IMAGE FLOW ESTIMATION USING FACET MODEL AND COVARIANCE PROPAGATION

MING YE and ROBERT M. HARALICK

Department of Electrical Engineering
University of Washington, Seattle, WA 98195, USA
E-mail: {*ming,haralick*}*@george.ee.washington.edu*

The accuracy of existing methods for gradient-based image flow estimation suffers from low-quality derivative estimation, lack of systematic error analysis, and heuristic selection.

In this paper we present an image flow estimator using the facet model and covariance propagation. The facet model provides high-quality derivatives, image noise variance estimates, and the effect of prefiltering at the same time. We propagate the covariance of the image data to the image flow vector, yielding a covariance matrix with each vector. From the raw flow field, a χ^2 test selects the estimates statistically significant from zero. This selection scheme successfully suppresses false alarms and bad estimates with a low misdetection rate. We further incorporate a flow field smoothness constraint to achieve higher motion field consistency. Experiments on both synthetic and real data, and comparison with other techniques show that at a 10% misdetection rate, our approach has an average error vector magnitude that is 25% less than that of Lucas and Kanade and a false alarm rate less than half of theirs.

Keywords: Facet model, covariance propagation, image flow, optic flow, hypothesis testing, error analysis.

1. INTRODUCTION

Optic flow is the projected 2D velocity arising from a 3D surface point of an object in motion relative to the camera or the observer. Image flow is the optic flow that can be estimated from image sequences. Various applications, such as motion detection, egomotion analysis, 3D motion inference, scene segmentation, and object structure or camera parameter analysis, require image flow as a fundamental measure. This makes accurate and efficient image flow estimation a critical issue.

Existing image flow estimation methods can be roughly divided into three categories, matching-based, frequency/phase-based and gradient-based methods.

Matching-based methods include feature-based and region-based methods. Feature-based methods locate and trace identifiable features of the image over time. They are robust to large motion and brightness variation. The problems with the methods are that the image flow field is usually very sparse and that the feature extraction and matching are difficult. Region-based methods locate the most similar region to the one under examination and calculate the flow vector from their displacement. These kinds of methods have been widely used in video coding. However its flow fields are still not dense enough. Both of these two subcategories have difficulty in measuring subpixel motion and the computation is too expensive. In terms of flow field accuracy and density, matching-based methods are not comparable with frequency-based or gradient-based methods.

Vision Interface — Real World Applications of Computer Vision (1999) 209–241

Frequency/phase-based methods use frequency/phase information from the output of velocity tuned filters. Although high accuracy on raw flow field estimates might be achieved,[1,11] they usually involve very intensive computation and meet with difficulties in selecting reliable estimates. The technique developed by Fleet and Jepson,[1] and that by Xiong and Shafer[11] are two of the representative ones with good performance reported.

Gradient-based methods compute 2D velocities from spatiotemporal intensity derivatives of image sequences. Because of their simple computation and favorable experimental results, gradient-based methods have received the most extensive study. A brief overview of gradient-based methods is given in Sec. 2.

There are four common steps of processing to gradient-based methods:

(i) prefiltering or smoothing image sequences to enhance the signal-to-noise-ratio;
(ii) estimating spatiotemporal derivatives;
(iii) organizing derivative information to form constraints and solving the constraint equations for the image flow vector;
(iv) selecting nonzero raw image flow estimates based on some kind of error analysis.

Much attention has been paid to Step (iii), developing the image flow estimation model. However, Steps (i), (ii) and (iv), the pre- and post-processings, to a great extent affect the overall performance of the algorithm.

Although quite a few existing gradient-based methods provide reasonable raw estimates, further improvement of accuracy is hindered by the following problems. Prefiltering, if any, involves additional work and has tuning parameters. Most methods use the neighborhood difference as the derivative estimate. This is highly sensitive to noise. Quantitative error analysis is seldom carried out. Without a reliable confidence measure, selecting good estimates is usually heuristic.

In this paper, we develop an image flow estimator involving local linear constraint equations, the facet model, covariance propagation and selection based on a statistical hypothesis testing.

The image flow constraint equations we use belong to the local optimization gradient-based category and involve both first- and second-order derivatives. They lay a sound base for the rest of the processing by providing good raw estimates.

The derivatives in the image flow constraint equations are estimated from the 3D cubic facet model. We assume that the unobserved noiseless intensity pattern in each small 3D neighborhood is a trivariate cubic polynomial in the (r.c.t) spatiotemporal domain and that the observation results from perturbing the true pattern by a small additive iid., Gaussian noise with zero mean and variance σ^2. We estimate the facet model parameters first and calculate the image flow vector from them. Compared with popular neighborhood difference approaches, the facet model approach achieves better accuracy and robustness because it uses the entire neighborhood gray-level information. The facet model automatically prefilters the neighborhood data avoiding any extra work or hand-tuned parameters used in other gradient-based techniques. It also provides neighborhood noise variance estimates, which can be used in the quantitative error analysis.

We assume the system is approximately linear for small perturbations. Under this assumption, the covariance matrix of the image flow estimate is a good measure of the error size. We partition the system into facet model fitting and image flow constraint solving and propagate the covariance through them one by one, finally obtaining a covariance matrix with each image flow estimate.

Under the assumption of Gaussian perturbations, we derive a χ^2 statistic as the confidence measure and conduct a hypothesis testing on it to select reliable estimates in a statistical sense. Experimental results show that the selection suppresses false alarms and bad estimates while preserving high accuracy and detection rates.

In the second phase, we incorporate a flow field smoothness constraint into our algorithm. Instead of estimating the image flow vector at each pixel independently from its neighboring vectors, we assume the flow vectors in each neighborhood are constant and solve a group of image flow constraint equations simultaneously. The flow field smoothness constraint dramatically improves the motion field consistency.

We show the advantages of our technique from experiments on both synthetic and real data. Comparisons with popular techniques, particularly the modified technique of Lucas and Kanade, the one having the best overall performance as reported by Barron,[2] are given as well.

At a 10% misdetection rate of nonzero image flow estimates, our approach yields an average error vector magnitude that is 25% less than that of Lucas and Kanade and a false alarm rate well less than half of theirs.

This paper is organized as follows. Section 2 gives a brief overview of typical gradient-based methods. Section 3 describes the Image Flow Constraint Equations we use. Section 4 introduces how we estimate derivatives from the cubic facet model. In Sec. 5 we do the covariance propagation through the image flow estimation system yielding a covariance matrix associated with each estimate. In Sec. 6 we use a χ^2 test to reliably select nonzero image flow estimates. The algorithm with the flow field smoothness constraint is developed in Sec. 7. Experimental results and analysis, and comparison with other techniques are presented in Sec. 8. Finally to make this paper more self-contained, Appendix A and B provide some related background knowledge on Discrete Orthonormal Polynomials and covariance propagation.

2. GRADIENT-BASED ESTIMATION METHODS

In this section we briefly describe the well-known Optic Flow Constraint Equation, which is the common base for all gradient-based methods. Then we divide gradient-based methods into three subcategories and give an introduction to each of them.

2.1. Optic Flow Constraint Equation

We represent the intensity of an image sequence by $I(x, y, t)$, where x, y, t are respectively the row, column and frame (time) indices. The following assumptions are made to derive the Optic Flow Constraint Equation: (i) the motion of objects is constant translation; (ii) the intensities of matching points on different frames are the same; (iii) $I(x, y, t)$ is differentiable in the spatiotemporal domain.

Consider a point moving from (x, y) to $(x+\Delta x, y+\Delta y)$ during the time interval $(t, t+\Delta t)$. Expanding its intensity at time $t+\Delta t$ about point (x, y, t) as Taylor series and omitting the terms higher than the first-order yields

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x, y, t) + \Delta x\frac{\partial I}{\partial x} + \Delta y\frac{\partial I}{\partial y} + \Delta t\frac{\partial I}{\partial t}.$$

Because

$$I(x, y, t) = I(x+\Delta x, y+\Delta y, t+\Delta t)$$

(Assumption ii), the above equation reduces to

$$\Delta x\frac{\partial I}{\partial x} + \Delta y\frac{\partial I}{\partial y} + \Delta t\frac{\partial I}{\partial t} = 0.$$

When $\Delta t \to 0$, it becomes

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0.$$

$(\frac{dx}{dt}, \frac{dy}{dt})$ is the image flow vector (u, v). Abbreviating $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\frac{\partial I}{\partial t}$ by I_x, I_y, I_t respectively, the well-known Optic Flow Constraint Equation (OFCE)

$$I_x u + I_y v + I_t = 0 \tag{1}$$

is obtained.

2.2. Subcategories

The two unknowns u and v cannot be solved for from the single equation OFCE. At least one more constraint is needed. According to different ways of organizing the constraints, gradient-based approaches are further divided into global optimization, local optimization and local parametric methods. First-order derivatives, second-order derivatives or both of them may be used in the constraints.

2.2.1. Global optimization

Global optimization techniques minimize an error function based upon the OFCE [Eq. (1)] and an assumption of local smoothness of image flow variations over the entire image sequence. A classic technique of this category is developed by Horn and Schunck.[7] They minimize an error over the entire image sequence

$$\varepsilon^2 = \iint (\alpha^2\varepsilon_c^2 + \varepsilon_b^2)dxdy$$

where

$$\varepsilon_b^2 = I_x u + I_y v + I_t$$

is actually the OFCE,

$$\varepsilon_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2$$

is the image flow field smoothness constraint, and α is the relative weight of these two factors.

The algorithm is implemented by iteration. The experimental results are reasonably good. The flow field density is high because of the global smoothness constraint. However it tends to oversmooth the flow field, does not perform well at boundaries and thus produces a lot of false alarms. It also meets with difficulties in choosing the weight α and a confidence measure to select reliable estimates.

2.2.2. Local optimization

Local optimization methods determine the image flow vector at each pixel independently from its neighboring flow field. They solve a group of linear constraint equations whose coefficients are derivatives evaluated at that pixel. Since estimates are not influenced by their neighbors, at the pixels for which the assumptions hold, the estimation accuracy is high, and motion boundaries and occlusions are preserved better. However most moving objects are large enough such that flow fields are usually smooth. Not taking this factor into account might hurt the robustness of the estimation.

The technique developed by Uras *et al.*[1] is a representative of this type of method. Uras assumes the first-order gradient is conserved for a moving point and solves for the image flow vector using two second-order constraints

$$I_{xx}u + I_{xy}v + I_{xt} = 0$$

$$I_{yx}u + I_{yy}v + I_{yt} = 0$$

when the Hessian H of $I(x, y, t)$ is nonsingular. For robustness, they divide the image into 8 × 8 regions. For each region they select the 8 estimates that best satisfy the constraint $\|(\Delta V)^T(\Delta I_x, \Delta I_y)^T\| \ll \|\Delta I_t\|$. Of these they choose the most reliable estimate as the velocity for the entire 8 × 8 region. In Barron's implementation, the determinant $\det(H)$ of the Hessian is used as the reliability measure.

This technique is reported to have competitive performance. However the restriction of constant motion in each 8 × 8 region makes the flow field blocky and limits the accuracy. Also the confidence measure $\det(H)$ turns out not to work very well.[1]

2.2.3. Local parametric methods

Local parametric approaches assume the image flow field in a small neighborhood is smooth so that (u, v) can be fit to a local model such as a low-order polynomial. In each neighborhood they first estimate the parameters of the model and then determine the image flow vectors from the parameters. These approaches can be looked on as a compromise of global and local optimization methods. They improve the robustness over local optimization methods, while they do not smear the motion field as much as global techniques do.

