial F}{\partial x} + (-v) \frac{\partial F}{\partial y}, \bar{H}_{000} \rangle \\
&= (-u) \langle \frac{\partial F}{\partial x}, \bar{H}_{000} \rangle + (-v) \langle \frac{\partial F}{\partial y}, \bar{H}_{000} \rangle \\
&= (-u) \langle F, \bar{H}_{100} \rangle + (-v) \langle F, \bar{H}_{010} \rangle \\
&= (-u) F_{100} + (-v) F_{010}
\end{aligned} \tag{14}$$

The derivation of (14) is based on Theorem 1. From (13) and (14), we derive the first order gradient constraint equation.

$$I_{001} + uI_{100} + vI_{010} = 0 \tag{15}$$

If a larger basis is used, we can have additional gradient constraint equations. For example, with $\bar{H}_{000}, \bar{H}_{100}, \bar{H}_{010}, \bar{H}_{001}, \bar{H}_{101}, \bar{H}_{011}, \bar{H}_{110}, \bar{H}_{200}, \bar{H}_{020}$, following similar reasoning, we have the second order gradient constraint equations:

$$\begin{aligned}
I_{101} + uI_{200} + vI_{110} &= 0 \\
I_{011} + uI_{110} + vI_{020} &= 0
\end{aligned} \tag{16}$$

Equations in (15) and (16) can be rewritten in a matrix form:

$$\begin{bmatrix} I_{100} & I_{010} \\ I_{200} & I_{110} \\ I_{110} & I_{020} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} I_{001} \\ I_{101} \\ I_{011} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

Realizing that $\{I_{ijk}\}$ is just the Gaussian derivatives of I , we note the similarity between (15), (16) and (2), (3), rewritten here:

$$\begin{aligned}
I_t + I_x u + I_y v &= 0 \\
I_{tx} + I_{xx} u + I_{yx} v &= 0 \\
I_{ty} + I_{xy} u + I_{yy} v &= 0
\end{aligned}$$

If the assumption of smooth scene and steady motion are not practical, we may use a larger polynomial basis and thus generate additional gradient constraint equations, while still maintaining a linear system of equations in u and v . More generally,

$$I_{i,j,k} + uI_{i+1,j,k-1} + vI_{i,j+1,k-1} = 0 \quad (18)$$

In fact, we can use as many equations as necessary for a particular image scene and motion.

Another approach is to use a nonlinear system to achieve higher accuracy. For example, we could add H_{002} to the previous second order basis. The additional equation will be

$$I_{002} - (I_{200}u^2 + 2I_{110}uv + I_{020}v^2) = 0 \quad (19)$$

The equation is not linear, so it might be necessary to use an iterative algorithm to search for the optimal solution.

3. Implementation

3.1 Localizing Computation

Recall that our motion model assumes that motion in the image is constant over time. To assure this is not violated, we should only use a small local neighborhood to compute optical flow for each pixel. In addition, we have to adapt the algorithm to the discrete case as required for the common image input. Therefore, we define local expansion coefficients $\{\hat{I}_{ijk}(x, y, t)\}$ to approximate $\{I_{ijk}\}$ and redefine the inner product as a local computation in V :

$$\begin{aligned} \hat{I}_{ijk}(x, y, t) &= \langle I(x, y, t), \bar{H}_{ijk} \rangle \\ &= \sum_V \sum \sum I(x + x_0, y + y_0, t + t_0) \bar{H}_{ijk}(x_0, y_0, t_0) G(x_0, y_0, t_0) \\ V &= \{ (x_0, y_0, t_0) \mid -a \leq x_0 \leq a, -b \leq y_0 \leq b, -c \leq t_0 \leq c \} \end{aligned} \quad (20)$$

Substituting $\{\hat{I}_{ijk}(x, y, t)\}$ for $\{I_{ijk}\}$ in (17), we have an overextended linear system:

$$\begin{bmatrix} \hat{I}_{100}(x, y, t) & \hat{I}_{010}(x, y, t) \\ \hat{I}_{200}(x, y, t) & \hat{I}_{110}(x, y, t) \\ \hat{I}_{110}(x, y, t) & \hat{I}_{020}(x, y, t) \end{bmatrix} \begin{bmatrix} u(x, y, t) \\ v(x, y, t) \end{bmatrix} + \begin{bmatrix} \hat{I}_{001}(x, y, t) \\ \hat{I}_{101}(x, y, t) \\ \hat{I}_{011}(x, y, t) \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

Note that we have reduced global computation in (17) to local computation in (21). Now for every

single image pixel, there is an associated linear system that can be solved for optical flow.

$$\text{Let } A = W \begin{bmatrix} \hat{I}_{100}(x, y, t) & \hat{I}_{010}(x, y, t) \\ \hat{I}_{200}(x, y, t) & \hat{I}_{110}(x, y, t) \\ \hat{I}_{110}(x, y, t) & \hat{I}_{020}(x, y, t) \end{bmatrix}, f = \begin{bmatrix} u(x, y, t) \\ v(x, y, t) \end{bmatrix}, b = W \begin{bmatrix} \hat{I}_{001}(x, y, t) \\ \hat{I}_{101}(x, y, t) \\ \hat{I}_{011}(x, y, t) \end{bmatrix}. \quad (22)$$

We introduce a weight matrix W to compensate the difference of norms between first and second order Hermite polynomials,

$$W = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_2 \end{bmatrix}, \quad (23)$$

Since $\{\hat{I}_{ijk}(x, y, t)\}$ is just a local approximation of $\{I_{ijk}\}$, we solve the overextended linear system for f (or u and v) in the least square error sense:

$$E = \min \|Af + b\|, \quad (24)$$

Using the QR decomposition method [Householder Triangularization], we derive the following:

$$A = QR, \text{ and } E = \min \|QRf + b\| = \min \|Rf + Q^H b\|, \text{ where } Q \text{ is unitary.} \quad (25)$$

$$R \text{ is an upper triangular matrix, denoted by } \begin{bmatrix} \lambda_1 & d \\ 0 & \lambda_2 \\ 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} R_s \\ 0 & 0 \end{bmatrix}, \quad (26)$$

where R_s is the upper square matrix.

Let $Q^H b$ be $\begin{bmatrix} b_s \\ r \end{bmatrix}$ where b_s is the upper 2-element vector. Equation (25) becomes

$$\begin{aligned} E &= \min (\|R_s f + b_s\| + r) \\ &= r \quad \text{if } R_s \text{ is not singular.} \end{aligned} \quad (27)$$

$$\text{The solution is computed from } R_s f + b_s = 0 \quad (28)$$

In the actual implementation, we use a floating point computation; as a result, R_s is rarely singu-

lar. However, the behavior of R_s determines the accuracy of the solution. Also the residual r of the overextended linear system plays an important role in determining the accuracy of the solution. We devote the next subsection to the discussion of accuracy of the computed optical flow and associated confidence measures.

