Proceedings of the International Conference on Pattern Recognition, Jerusalem, Israel, October 1994.

A Generalized Motion Model for Estimating Optical Flow 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)
Blg. 220, Rm B124, Gaithersburg, MD 20899
[†]Center for Automation Research/Department of Electrical Engineering,
University of Maryland, College Park 20742

Abstract

Most optical flow algorithms assume local 2-D translational motion. For image sequences with significant expansion or rotation, these methods are not adequate. We present a new model that encompasses both translation and expansion, as induced by any 3-D relative translation. The formulation of the model requires an extra parameter for expansion. Therefore, we establish an algorithm based on Hermite polynomial differentiation filters, whose orthogonality and Gaussian derivative properties insure numerical stability. In this way, we extend the filter design to higher orders so as to improve accuracy in estimation. The use of higher order differentiation filters is justified by the algorithm's reliable performance. We tested our algorithm using an evaluation scheme established by Barron et al., involving extensive comparisons with other existing algorithms. The tests show that our algorithm performs consistently well over a wide variety of images, especially diverging ones.

1. Introduction

Most optical flow algorithms assume a simple translational image motion in local image neighborhood as in (1):

$$I(x, y, t) = F(x-ut, y-vt) \tag{1}$$

This assumption is only acceptable if the neighborhood is small. Unfortunately, a small local neighborhood limits the information that can be used for estimation and also the extent of smoothing. We are thus motivated to find a more general motion model that represents real world situations and meets practical computational requirements.

Generally, image plane motion in the form of translation, divergence, curl, and deformation all need to be modelled. We take a step forward by modeling motion that is both translational and expanding (contracting), as depicted in Fig 1. We formalize the model in the next section.

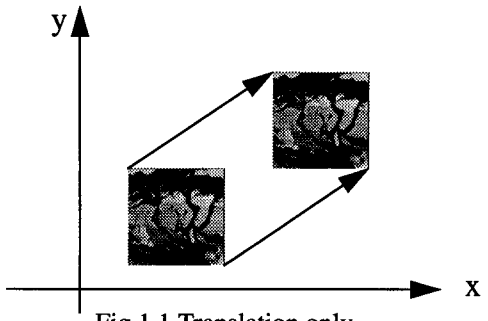

Fig 1.1 Translation only

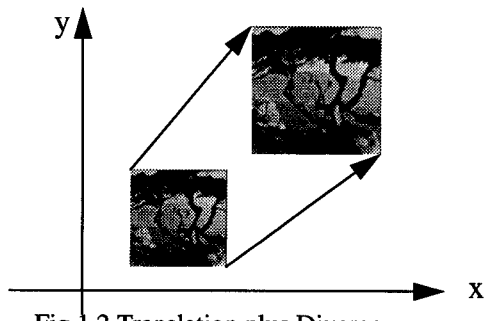

Fig 1.2 Translation plus Divergence

However, this model poses a new difficulty in the computation since we need to solve not only for translation parameters but also an expansion parameter. To overcome this, we must use more filters to derive additional equations. The selection of the filters determines the reliability of the flow results. We formulate a theory of derivative filters using 3-D Hermite polynomials and develop an algorithm based on the theory. There are three reasons to justify the choice of 3-D Hermite polynomial filters: first, the orthogonality and Gaussian properties of the filters insure numerical stability; second, it is extensible to the higher order derivatives that we desire; and third, numerous physiological models[7,13,23] support the theory that the visual receptive field can be modeled by Gaussian derivatives of various widths.

Finally, to demonstrate the reliability of our flow results, we evaluate our algorithm using the scheme established by Barron et al.[2] Extensive comparisons with existing algorithms show that our algorithm performs consistently well over a wide set of images.

2. The Generalized Motion Model

We extend the image motion from simple translation to translation and expansion. A 3-D point at position $\vec{P} = (X, Y, Z)$ *, under perspective projection, projects to a point in 2-D image plane, (x, y) ,

$$\begin{aligned} x &= fX/Z \\ y &= fY/Z, \text{ where } f \text{ is the focal length of the projection.} \end{aligned} \tag{2}$$

Suppose there is relative 3-D translational motion $\vec{P}(t) = (X + U_X t, Y + U_Y t, Z - U_Z t)$. A conventional algorithm would assume small translation $U_Z t$ relative to the distance Z and arrive at equation (1)†. While preserving the assumption, we still desire to account for the expansion effect of this micro translation. Hence, $(x(t), y(t))$ becomes

$$\begin{aligned} x(t) &= f(X + U_X t) / (Z - U_Z t) \\ y(t) &= f(Y + U_Y t) / (Z - U_Z t) \, . \end{aligned} \tag{3}$$

Brightness constancy and (3) yield

$$I(x, y, t) = F\left(x\left(1 - \frac{U_Z}{Z}t\right) - \frac{fU_X}{Z}t, y\left(1 - \frac{U_Z}{Z}t\right) - \frac{fU_Y}{Z}t\right). \tag{4}$$

Equation (4) is the formulation of our new motion model, for which the optical flow is derived as

$$(u, v) = \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}\right) = \left(\frac{fU_X}{Z - U_Z t} + \frac{U_Z x}{Z - U_Z t}, \frac{fU_Y}{Z - U_Z t} + \frac{U_Z y}{Z - U_Z t}\right) \approx \left(\frac{fU_X}{Z} + \frac{U_Z x}{Z}, \frac{fU_Y}{Z} + \frac{U_Z y}{Z}\right) \tag{5}$$

Let the expansion $\frac{U_Z}{Z}$ be denoted by s, and the 2-D translation velocity $\left(\frac{fU_X}{Z}, \frac{fU_Y}{Z}\right)$ by (p, q) .