A simple local parametric scheme assumes the velocity is constant in each neighborhood. The technique developed by Lucas and Kanade[4] uses this scheme. It solves for the constant velocity in each small neighborhood Ω by minimizing

$$\sum_{\Omega} W^2(x,y)[I_x u + I_y v + I_t]^2$$

where $W(x,y)$ denotes a weight function that shows more influence of the center of the neighborhood. The solution to the minimization problem is given by

$$A^T W^2 A V = A^T W^2 b$$

where for n points $p_i \in \Omega$ at a single time t,

$$A = \begin{pmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_n) & I_y(p_n) \end{pmatrix} \qquad b = -\begin{pmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_n) \end{pmatrix}$$

and $W = \mathrm{diag}(W(p_1), \ldots, W(p_n))$. When $A^T W^2 A$ is nonsingular, V has a closed form solution

$$V = (A^T W^2 A)^{-1} A^T W^2 b\,.$$

Simoncelli *et al.*[10] arrive at a similar result through Maximum Likelihood Estimation. They model the OFCE [Eq. (1)] using Gaussian distribution errors on gradient measurements and a Gaussian prior distribution on velocity V, and derive the mean and covariance of the image flow vector by a Bayesian approach. Although the accuracy of the raw estimate does not improve much, they provide a covariance matrix with each estimate as the error indicator. Barron further suggests that the smaller eigenvalue of $A^T W^2 A$ can be used as the confidence measure to select the good estimates.

This technique is reported to be the most efficient one among nine popular techniques.[1] However as we will show in Sec. 8, its confidence measure and selection scheme are still not reliable. One of the major reasons is, they *empirically* choose the three variances of the assumed Gaussian variables instead of determining them by propagating the error from the image sequence.

3. IMAGE FLOW CONSTRAINT EQUATIONS

The image flow constraint equations we use were first described by Haralick and Lee.[4] This section illustrates how the estimation scheme works (Fig. 1) and explains the derivation of the constraint equations.

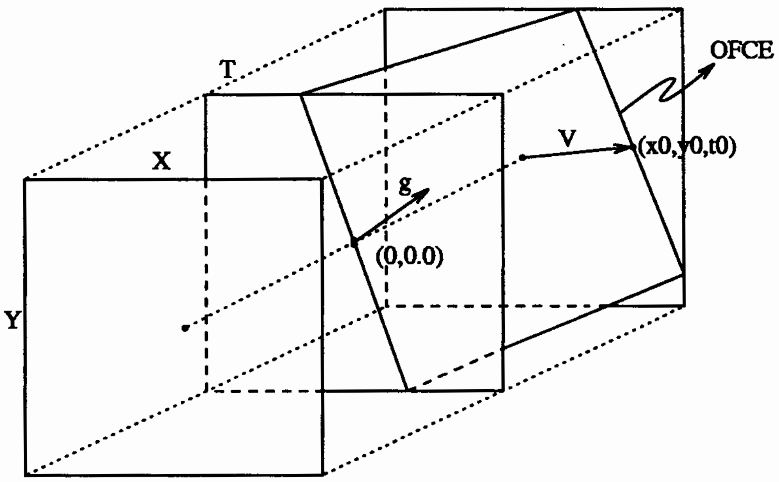


Fig. 1. Interpretation of image flow constraint equations.

Let us consider a point moving with 2D velocity $V = (u, v)'$ in an image sequence. Suppose the point is at the origin $(0, 0, 0)$ in the 3D space of row x, column y and time t. After a time interval t_0, the point arrives at the location (x_0, y_0, t_0). We define a neighborhood N of size $X \times Y \times T$ centered at $(0, 0, 0)$ such that $(x_0, y_0, t_0) \in N$. Once we have located the point (x_0, y_0, t_0), the image flow vector is readily available as $V = (x_0/t_0, y_0/t_0)'$.

We assume:

(i) the underlying 3D intensity value of the neighborhood N is a continuous function $I(x, y, t)$;

(ii) a point retains its intensity value during motion, i.e.

$$I(x_0, y_0, t_0) = I(0, 0, 0); \tag{2}$$

(iii) the motion is translational without acceleration;

(iv) there is sufficient intensity variation along the direction of motion (no aperture problem).

We first construct an isocontour plane at the origin

$$xI_x + yI_y + tI_t = 0\,.$$

The plane is isocontour in the sense that it is orthogonal to $g = (I_x, I_y, I_t)$, the gradient vector of $I(x, y, t)$ at $(0, 0, 0)$. Intersecting the plane with the frame at t_0 and dividing both sides by t_0 yields

$$\frac{x}{t_0}I_x + \frac{y}{t_0}I_y + I_t = 0\,.$$

Noting that $(\frac{x_0}{t_0}, \frac{y_0}{t_0}) = (u, v)$, this equation is the well-known Optic Flow Constraint Equation (OFCE). Thus our estimation scheme actually interprets the OFCE as the intersection line of the isocontour plane with an image frame in the neighborhood. Finally the point (x_0, y_0, t_0) is located on the intersection line as the one of the same intensity value as the point $(0,0,0)$.

Now let us quickly go over the derivation of the constraint equations from the above estimation scheme.

Expanding both sides of Eq. (2) as Taylor series around $(0,0,0)$ and neglecting the third- and higher-order terms yields

$$I(x,y,t) = I(0,0,0)+xI_x+yI_y+tI_t+\frac{x^2}{2}I_{xx}+xyI_{xy}+\frac{y^2}{2}I_{yy}+xtI_{xt}+ytI_{yt}+\frac{t^2}{2}I_{tt}\,. \tag{3}$$

Matching of spatial intensity patterns around the two corresponding points results in

$$I_x(x,y,t) = I_x(0,0,0) = I_x$$
$$I_y(x,y,t) = I_y(0,0,0) = I_y\,.$$

Since the motion is uniform with no acceleration, the temporal derivative of the intensity must match. It yields

$$I_t(x,y,t) = I_t(0,0,0) = I_t\,.$$

Applying these constraints to Eq. (3) we have

$$xI_x + yI_y + tI_t + \frac{x^2}{2}I_{xx} + xyI_{xy} + \frac{y^2}{2}I_{yy} + xtI_{xt} + ytI_{yt} + \frac{t^2}{2}I_{tt} = 0 \tag{4}$$
$$xI_{xx} + yI_{xy} + tI_{xt} = 0 \tag{5}$$
$$xI_{yx} + yI_{yy} + tI_{yt} = 0 \tag{6}$$
$$xI_{tx} + yI_{ty} + tI_{tt} = 0\,. \tag{7}$$

Multiplying Eqs. (5)–(7) by x, y, t respectively and adding them together produces

$$x^2I_{xx} + 2xyI_{xy} + c^2I_{yy} + 2xtI_{xt} + 2ytI_{yt} + t^2I_{tt} = 0\,.$$

Substituting the above equation back into Eq. (4) we obtain

$$xI_x + yI_y + tI_t = 0,$$

the OFCE. Thus, the technique of using Eqs. (1) and (2) in essence works because it assumes that all first partials match. Combining Eqs. (1), (5)–(7) yields the image flow constraint equations we use.

Solving the over-constraint equations is equivalent to find the $V = (u,v)'$ minimizing

$$\|AV - b\|^2 \tag{8}$$

where

$$A = \begin{pmatrix} I_x & I_y \\ I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \\ I_{tx} & I_{ty} \end{pmatrix} \quad b = -\begin{pmatrix} I_t \\ I_{xt} \\ I_{yt} \\ I_{tt} \end{pmatrix}.$$

Once the partial derivatives are available, (u, v) can be solved for by standard least-square procedures. If $A'A$ is nonsingular, we can write the explicit solution in the least-square sense as

$$V = (A'A)^{-1}(A'b). \tag{9}$$

Before evaluating the matrix inverse $(A'A)^{-1}$, we first examine the condition of $A'A$ by looking at its determinant. If the determinant is smaller than a certain threshold, $A'A$ is close to singular. In such cases, the estimation becomes extremely unstable and we directly set $V = 0$. We set the threshold to 10^{-5} in our experiments.

4. FACET MODEL AND DERIVATIVE ESTIMATES

The coefficients of the Image Flow Constraint Equations [Eq. (8)], which are derivatives of the neighborhood data evaluated at the origin, should be determined before the equations are solved. The quality of the derivatives greatly affects the the estimation accuracy. We extract the derivatives from the 3D cubic facet model. The facet model yields high-quality derivatives, prefilters the image data, and provides image noise variance estimates simultaneously. This section discusses how derivatives are obtained from the facet model and the advantages of this approach.

4.1. Facet Model Concepts

The facet model principle states that the image can be thought of as an underlying piecewise continuous gray-level intensity surface. The observed digital image is a noisy, discrete sampling of a distorted version of this surface. A small neighborhood of any pixel is completely characterized by two models, one describing what the general form of the surface would be in the neighborhood if there were no noise, and the other describing what the noise and distortion do to the assumed form. Once the two models are well defined, processing can be applied to the noise-free model, and the impact of the noise is examined quantitatively.[5]

Usually the facet model is defined on an image, a 2D domain. In our case it is generalized to the image sequence which is a 3D domain.

Low-order polynomials are the most commonly used general forms for the facet model. We use a 3D cubic polynomial in our algorithm. "Cubic" means the highest-order of any term is 3.

To ensure that the center of the neighborhood lies at the origin, we require the neighborhood sizes to be odd numbers. As the smallest neighborhood size for an 1D cubic polynomial is 5, the 3D neighborhood at least has the size of 5 × 5 × 5.

4.2. Canonical Cubic Facet Model

We define a noise-free facet model in each 3D neighborhood N by a canonical cubic polynomial of row, column and time index x, y, t

$$\begin{aligned} I(x,y,t) = {} & a_1 + a_2x + a_3y + a_4t + a_5x^2 + a_6xy + a_7y^2 + a_8yt \\ & + a_9t^2 + a_{10}xt + a_{11}x^3 + a_{12}x^2y + a_{13}xy^2 + a_{14}y^3 + a_{15}y^2t \\ & + a_{16}yt^2 + a_{17}t^3 + a_{18}x^2t + a_{19}xt^2 + a_{20}xyt, \quad (x,y,t) \in N. \end{aligned} \tag{10}$$

We assume the noise in each neighborhood an iid., additive Gaussian variable with zero mean and small variance σ^2.

The coefficient vector a minimizes

$$\|Da - J\|^2 \tag{11}$$

where

$$D = \begin{pmatrix} 1 & x_1 & y_1 & t_1 & \dots & x_1y_1t_1 \\ 1 & x_1 & y_2 & t_1 & \dots & x_1y_2t_1 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_1 & y_1 & t_2 & \dots & x_1y_1t_2 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_2 & y_1 & t_1 & \dots & x_2y_2t_1 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_X & y_Y & t_T & \dots & x_Xy_Yt_T \end{pmatrix} \quad J = \begin{pmatrix} J_1 \\ J_2 \\ \vdots \\ J_N \end{pmatrix}.$$

D is the design matrix of the canonical cubic facet fitting problem, with each row the twenty bases evaluated at the point (x_i, y_j, t_k). J_n is the intensity value at this point.

Since $D'D$ is nonsingular, the Least-Square solution to a is

$$a = (D'D)^{-1}D'J\,,$$

which is just a linear transform of the neighborhood data J.