3.2 Confidence Measures

Optical flow computed from noisy data is unreliable and there should be a mechanism to reject unreliable or poor flow estimates. The confidence measure is defined as a quantitative value that indicates the degree of confidence in the quality of the computed result. Our algorithm provides ample information about the behavior of the system equations. It is then shown that this information signifies certain image phenomena, e.g., occlusion, which present difficulties for optical flow computation. Therefore, it can be utilized as confidence measures. We first analyze some potential confidence measures in the following subsections.

3.2.1 Residual

The residual of our algorithm is $\|Af + b\|$ or $r (=E)$ (27). The residual of an overextended linear system indicates the degree to which the equations disagree with one another. Recalling (15) and (16), if $\{I_{ijk}\}$ are computed exactly, these equations should hold. The reason for the existence of residual lies in the approximation error of $\{\hat{I}_{ijk}(x, y, t)\}$. A high approximation error may indicate one of three problems:

1. The assumption of constant motion is violated in the window V . It is possible that the window covers more than one moving object. Occlusion and multiple independently moving objects in a window can cause this problem.
2. The assumption of constant image brightness is violated. It is not unusual for the brightness of an object to change when the viewing angle changes due to relative motion. In addition, the observing camera may adjust the picture balance for different scenes, resulting in change of object brightness. Similar results can be caused by the

shadow of a moving object.

3. Truncation error. Truncation errors are introduced when we use a small window to compute $\{\hat{I}_{ijk}(x, y, t)\}$. Within the small window, the Hermite polynomials are no longer orthogonal and the expansion coefficients are not accurate.

We can model the above errors as perturbations to the linear system (Appendix B):

$$\tilde{E} = \min \|(A + N)\tilde{f} + (b + \Delta b)\|, \text{ where } N \text{ and } \Delta b \text{ denote errors.} \quad (29)$$

We prove in Appendix B the following analytical results:

$$\Delta f = \tilde{f} - f \approx -\left(A^T A\right)^{-1} A^T (Nf + \Delta b) \quad (30)$$

$$\tilde{r} = \tilde{E} \approx \left\| \left(I - A \left(A^T A \right)^{-1} A^T \right) (Nf + \Delta b) \right\| \quad (31)$$

Note that the expressions of both optical flow error Δf (30) and residual \tilde{r} (31) contain the error vector $(Nf + \Delta b)$. Even so, it may be deceptive to claim that the residual is proportional to the optical flow error because the error vector is mapped by different matrices $\left(\left(A^T A \right)^{-1} A^T, I - A \left(A^T A \right)^{-1} A^T \right)$. We further show that the matrix $I - A \left(A^T A \right)^{-1} A^T$ in the residual expression has only one nontrivial eigenvalue, which happens to be 1. This makes it more likely for the error vector to be mapped to a small vector. Nonetheless, it is clear that a large residual certainly means there are errors and the optical flow result may be inaccurate.

Note that the three problems mentioned above suggest contradictory choices for the window size. With larger windows, problems 1 and 2 may be aggravated; with smaller windows, problem 3 may become intolerable.

Using $|1/r|$ as a confidence measure, we can locate these problems and eliminate unreliable results.

3.2.2 Condition Number

The condition number of R_s , denoted by $\kappa(R_s)$, is defined as $\|R_s\| \|R_s^{-1}\|$ and can be shown to be

$\frac{|\lambda|_{max}}{|\lambda|_{min}}$, where λ_{max} , λ_{min} are eigenvalues of R_s .

A condition number measures the extent to which a linear system maps the input error into output error, or in brief, the numerical instability of the system. The higher the condition number, the more ill-conditioned a system is. Recalling (24), we regard b as the input of the system and f the output. There are inherent errors in the elements of b . We certainly do not consider f which contains errors magnified by an ill-conditioned A reliable. Since matrix A is concerned with the image texture only and not with motion, we find the correspondence between a high condition number and the following two scenarios of the image neighborhood:

1. When there is a steep edge in the $x(y)$ direction (Fig 1.1), so that the first order and second order derivatives are very large for $x(y)$ and small for $y(x)$.
2. When there is a lack of texture in a direction (Fig 1.2), so that the derivatives in the x direction are approximately proportional to the derivatives in the y direction, i.e.

$$I(x, y) \approx I(kx + y) .$$

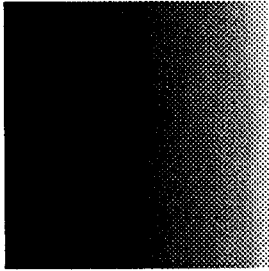


Fig 1.1 Smoothed steep edge.

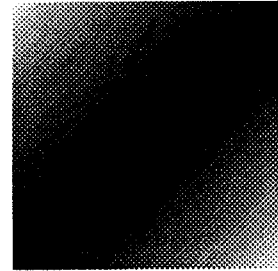


Fig 1.2 Lack of texture in $x+y$ direction.

The above two findings are easily verified from the QR decomposition process as explained in Appendix D.

If there is motion in the area where one of the two scenarios dominate, then it corresponds to what is known as the aperture problem. Since we will not be able to recover the velocity accurately with any local computation, we may as well eliminate the results.

So $1/\kappa(R_s)$ can be used as a confidence measure.

3.2.3 Determinant

The determinant of R_s is $\lambda_1 \cdot \lambda_2$. In solving (28) or $f = -R_s^{-1} \cdot b_s$, the determinant plays an important role in the matrix inverse. Since we use the QR decomposition method, Q is unitary (orthonormal projection) so the behavior of R_s is similar to the original A . Looking at (24), a small determinant of R_s indicates one of the following two scenarios:

1. The two columns of A are close to being linearly dependent. This is the same as the second scenario in the discussion of condition number. In fact, we prove in Appendix E that low determinant due to linear dependency also causes high condition number.
2. All the elements of A are very small. This corresponds to a uniform brightness area, e.g. blue sky.

As noted before, the above corresponds to the general case of the aperture problem. It is interesting to note that Barron, Fleet, and Beauchemin [4] recognize the determinant as a better confidence measure in the application of the Uras et al. [22] optical flow algorithm than the condition number as used in the original paper. Our analysis using the principles of linear algebra agrees with their empirical findings.

So we shall use $|\det(R_s)|$ as a confidence measure.

3.2.4 Integration of Confidence Measures

Based on the above analysis, we choose a combination of confidence measures according to the nature of a given image sequence.

If the image sequence contains numerous moving objects or the brightness changes significantly, residuals should be used as confidence measures. The residual is unique in the sense that it captures the three problems in 3.2.1, which no other measure does.

Condition number and determinant have something in common although they may capture differ-

ent scenarios. Together they signify the relationship between numerical instability and the aperture problem. We suggest the multiplicative combination of these two, $\det(R_s)/\kappa(R_s)$, which is equivalent to using $|\lambda|_{min}$. This has been proposed by Giroi et al.[8] in a similar context and was used in Barron’s implementation [4] of Lucas and Kanade’s optical flow algorithm. In our algorithm, it simply means the smaller of the derivatives in x and y directions. We shall use it to capture the aperture problems and to avoid numerical instability.