---

*. In an observer-centered coordinate frame; Z is the axis along the line of sight.

†. Using the brightness constancy assumption.

Rewrite (4) and (5) as

$$I(x, y, t) = F(x(1-st) - pt, y(1-st) - qt) \tag{6}$$

$$(u, v) = (p + sx, q + sy) \tag{7}$$

Intuitively, (7) states that optical flow has two components: translation (p, q), and expansion (sx, sy). Translation is induced by (U_X, U_Y) only, and expansion by U_Z only. Note that when $s = 0$, then $(u, v) = (p, q)$ and (6) becomes (1).

To complete the model, note that the Focus of Expansion (FOE) is not always at the image origin. In fact, when we perform local estimation with this motion model, the position of the FOE relative to the origin, which is the center of the current window, is changing with a moving window. Therefore, (6) and (7) should be rewritten as

$$I(x, y, t) = F((x-x_0)(1-st) - pt + x_0, (y-y_0)(1-st) - qt + y_0) \tag{8}$$

$$(u, v) = (p + s(x-x_0), q + s(y-y_0)), \tag{9}$$

where (x_0, y_0) is the location of the FOE relative to the current window center. Since we will find (u, v) as a whole, the introduction of (x_0, y_0) does not complicate the computation.

In the model, there are expansion as well as flow parameters to be estimated. It certainly demands more elegant techniques to achieve high accuracy relative to the conventional algorithms. Next, we introduce the theory of Hermite polynomial filters and its application in our model.

3. Hermite Polynomial Filters

3.1 Hermite Polynomials

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

$$\frac{d^2 H_n}{dx^2} - 2x\frac{dH_n}{dx} + 2nH_n = 0. \tag{10}$$

The $H_n(x)$ are derived by Rodrigues' formula [9]

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}. \tag{11}$$

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

$$\begin{aligned} H_{n+1}(x) &= 2xH_n(x) - 2nH_{n-1}(x) \\ H_0(x) &= 1 \\ H_1(x) &= 2x \end{aligned} \tag{12}$$

By substituting *G(x)* for e^{-x^2} in (11), we generalize to Hermite polynomials with respect to the Gaussian function. Let these Hermite polynomials be denoted by $\bar{H}_n(x)$

$$\bar{H}_n(x) = (-1)^n G^{-1}(x) \frac{d^n}{dx^n}(G(x)) \tag{13}$$

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

$$\bar{H}_n(x) = \left(\frac{1}{2^{1/2}\sigma}\right)^n H_n\left(\frac{x}{2^{1/2}\sigma}\right) \tag{14}$$

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[9]:

$$\langle \bar{H}_m, \bar{H}_n \rangle = \sigma^{-2n} n! \delta_{mn}, \tag{15}$$

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

$$\bar{H}_{ijk}(x, y, t) = \bar{H}_i(x) \cdot \bar{H}_j(y) \cdot \bar{H}_k(t) \tag{16}$$

3.2 Derivation of Gradient Constraint Equations

One of the most important properties of Hermite polynomials is the property of Gaussian derivatives. It is with the aid of this property that we are able to establish gradient constraint equations. This property is manifested in the following theorem[14].

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

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

$$\text{Then } I_k = \langle I, \bar{H}_k \rangle = \langle I^{(k)}, \bar{H}_0 \rangle \text{ where } \bar{H}_0(x) = 1 \text{ and } I^{(k)} = \frac{d^k I}{dx^k}. \tag{17}$$

Recall our motion model,

$$I(x, y, t) = F((x - x_0)(1 - st) - pt + x_0, (y - y_0)(1 - st) - qt + y_0),$$

Expand both sides with Hermite polynomials,

$$\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{k=0}^{\infty} I_{ijk} \frac{\bar{H}_{ijk}}{\|\bar{H}_{ijk}\|^2} = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{k=0}^{\infty} F_{ijk} \frac{\bar{H}_{ijk}}{\|\bar{H}_{ijk}\|^2} \text{ then } I_{ijk} = \langle I, \bar{H}_{ijk} \rangle = F_{ijk} = \langle F, \bar{H}_{ijk} \rangle \tag{18}$$

Equating I_{ij1} to F_{ij1} and using Theorem 1, we derive (see Appendix A)

$$I_{ij1} \approx -uI_{(i+1)j0} - vI_{i(j+1)0} - (i+j)sI_{ij0} \text{ where } (u, v) \text{ are defined in (9).} \tag{19}$$

The extensibility of Hermite polynomial filters to higher order is clear from equation (19). It may be observed that orthogonality of the filters makes it numerically stable, Gaussian smoothing improves resistance to noise, and all the equations are linear. The theory indeed suggests local, simple and parallel computation, as presented in the next section.

