EUROPEAN CONFERENCE ON COMPUTER VISION, pp. 124-134, ANTIBES, FRANCE, APRIL 1990

PARALLEL COMPUTATION OF OPTIC FLOW

Shaogang Gong
Computer Science Department
Queen Mary College, University of London
Mile End Road
London E1 4NS, UK

Michael Brady
Department of Engineering Scien e
Oxford University
Parks Road
Oxford OX1 3PJ, UK

Abstract

Both the tangential and normal components of the flow can be computed reliably where the image Hessian is well-conditioned. A fast algorithm to propagate flow along contours from such locations is proposed. Experimental results for an intrinsically parallel algorithm for computing the flow along zero-crossing contours are presented.

1 Introduction

An algorithm for the computation of optic flow should satisfy (at least) the following two conditions: (i) it should estimate flow vectors *accurately*, particularly if it is to be used to determine usable three-dimensional scene reconstructions; and (ii) it should provably and demonstrably compute flow vectors *fast*. As a result of research over the past decade or so, many methods have been proposed to compute optic flow. Most time and effort has been concentrated on achieving the first of the above properties: correctness and accuracy. Murray and Buxton (1989) forthcoming book, as well as Scott (1986) and Gong (1989b) survey previous work. Many algorithms operate on the basis of the motion constraint equation, which corresponds to a Taylor's series expansion of the image function $I(x,y,t)$ up to first order. In that case, the computation of optic flow is *under-constrained*, and additional smoothness constraints are imposed. These are implemented as local weighted averaging of flow estimates. Inevitably, the resulting algorithms are inherently *slow*, and this prevents them from being useful in practice.

In earlier work (Gong 1988; Gong 1989a) we proposed an additional constraint, which we call the *Curved Motion Constraint Equation*, to develop Hildreth's scheme (Hildreth 1984) for estimating the flow along zero-crossing contours. We restrict attention to zero crossings of a Laplacian operator. In that case, assume that: (i) the intensity function is spatio-temporally differentiable up to second order, and (ii) third order derivatives of the intensity function can be ignored. Then it can be shown that

$$(\ ^{\top}\mathbf{Hn})(\mathbf{t}^{\top}\mathbf{H}\mu) + (\ ^{\top}\mathbf{Hn})(\nabla I_t \cdot \mathbf{t}) + (\mathbf{t}^{\top}\mathbf{Ht})(\nabla I_t \cdot \mathbf{n}) \quad \frac{}{3}det(\mathbf{H})(\mathbf{n}\cdot\mu) = 0 \qquad (1)$$

By expanding μ in the local $(\mathbf{t},\mathbf{n})$ coordinate frame, we deduce that $(\mathbf{t}\cdot\mu)$ (and hence the full flow) can be estimated in a well-conditioned manner wherever $(\mathbf{t}^{\top}\mathbf{Hn})$ and $(\mathbf{t}^{\top}\mathbf{Ht})$ are significant. We extend Hildreth's scheme to take account of the initial well-conditioned estimates of the tangential flow. We minimise

$$\Theta = \int [(\frac{\partial \mu_x}{\partial s})^2 + (\frac{\partial \mu_y}{\partial s})^2] ds + \alpha \int [\mathbf{n} \cdot \mu - \mu^{\perp}]^2 ds + \beta \int [\mathbf{t} \cdot \mu - \mu^{\top}]^2 ds. \quad (2)$$

Here $\beta(s)$ is a Lagrange multiplier that expresses confidence in the tangential flow estimates $\mu^{\top}$, and is a function of the local Hessian matrix, $\beta = \frac{det\mathbf{H}}{\epsilon}$, where ϵ is the condition number of the Hessian matrix.

In this paper, we show how to adapt a mixed wave-diffusion algorithm (Scott, Turner and Zisserman 1988) to propagate quickly the full flow from known edge loci to all other edge locations at which only the normal component is known initially. We demonstrate the robust performance of an implemented algorithm on a number of real image sequences and compare its performance with Hildreth's scheme. We have designed a parallel implementation of our algorithm for a network of Transputers; but for reasons of space it will be reported elsewhere.

2 Propagating Flow

The time required by Hildreth's algorithm (and by the improved version using the Curved Motion Constraint Equation that we described above), is mostly spent on iterations of the conjugate gradient subroutine, which is inherently sequential. To overcome this problem, we have developed a novel, intrinsically local, propagation mechanism, which is described in this Section.