All the above mentioned problems are not unique to our algorithm. In fact, many of them are common to other optical flow algorithms. However, we believe our algorithm is the first serious effort to capture most of the problems in optical flow computation in a quantitative way.

4. Experiments

Based on the work of Barron, Fleet, and Beauchemin[4], we conducted extensive comparisons of our algorithm with other current optical flow algorithms, including ones by Horn and Schunck[14], Lucas and Kanade[17], Uras et al. [22], Nagel[19], Anandan[2], Singh[20], Heeger[12], Waxman et al. [24], Fleet and Jepson[7]. The synthetic image sequences we used for comparison are Sinusoid, Translating Tree, Diverging Tree, and Yosemite Fly-by. All of the above come with optical flow ground truth. The real image sequences we used for demonstration are SRI Trees, Rubik Cube, and Hamburg Taxi. All of these images were provided by J.L. Barron.

The error statistic utilized is the angle error between the computed optical flow time-space direction $(u_e, v_e, 1)$ and the ground truth flow time-space direction $(u_c, v_c, 1)$ averaged over the whole image. Note that the 3D directions are different from the intuitive 2D flow directions. It takes into account the magnitude as normalized with respect to time. Refer to [4] for more details.

We recognize the importance of optical flow magnitude for subsequent applications, e.g. time to contact. To separate magnitude error from angle error, we listed the magnitude error, defined as the difference of the magnitudes of computed flow and ground truth flow, scaled by the ground truth flow. But there are no comparison data available.

In order to make extensive comparisons, we implemented our algorithm in such a way that a threshold on a specified confidence measure can be used to eliminate unreliable flow values, thus controlling the number of points, or density, for which computed flow field values are available. In all the tables in the following subsections, the first column is density, the second and third columns present the error statistics for our algorithm for that density, and the fourth and fifth columns present the error statistics for a given comparison algorithm for the same density. The error statistics and associated density for the comparison algorithms were obtained from Barron et al. [4]. For a single technique with multiple rows of data, different threshold values are used in the algorithm to produce multiple densities of output. For the actual threshold values of the comparison algorithms, refer to Barron et al.[4].

The results show that our algorithm performs consistently well over a wide set of images, if not the best in every single sequence. On the other hand, most of the other techniques included in comparisons perform well only over a narrow set of images.

4.1 Sinusoid

Sinusoid is a synthetic image sequence (Fig 2) generated by a spatial sinusoidal wave traversing toward the upper right side. For our method we chose a window size large enough ($17 \times 17 \times 7$ for x, y, t) to prevent aliasing. Fig 3.1 shows the true optical flow for sinusoid, while Fig 3.2 shows the flow computed with our method. The output density in Fig 3.2 is 100%. $|1/r|$ was used as the confidence measure in Table 1. Our algorithm performs far better than all of the other algorithms except Fleet and Jepson's. The average magnitude error is 1.83% for 100% density.

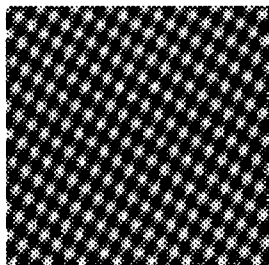


Fig 2. Spatial sinusoid

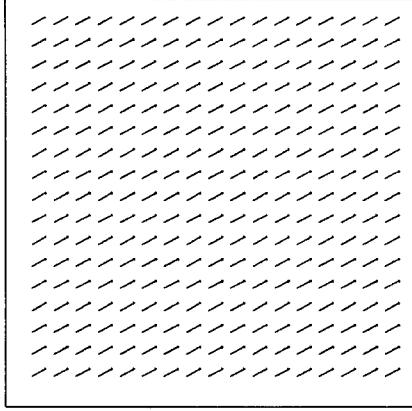


Fig 3.1 True optical flow for sinusoid

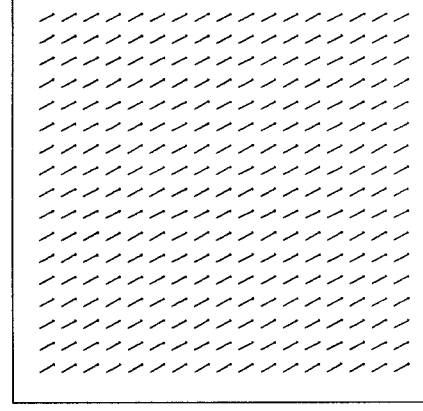


Fig 3.2 Computed optical flow (100%)

Table 1: Summary of Sinusoid Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
100%	0.65°	0.25°	4.19°	0.50°	Horn & Schunck (original unthresholded)
			2.55°	0.59°	Horn & Schunck (modified unthresholded)
			2.47°	0.16°	Lucas and Kanade (unthresholded)
			2.59°	0.71°	Uras et al. (unthresholded)
			2.55°	0.93°	Nagel
			30.80°	5.45°	Anandan
			2.24°	0.02°	Singh (step 1 unthresholded)
			0.03°	0.01°	Fleet and Jepson
12.8%	0.65°	0.26°	64.26°	26.14°	Waxman et al.

4.2 Translating And Diverging Tree

The translating and diverging tree sequences are two realistic synthetic sequences simulating the motion of simple translation and expansion, respectively, of a poster (Fig 4). The window size used in our method is 19x19x11 for translating tree and 17x17x9 for diverging tree. Due to the lack of texture in some background areas, we used $|det(R_s)|$ as the confidence measure. The magnitude error for translating tree is 2.37% for 100% density and 1.23% for 50% density; for diverging tree, it is 2.08% for 100% density and 1.51% for 50% density. Fig 5 and Fig 6 show the results. The output density for Fig 5.2 and Fig 6.2 is 90%. Error statistics are shown in Table 2 and Table 3. Note that when the density is low (<10%), the error increases. It is because the area with high $|det(R_s)|$ corresponds to high contrast corners and may well represent motion bound-

aries. Only Uras's and Fleet and Jepson's algorithms perform better than ours in terms of average error for translating tree. For diverging tree, our algorithm's performance is good but not outstanding. It is due to the fact that the velocity variation is large in a local window.