4. Implementation

4.1 Algorithm

From (19), we derive six equations up to the third order. Within a 3-D local window, we estimate $\{I_{ijk}\}$ with the discrete approximation $\{\hat{I}_{ijk}(x, y, t)\}$, and write the equations in the matrix form

for obtaining a linear least square estimate:

$$E = min\|Af + b\|, \text{ where} \tag{20}$$

$$A = W\begin{bmatrix} \hat{I}_{100} & \hat{I}_{010} & 0 \\ \hat{I}_{200} & \hat{I}_{110} & \hat{I}_{100} \\ \hat{I}_{110} & \hat{I}_{020} & \hat{I}_{010} \\ \hat{I}_{300} & \hat{I}_{210} & 2\hat{I}_{200} \\ \hat{I}_{210} & \hat{I}_{120} & 2\hat{I}_{110} \\ \hat{I}_{120} & \hat{I}_{030} & 2\hat{I}_{020} \end{bmatrix}, f = \begin{bmatrix} u \\ v \\ s \end{bmatrix}, b = W\begin{bmatrix} \hat{I}_{001} \\ \hat{I}_{101} \\ \hat{I}_{011} \\ \hat{I}_{201} \\ \hat{I}_{111} \\ \hat{I}_{021} \end{bmatrix}, W = \begin{bmatrix} w_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_6 \end{bmatrix} \tag{21}$$

W is a weight matrix that is used to compensate for the different scales of equations due to different Hermite polynomial norms. We make the following selection:

$$w_1 = \|\bar{H}_{001}\|, w_2 = \|\bar{H}_{101}\|, w_3 = \|\bar{H}_{011}\|, w_4 = \|\bar{H}_{201}\|, w_5 = \|\bar{H}_{111}\|, w_6 = \|\bar{H}_{021}\| \tag{22}$$

For the center pixel of the window, we solve (20) using QR decomposition.

$$A = QR, \text{ and } E = min\|QRf + b\| = min\|Rf + Q^H b\|, \text{ where } Q \text{ is unitary.} \tag{23}$$

R can be denoted by $\begin{bmatrix} R_s \\ 0 \end{bmatrix}$, where R_s is an upper triangular matrix; and $Q^H b$ is $\begin{bmatrix} b_s \\ r \end{bmatrix}$, correspondingly. Equation (23) becomes

$$\begin{aligned} E &= min\,(\|R_s f + b_s\| + r) \\ &= r \qquad \text{if } R_s \text{ is not singular.} \end{aligned} \tag{24}$$

$$\text{The solution is computed from } R_s f + b_s = 0 \tag{25}$$

Note that the above process can run on all image pixels in a parallel fashion.

In the actual implementation, we use floating point computation; as a result, R_s is rarely singular. However, the behavior of R_s determines the accuracy of the solution. The residual r is also a good indicator of the reliability of the solution. We devote the next subsection to the discussion of the accuracy of the computed optical flow and associated confidence measures.

4.2 Confidence Measures

Our algorithm provides ample information about the behavior of the system equations. It includes the residual r, the condition number and the determinant of R_s. They can be shown [14] to signify certain image phenomena, e.g., occlusion, the aperture problem, etc., which present difficulties for optical flow computation. Therefore, they can be utilized to locate high error areas and suggest subsequent improvement methods. For the sake of evaluation in Section 6, we simply use them as confidence measures or threshold values to extract more accurate data. Following is how we integrate these different confidence measures.

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 these problems that no other measure does.

The condition number and the determinant together signify the relationship between numerical instability and the potential aperture problem. We suggest the multiplicative combination of these two, $det(R_s)/\kappa(R_s)$. In the conventional algorithm where there are only two flow parameters (u, v), this is equivalent to using $|\lambda|_{min}$ [14]. In our algorithm, $|\lambda|_{min}$ sometimes shows even stronger performance than either $det(R_s)$ or $1/\kappa(R_s)$ empirically. The use of $|\lambda|_{min}$ has been proposed by Girosi et al.[6] in a similar context and was used in Barron's implementation [2] of Lucas and Kanade's optical flow algorithm.

5. Previous Approaches And Our Contributions

In his approach to a generalized gradient method for optical flow, Srinivasan [19] demonstrated his algorithm on a simple translational motion model, and also on a simple rotation and expansion model, in which the focus of expansion (and rotation) is assumed to be known. The limitation of his algorithm is that it did not deal with translation plus expansion motion. In fact, "Erroneous results can occur if a translatory motion is superimposed upon the rotation or expansion"[19]. On

the contrary, our algorithm not only computes flow but also expansion at the same time.

Campani and Verri [4] recognized the limitations of the uniform flow field assumption in image neighborhood and the noise sensitivity of pointwise flow estimates. They extended the conventional motion with a weaker constraint, namely, flow generated by a rigid body motion is a piecewise linear vector field in a relatively large patch. It was formulated as, in their notation,

$$\mathbf{v} = \mathbf{v}_0 + \mathbf{M}(\mathbf{x} + \mathbf{x}_0) \quad \text{where } \mathbf{M}_{ij} = \frac{\partial v_i}{\partial x_j},\ i,j = 1,2. \tag{26}$$

Note that the linear optical flow field suffices to model elementary local image motions including translation, expansion, and rotation. They estimated flow in a patchwise manner with the first order gradient constraint equations plus the linear flow field constraint. The flow results were used to demonstrate qualitative motion analysis and applications.