With the assumption of equal-variance Gaussian noise, the optimization problem in Eq. (11) can be interpreted as minimizing the chi-square statistic

$$\left\|\frac{Da - J}{\sigma}\right\|^2$$

with $N - 20$ degrees of freedom. As a constant, the noise standard deviation σ has no impact on the minimization problem and thus can be left out when solving for a.

The mean of a chi-square statistic is equal to its degrees of freedom. Here

$$E\left\{\left\|\frac{Da - J}{\sigma}\right\|^2\right\} = N - 20\,.$$

This makes

$$\hat{\sigma}^2 = \frac{1}{N - 20}\| Da - J\|^2$$

an unbiased estimate of the true neighborhood noise variance σ^2.

4.3. DOP Cubic Facet Model

Directly solving for a and σ^2 is computationally expensive. For example, when the neighborhood size is $5 \times 5 \times 5$, each a_i needs 125 multiplications and 124 additions in the linear transform. To improve efficiency, we do not use the brute-force approach, but adopt the Discrete Orthonormal Polynomial (DOP) approach.

For a better understanding on the DOP cubic facet model, an introduction to the discrete orthonormal polynomial and its fast implementation is given in Appendix A.

The major difference between the canonical and DOP model lies in what bases they use. In the DOP approach, we generate three 1D DOPs and use a subset of their tensor product to form the twenty bases. Replacing the canonical bases with the DOP bases in Eq. (11), we obtain the design matrix D_d of the DOP facet fitting problem. D_d is column-wise orthonormal in the sense that $D'_d D_d = I$.

Estimating the coefficients of the DOP cubic facet model corresponds to finding k to minimize the fitting error

$$e^2 = \|D_d k - J\|^2 .$$

Solving for k in the least-square sense yields

$$k = (D'_d D_d)^{-1} D'_d J = D'_d J ,$$

which is an orthonormal transform on the neighborhood data vector. We do not directly evaluate the expression for k, but take the advantage of the DOP's separability and cut down the amount of computations by a factor greater than 10.

The fitting error is calculated as the energy difference between the neighborhood data and the DOP coefficients. The image noise variance is estimated from the fitting error by

$$\hat{\sigma}^2 = \frac{1}{N-20} \left(\|J\|^2 - \|k\|^2\right) .$$

Noticing that two neighborhoods whose centers are in the same row and immediately adjacent are different only by 25 pixels, $\|J\|^2$ can be calculated recursively. Compared with the brute-force approach, this reduces the amount of computation to less than 1/50.

4.4. From DOP to Canonical Parameters

We have obtained the DOP parameter k, while what we need for the IFCEs are the canonical parameters a_2 through a_{10}. Converting k to the canonical parameter vector $X = (a_2, a_3, \ldots, a_{10})^T$ is merely a sparse linear transform $X = Lk$.

For the cubic facet model defined in a $5 \times 5 \times 5$ neighborhood, the transform is represented as

$$
\begin{aligned}
a_2 &= c_1k_2 + c_2k_{11} + c_3k_{13} + c_4k_{19}\\
a_3 &= c_5k_3 + c_6k_{12} + c_7k_{14} + c_8k_{16}\\
a_2 &= c_9k_4 + c_{10}k_{15} + c_{11}k_{17} + c_{12}k_{18}\\
a_5 &= c_{13}k_5\\
a_6 &= c_{14}k_6\\
a_7 &= c_{15}k_7\\
a_8 &= c_{16}k_8\\
a_9 &= c_{17}k_9\\
a_{10} &= c_{13}k_{10}
\end{aligned}
$$

where c_i's are all constants. In this case, the transform matrix L is row-wise orthogonal because no k_i appears more than once in L. The fact that LL' is a constant diagonal matrix simplifies the calculation of the covariance matrix of the facet model for this case.

4.5. Derivatives and Solving the IFCEs in a

The coefficients in the IFCEs are scaled canonical cubic facet parameters. We substitute the parameters into Eq. (8) obtaining the Image Flow Constraint Equations which we actually use. The A and b are expressed in terms of a as

$$
A = \begin{pmatrix} a_2 & a_3 \\ 2a_5 & a_6 \\ a_6 & 2a_7 \\ a_{10} & a_8 \end{pmatrix} \; b = - \begin{pmatrix} a_4 \\ a_{10} \\ a_8 \\ 2a_9 \end{pmatrix} . \tag{12}
$$

When $A'A$ is nonsingular, V is evaluated as

$$
V = (A'A)^{-1}A'b\,.
$$

Defining

$$
\begin{aligned}
p_1 &= a_2^2 + 4a_5^2 + a_6^2 + a_{10}^2\\
p_2 &= a_2a_3 + 2a_5a_6 + 2a_6a_7 + a_8a_{10}\\
p_3 &= a_3^2 + a_6^2 + 4a_7^2 + a_8^2\\
p_4 &= a_2a_4 + 2a_5a_{10} + a_6a_8 + 2a_9a_{10}\\
p_5 &= a_3a_4 + a_6a_{10} + 2a_7a_8 + 2a_9a_9\,,
\end{aligned}
$$

yields

$$
A'A = \begin{pmatrix} p_1 & p_2 \\ p_2 & p_3 \end{pmatrix} \; A'b = - \begin{pmatrix} p_4 \\ p_5 \end{pmatrix}
$$

and

$$V = \frac{1}{p_1p_3 - p_2^2}\begin{pmatrix} p_2p_5 - p_3p_4 \\ p_2p_4 - p_1p_5 \end{pmatrix}.$$

4.6. Advantages of Facet Model in Image Flow Estimation

Derivative calculation is the lowest-level processing for gradient-based image flow estimation. It greatly affects the efficiency of the entire algorithm. People usually use simple neighborhood differences as derivative estimates. Such schemes are highly sensitive to noise. Barron[1] claims that higher accuracy can be achieved using the four-point central difference mask $(-1, 8, 0, -8, 1)/12$. However we have proven that applying this mask actually gives us the parameter of the first-order term of an 1D canonical cubic polynomial on a neighborhood of size 5. This means, in a $5 \times 5 \times 5$ neighborhood, this method estimates the first-order derivatives using only $13/125$ of the data. The accuracy and robustness of this approach is problematic. We implement an image flow estimator which is the same as our estimator except it uses the four-point central difference mask approach for derivatives. Experiments show that on our synthetic sphere sequences, it produces about twice the average error vector magnitude as our estimator, and on real data it gives much worse visual effects.

As Barron reports in Ref. 1, an additional prefilter or smoother prior to the estimator can significantly improve the derivative accuracy. He applies a spatiotemporal Gaussian prefilter to the gradient-based techniques. However the improvement comes with extra computation. Furthermore, the variance σ^2 of this prefilter are chosen empirically, although he claims the value $\sigma = 1.5$ works generally well. Since fifteen frames are required for the typical σ value, when the number of frames available is small, this method will be a problem. As well, if the total neighborhood size is kept constant, the facet model derivative estimator will in fact have lower variance than those obtained by a prefiltering approach.

In the facet model, we first estimate the true neighborhood data from the noisy observation and then calculate the derivatives from the estimates. This scheme in effect works as a smoothing filter. No extra work or setting hand-tuned parameters is needed. The facet model allows different noise variance in each neighborhood. This adaptive scheme is more appropriate than sweeping the same Gaussian mask across the entire sequence.

The facet model also provides the image noise variance estimate in each neighborhood. It is a quantitative evaluation to the input noise and enables us to estimate the perturbation on the image flow vector in a systematic way.

5. ERROR ANALYSIS THROUGH COVARIANCE PROPAGATION

We conduct covariance propagation analysis to keep track of the error in the estimated image flow. We propagate all sources of errors to image flow vectors, obtaining a covariance matrix with each of them. And then a χ^2 statistic is obtained from the covariance matrix as the estimation confidence measure. Appropriately

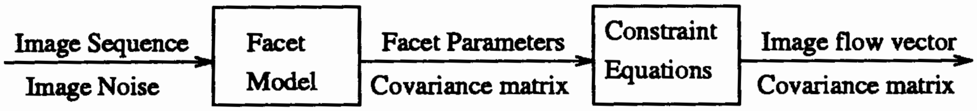


Fig. 2. Block diagram of our algorithm

thresholding the χ^2 statistic allows us to select the reliable estimates at a specified significance level. The selection has a clear statistical meaning and is therefore widely applicable.

A complete introduction to the covariance propagation theory is given in Ref. 6. Appendix B describes the part pertinent to our algorithm.

To propagate covariance through the system (Fig. 2), two assumptions are made: (i) the error only comes from the image noise, which is an additive iid., Gaussian variable with zero mean and variance σ^2, and (ii) the system is approximately linear for a small perturbation σ. As the system is composed of two cascaded subsystems, covariance propagation has two steps.

5.1. Covariance Propagation Through the Facet Model

The input to the facet model fitting step is the neighborhood data vector J, and the output is the canonical cubic facet parameter vector $a = (a_1, a_2, \ldots, a_{20})$. Only nine of the twenty parameters are used in the IFCEs. We denote them by the vector $X = (a_2, a_3, \ldots, a_{10})$. Under the assumption of the noise model, the covariance matrix of J is $\sigma^2 I$.

Calculating the covariance matrix Σ_a of a is trivial, because going from J to a is merely a linear transform. Denoting by 'Δ' the error between the estimate (observation) and the true value,

$$\begin{aligned}\Sigma_a &= E\{\Delta a \Delta a'\}\\ &= E\{(D'D)^{-1}D'\Delta J \Delta J' D(D'D)^{-1}\}\\ &= (D'D)^{-1}D'E\{\Delta J\Delta J'\}D(D'D)^{-1}\\ &= \sigma^2(D'D)^{-1}\,.\end{aligned}$$

The covariance matrix of X, Σ_X, is easily found as a 9×9 submatrix of Σ_a.

The DOP approach helps us gain more insight to the matrix $(D'D)^{-1}$ for simplifying the implementation. Recall that the DOP parameter vector k comes from an orthonormal transform on the image data vector J

$$k = D'_d J\,.$$

Hence the covariance matrix of k is the same as that of J because

$$\Sigma_k = E\{\Delta k \Delta k'\} = E\{D'_d \Delta J \Delta J' D_d\} = D'_d E\{\Delta J \Delta J'\} D_d = \sigma^2 I\,.$$

We apply a linear transform $X = Lk$ to obtain X. The covariance matrix of X is

$$\Sigma_X = E\{\Delta X \Delta X'\} = E\{L\Delta k\Delta k' L'\} = LE\{\Delta k \Delta k'\}L' = \sigma^2 LL' .$$

For the cubic facet model defined on the $5 \times 5 \times 5$ neighborhood, LL' happens to be a diagonal matrix (Sec. 4.4). Therefore Σ_X is also diagonal. This property allows us to convert matrix multiplications to scalar products.

With X and Σ_X available, we now proceed to propagate covariance through the image flow constraint equations.

5.2. Covariance Propagation Through the IFCEs

The input to the image flow constraint equations is the canonical facet parameter vector X with its covariance Σ_X, and the output is the image flow vector $V = (u, v)'$ with its covariance matrix Σ_V. Estimating V from X is no longer a simple linear transform. Therefore we apply the implicit covariance propagation theory[6] to find Σ_V.

$F(X, V)$, the criterion function to estimate V, is defined by

$$F(X, V) = (AV - b)'(AV - b) \tag{13}$$

where A, b are given in Eq. (12).