Fig 4. Translating and diverging tree

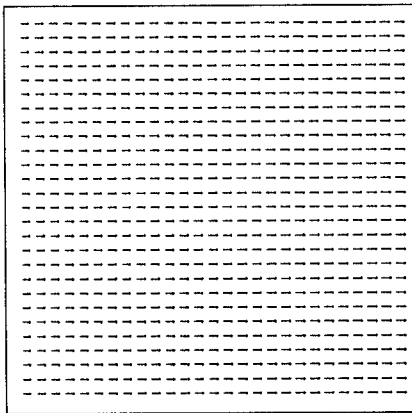


Fig 5.1 True flow for translating tree

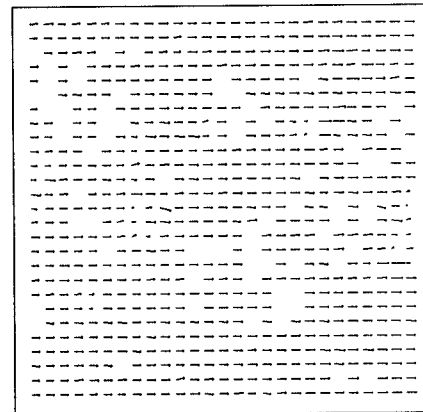


Fig 5.2 Computed flow for translating tree (90%)

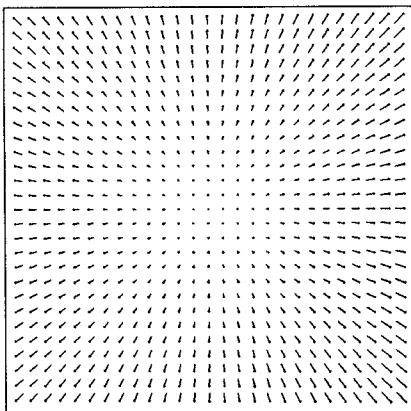


Fig 6.1 True flow for diverging tree

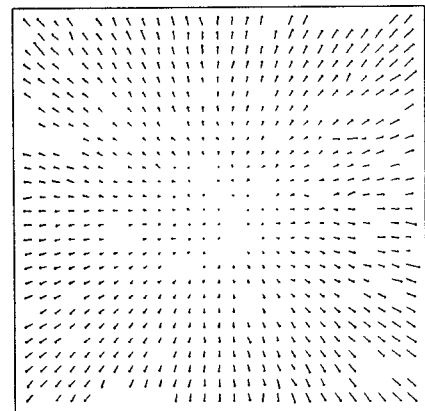


Fig 6.2 Computed flow for diverging tree (90%)

Table 2: Summary of Translating Tree Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
100%	1.05°	1.45°	38.72°	27.67°	Horn & Schunck (original unthresholded)
			2.02°	2.27°	Horn & Schunck (modified unthresholded)
			0.62°	0.52°	Uras et al. (unthresholded)
			2.44°	3.06°	Nagel
			4.54°	3.10°	Anandan
			1.64°	2.44°	Singh (step 1 unthresholded)
			1.25°	3.29°	Singh (step 2 unthresholded)
99.6%	1.04°	1.42°	1.11°	0.89°	Singh (step 2)
74.5%	0.72°	0.82°	0.32°	0.38°	Fleet and Jepson
53-57%	0.66°	0.77°	32.66°	24.50°	Horn & Schunck (original)
			5.63°	2.78°	Heeger (level 1)
			1.89°	2.40°	Horn & Schunck (modified)
49.7%	0.65°	0.75°	0.23°	0.19°	Fleet and Jepson
44.2%	0.63°	0.69°	8.50°	13.50°	Heeger (level 0)
40-42%	0.62°	0.66°	0.46°	0.35°	Uras et al.
			0.72°	0.75°	Singh (step 1)
			0.66°	0.67°	Lucas and Kanade
26.8%	0.60°	0.59°	0.25°	0.21°	Fleet and Jepson
13.1%	0.58°	0.51°	0.56°	0.58°	Lucas and Kanade
1.9%	0.70°	0.60°	6.66°	10.72°	Waxman et al.

Table 3: Summary of Diverging Tree Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
100%	2.91°	2.89°	12.02°	11.72°	Horn & Schunck (original unthresholded)
			2.55°	3.67°	Horn & Schunck (modified unthresholded)
			4.64°	3.48°	Uras et al. (unthresholded)
			2.94°	3.23°	Nagel
			7.64°	4.96°	Anandan
			17.66°	14.25°	Singh (step 1 unthresholded)
			8.60°	4.78°	Singh (step 2 unthresholded)
99%	2.83°	2.57°	8.40°	4.78°	Singh (step 2)
73.8%	2.43°	1.97°	4.95°	3.09°	Heeger (combined)
60-61%	2.32°	1.84°	0.99°	0.78°	Fleet and Jepson
			8.93°	7.79°	Horn & Schunck (original)
			3.83°	2.19°	Uras et al.
46-48%	2.26°	1.84°	2.50°	3.89°	Horn & Schunck (modified)
			0.80°	0.73°	Fleet and Jepson
			1.94°	2.06°	Lucas and Kanade
28.2%	2.29°	1.92°	0.73°	0.46°	Fleet and Jepson

Table 3: Summary of Diverging Tree Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
24.3%	2.29°	1.91°	1.65°	1.48°	Lucas and Kanade
4.9%	2.52°	2.33°	13.69°	11.83°	Waxman et al.
3.9%	2.51°	2.27°	5.62°	6.16°	Singh (step 1)

4.3 Yosemite Fly-by

The Yosemite Fly-by sequence is a synthetic realistic image sequence (Fig 7). The flight scene is simulated from actual aerial photos and digital-terrain maps plus artificial sky and clouds. Since the clouds in the sky change brightness over time, it presents difficulties for our method. Based on our previous analysis, we used $|1/r|$ as the confidence measure to eliminate those data points which correspond to a large blank area in the sky and at motion boundaries in Fig 8.2. Since the motion is rather fast in some areas, we used a larger window (21x21x7). Error statistics are shown in Table 4. The magnitude error is 27.2% for 100% density and 9.27% for 50% density. Again, the clouds account for the large magnitude error. The output density for Fig 8.2 is 75%. Only Lucas and Kanade's algorithm performs better than ours. From this we believe our algorithm should work well with real images. This will be shown in the next subsection.



Fig 7. Yosemite fly-by image

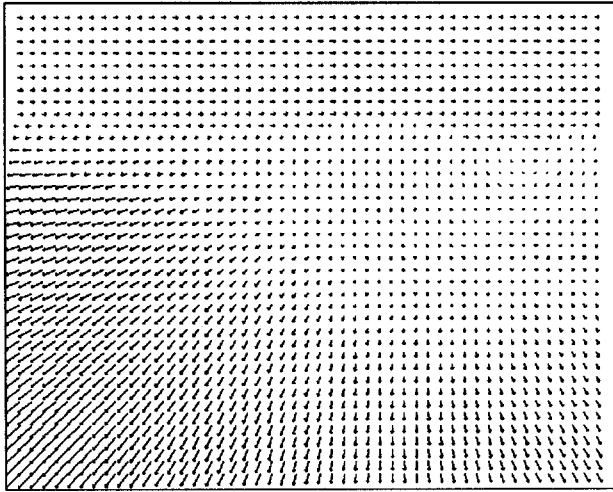


Fig 8.1 True optical flow field for Yosemite fly-by

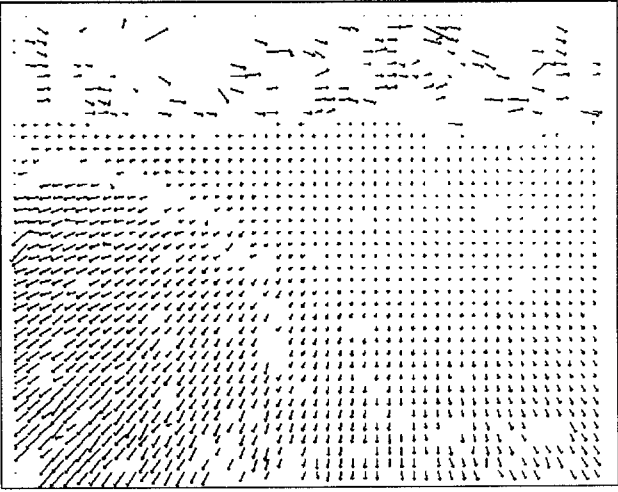


Fig 8.2 Computed optical flow for Yosemite fly-by (75%)