Our work is initiated with the same motivation. Our new motion model also prescribes a piecewise linear flow field as can be seen from equation (7). We both use a relatively larger window to carry out flow estimation, and both algorithms compute divergence* in addition to flow.

However, there are fundamental differences in our approaches. Imposing constraints on the solution (flow) is a powerful engineering technique, but we are tempted to model the physical phenomena (motion) in a more comprehensive way and let the solution reflect the constraint by itself. Modeling a flow field is essentially a 2-D process, whereas modeling motion is a 3-D process, that is, motion over the time domain, which is relatively difficult, but there is a reward: we can impose temporal smoothing in an integrated theoretical framework based on Hermite polynomials. Moreover, a major algorithmic advantage of our approach is that we are still able to perform pointwise flow estimation†, with which our algorithm computes more accurate flow because more information is used per pixel.

---

*. In our algorithm, divergence can be computed from expansion s. Campani and Verri's algorithm is also capable of computing curl and deformation.

†. It is also feasible to implement our algorithm differently to perform patchwise flow estimation, whereas it is not possible for their algorithm to perform pointwise estimation.

Fleet and Jepson also attempted to cope with non-translational motion in [5]. They showed that the phase response, instead of the amplitude response, of the velocity-tuned filters is robust to image affine transformation or photometric deformation. Their algorithm is based on constant phase contours and tends to be more accurate but produces sparse flow field. On the contrary, we can often output dense results, which are very handy as far as motion segmentation is concerned.

Workhoven and Koenderink's [22] is our theoretical predecessor. They assumed an infinitesimal affine flow filed and derived similar motion equations. Again, we start with a 3-D motion model and contribute a numerically more feasible algorithm, as supported by experimental results.

6. Experiments

Based on the work of Barron, Fleet, and Beauchemin[2], we conducted extensive comparisons between our algorithm and other current optical flow algorithms, including those by Horn and Schunck[11], Lucas and Kanade[15], Uras et al. [20], Nagel[17], Anandan[1], Singh[18], Heeger[10], Waxman et al. [21], Fleet and Jepson[5]. The synthetic image sequences we used for comparison are Sinusoid, Translating tree, Diverging tree, and Yosemite fly-by. The real image sequences we used for demonstration are NASA and HMMWV. 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. Refer to [2] for more details. The error statistics and associated density for the comparison algorithms were obtained from Barron et al. [2].

6.1 Sinusoid

This is a synthetic image sequence (Fig 2) of a spatial sinusoidal wave traversing toward the upper right side. For our method we chose a window size large enough (17x17x7 for x,y,t) to prevent

*. Except HMMWV, which was provided by Gombash of Army Research Lab, and stabilized by Dr. Q. Zheng at the Center for Automation Research, University of Maryland.

aliasing. $|1/r|$ was used as the confidence measure in Table 1. Fig 3.1 shows the true optical flow, while Fig 3.2 shows the flow computed with our method. Our algorithm performs better than all of the other algorithms except Fleet and Jepson's.

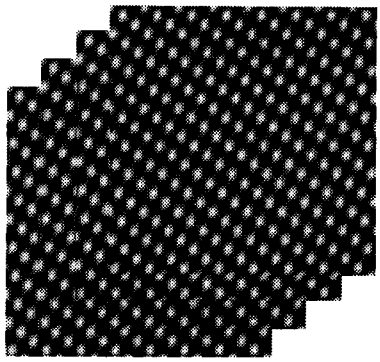

Fig 2. Traversing sinusoid

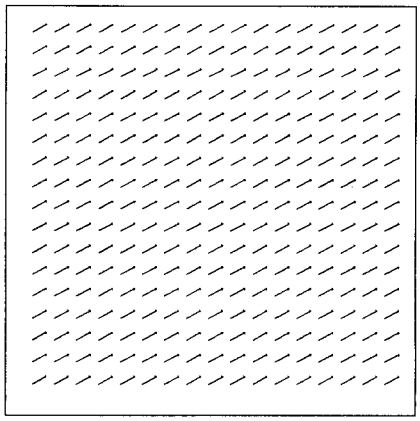

Fig 3.1True optical flow for sinusoid

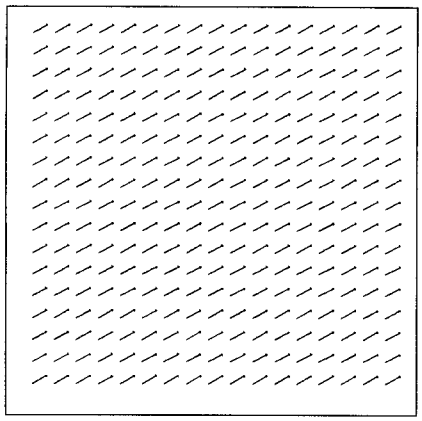

Fig 3.2 Computed optical flow (100%)

Table 1: Summary of Sinusoid Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
100%	0.63°	0.08°	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%	0.63°	0.08°	64.26°	26.14°	Waxman et al.