We take partial derivatives of $F(X, V)$ with respective to V forming the gradient vector $g(X, V)^{2\times 1} = 2A'(AV - b)$.

The second-order partial derivatives of F with respect to V and X are respectively

$$\frac{\partial g(X, V)}{\partial V}^{2\times 2} = 2A'A \tag{14}$$

$$\frac{\partial g(X, V)}{\partial X}^{9\times 2} = 2\times \begin{pmatrix} 2a_2u + a_3v + a_4 & a_3u \\ a_2v & a_2u + 2a_3v + a_4 \\ a_2 & a_3 \\ 8a_5u + 2a_6v + 2a_{10} & 2a_6u \\ 2v(a_5 + a_7) + 2a_6u + a_8 & 2u(a_5 + a_7) + 2a_6v + a_{10} \\ 2a_6v & 2a_6u + 8a_7v + 2a_8 \\ a_6 + a_{10}v & 2a_7 + a_{10}u + 2a_8v + 2a_9 \\ 2a_{10} & 2a_8 \\ 2a_5 + 2a_{10}u + a_8v + 2a_9 & a_6 + a_8u \end{pmatrix} . \tag{15}$$

Based on Eq. (20), the covariance of the random perturbation ΔV is

$$\Sigma_{\Delta V} = \left(\frac{\partial g}{\partial V}\right)^{-1} \frac{\partial g}{\partial X}' \Sigma_{\Delta X} \frac{\partial g}{\partial X} \left(\frac{\partial g}{\partial V}\right)^{-1} .$$

Since in our algorithm zero-mean Gaussian noises are assumed ($E\{\Delta V\} = E\{\Delta X\} = 0$), it satisfies that $\Sigma_{\Delta V} = \Sigma_V$ and $\Sigma_{\Delta X} = \Sigma_X$. Not knowing the true values of

X, V, we approximate Σ_V by substituting $\hat{X}, \hat{V}$ for X, V

$$\hat{\Sigma}_{\hat{V}} = \left(\frac{\partial g}{\partial V}(\hat{X}, \hat{V})\right)^{-1} \frac{\partial g'}{\partial X}(\hat{X}, \hat{V}) \Sigma_{\hat{X}} \frac{\partial g}{\partial X}(\hat{X}, \hat{V}) \left(\frac{\partial g}{\partial V}(\hat{X}, \hat{V})\right)^{-1} \tag{16}$$

Up to this point, we have accomplished the covariance propagation from the input image sequences to the image flow estimates.

6. SELECTION BY χ^2 TESTING

The statistical meaning of the selection scheme is that a velocity estimate is taken to be nonzero only if it is statistically significantly different from zero, or to say, it is taken to be nonzero only if it is large enough relative to its variance. A physical interpretation of this selection scheme is that only the estimate with a small relative error is taken to be good.

We assume the image noise is Gaussian and the system is approximately linear for small variances. Therefore the image flow estimate $\hat{V}$ is a Gaussian variable with mean the true 2D velocity V and covariance Σ_V. When the noise is small, $\hat{\Sigma}_{\hat{V}}$ is a good estimate of Σ_V. We further assume the perturbation on the two components u, v are independent and identically distributed. Under this assumption, Σ_V is simplified to $\sigma^2 I^{2\times 2}$, where σ^2 is the common variance of u and v. We further approximate σ^2 by $(\hat{\sigma}_u^2 + \hat{\sigma}_v^2)/2$, the average of the diagonal entries in $\hat{\Sigma}_{\hat{V}}$. Finally the hypothesis testing is described as below.

$H_0 : V = 0, \quad H_1 : V \neq 0$

Test statistic:

$$T = \hat{V}'\hat{\Sigma}_{\hat{V}}^{-1}\hat{V}\,. \tag{17}$$

Because of the large number of degrees of freedom associated with $\hat{\sigma}_u^2 + \hat{\sigma}_v^2$, the distribution of T is χ_2^2, a central χ^2 with two degrees of freedom.

A single-sided χ^2 test is conducted on T to reject the false alarms. For any specified significance level α there is associated a χ^2 value T_t, which is used as the threshold. H_0 is accepted when $T < T_t$ or H_1 is accepted otherwise. Whenever H_0 is accepted a false alarm is claimed and the estimate is set to 0.

The assumptions for the Image Flow Constraint Equations do not hold at places of homogeneity, moving boundaries and abrupt intensity change. The χ^2 test can detect these situations and then reject the image flow estimates. Because the 3D cubic facet model is insufficient to describe neighborhoods at moving boundaries or abrupt intensity change, the variance estimates σ^2 become very large at such places. Large variances will propagate to the flow vectors to produce large σ_u^2, σ_v^2. When aperture problems occur, the matrix $A'A$ in the IFCEs become close to singular. This causes huge variances of flow vectors. As it is clear from Eq. (17), large variances make T very small. In the hypothesis testing, H_0 will be accepted and the estimates will be set to 0. Therefore the χ^2 test also rejects bad estimates. Since V will be eventually set to 0 when $A'A$ is close to singular, we detect this situation early and directly set $V = 0$, skipping the equation solving, covariance propagation and selection steps.

7. FLOW FIELD SMOOTHNESS CONSTRAINT

We call the algorithm developed in Secs. 3–6 Algorithm I. It estimates each image flow vector independently from its neighboring vectors. Actually in most cases moving objects are larger than one pixel and the motion field is smooth at most places. Taking this into account can improve the estimation robustness. Therefore we add a flow field smoothness constraint to Algorithm I and form Algorithm II.

7.1. Algorithm II

Now we assume that the image flow vectors in each small 3D flow field regularization neighborhood are constant. It is worth noting that just to solve for the raw flow field, the size of this neighborhood can be any reasonable combination, for example, $3 \times 3 \times 3$, or $5 \times 5 \times 1$. But currently our implementation of covariance propagation through Algorithm II depends on the choice of this size, so in the following explanation we set it to be $5 \times 5 \times 1$ (row by column by frame). Also to facilitate understanding, we take the facet model neighborhood size to be $5 \times 5 \times 5$. Thus the image data support for each velocity estimate becomes $9 \times 9 \times 5$.

We traverse the 25 positions within the regularization neighborhood in a left–right and top–down manner and number them by $i, i = 1, 2, \ldots, 25$. Accordingly, Algorithm I's image flow constraint equation at the ith position is expressed as $A_i V = b_i$.

We want to determine the motion vector at the center, namely the 13th position. Under the assumption of constant flow in the neighborhood, we stack the 25 sets of IFCEs together to form the new constraint equation

$$A_s V = b_s$$

where

$$A_s^{100\times 2} = \begin{pmatrix} A_1^{4\times 2} \\ A_2^{4\times 2} \\ \vdots \\ A_{25}^{4\times 2} \end{pmatrix} \quad \text{and} \quad b_s^{100\times 1} = \begin{pmatrix} b_1^{4\times 1} \\ b_2^{4\times 1} \\ \vdots \\ b_{25}^{4\times 1} \end{pmatrix}.$$

Similarly to the Algorithm I, when $A_s' A_s$ is nonsingular, the explicit Least-Square solution is

$$V = (A_s' A_s)^{-1} A_s' b_s .$$

If $A_s' A_s$ is close to singular (the magnitude of its determinant is less than a certain threshold), we directly set $V = 0$.

Algorithm II is less sensitive to noise than Algorithm I. The flow field consistency improvement is clear by comparing Figs. 6 and 8.

Algorithm II involves more computation, and it makes motion near boundaries more smeared. However these problems are negligible when overall accuracy is the primary concern.

7.2. Covariance Propagation in Algorithm II

Algorithm II is also composed of two subsystems (Fig. 2) just as Algorithm I, but the input sizes of these systems have changed. Now for each velocity estimate the input to the facet model step is a $9 \times 9 \times 5$ block of image data, and the input of the IFCEs (output of the facet model) becomes $X_s = (X_1', X_2', \ldots, X_{25}')'$.

By exploiting the regularity in forming the new constraint, covariance propagation through the IFCE step is easily done.

7.2.1. Propagating through the IFCEs

The criterion function we minimize for V is

$$F_s(X_s, V) = (A_s V - b_s)'(A_s V - b_s)\,.$$

The gradient of F_s with respect to V is

$$\begin{aligned} g_s^{2\times 1}(X_s, V) &= 2A_s'(A_s V - b_s) \\ &= 2(A_1' A_2' \ldots A_{25}')^{2\times 100} \begin{pmatrix} A_1 V - b_1 \\ A_2 V - b_2 \\ \vdots \\ A_{25} V - b_{25} \end{pmatrix}^{100\times 1} \\ &= 2\sum_{i=1}^{25} A_i'(A_i V - b_i)\,. \end{aligned}$$

The second-order derivative of F_s with respect to V and X are

$$\begin{aligned} \frac{\partial g_s^{2\times 2}(X_s, V)}{\partial V} &= 2A_s' A_s \\ &= 2\sum_{i=1}^{25} A_i' A_i \end{aligned}$$

$$\frac{\partial g_s^{2\times 1}(X_s, V)}{\partial X_s^{(9\times 25)\times 1}} = \begin{pmatrix} \dfrac{\partial g_s(X_s, V)}{\partial X_1}^{9\times 2} \\ \dfrac{\partial g_s(X_s, V)}{\partial X_2}^{9\times 2} \\ \vdots \\ \dfrac{\partial g_s(X_s, V)}{\partial X_{25}}^{9\times 2} \end{pmatrix}^{225\times 2}$$

where each $\frac{\partial g_s(X_s,V)}{\partial X_i}$ is the same as Eq. (15) except that all facet parameters now become functions of i.

Similarly,

$$\Sigma_{\Delta V} = \left(\frac{\partial g_s}{\partial V}\right)^{-1} \frac{\partial g_s}{\partial X_s}' \Sigma_{\Delta X_s} \frac{\partial g_s}{\partial X_s} \left(\frac{\partial g_s}{\partial V}\right)^{-1},$$

and we finally estimate $\hat{\Sigma}_{\hat{V}}$ by substituting $\hat{V}$ and $\hat{X}_s$ into the expression. The only unknown in the expression is $\Sigma_{\Delta X_s}$, which we will estimate in the section below.

7.2.2. Propagating through the facet model

Now X_s has the dimension 225×1 $((9 \times 25) \times 1)$ and its covariance matrix Σ_{X_s} has the dimension 225×225. We divide Σ_{X_s} into $25 \times 25 = 625$ submatrices, each of which, C_{ij}, is a 9×9 covariance matrix of the two parameter vectors $X_i, X_j, i, j = 1, 2, \ldots, 25$. We will estimate these submatrices one by one and then put them together to form Σ_{X_s}. Section 5.1 has already discussed how to calculate each diagonal submatrix C_{ii}. The problem lies in determining the off-diagonal submatrices, because as every two $5 \times 5 \times 5$ neighborhoods overlap each other in the $9 \times 9 \times 5$ support, two sets of facet parameters $X_i, X_j (i \neq j)$ are not independent.

Let us see how X_i and X_j are related to each other.

Recall that the facet parameter vector X is a linear transform of the neighborhood data vector J, i.e. $X = PJ$, where $P^{9 \times 125}$ is composed of the 2nd to the 10th rows of $(D'D)^{-1}D'$. Clearly there are many ways that we can traverse the neighborhood data to form the vector J. But for each traversing order, there is only one column ordering of P which leads to the correct X. To indicate the dependency of P's arrangement on J's, we denote P by P_J instead.