Table 4: Summary of Yosemite Fly-by Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
100%	10.17°	13.61°	32.43°	30.28°	Horn & Schunck (original unthresholded)
			11.26°	16.41°	Horn & Schunck (modified unthresholded)
			10.44°	15.00°	Uras et al. (unthresholded)
			11.71°	10.59°	Nagel
			15.84°	13.46°	Anandan
			18.24°	17.02°	Singh (step 1 unthresholded)
			13.16°	12.07°	Singh (step 2 unthresholded)
97.8%	9.73°	13.21°	12.9°	11.57°	Singh (step 2)
64.2%	5.14°	7.67°	20.89°	34.26°	Heeger (level 0)
59.6%	4.95°	7.42°	25.41°	28.14°	Horn & Schunck (original)
44.8%	4.59°	6.90°	11.74°	19.04°	Heeger (combined)
33-35%	4.42°	6.72°	4.10°	9.58°	Lucas and Kanade
			4.29°	11.24°	Fleet and Jepson
			5.48°	10.41°	Horn & Schunck (modified)
30.6%	4.40°	6.70°	4.95°	12.39°	Fleet and Jepson
15%	4.34°	6.57°	10.51°	12.11°	Heeger (level 1)
			6.74°	16.01°	Uras et al.
8.7%	4.40°	6.76°	3.05°	7.31°	Lucas and Kanade
7.4%	4.45°	6.88°	20.32°	20.60°	Waxman et al.
2.4%	4.90°	7.60°	11.51°	11.83°	Heeger (level 2)
2.2%	4.85°	7.51°	16.29°	25.70°	Singh (step 1)

4.4 Real Images

Current optical flow algorithms often have difficulty with real image sequences. The reasons include camera jitter, nonrigidity of objects, and brightness variations due to changes in lighting or adjusted camera settings. Here we show that our confidence measures can eliminate many of the unreliable flow values. The optical flow output of the following subsections has undergone thresholding using two confidence measures, $|1/r|$ and $|\lambda|_{min}$.

4.4.1 SRI Trees

The SRI trees sequence (Fig 9.1) is a scene of stationary trees taken from a camera moving laterally. The tree in the center of the scene is closer to the camera and therefore generates large optical flow, as shown in the computed optical flow field (Fig 9.2). The noise in this sequence is significantly larger than the others. The output density for this sequence is 71%.



Fig 9.1 SRI trees image

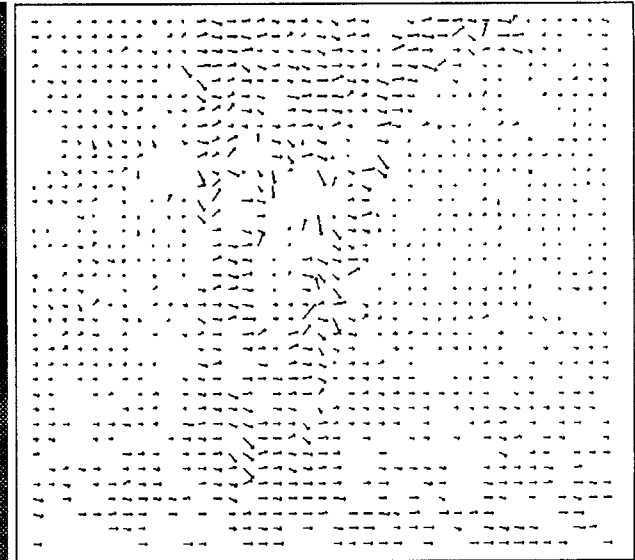


Fig 9.2 Computed optical flow field for SRI trees

4.4.2 Rubik Cube

The Rubik cube sequence (Fig 10.1) is generated by a stationary camera looking at a rotating platform with a Rubik cube on it. Note that the output in the area without texture is eliminated by thresholding on the confidence measure and the flow field is well organized and corresponds to moving objects (Fig 10.2). The output density for this sequence is 47%.

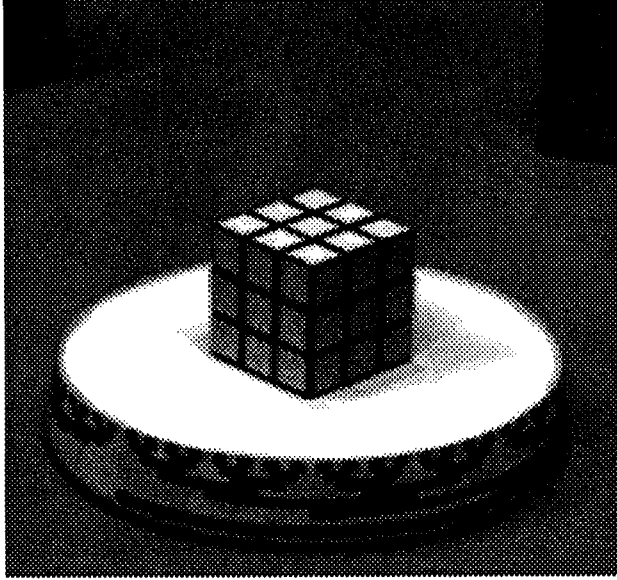


Fig 10.1 Rubik Cube image

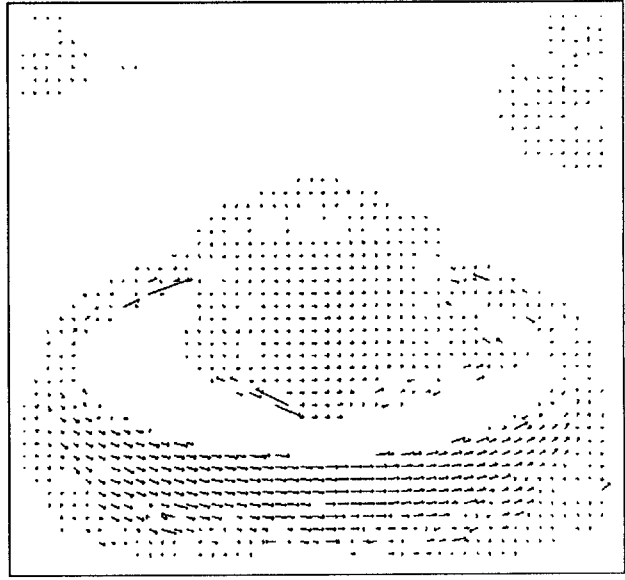


Fig 10.2 Computed optical flow field for Rubik Cube

4.4.3 Hamburg Taxis

The Hamburg taxis sequence (Fig 11.1) contains three independently moving vehicles in front of a stationary background. Our algorithm captures all the three moving objects, as shown in the flow field (Fig 11.2). The output density of the sequence is 50%.