6.2 Translating and Diverging Tree

The translating and diverging tree sequences are two realistic synthetic sequences simulating the motion of simple translation (Fig 4.1) and expansion (Fig 4.2), respectively, of a poster. The window size used in our method is 19x19x11 for the translating tree and 17x17x9 for the diverging

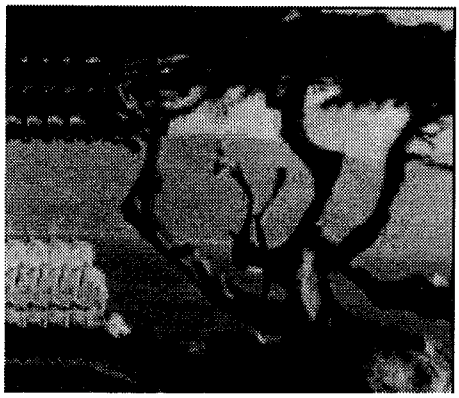

Fig 4.1 Translating tree

Fig 5.2 Diverging tree

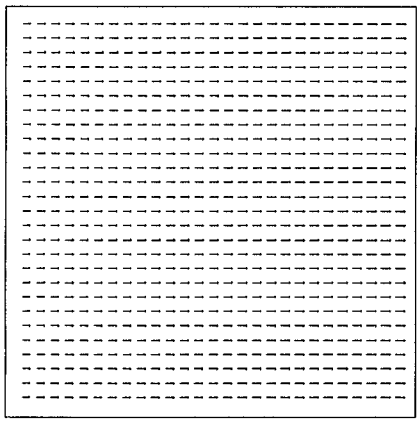

Fig 5.1 True flow for translating tree

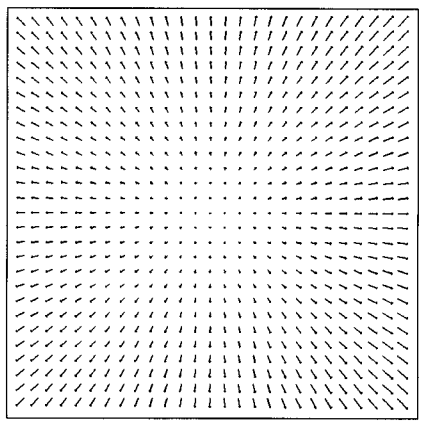

Fig 5.2 True flow for diverging tree

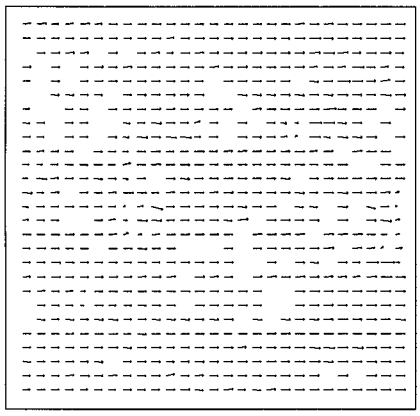

Fig 6.1 Computed flow for translating tree (90%)

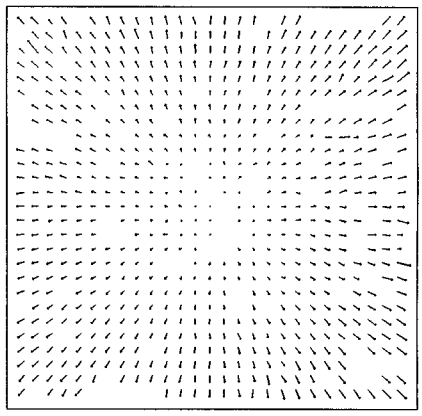

Fig 6.2 Computed flow for diverging tree (90%)

tree. Due to the lack of texture in some background areas, we used $|\lambda|_{min}$ as the confidence measure. Fig 5 and Fig 6 show the results. Only Uras's and Fleet and Jepson's algorithms perform better than ours for the translating tree sequence (Table 2, page 13). For the diverging tree sequence, our results are second only to Fleet and Jepson's (Table 3, page 13).

Table 2: Summary of Translating Tree Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
100%	0.92°	0.94°	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%	0.91°	0.92°	1.11°	0.89°	Singh (step 2)
74.5%	0.69°	0.51°	0.32°	0.38°	Fleet and Jepson
53-57%	0.59°	0.39°	32.66°	24.50°	Horn & Schunck (original)
			5.63°	2.78°	Heeger (level 1)
			1.89°	2.40°	Horn & Schunck (modified)
49.7%	0.57°	0.37°	0.23°	0.19°	Fleet and Jepson
44.2%	0.55°	0.34°	8.50°	13.50°	Heeger (level 0)
40-42%	0.53°	0.33°	0.46°	0.35°	Uras et al.
			0.72°	0.75°	Singh (step 1)
			0.66°	0.67°	Lucas and Kanade
26.8%	0.48°	0.28°	0.25°	0.21°	Fleet and Jepson
13.1%	0.42°	0.24°	0.56°	0.58°	Lucas and Kanade
1.9%	0.35°	0.19°	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%	1.84°	1.33°	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%	1.82°	1.28°	8.40°	4.78°	Singh (step 2)