In the ith and jth neighborhoods, we have $X_i = P_{J_i} J_i$ and $X_j = P_{J_j} J_j$, where J_i and J_j have some data in common. Let J_{ij} be the union of J_i and J_j. We can find two linear operators B_i, B_j which make $X_i = B_i J_{ij}$, $X_j = B_j J_{ij}$ hold. Since J_{ij} has more elements than J_i (J_j), B_i (B_j) has more columns than P_{J_i} (P_{J_j}). To keep the identity $B_i J_{ij} = P_{J_i} J_i$ $(B_j J_{ij} = P_{J_j} J_j)$, we need to insert some zero columns into P_{J_i} (P_{J_j}) to form B_i (B_j). Those zero columns should be at the places whose corresponding J_{ij} terms do not belong to J_i (J_j). In our implementation, J_i has the common data at the rear, J_j has them at the head, and J_{ij} has them in the middle. Therefore B_i is formed by padding a zero block to the right of P_{J_i}, $B_i = (P_{J_i} \,|\, \mathbf{0})$, and similarly $B_j = (\mathbf{0} \,|\, P_{J_j})$. Figure 3(a) illustrates the two neighborhoods and our data traversing order. Neighborhood i, j are partitioned into three 3D blocks. In each block, we scan the first frame in a zig-zag manner, and then the second frame, ..., when this block is finished, we move on to the next one.

We assume the noise on J_{ij} are additive iid with zero mean and variance σ_{ij}^2, and σ_{ij}^2 can be approximated by $(\sigma_i^2 + \sigma_j^2)/2$. Then the covariance matrix of X_i, X_j is

$$C_{ij} = E\{\Delta X_i \Delta X_j'\} = E\{B_i \Delta J_{ij} \Delta J_{ij}' B_j'\} = \sigma_{ij}^2 B_i B_j' .$$

We do not need to repeat zero padding and matrix multiplication to compute all $B_i B_j'$ in each neighborhood, because $B_i B_j'$ only has a small number of forms, which can be determined before the covariance propagation procedure. Figure 3(b) shows a $9 \times 9 \times 5$ support and three neighborhoods i, j, k. The relative position of k to j is the same as that of j to i. We can traverse j, k in the same way as we traverse i, j. Each traversing order corresponds with one pair of B_i and B_j, and hence a unique $B_i B_j'$. So i, j and j, k have the same matrix $B_i B_j'$. In our case we observe that

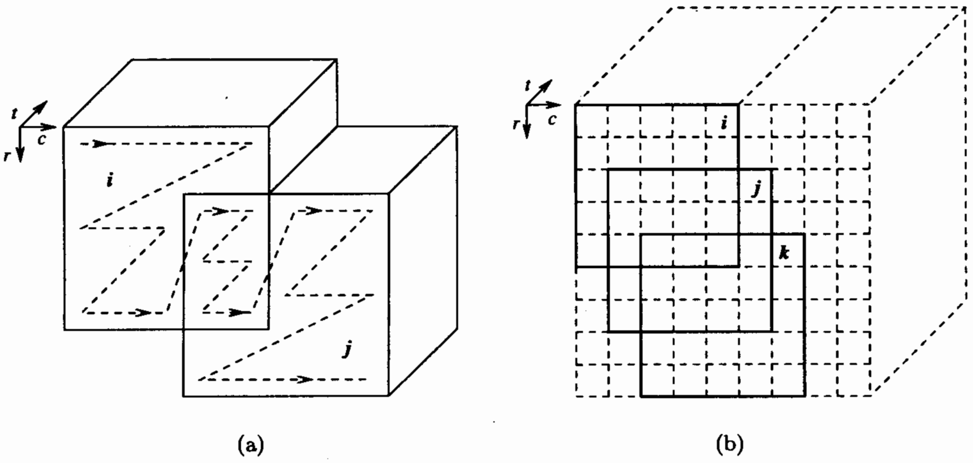


Fig. 3. Relationship between two facet fitting neighborhoods. (a) Our data traversing order and (b) B_iB_j' is decided by neighborhood relative position.

two neighborhoods have totally 41 possible relative positions. Therefore B_iB_j' can only take 41 forms, which we can determine beforehand and put in a lookup table. When we actually evaluate C_{ij}, we just look up the table for the appropriate B_iB_j' and multiply it by the scalar $(\sigma_i^2 + \sigma_j^2)/2$. With all C_{ij}, $i, j = 1, 2, \ldots, 25$ available, we assembly them together yielding Σ_{X_s}. And then we can proceed to evaluate the velocity covariance matrix $\hat{\Sigma}_{\hat{V}}$.

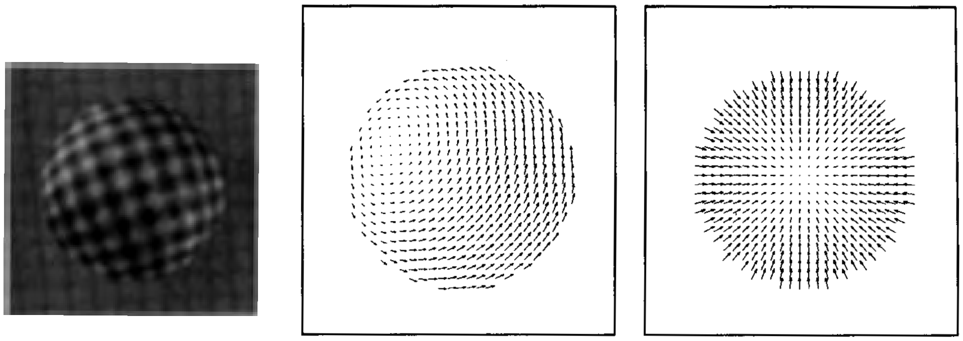
8. EXPERIMENTS AND ANALYSIS

We conducted both qualitative and quantitative experiments to evaluate our algorithms. Barron gives a comparison among nine representative image flow estimation techniques on various synthetic and real image sequences in Ref. 1. Of the nine techniques, the one developed by Lucas and Kanade (LK) is the best in terms of overall performance.[2] The one by Fleet and Jepson (FJ), and that by Uras *et al.* (Uras) rank as the second and the third, respectively. The technique of Horn and Schunck (HS) is probably the earliest one with reasonably good performance. We will compare our algorithms mainly with LK. Results of FJ, Uras and HS will be shown on the TAXI sequence as well.

For all the experiments presented here, we choose the facet model fitting neighborhood size to be 5 × 5 × 5 and the smoothness constraint neighborhood size 5 × 5 × 1.

8.1. Test Data

We generate the sequences "ROT", of a rotating sphere, and "DIV", of a diverging sphere for quantitative evaluation and comparison. Their central frames and ground-truth flow fields are shown in Fig. 4. Details about the synthesis scheme are given in Ref. 14. Discussion of results on the synthetic data is in Sec. 8.3.

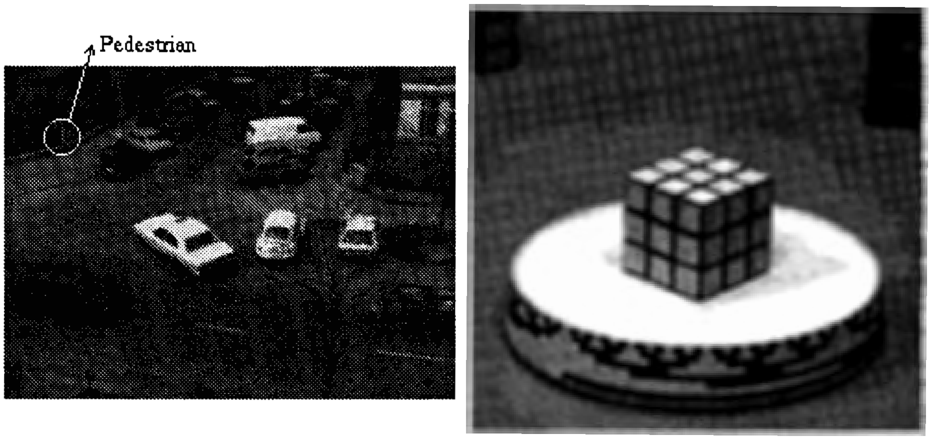


(a) Central frame

(b) ROT

(c) DIV

Fig. 4. Synthetic sequences and their true flow fields.



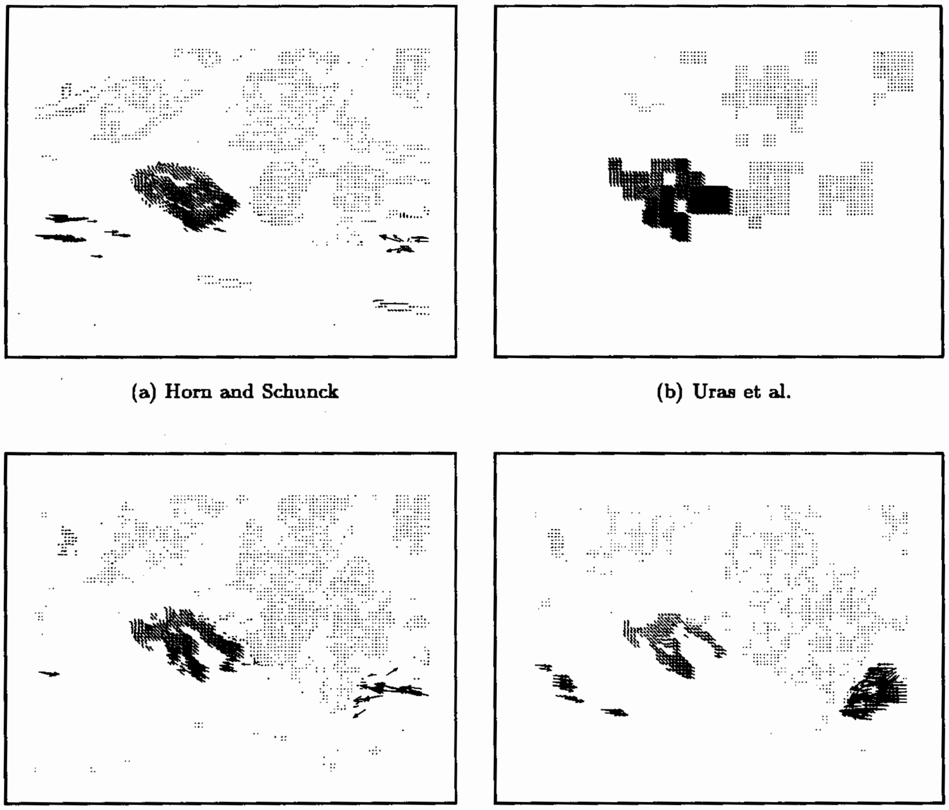
(a) TAXI

(b) RUBIK

Fig. 5. Central frames of TAXI and RUBIK sequences.

Results of two real image sequences "TAXI" [Fig. 5(a)] and "RUBIK" [Fig. 5(b)] from Ref. 1 will be shown in Secs. 8.2 and 8.3. In TAXI there are four moving objects: a taxi turning the corner, a car in the lower left, driving from the left to the right, a van in the lower right driving from the right to the left, and a pedestrian in the upper left walking across the street from the right to the left [circled and marked in Fig. 5(a)]. 2D speeds of the four moving objects are approximately 1.0, 3.0, 3.0 and 0.3 pixel/frame respectively. The background scene is still. In RUBIK, on a still background, a Rubik's cube is rotating counterclockwise on a turntable. The 2D velocity of the cube is between 0.2 and 0.5 pixel/frame, and that of the turntable is between 1.2 and 1.4 pixel/frame.