Even using the Curve Motion Constraint Equation, full flow estimates are still restricted to a few locations along image curves, and so the minimisation computation suggested by Hildreth is ill-posed. An additional smoothness constraint is required to guarantee that a minimum is reached. Inspired by (Yuille 1988), we develop a modified one-dimensional Motion Coherence Theory which captures the essence of our problem. In order to have a smooth flow field, the functional to be minimised in terms of both flow components $\mu(s)$ along a curve C can be formulated as:

$$E_{1D} = \int [M(\mu(s)) - \mathbf{M}_{\mu}(s)]^2 ds + \lambda \int \sum_{m=0}^{\infty} c_m (\frac{\partial^m \mu(s)}{\partial s^m})^2 ds \quad (3)$$

where $\lambda \geq 0$ is a Lagrange multiplier, the constants $c_m \geq 0$ weight the various derivative (smoothness) terms, and $\mathbf{M}_{\mu}(s)$ are the initial local measurements of the continuous flow field along a curve. (Note that in Yuille's original formulation, the data points $M(\mathbf{U}_i)$ are at sparse image locations.)

Equation (3) is reminiscent of Hildreth's scheme, which in fact, it generalises. In that case, $c_1 = 1$, and all other c_i are zero. Using the Curve Motion Constraint Equation, instead of $\mu(s) \cdot \mathbf{n}(s)$, as used by Hildreth, the local flow field measurement $M(\mu(s))$ is $\mu(s)$. For $c_1 = 1$, the Euler--Lagrange equation is:

$$\lambda(\frac{\partial^2 \mu(s)}{\partial s^2}) = [\mu(s) - \mathbf{M}_{\mu}(s)]$$

Now $\mathbf{M}_{\mu}(s)$ is the initial local measurement of the flow field, that is $\mu_0(s) = \mathbf{M}_{\mu}(s)$. Iterating,

$$\mu(s) - \mu_0(s) = d\mu(s) = \frac{\partial \mu(s,t)}{\partial t} \triangle t$$

As usual, we treat $\triangle t$ as a unit time interval between successive iterations of the procedure. It follows that the Euler-Lagrange equation of E_{cf} can be re-written as the diffusion equation:

$$\lambda(\frac{\partial^2 \mu(s,t)}{\partial s^2}) = \frac{\partial \mu(s,t)}{\partial t} \tag{4}$$

In a diffusion equation, the physical interpretation of λ has dimensions of length^2 $\times$ time^{-1}. Therefore, a combination of variables s and t with λ gives a dimensionless parameter: $\eta = s^2/\lambda t$. Then

$$\mu(s,t) = K \int_0^{\frac{1}{2\lambda}(st^{-1/2})} G_\sigma(\nu) d\nu \tag{5}$$

where $K = \sqrt{\pi}B$ is a constant. It follows that to construct the flow field μ along a curve by minimising the above functional is to propagate the initial local measurements of flow by a Gaussian interaction. The problem with diffusion processes is that they are too slow. This is clear from the above equation, which shows that that the flow field is a function of $st^{-1/2}$. That is, points along s at time t, such that $st^{-1/2}$ has a particular value, should all have the same flow vector. In other words, at any specific time t, there is a particular flow vector which has moved along the positive (as $0 \leq s \leq N$ where $N \geq 0$) curve direction, with a distance proportional to *the square root of the time*. This means that the speed of propagation of a flow vector is $c = \frac{\varphi}{\sqrt{t}}$, where φ is a constant corresponding to the strength of the flow vector being propagated. This speed decreases according to the root of the time and, since bigger φ means faster c, larger flow vectors propagate faster. On the other hand, as the diffusion progresses, the φ of flow vectors being propagated gets smaller as time t elapses.

To address this problem, (Scott, Turner and Zisserman 1988) have proposed the use of a mixed wave-diffusion process. They were primarily concerned with the computation of symmetry sets (such as the symmetric axis transform, and smoothed local symmetries) for two-dimensional visual shape representations. We show that their technique can be adapted to propagate quickly flow between locations at which the curved motion constraint equation is well-conditioned.

A wave propagates information at some constant speed C. The wave equation needs boundary conditions that specify $\mu(s,0)$ and its velocity $\partial\mu(s,0)/\partial t = \Psi(s)$ (say) at time $t = 0$. In our case, $\mu(s,0) = \mathbf{M}_\mu(s) = \mu_0(s)$, and $\Psi(s) = 0$. The (d'Alembert) solution to the wave equation is given by:

$$\mu(s,t) = \frac{1}{2}[\mu_0(s - Ct) + \mu_0(s + Ct)] + \frac{1}{2C}\int_{x-Ct}^{x+Ct} \Psi(\xi) d\xi$$

which, in our case reduces to:

$$\mu(s,t) = \frac{1}{2}[\mu_0(s - Ct) + \mu_0(s + Ct)] \tag{6}$$

This means that the every flow vector that is measured initially is propagated in both directions along the curve, each at half the strength.