Table 3: Summary of Diverging Tree Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
73.8%	1.59°	1.12°	4.95°	3.09°	Heeger (combined)
60-61%	1.49°	1.02°	0.99°	0.78°	Fleet and Jepson
			8.93°	7.79°	Horn & Schunck (original)
			3.83°	2.19°	Uras et al.
46-48%	1.40°	0.92°	2.50°	3.89°	Horn & Schunck (modified)
			0.80°	0.73°	Fleet and Jepson
			1.94°	2.06°	Lucas and Kanade
28.2%	1.28°	0.79°	0.73°	0.46°	Fleet and Jepson
24.3%	1.24°	0.77°	1.65°	1.48°	Lucas and Kanade
3.9-4.9%	1.09°	0.66°	13.69°	11.83°	Waxman et al.
			5.62°	6.16°	Singh (step 1)

6.3 Yosemite Fly-by

The Yosemite Fly-by sequence is a realistic synthetic image sequence (Fig 7). The flight scene is simulated from actual aerial photos and digital-terrain maps, and artificial sky and clouds. Since the clouds in the sky change brightness over time, it poses difficulties for all algorithms. 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, page 14. Again, the clouds account for the large magnitude error. Our algorithm is the best. From this we believe our algorithm should work well with real images, as shown later.

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%	9.06°	13.23°	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%	8.46°	12.43°	12.9°	11.57°	Singh (step 2)
64.2%	3.70°	5.84°	20.89°	34.26°	Heeger (level 0)

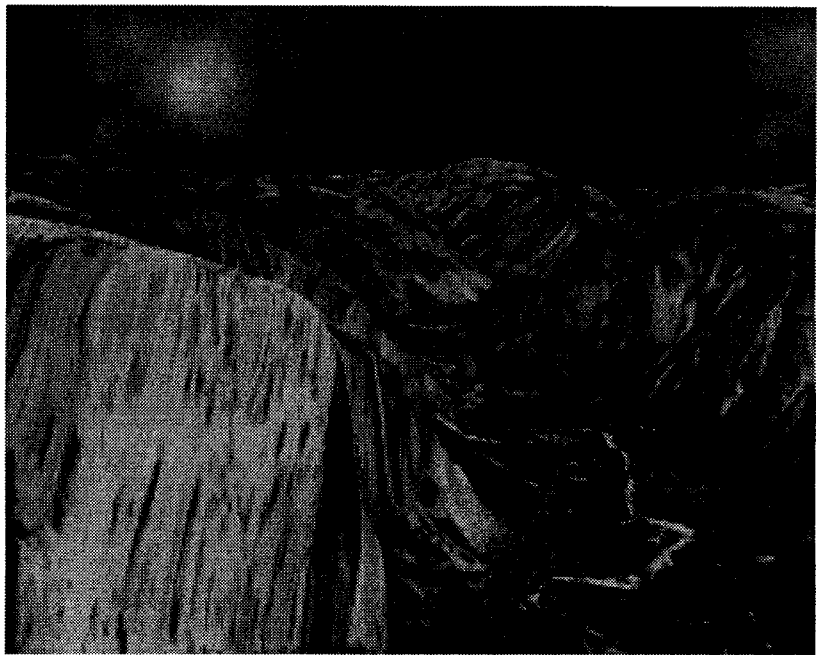

Fig 7. Yosemite fly-by image

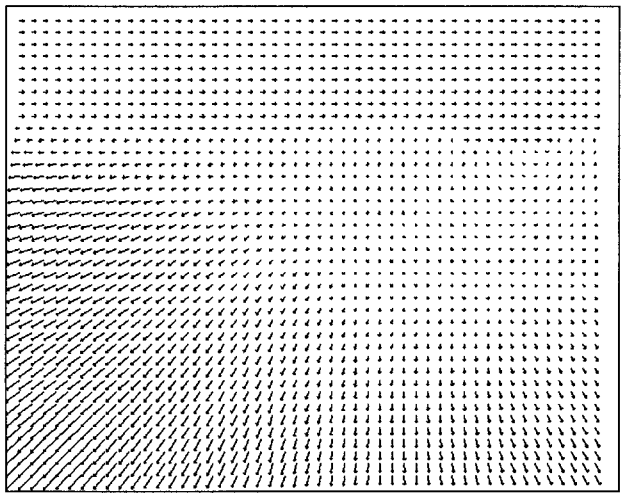

Fig 8.1 True optical flow field for Yosemite fly-by

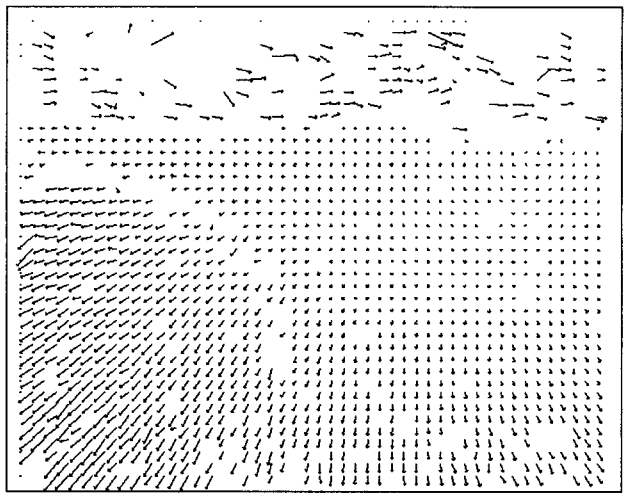

Fig 8.2 Computed optical flow for Yosemite fly-by (75%)