The results of HS, Uras, LK and FJ on TAXI are shown in Fig. 6 for future comparison.



(a) Horn and Schunck

(b) Uras et al.

(c) Lucas and Kanade

(d) Fleet and Jepson

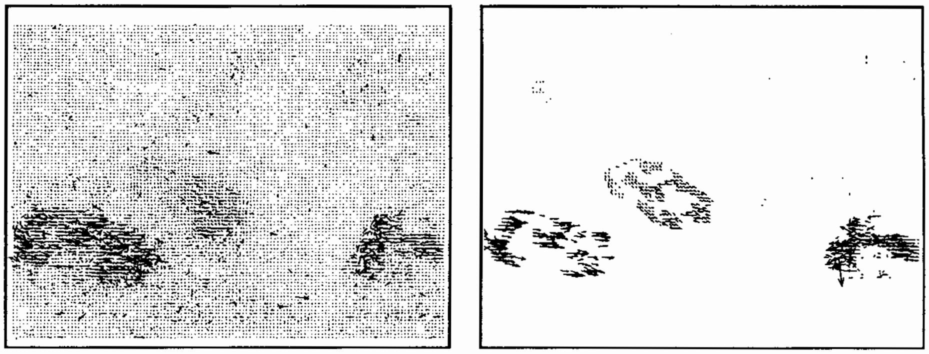
Fig. 6. Results of four representative techniques on TAXI.

8.2. Results of Algorithm I

As we can see from Refs. 1 and 11 etc., many techniques suffer from severe false alarms on TAXI. After selection, many false alarms are still there while some correct motion vectors such as those of the two black cars are gone. So far nearly no result reported in the literature successfully discriminates the motion vectors of the pedestrian from the background false alarms.

Our raw flow field [Fig. 7(a)] also has a lot of false alarms. But after χ^2 selection with a significance level $\alpha = 0.005$, almost all false alarms are eliminated and the motion vectors of the four moving objects including the pedestrian stand out [Fig. 7(b)].

If a selection scheme is reliable, the more rigorous the selection becomes, higher accuracy should result. When we decrease α to a certain value, all the false alarms are gone. Further decreasing rejects all the vectors inconsistent with the prior knowledge. At last when α is very close to 0, only one vector is left, which is on white car at the center. The observation is also clear from Fig. 9 (Algorithm II). Similar experiments were conducted on LK's selector. When the threshold increases,



(a) Raw flow field

(b) χ^2 selection, $\alpha = 0.005$

Fig. 7. Results of TAXI from Algorithm I.

the vectors of the two black cars quickly go away. Finally a high threshold only leaves three vectors in the flow field, which are all background false alarms.

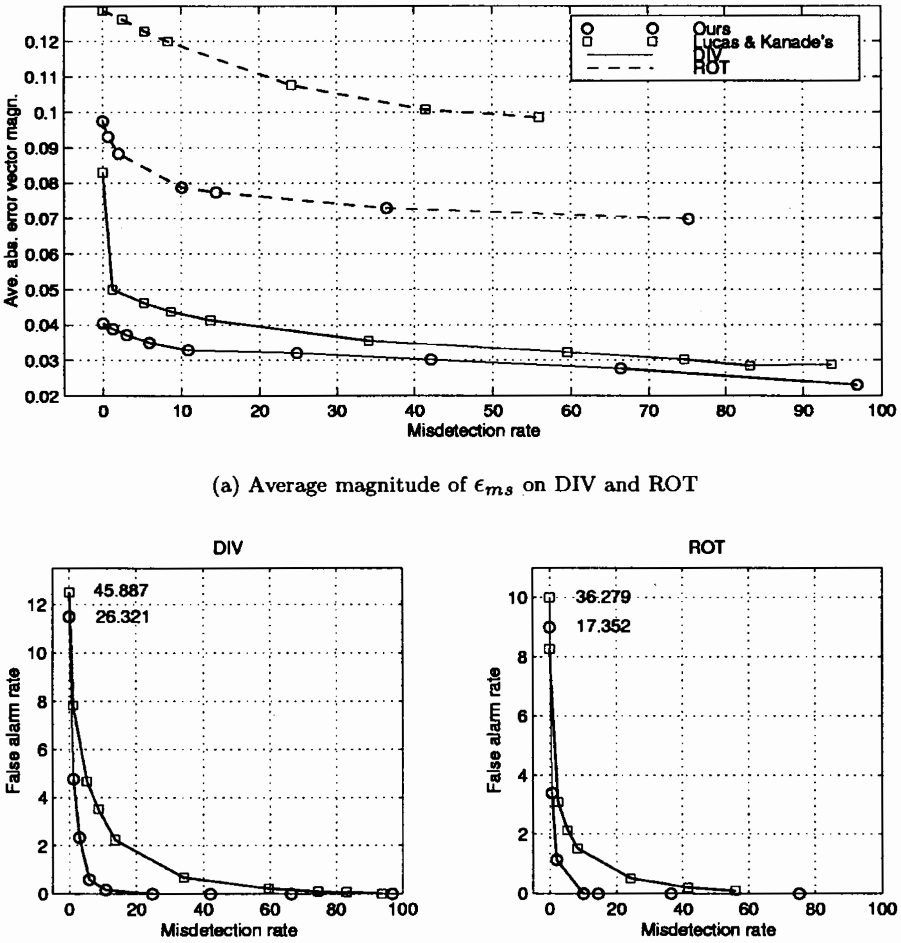
8.3. Results of Algorithm II

We compare the performance of Algorithm II with that of LK in terms of False Alarm Rate (FAR), Misdetection Rate (MR), and Average Error Vector Magnitude (AEVM) on our synthetic sphere data. A false alarm is defined as the case in which a nonzero vector is claimed at a pixel of no motion. A misdetection is the case in which no motion is detected at a pixel of nonzero motion vector. The error vector is defined as the difference vector of the true and the detected motion vector at places where both of them are nonzero. False alarm rate (misdetection rate) is defined as the number of false alarms (misdetections) divided by the number of actual vectors in the flow field.

The AEVM-MR and FAR-MR curves of Algorithm II and LK's are shown in Fig. 8. Circles and squares represent our and LK's techniques respectively. To exhibit the details, large FAR's are not shown with the literal scale but marked with their values instead.

On these two sequences our algorithm achieves higher accuracy. Especially on ROT, the average vector error magnitude of LK with the misdetection rate as high as 56% is still larger than that of our raw estimate. As to the false alarm rates on the raw flow field, our results are less than half of those of LK. When the misdetection rate increases, the false alarm rate of our algorithm quickly decreases to 0, while those of LK remain nonzero even with the misdetection rates as high as 56% on ROT and 94% on DIV.

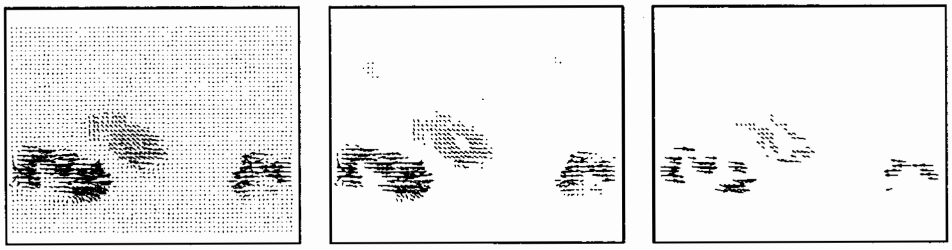
The raw flow field, the flow field selected with $\alpha = 0.005$ and $\alpha < 10^{-14}$ on TAXI from Algorithm II are shown in Fig. 9. The improvement of accuracy and motion field consistency over that of Algorithm I is dramatic. Compared with the results on TAXI reported in the literature, our algorithm gives the best performance.



(a) Average magnitude of ϵ_{ms} on DIV and ROT

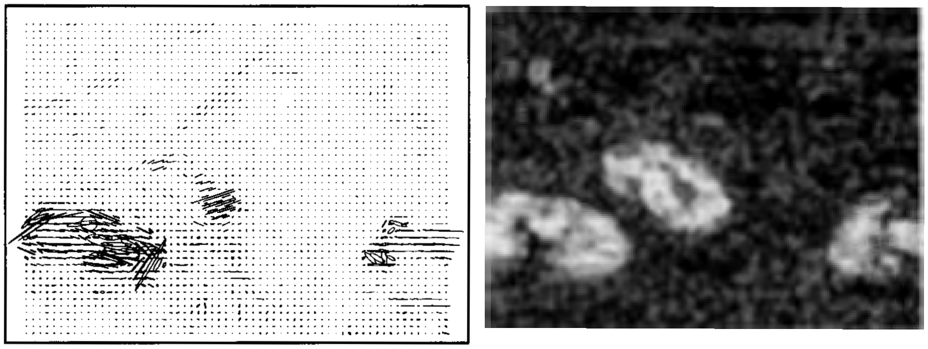
(b) False alarm — misdetection on DIV and ROT

Fig. 8. Quantitative comparison between Algorithm II and LK.



(a) Raw flow field (b) $\alpha = 0.005$ (c) $\alpha < 10^{-14}$

Fig. 9. Effect of decreasing significance level of χ^2 test on TAXI (Algorithm II).



(a) Covariance field

(b) χ^2 map

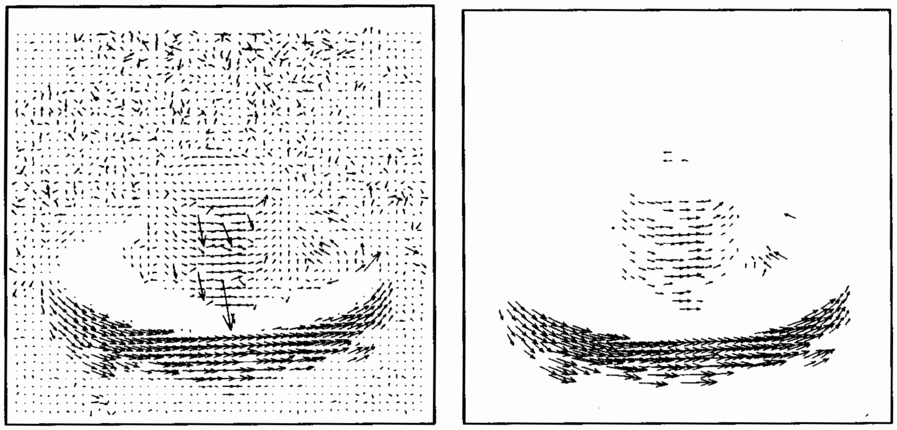
Fig. 10. Covariance field and χ^2 map of TAXI.

We show the covariance field of TAXI in Fig. 10(a). Each covariance matrix is represented as an ellipse with the lengths of axes as its eigenvalues and the axes directions are its eigenvectors. The aperture problem at the places of boundaries, such as the car window and the white lines, are detected by the very slim ellipses along the small-gradient directions. The lower-right car is occluded by a tree and hence the variances of its motion are large.

In this flow field, large motion is associated with large uncertainties. This gives one possible reason why LK's selection scheme does not work properly. It uses the smaller eigenvalue of the covariance matrix as the confidence measure,[1] while as we see in this covariance field the background false alarms have the smallest eigenvalues.