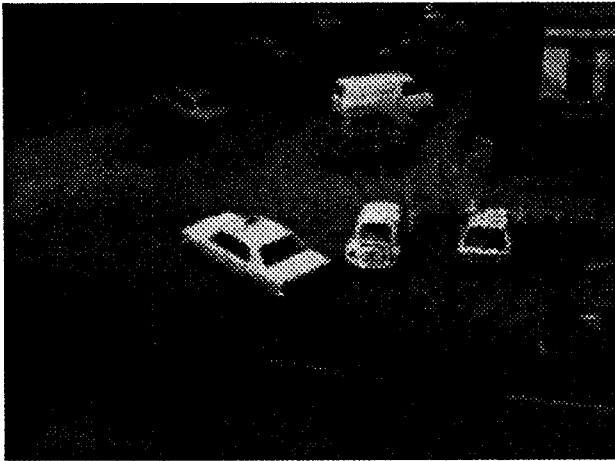


Fig 11.1 Hamburg taxi image

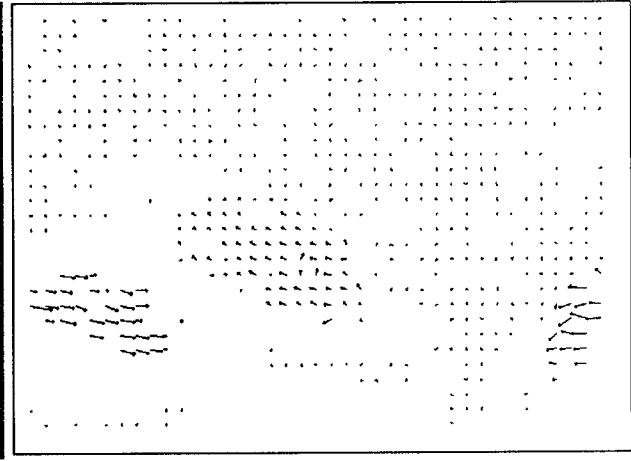


Fig 11.2 Computed optical flow for Hamburg taxis

5. Conclusion

Some claim that the optical flow problem is ill-posed. We find that as long as the computation is local and robust numerical techniques are employed to provide sufficient information about the

reliability of the result, then the extraction of reliable optical flow values is possible and the computed flow field should be useful for other applications such as obstacle avoidance, 3-D scene reconstruction, etc. The algorithm presented here is an attempt in this direction. We have demonstrated that our algorithm compares very favorably with other existing algorithms and performs consistently well over a variety of synthetic and real images.

Appendix. A

We prove Theorem 1 as follows:

Proof: The first equality comes from the orthogonality of $\{\bar{H}_n(x)\}$. We now prove the second equality, which claims that the scalar product of a function and the k th Hermite polynomial is equal to the scalar product of the k th derivative of the function and 1.

$$\begin{aligned}
\langle I, \bar{H}_k \rangle &= \int_{-\infty}^{\infty} G(x) I(x) \bar{H}_k(x) dx \\
&= \int_{-\infty}^{\infty} G(x) I(x) (-1)^k G^{-1}(x) \frac{d^k G(x)}{dx^k} dx \\
&= (-1)^k \int_{-\infty}^{\infty} I(x) \frac{d^k G(x)}{dx^k} dx \\
&= (-1)^k I(x) \frac{d^{k-1} G(x)}{dx^{k-1}} \Big|_{-\infty}^{\infty} + (-1)^{k-1} \int_{-\infty}^{\infty} \frac{dI(x)}{dx} \frac{d^{k-1} G(x)}{dx^{k-1}} dx \\
&= \int_{-\infty}^{\infty} I^{(1)}(x) (-1)^{k-1} \frac{d^{k-1} G(x)}{dx^{k-1}} dx \\
&= \langle I^{(1)}, \bar{H}_{k-1} \rangle \\
&\vdots \\
&= \langle I^{(k)}, \bar{H}_0 \rangle
\end{aligned}$$

□

Appendix. B

First, let us model the error due to brightness change. The context (Section 3.2.1) has stated the

two possible reasons for brightness change. One is referred to as reflection; the other is referred to as glooming. Regarding them as additive noise, we model the image intensity as:

$$\tilde{I}(x, y, t) = I(x, y, t) + N(x, y, t) \quad (32)$$

As far as the image sequence is concerned, the difference between reflection and glooming is that reflection can be represented by impulse noise and glooming can be represented by sustained noise with respect to time (Fig 12).

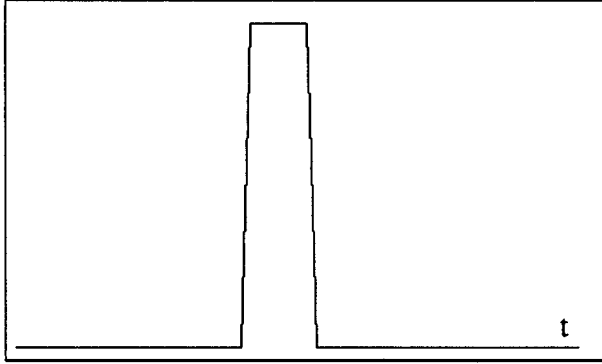


Fig 12.1 Reflection noise in the time dimension.

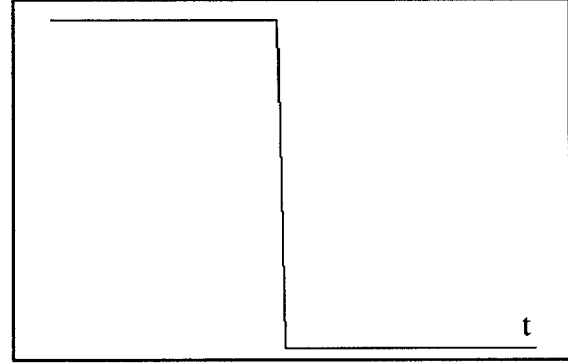


Fig 12.2 Glooming noise in the time dimension.

So for reflection noise, assuming there is a single frame of strong reflection, we can simplify (32) as:

$$\tilde{I}(x, y, t) = \begin{cases} I(x, y, t) & t \neq 0 \\ I(x, y, t) + r(x, y) & t = 0 \end{cases} \quad (33)$$

The expansion coefficients are

$$\begin{aligned} \tilde{I}_{ijk}(x, y, t) &= \sum_V \sum \sum \tilde{I}(x + x_0, y + y_0, t + t_0) H_{ijk}(x_0, y_0, t_0) G(x_0, y_0, t_0) \\ &= \hat{I}_{ijk}(x, y, t) + \sum_{x_0} \sum_{y_0} r(x + x_0, y + y_0) (H_{ijk}(x_0, y_0, -t) G(x_0, y_0, -t)) \end{aligned} \quad (34)$$

Realizing that the zeroth Hermite polynomial H_0 is an even function and the first Hermite polynomial H_1 is an odd function (Fig 13), we know that when t is equal to 0, the noise affects $\tilde{I}_{ij0}(t)$ more than $\tilde{I}_{ij1}(t)$ and when t is not equal to zero but small, then the noise affects $\tilde{I}_{ij1}(t)$ more than $\tilde{I}_{ij0}(t)$. The above statement will hold true even with multiple frames of strong reflection.