Table 4: Summary of Yosemite Fly-by Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
59.6%	3.42°	5.36°	25.41°	28.14°	Horn & Schunck (original)
44.8%	2.91°	4.44°	11.74°	19.04°	Heeger (combined)
33-35%	2.66°	4.04°	4.10°	9.58°	Lucas and Kanade
			4.29°	11.24°	Fleet and Jepson
			5.48°	10.41°	Horn & Schunck (modified)
30.6%	2.62°	3.96°	4.95°	12.39°	Fleet and Jepson

Table 4: Summary of Yosemite Fly-by Error Statistics

Density	Our Algorithm		Other Algorithm		
	Average Error	Standard Deviation	Average Error	Standard Deviation	Technique by
15%	2.45°	3.83°	10.51°	12.11°	Heeger (level 1)
			6.74°	16.01°	Uras et al.
8.7%	2.44°	3.95°	3.05°	7.31°	Lucas and Kanade
7.4%	2.40°	3.86°	20.32°	20.60°	Waxman et al.
2.2-2.4%	2.37°	4.00°	11.51°	11.83°	Heeger (level 2)
			16.29°	25.70°	Singh (step 1)

6.4 Comparison Summary

We also compare the results with those of the Hermite polynomials algorithm based on translation motion. As expected, for dense output, it shows an 11%~37% drop* in average angle error for diverging images (Diverging trees, Yosemite) while showing no significant improvement on translating images. Fleet and Jepson's algorithm is also very competitive. But it does not always have the capability of producing 100% density of output. Furthermore, our relatively small standard deviation reflects noise insensitivity, which can be attributed to using 3-D Hermite polynomials. To conclude, our algorithm performs well over a wide set of images, especially diverging images, it offers flexibility in terms of output density, and it is relatively less sensitive to noise.

6.5 Real Images Demonstration

Current optical flow algorithms often have difficulty with real image sequences. The reasons include camera jitter, nonrigidity of objects, and brightness changes. We demonstrate our algorithm with two diverging real images, Fig 10 and Fig 9. The nature of coherent or "clean" flow field is evident. The optical flow output of these images has undergone thresholding based on two confidence measures, $|1/r|$ and $|\lambda|_{min}$.

*. For other than 100% density data, the drop in error reaches 57%. This is due to better a confidence measure.

Fig 9.1 HMMWV sequence

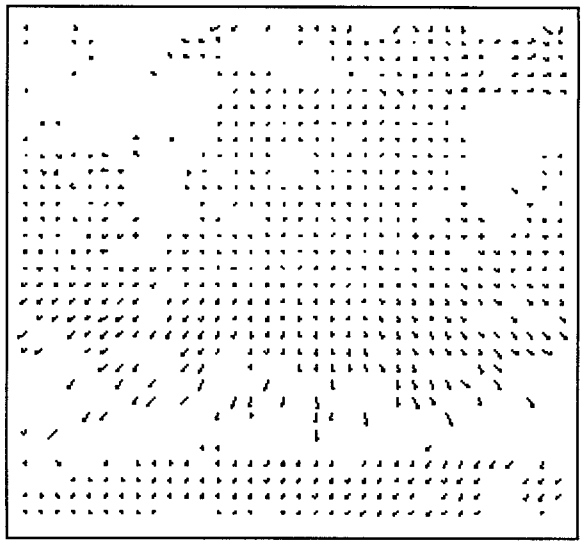

Fig 9.2 HMMWV flow field (64%)

Fig 10.1 NASA sequence

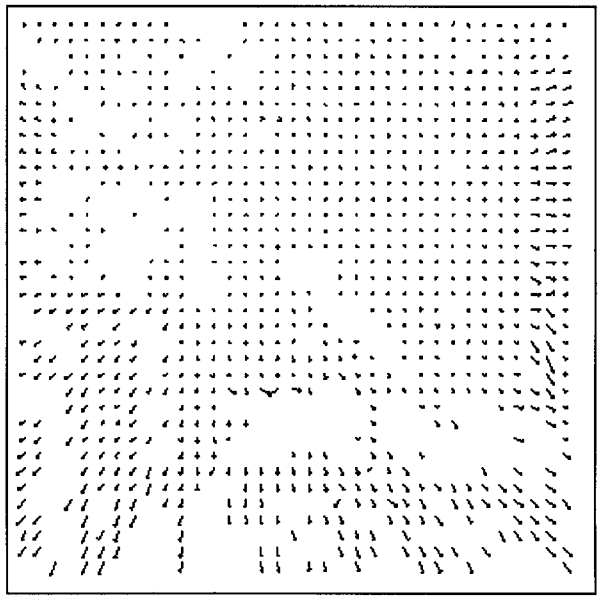

Fig 10.2 NASA flow field (75%)

7. Conclusion

The optical flow problem is difficult and sometimes ill-posed; the simple translational motion model is problematic; and gradient based methods often suffer from noise sensitivity in numerical differentiation. We tackle these problems with a combined effort of a generalized motion model

and Hermite polynomial filters. The new model enables us to estimate more reliable flow, but also demands higher order differentiation. Hermite polynomial theory comes to the rescue with its many elegant properties, including orthogonality, extensibility, Gaussian smoothing, etc. Contrary to general belief, the behaviors of these high order (up to 3) differentiation filters are quite insensitive to noise. This is justified by the excellent performance of the flow results. Simplicity adds yet another dimension to the strength of this algorithm—making real-time implementation possible. We realize that a general 3-D motion including rotation is yet to be modeled so that optical flow can be estimated even more accurately. We hope that this work prompts more efforts in this direction.