Figure 10(b) shows the map of test statistic T. The brighter intensity value represents the larger T. There are four bright spots at the places of four moving objects. Comparing the map with the covariance field we can see the χ^2 test statistic is a better confidence measure than those extracted only from the covariance matrix. It emphasizes the importance of using covariance matrix appropriately.

The RUBIK sequence is quite challenging in the sense that many sources of error are present. It has large image noise, as we can tell from severe background false alarms. The top of the turntable is a homogeneous region. The cube has many black stripes on the surface, which cause abrupt intensity changes and aperture problems. Our algorithm detects the homogeneity from the ill-conditioned matrix $A'_s A_s$ and set the estimates 0. The abrupt intensity change and aperture problem at the black stripes produces erroneous vectors. But they are associated with very large variance and thus rejected in the χ^2 test. Almost all the background false alarms disappear after the selection. Compared with those given in Ref. 1, our result (Fig. 11) on this sequence is among the best.



(a) Raw flow field

(b) $\alpha = 2.06 \times 10^{-9}$

Fig. 11. Results of RUBIK from Algorithm II.

9. CONCLUSIONS AND FUTURE WORK

There are two major contributions of our image flow estimation technique. First, the facet model is used to provide high-quality derivative estimates, image noise variance estimation and the effect of prefiltering simultaneously. Secondly, covariance propagation enables a systematic error analysis and a statistically meaningful selection scheme.

Experiments show our algorithms successfully reject false alarms and bad estimates while achieving a low misdetection rate and high accuracy for nonzero image flow vectors. Quantitative and qualitative comparison between our approach with other competitive techniques makes the advantages obvious.

There are quite a few directions we may pursue in future work. The smoothness constraint has proven a success in achieving high accuracy, although only the simplest assumption of constant local flow is made. We are considering to add weights when combining the image flow constraint equations. We also expect higher accuracy with a linear local flow model. We will propagate the covariance through Lucas and Kanade's algorithm to gain more insight to the comparison. This work will also enable us to study the robustness of different algorithms. The assumption of iid Gaussian image noise might oversimplify many real cases. Now we are trying to characterize the noise as colored Gaussian.

APPENDIX

A. DISCRETE ORTHONORMAL POLYNOMIALS (DOPS)

This section gives a description on the discrete orthonormal polynomials mainly based on Chapter 8 in Ref. 5. We start from the concept, generation and use of 1D DOPs, proceed by the properties and generation of N–D DOPs, at last explain the fast implementations by use of separability and recursion. The contents are important for a better understanding on the DOP approach in Sec. 4.3.

A.1. 1D DOPs

The 1D $(N-1)$th order Discrete Orthonormal Polynomial $P_n(r)$, $n = 0, \ldots, N-1$ is defined on a symmetric discrete integer index set R. R is symmetric in the sense that $r \in R$ implies $-r \in R$. The number of elements in R is N. $P_n(r)$ is orthonormal because

$$\sum_{r \in R} P_n(r) P_m(r) = 0$$

$$\sum_{r \in R} P_n(r) P_n(r) = 1, \ \forall m, \ n \in [0, N-1], \ m \neq n\,.$$

Discrete orthogonal polynomial $p_n(r)$'s can be recursively generated by the relation

$$p_{n+1}(r) = r p_n(r) - \beta_n p_{n-1}(r)$$

where

$$\beta_n = \frac{\sum_{r \in R} r p_n(r) p_{n-1}(r)}{\sum_{r \in R} p_{n-1}^2(r)}$$

$p_0(r) = 1$ and $p_1(r) = r$.

The discrete orthonormal polynomials $P_n(r)$'s are obtained by normalizing the discrete orthogonal polynomials on their index set

$$P_n(r) = \frac{p_n(r)}{\sqrt{\sum_{s \in R} p_n^2(s)}}\,.$$

Here is an example of the 1D 3rd-order discrete orthogonal polynomials

Index Set	DOP
$\{-2, -1, 0, 1, 2\}$	$\left\{1, r, r^2 - 2, r^3 - 3.4r, \ r^4 - \frac{31}{7} r^2 + \frac{72}{35},\right\}$

and the corresponding discrete orthonormal polynomials (DOPs)

Index Set	DOP
$\{-2, -1, 0, 1, 2\}$	$\left\{\frac{1}{\sqrt{5}}, \frac{r}{\sqrt{10}}, \frac{1}{\sqrt{14}}(r^2 - 2), \frac{\sqrt{10}}{12}(r^3 - 3.4r), \ \frac{\sqrt{70}}{24}\left(r^4 - \frac{31}{7} r^2 + \frac{72}{35}\right)\right\}$

A.2. 1D Cubic Curve Fitting

Let a data value $d(r)$ be observed for each $r \in R$. The exact fitting problem is to determine the coefficients $k_0, \ldots, k_{N-1}$ such that

$$d(r) = \sum_{n=0}^{N-1} k_n P_n(r), \ r \in R.$$

The approximate fitting problem is to determine $k_0, \ldots, k_{K-1}$, $K \leq N$ such that

$$e^2 = \sum_{r \in R} [d(r) - \sum_{n=0}^{K-1} k_n P_n(r)]^2$$

is minimized. In either case the solution is

$$k_n = \sum_{r \in R} P_n(r) d(r).$$

The exact fitting coefficient and the least-square coefficients are identical for $n = 0, \ldots, K-1$.

It is obvious that each k_n is actually a dot-product of $d(r)$'s and $P_n(r)$'s for $r \in R$. Thus to calculate k_n, we just need to convolve the mask $P_n(r), r \in R$ with the neighborhood data.

The approximate fitting can be thought of as a projection from a N-dimensional space to a K-dimensional space spanned by the K DOP basis vectors. The residual error e^2 is the energy residing in the complement space of dimension $N - K$. Thus instead of calculating e^2 in the brute force manner, we can compute it as the energy difference between the two spaces

$$e^2 = \sum_{r \in R} d^2(r) - \sum_{r \in K} k_r^2.$$

This approach greatly reduces the computational complexity.

A.3. 2D and Higher Dimensional DOPs

2D DOPs can be created from two sets of 1D DOPs by taking their tensor products. Let $\{P_0(r), \ldots, P_{N-1}(r)\}$ and $\{Q_0(c), \ldots, Q_{M-1}(c)\}$ be two sets of DOPs defined on R and C respectively. Then their tensor product is a set of 2D DOPs defined on $R \times C$. A simple proof is given below.

$$\sum_{r \in R} \sum_{c \in C} P_i(r) Q_j(c) P_n(r) Q_m(c) = \sum_{r \in R} P_i(r) P_n(r) \sum_{c \in C} Q_j(c) Q_m(c) = 0$$

$$\sum_{r \in R} \sum_{c \in C} P_i(r) Q_j(c) P_i(r) Q_j(c) = \sum_{r \in R} P_i(r) P_i(r) \sum_{c \in C} Q_j(c) Q_j(c) = 1$$

where $i \neq n, j \neq m$.

Similarly, all the DOPs of dimension higher than 2 can be generated from 1D DOPs using tensor products. The way of computing the fitting error from the residual energy can also be generalized to higher dimensional cases.

A.4. Computational Efficiency

The DOP facet model is superior to the canonical model because of the computational efficiency. Significant savings of computation result from the use of separability and recursion.

Any N–D ($N \geq 2$) DOP can be expressed as a tensor product of N 1D DOPs, as it is clear from the procedure we generate it. To estimate the coefficient of an N–D DOP, rather than applying an N–D mask on the data, we can separate the N–D mask to N 1D masks and apply them along each of the N directions respectively.

Let us consider the typical case in which the neighborhood size is $W \times W \times W$ and the image sequence size is $N \times N \times W$.

In the brute force approach, every coefficient takes W^3 multiplications in each neighborhood. There are totally $(N-W+1)^2$ neighborhoods. Therefore the number of multiplications is $(N-W+1)^2W^3$.

Now let us see what happens when separability is used. Masking the sequence by the 1D mask along the time axis needs N^2W multiplications. It yields one frame of temporary result. Masking this frame by the 1D mask along the column axis takes $N(N-W+1)W$ multiplications. This produces a frame of size $N(N-W+1)$. Finally masking this frame by the 1D mask along the row axis yields the coefficient corresponding with these three 1D masks, with $(N-W+1)^2W$ multiplications. The total number of multiplications is $N^2W + N(N-W+1)W + (N-W+1)^2W$.

It is easier to see the difference from the example. When $N = 256, W = 5$, the brute-force approach involves 7,938,000 multiplications, while the one with separability only involves 967,760. The savings is about eight times.

It is important to work on the dimensions in the correct order to achieve the best saving. Separated 1D convolutions should always go from the dimension of smaller size to those of larger sizes. The proof is omitted here.

Another significant saving comes from reusing results of intermediate frames. For the 3D cubic DOPs, there are four 1D bases of order 0 through 3 on each dimension. Each of the twenty DOP bases is a combination of three of these twelve 1D bases whose order sum do not exceed 3, for instance, the DOP basis with the highest-order term rct is a combination of the 1D DOP bases of the highest-order terms r, c, t respectively. Therefore two 3D bases may have one or two dimensions in common. For instance, both r and r^2 have the 0th order terms on the time and column axes. Thus we can obtain the two coefficients on the same intermediate frame resulting from masking on the common bases.

The trouble with this scheme might be the peak storage. However in our case it is not a big problem. Applying the four masks along the time axis gives us four frames of size N^2. Under the constraint of order sum less than or equal to 3, there is a total of 10 frames of size $N(N-W+1)$ from the row-wise masking. The peak intermediate storage is $4N^2 + 10N(N-W+1)$, about 3.6M for $N = 256, W = 5$. This is not bad at all.

Use of separability in the above two ways brings more than 10 times saving on the amount of computation for each parameter estimating. Further saving can be achieved by calculating the residual error in a recursive way.

Let us consider the 3D cubic model in a $5 \times 5 \times 5$ neighborhood. The number of bases is 20. If we directly evaluate $e^2 = (D_dk - J)^2$ in each neighborhood, $125 \times 20 + 125 = 2625$ multiplications are needed. While if we compute e^2 by $\|J\|^2 - \|k\|^2$, it reduces to $125 + 20 = 145$ multiplications. When the $5 \times 5 \times 5$

mask is moving in the image sequence in the way of progressive scanning, only 25 pixels are different between two neighborhoods whose centers are in the same row and immediately adjacent. Therefore except for the ones at the beginning of the row, we only need 25 dot-products for each neighborhood to evaluate $\|J\|^2$. The number of multiplications further reduces to about 45 on the average.

With the increase of the index set size or the dimension, the improvement of the computational efficiency becomes even more substantial.

B. RELATED THEORY OF COVARIANCE PROPAGATION

Reference 6 gives a comprehensive description to the Theory of Covariance Propagation in Computer Vision. This section introduces the part related to our image flow estimation algorithms to facilitate understanding Secs. 5 and 7.2.

The goal of covariance propagation is to propagate the input error step by step through a system to obtain the perturbation on the output. When the random perturbations on both the input and the output of a system are approximately additive, one basic measure of the random perturbation size is given by the covariance matrix of the estimate. As long as the system is approximately linear for small perturbations, given the covariance matrix of the input, we can determine the covariance matrix of the output by analytical covariance propagation.

There are two ways to do covariance propagation through a system. One is the explicit way, the other is the implicit way, which is first introduced into the computer vision field by Ref. 6.