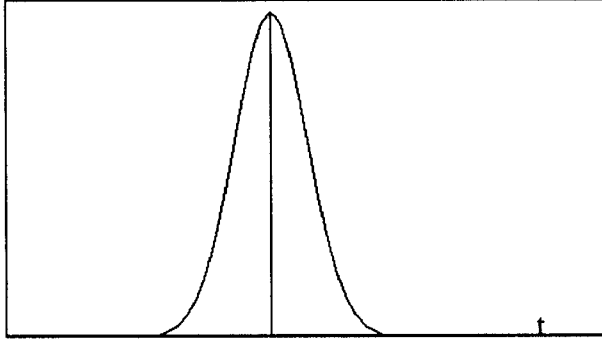


Fig 13.1 $\bar{H}_0(t) G(t)$ is an even function.

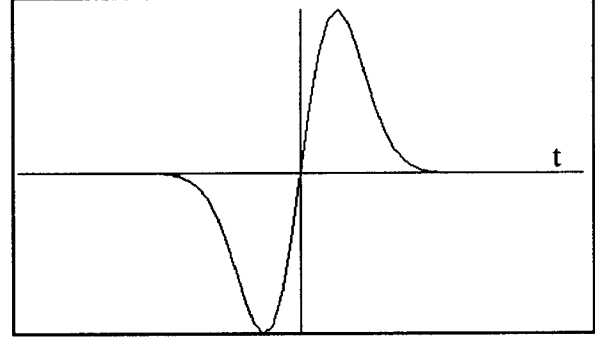


Fig 13.2 $\bar{H}_1(t) G(t)$ is an odd function.

Note that (22) indicates that b consists of $\tilde{I}_{ij1}(t)$ only and A consists of $\tilde{I}_{ij0}(t)$ only. We conclude here that when a strong reflection happens in the middle frame of the sequence, the matrix A is perturbed more; otherwise vector b is perturbed more.

For glooming noise, assuming that from a certain point of time on, the brightness change for some objects, we model the image intensity as follows:

$$\tilde{I}(x, y, t) = \begin{cases} I(x, y, t) & t < 0 \\ I(x, y, t) + g(x, y) & t \geq 0 \end{cases}, \quad (35)$$

The noise $g(x, y)$ may well be proportional to $I(x, y)$ in the real world though.

Contrary to the case of reflection noise, from the even and odd properties of Hermite polynomials and similar reasoning to that above, we conclude that when t is equal to or close to 0, the noise affects both A and b , and for other values of t , the effect is negligible. This fact can be easily seen by aligning Fig 12.1(2) and Fig 13.1(2) correspondingly.

Now if the source of error is independently moving objects (objects A and B) in a small window or occlusion, the image intensity is

$$\tilde{I}(x, y, t) = \begin{cases} I(x, y, t) & (x, y, t) \in V_A \\ I_B(x, y, t) & (x, y, t) \in V_B \end{cases}, \quad (36)$$

where V_A (V_B) contains the image of object A(B). Suppose object A occupies a larger area in the window and we intend to estimate the velocity of A. Regarding the image of object B as a source

of noise, we can model $I_B(x, y, t)$ as an extension of object A's image $I(x, y, t)$ plus noise $I_B(x, y, t) - I(x, y, t)$.

For most kinds of noise, the following model may be adequate:

$$\tilde{E} = \min \| (A + N)f + (b + \Delta b) \| \text{ as written in (29).}$$

We conclude this appendix with a claim that the perturbation caused by multiple moving objects is usually greater than that caused by brightness change. It is often more random and covers a larger volume of the 3D spatio-temporal image.

Appendix. C

Let A and b , defined in (22), contain no noise and let the noise is modelled as in (29). Then,

$$E = Af + b = 0 \text{ and } f = -(A^T A)^{-1} A^T b. \quad (37)$$

Let the new optical flow be \tilde{f} and new residual be \tilde{E} and assume that $N \ll A$ and $\Delta b \ll b$ element-wise. Then

$$\tilde{f} = -[(A + N)^T (A + N)]^{-1} (A + N)^T (b + \Delta b), \text{ and} \quad (38)$$

$$\begin{aligned} [(A + N)^T (A + N)]^{-1} &\approx (A^T A [I + (A^T A)^{-1} (A^T N + N^T A)])^{-1} \\ &\approx [I - (A^T A)^{-1} (A^T N + N^T A)] (A^T A)^{-1}, \text{ so} \end{aligned} \quad (39)$$

$$\tilde{f} \approx - (A^T A)^{-1} A^T b + (A^T A)^{-1} (A^T N + N^T A) (A^T A)^{-1} A^T b + (A^T A)^{-1} A^T \Delta b - (A^T A)^{-1} A^T \Delta b.$$

Using (37), this can be simplified as follows:

$$\tilde{f} \approx f - (A^T A)^{-1} A^T N f - (A^T A)^{-1} A^T \Delta b \text{ and } \Delta f \approx (A^T A)^{-1} A^T N f - (A^T A)^{-1} A^T \Delta b.$$

For the residual, substituting \tilde{f} into (29), and using (37), we have

$$\begin{aligned} \tilde{E} &\approx \| (A + N)f - A (A^T A)^{-1} A^T N f - A (A^T A)^{-1} A^T \Delta b + b + \Delta b \| \\ &\approx \| (I - A (A^T A)^{-1} A^T) (Nf + \Delta b) \| \end{aligned} \quad \text{as in (31).}$$

To understand \tilde{E} better, we analyze matrix $I - A (A^T A)^{-1} A^T$, denoted by K .

Let $a = \hat{I}_{110}^2 - \hat{I}_{200}\hat{I}_{020}$, $b = \hat{I}_{100}\hat{I}_{020} - \hat{I}_{010}\hat{I}_{110}$, $c = \hat{I}_{010}\hat{I}_{200} - \hat{I}_{100}\hat{I}_{110}$, then

$$K = I - A(A^T A)^{-1}A^T = \frac{1}{a^2 + b^2 + c^2} \begin{bmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{bmatrix} \quad (40)$$

The eigenvalues of matrix K are (0,0,1), which means that it maps any vector to one single direction, specified by the eigenvector corresponding to the nontrivial eigenvalue.