Acknowledgment

We would like to thank J.L. Barron for providing the images and comparison results on which our experiments are based. We would also like to thank Dr. Q. Zheng for stabilizing the HMMWV sequence.

Appendix. A

We prove equation

$$I_{ij1} \approx -uI_{(i+1)j0} - vI_{i(j+1)0} - (i+j)\,sI_{ij0}.$$

First, from (12) and (14), we establish

$$x\overline{H}_n(x) = \sigma^2\overline{H}_{n+1}(x) + n\overline{H}_{n-1}(x). \tag{27}$$

Equating I_{ij1} to F_{ij1} and using Theorem 1,

$$\begin{aligned} I_{ij1} = F_{ij1} &= \langle F, \overline{H}_{ij1}\rangle = \langle \frac{\partial F}{\partial t}, \overline{H}_{ij0}\rangle \\ &= -\langle \left(\frac{s(x-x_0)+p}{1-st}\right)\frac{\partial I}{\partial x} + \left(\frac{s(y-y_0)+q}{1-st}\right)\frac{\partial I}{\partial y}, \overline{H}_{ij0}\rangle \end{aligned} \tag{28}$$

Practically, $s \ll 1$ so $1 - st \approx 1$. Equation (28) can be approximated by

$$-\langle (s(x-x_0)+p)\frac{\partial I}{\partial x} + (s(y-y_0)+q)\frac{\partial I}{\partial y}, \overline{H}_{ij0}\rangle \text{ or } -\langle (sx+u)\frac{\partial I}{\partial x} + (sy+v)\frac{\partial I}{\partial y}, \overline{H}_{ij0}\rangle \tag{29}$$

Using Theorem 1 again, we derive

$$-uI_{(i+1)j0} - vI_{i(j+1)0} - s\langle\frac{\partial I}{\partial x}, x\bar{H}_{ij0}\rangle - s\langle\frac{\partial I}{\partial y}, y\bar{H}_{ij0}\rangle. \tag{30}$$

Equation (30) and (27) yield

$$-uI_{(i+1)j0} - vI_{i(j+1)0} - (i+j)\,sI_{ij0} - s\sigma^2 (I_{(i+2)j0} + I_{i(j+2)0}) \tag{31}$$

The last term in (31) involves higher order differentiation, which often suffers from quantization error due to limited filter support. Furthermore, it is very small in smooth images. We choose to ignore it in practice. Hence, we have proved equation (19).

Reference

[1] Anandan, P., "Measuring Visual Motion from Image Sequences", Ph.D. Thesis, COINS TR 87-21, University of Massachusetts, Amherst MA, 1987.

[2] 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.

[3] 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.

[4] Campani, M. and Verri, A., "Motion Analysis from First-Order Properties of Optical Flow", Computer Vision, Graphics, and Image Processing: Image Understanding, vol. 50, No. 1, pp. 90-107, July 1992.

[5] 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.

[6] Girosi, F., Verri, A., Torre, V., "Constraints for the Computation of Optical Flow", Proceedings of IEEE Workshop on Visual Motion, Irvine, CA, pp. 116-124, 1989.

[7] 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.

[8] 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.

[9] 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.

[10] Heeger, D. J., "Optical Flow Using Spatiotemporal Filters", International Journal Of Computer Vision, vol. 1, pp. 279-302, 1988.

[11] Horn, B. K. P. and Schunck, B. G., "Determining Optical Flow", Artificial Intelligence, vol. 17, pp. 185-204, 1981.

[12] 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.

[13] Koenderink, J.J. and Van Doorn, A.J., "Representation of Local Geometry in The Visual System", Biological Cybernetics, vol. 55, pp. 367-375, 1987.

[14] Liu, H., Hong, T., Herman, M., Chellappa, R., "A Reliable Optical Flow Algorithm Using 3-D Hermite Polynomials", NIST-5333, December, 1993.

[15] 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.

[16] 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.

[17] 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.

[18] Singh, A., "An Estimation-Theoretic Framework for Image-Flow Computation", Proceedings of International Conference on Computer Vision, pp. 169-177, 1990.

[19] Srinivasan, M.V., "Generalized Gradient Schemes for the Measurement of Two-Dimensional Image Motion", Biological Cybernetics, vol. 63, pp. 421-431, 1990.

[20] Uras S., Girosi F., Verri A., Torre V. "A Computational Approach to Motion Perception", Biological Cybernetics, vol. 60, pp. 79-97, 1988.

[21] 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.

[22] Werkhoven, P., Koenderink, J.J., "Extraction of Motion Parallax Structure in the Visual System I", Biological Cybernetics, vol. 63, pp. 185-191, 1990.

[23] 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.