B.1. Explicit Covariance Propagation

The output Θ of a system is related to the input X by $\Theta = f(X)$. If f is an explicit function of X, we can expand $f(X)$ as Taylor series, neglect all the terms of order higher than one, and then relate the small perturbation on Θ to that on X by

$$\Delta\Theta = \frac{d\Theta'}{dX}\Delta X.$$

With the covariance matrix of the input $\Sigma_{\Delta X\Delta X}$ known, the covariance of Θ is

$$\begin{aligned}\Sigma_{\Delta\Theta\Delta\Theta} &= E[\Delta\Theta\Delta\Theta'] \\ &= \frac{d\Theta'}{dX}E[\Delta X\Delta X']\frac{d\Theta}{dX} \\ &= \frac{df(X)'}{dX}\Sigma_{\Delta X\Delta X}\frac{df(X)}{dX}\,.\end{aligned}$$

The derivation of $\Sigma_{\Delta\Theta\Delta\Theta}$ requires $d\Theta/dX$ and $\Sigma_{\Delta X\Delta X}$, functions of the true input X be known, while only $\hat{X} = X + \Delta X$ is observed. When ΔX is small, substituting $\hat{X}$ for X in the two functions give reasonable estimates of the true values. Thus finally the covariance matrix of the output is estimated as

$$\hat{\Sigma}_{\Delta\Theta\Delta\Theta} = \frac{df(\hat{X})'}{d\hat{X}}\hat{\Sigma}_{\Delta X\Delta X}\frac{df(\hat{X})}{d\hat{X}}. \tag{B.1}$$

9.1. Implicit Covariance Propagation

Many steps in a system cannot be expressed as explicit functions of the input and the output. Instead, a certain criterion function $F(X, \Theta)$ is minimized to solve for Θ in a least-square sense. In such a context, it might be very difficult to find $\frac{d\Theta}{dX}$ directly. If ideally the minimum of $F(x, \Theta)$ is 0, we can use the implicit method introduced by Ref. 6 to do the covariance propagation.

The basic idea of the implicit way is to find $\frac{d\Theta}{dX}$ indirectly from an implicit function $g(\Theta, X)$ which relates Θ and X to each other. If $\partial g/\partial\Theta$ and $\partial g/\partial X$ are computable, then

$$\frac{d\Theta}{dX} = \left(\frac{\partial g}{\partial\Theta}\right)^{-1}\frac{\partial g}{\partial X}\,.$$

Minimizing $F(X, \Theta)$ requires $\partial F(X, \Theta)/\partial\Theta$ equal to zero. Hence

$$g(X, \Theta) = \frac{\partial F(X, \Theta)}{\partial\Theta} = 0$$

is an desired implicit function relating Θ and X together.

To minimize $F(X, \Theta)$, the solution $\hat{\Theta} = \Theta + \Delta\Theta$ must be a zero of $g(\hat{X}, \hat{\Theta})$. Now taking a Taylor series expansion of g around (X, Θ) we obtain to the first-order approximation:

$$g(X + \Delta X, \Theta + \Delta\Theta) = g(X, \Theta) + \frac{\partial g}{\partial X}'(X, \Theta)\Delta X + \frac{\partial g}{\partial\Theta}'(X, \Theta)\Delta\Theta.$$

Because $\Theta + \Delta\Theta$ extremizes $F(X + \Delta X, \Theta + \Delta\Theta)$, $g(X + \Delta X, \Theta + \Delta\Theta) = 0$. Also, since Θ extremizes $F(X, \Theta)$, $g(X, \Theta) = 0$. Thus to a first-order approximation,

$$0 = \frac{\partial g}{\partial X}'(X, \Theta)\Delta X + \frac{\partial g}{\partial\Theta}'(X, \Theta)\Delta\Theta\,.$$

Since the relative extremum of F is a relative minimum, the matrix

$$\frac{\partial g}{\partial\Theta}(X, \Theta) = \frac{\partial F^2}{\partial^2\Theta}(X, \Theta)$$

must be positive definite for all (X, Θ). This implies that $\frac{\partial g}{\partial\Theta}(X, \Theta)$ is non-singular and have the inverse $(\frac{\partial g}{\partial\Theta})^{-1}$. Hence

$$\Delta\Theta = -\left(\frac{\partial g}{\partial\Theta}(X, \Theta)\right)^{-1}\frac{\partial g'}{\partial X}(X, \Theta)\Delta X$$

noting that $(\frac{\partial g}{\partial\Theta})^{-1}$ is symmetric. This expression relates to how the random perturbation ΔX on X propagates to the random perturbation $\Delta\Theta$ on Θ. If the expected value of ΔX, $E[\Delta X]$, is zero, then from this relation we see $E[\Delta\Theta]$ will also be zero, to a first-order approximation.

This relation permits us to calculate the covariance of the random perturbation $\Delta\Theta$

$$\begin{aligned}\Sigma_{\Delta\Theta} &= E[\Delta\Theta\Delta\Theta'] \\ &= \left(\frac{\partial g}{\partial\Theta}\right)^{-1}\frac{\partial g}{\partial X}'\Sigma_{\Delta X}\frac{\partial g}{\partial X}\left(\frac{\partial g}{\partial\Theta}\right)^{-1}.\end{aligned} \tag{B.2}$$

To the extent that the first-order approximation is good, (i.e. $E[\Delta\Theta] = 0$), it exists $\Sigma_{\hat{\Theta}} = \Sigma_{\Delta\Theta}$.

The way in which we have derived the covariance matrix for $\Delta\Theta$ based on the covariance matrix for ΔX requires that the matrices

$$\frac{\partial g}{\partial \Theta}(X,\Theta) \text{ and } \frac{\partial g}{\partial X}(X,\Theta)$$

be known. But the true values X and Θ are not observed. Only $X + \Delta X$ and $\Theta + \Delta\Theta$ are available. If we want to determine the estimate $\hat{\Sigma}_{\hat{\Theta}}$ for the covariance matrix $\Sigma_{\hat{\Theta}}$, we can proceed by expanding $g(X,\Theta)$ around $g(X+\Delta X, \Theta+\Delta\Theta)$.

$$\begin{aligned} g(X,\Theta) = g(X+\Delta X,\Theta+\Delta\Theta) &- \frac{\partial g}{\partial X}'(X+\Delta X,\Theta+\Delta\Theta)\Delta X \\ &- \frac{\partial g}{\partial \Theta}'(X+\Delta X,\Theta+\Delta\Theta)\Delta\Theta\,. \end{aligned}$$

Because $g(X,\Theta) = g(X+\Delta X,\Theta+\Delta\Theta) = 0$ and $\frac{\partial g}{\partial \Theta}'(X+\Delta X,\Theta+\Delta\Theta)$ is non-singular,

$$\Delta\Theta = -\left(\frac{\partial g}{\partial \Theta}(X+\Delta X,\Theta+\Delta\Theta)\right)^{-1}\frac{\partial g}{\partial X}'(X+\Delta X,\Theta+\Delta\Theta)\Delta X\,.$$

This motivates the estimator $\hat{\Sigma}_{\Delta\Theta}$ for $\Sigma_{\Delta\Theta}$ defined by

$$\hat{\Sigma}_{\Delta\Theta} = \left(\frac{\partial g}{\partial \Theta}(\hat{X},\hat{\Theta})\right)^{-1}\frac{\partial g'}{\partial X}(\hat{X},\hat{\Theta})\Sigma_{\Delta X}\frac{\partial g}{\partial X}(\hat{X},\hat{\Theta})\left(\frac{\partial g}{\partial \Theta}(\hat{X},\hat{\Theta})\right)^{-1} \qquad \text{(B.3)}$$

To the extent that the first-order approximation is good, $\hat{\Sigma}_{\hat{\Theta}} = \hat{\Sigma}_{\Delta\Theta}$.

REFERENCES

1. J. L. Barron, D. J. Fleet and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.* **12**, 1 (1994) 43–77.
2. J. L. Barron, S. S. Beauchemin and D. J. Fleet, "On optical flow," *Proc. Sixth Int. Conf. Artificial Intelligence and Information-Control Systems of Robots*, Smolenice Castle, Slovakia, September 12–16, 1994, pp. 3–14.
3. S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Comput. Surv.* **27**, 3 (1995) 433–467.
4. R. M. Haralick and J. S. Lee, "The facet approach to optic flow," *Proc. Image Understanding Workshop*, VA, 1983.
5. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, 1992.
6. R. M. Haralick, "Propagating covariance in computer vision," *Int. J. Pattern Recognition and Artificial Intelligence* **10**, 5 (1996) 561–572.
7. B. K. P. Horn and B. G. Schunck, "Determining optic flow," *Artif. Intell.* **17** (1981) 185–203.
8. J. K. Kearney, W. B. Thompson and D. L. Boley, "Optical flow estimation: an error analysis of gradient-based methods with local optimization," *IEEE Trans. Patt. Anal. Mach. Intell.* **9**, 2 (1987) 229–244.

9. H. H. Nagel, "On the estimation of optical flow: relations between different approaches and some new results," *Artif. Intell.* **33** (1984) 299–324.
10. E. P. Simoncelli, E. H. Adelson and D. J. Heeger, "Probability distributions of optical flow," *Proc. 1991 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Maui, HI, June 3–6, 1991, pp. 310–315.
11. Y. Xiong and S. A. Shafer, "Moment and Hypergeometric Filters for High Precision Computation of Focus, Stereo and Optical Flow," *Int. J. Comput. Vis.* **22**, 1 (1997) 25–59.
12. M. Ye and R. M. Haralick, "Image flow estimation using facet model and covariance propagation," *Vision Interface'98*, Vancouver, British Columbia, June 17–20, 1998, pp. 51–58.
13. M. Ye and R. M. Haralick, "Image flow estimation using facet model and covariance propagation," ISL Technical Report, 1998.
14. M. Ye and R. M. Haralick, "Synthesizing test image sequences: translating and rotating spheres," ISL Technical Report, 1998.

Ming Ye received the B.S.E.E. degree with honors from the University of Electronic Science and Technology of China in 1997. After that she joined the Intelligence System Laboratory of the University of Washington as a research assistant and obtained the M.S.E.E. degree in 1999. Now she is working for the Ph.D. degree in the University of Washington. She is a student member of IEEE.

Her research interests include computer vision, quantitative performance analysis, image processing and video technology. She is currently working on image noise characterization, image flow estimation and its performance evaluation through covariance propagation.

Robert M. Haralick occupies the Boeing Clairmont Egtvedt Professorship in the Department of Electrical Engineering at the University of Washington. Prof. Haralick is responsible for developing the gray-scale co-occurrence texture analysis technique and the facet model technique for image processing. He has developed shape analysis and extraction techniques using mathematical morphology, the morphological sampling theorem, and fast recursive morphology algorithms. In the area of document image understanding, Prof. Haralick, along with Prof. Ihsin Phillips, developed a comprehensive ground-truth set of over 1500 document images most in English and some in Japanese. He has also developed algorithms for document image skew angle estimation, zone delineation, word and text line bounding box delineation. His most recent research is in the area of computer vision performance characterization.

Prof. Haralick is a Fellow of IEEE for his contributions in computer vision and image processing and a Fellow of IAPR for his contributions in pattern recognition, image processing, and for service to IAPR. He has published over 470 papers and has just completed his term as the president of the International Association for Pattern Recognition.