Appendix. D

We show that a high condition number of matrix R_s corresponds to one of the two scenarios in Section 3.2.2. Recalling (26), with Q unitary (length-preserving and relative angle-preserving), we have $|\lambda_1| = \|(I_{100}, I_{200}, I_{110})^T\|$, and $\sqrt{\lambda_2^2 + d^2} = \|(I_{010}, I_{110}, I_{020})^T\|$. A high condition number means either 1. $|\lambda_2| \gg |\lambda_1|$ or 2. $|\lambda_2| \ll |\lambda_1|$. In case 1, it is apparent that the second column of matrix A $(I_{010}, I_{110}, I_{020})^T$ is much larger than the first column $(I_{100}, I_{200}, I_{110})^T$. This means that the derivatives in the direction of y are much larger than the derivatives in the direction of x . This corresponds to the first scenario in 3.2.2. In case 2, there are two possibilities: one is that d is also very small, the other is that d is comparable with λ_1 . If both λ_2 and d are small, we have a case opposite to case 1. This also corresponds to the first scenario in 3.2.2. If d is not small, it indicates that the two column vectors of matrix A are close to being linearly dependent. It means that the derivatives in the x direction are proportional to the derivatives in the y direction. This corresponds to the second scenario in 3.2.2.

Appendix. E

We show that for any 2x2 matrix R , a small determinant relative to the elements of the matrix results in a high condition number, i.e. if $|\det(R)| \ll |\text{Tr}(R)|$, then $\text{cond}(R)$ will be large.

$cond(R) = \frac{|\lambda|_{max}}{|\lambda|_{min}}$, let $Tr(R) > 0$ without loss of generality, then

$$\begin{aligned} cond(R) &= \frac{|Tr(R) + \sqrt{Tr(R)^2 - 4det(R)}|}{|Tr(R) - \sqrt{Tr(R)^2 - 4det(R)}|} \approx \frac{|Tr(R) + (Tr(R) - 2det(R))|}{|Tr(R) - (Tr(R) - 2det(R))|} \\ &= \frac{Tr(R) - det(R)}{|det(R)|} \approx \frac{Tr(R)}{|det(R)|} \end{aligned}$$

as can be seen from the assumption that the condition number is very large.

Acknowledgment

We would like to thank J.L. Barron for providing all the images and comparison results on which our experiments are based.

References

- [1] Adelson, E. H. and Bergen, J. R., "The Extraction of Spatiotemporal Energy in Human and Machine Vision", Proceedings of IEEE Workshop on Visual Motion, Charleston SC, pp.151-155, 1986.
- [2] Anandan, P., "Measuring Visual Motion from Image Sequences", Ph.D. Thesis, COINS TR 87-21, University of Massachusetts, Amherst MA, 1987.
- [3] Barnard, S. T. and Thompson, W. B., "Disparity Analysis of Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, pp. 333-340, 1980.
- [4] Barron, J. L., Fleet, D. J., and Beauchemin, S. S., "Performance of Optical Flow Techniques", RPL-TR-9107, Department of Computer Science, Queens University, 1992.
- [5] Barron, J. L., Fleet, D. J., and Beauchemin, S. S., "Performance of Optical Flow Techniques", to be published in International Journal of Computer Vision, 1993.
- [6] Buxton, B. F. and Buxton, H., "Computation of Optical Flow from The Motion of Edge Features in Image Sequences," Image and Vision Computing, vol. 2, pp. 59-74, 1984.
- [7] Fleet, D.J. and Jepson, A.L., "Computation of Component Image Velocity from Local Phase Information," International Journal of Computer Vision, vol. 5, pp. 77-104, 1990.
- [8] Giroi, F., Verri, A., Torre, V., "Constraints for the Computation of Optical Flow," Proceed-

- ings of IEEE Workshop on Visual Motion, Irvine, CA, pp. 116-124, 1989.
- [9] Grzywacz, N.M. and Yuille A.L., "A Model for Estimate of Local Image Velocity by Cells in The Visual Cortex," Proceedings of the Royal Society of London, vol. A 239, pp. 129-161, 1990.
 - [10] Haralick, R. M. and Lee, J. S., "The Facet Approach to Optic Flow," Proceedings of DARPA Image Understanding Workshop, Arlington VA, pp. 84-94, 1983.
 - [11] Hashimoto M, and Sklansky, J. "Multiple-Order Derivatives for Detecting Local Image Characteristics," Computer Vision, Graphics, and Image Processing, vol. 39, pp. 28-55, 1987.
 - [12] Heeger, D. J., "Optical Flow Using Spatiotemporal Filters", International Journal Of Computer Vision, vol. 1, pp. 279-302, 1988.
 - [13] Heeger, D.J., "Model for the Extraction of Image Flow," Journal of the Optical Society of America, vol. A4, pp. 1455-1471, 1987.
 - [14] Horn, B. K. P. and Schunck, B. G., "Determining Optical Flow," Artificial Intelligence, vol. 17, pp. 185-204, 1981.
 - [15] Kearney, J. K., Thompson, W. B. and Boley, D. L., "Optical Flow Estimation: An Error Analysis of Gradient Based Methods With Local Optimization," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, pp. 229-244, 1987.
 - [16] Koenderink, J.J. and Van Doorn, A.J., "Representation of Local Geometry in The Visual System," Biological Cybernetics, vol. 55, pp. 367-375, 1987.
 - [17] Lucas, B. D. and Kanade, T., "An Iterative Image Registration Technique with An Application to Stereo Vision", Proceedings of DARPA Image Understanding Workshop, pp.121-130, 1981.
 - [18] Nagel, H. H., "Constraints for the Estimation of Displacement Vector Fields from Image Sequences", Proceedings of International Joint Conference on Artificial Intelligence, pp. 945-951, 1983.
 - [19] Nagel, H. H., "Displacement Vectors Derived from Second-order Intensity Variations in Image Sequences", Computer Vision, Graphics and Image Processing, vol. 21, pp. 85-117, 1983.
 - [20] Singh, A., "An Estimation-Theoretic Framework for Image-Flow Computation", Proceedings of International Conference on Computer Vision, pp. 169-177, 1990.
 - [21] Singh, A., "Optical Flow Computation, A Unified Perspective", IEEE Computer Society Press, 1991.

- [22] Thompson, W.B., Barnard, S.T., "Low-Level Estimation and Interpretation of Visual Motion", *Computer*, vol.14, pp. 20-27, 1981.
- [23] Uras S., Girosi F., Verri A., Torre V. "A Computational Approach to Motion Perception", *Biological Cybernetics*, vol. 60, pp. 79-97, 1988.
- [24] Watson, A. B. and Ahumada, A. J., "Model of Human Visual-motion Sensing." *Journal of the Optical Society of America A*, vol. 2, pp. 322-342, 1985.
- [25] Waxman, A.M., Wu J., Bergholm F. "Convected Activation Profiles and receptive Fields for Real Time measurement of Short Range Visual Motion", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, pp. 717-723 1988.
- [26] Wong, R. Y. and Hall E. L., "Sequential Hierarchical Scene Matching", *IEEE Transactions on Computers*, vol. C-27, pp.358-366, 1978.
- [27] Young, R. A., "Simulation of Human Retinal Function with The Gaussian Derivative Model", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, pp. 564-569, 1986.