Theoretically, this is a very attractive property as the speed of the wave can be chosen to be much faster than the diffusion equation. However, a wave equation itself does not impose any

constraint through the process and, furthermore, numerically, there is another problem associated with wave which can be seen from the following form of equation (6):

$$\begin{aligned} \mu(s,t+1)' &= \mu(s,t)' + C^2 \frac{\partial^2 \mu(s,t)}{\partial s^2} \\ \mu(s,t+1) &= \mu(s,t)' + \mu(s,t) \\ (t &= 0,1,2,\ldots,\infty) \end{aligned} \tag{7}$$

Here $\mu(s,t)$ is the wave displacement, $\mu(s,t)'$ and $\mu(s,t+1)'$ represent the velocities for the wave equation at time t and $t+1$ at a curve location s. Because of quantisation errors and noise in the digital image, the second order partial derivatives $\frac{\partial^2 u_i}{\partial s^2}$ are unstable. This causes unpredictable pulses in the velocity, which are then propagated. Therefore, propagation according to a wave equation is unstable for digitised images unless the chosen propagation speed C is *small*. Smaller C compensates for errors in the computation of the partial derivatives.

The instability of wave processing comes from second order spatial differentiation and furthermore, this second order differentiation only affects non-linear changes in the displacement. Thus, in order to reduce the sensitivity of the velocity computation, we smooth any *large* non-linear changes in the displacement caused by noise. From the earlier discussion of the diffusion equation, it is clear that diffusion is well suited to that task. Following (Scott, Turner and Zisserman 1988), we propose a combined procedure such that every iteration of a wave propagation is followed by an iteration of a diffusion process. This combined procedure has the following desirable features:

1. It quickly damps out the violent non-linear changes that are most likely to be caused by quantisation errors and noise in the displacement. This stablises the propagation process, and enables a reasonably fast wave speed to be used. On the other hand, a faster wave speed implies more smoothing between each successive iteration of the propagation.

2. It imposes a consistent smoothing on the flow vectors through the propagation. Although this side-effect of smoothing the flow vectors does not apply any constraint explicitly and it will not constrain the flow field to be interpolated under the smoothness constraint, the flow field is modified towards the desired distribution. It reduces the task of smoothing in the next step to be carried out by a diffusion.

3. It propagates a weaker influence from any source if the distance along the curve from this source is greater. This is certainly an improvement over a pure wave propagation which would propagate the same strength of data, half of the original, irrespective of the distance from the source. Actually, this is another way to see that the combined procedure carries out a weak smoothing interpolation.

The combined wave-diffusion propagation does not impose a smoothness constraint or a Gaussian interaction between the flow vectors. With fixed boundary values, it will carry on indefinitely. We need to stop the wave-diffusion propagation when the data at the two ends of the curve reach

each other simultaneously. For a pure wave, the propagation speed is C. For a modified wave--diffusion, which smooths the flow while it is propagating values, the speed of propagation is faster than C. From experiments, we find that the propagation speed and the time at which to stop the processing can be approximated heuristically by:

$$C_{wd} = C + \frac{1}{log_{10}l} \text{ and } T_{wd} = \frac{l}{C + 1/log_{10}l} \tag{8}$$

where l is the length of the image curve.

3 Experimental Evaluation

In order to evaluate our algorithm, we have tested it on dozens of image sequences of real moving objects. We compare optic flows computed by our algorithm (curved motion constraint extension addition to Hildreth, plus mixed wave-diffusion propagation) with those computed by a reimplementation of Hildreth's method. First, we estimate the optic flow for two toy cars moving in the same direction, where the larger one moves with greater speed. This motion takes place parallel to the image plane (see (a) in figure 1).

It is clear from (b) and (d) in figure 1 that the derivative based local computations are noise sensitive, especially for the second order based local tangential flow. Median filtering is crucial for the interpolation of the tangential flow (see (c) and (e) in figure 1). The flow fields shown in (f) and (g) of figure 1 are computed respectively by our scheme and by Hildreth's. Both schemes detect the greater speed of the larger car, but Hildreth's scheme imposes stronger smoothness on the flow field. As a result, it computes more accurate magnitudes of the flow vectors in this case, but it is also more noise sensitive. It also makes errors at the points indicated by rectangles in the figure.

As a second example, in figure 2, the optic flow estimated for a hand moving in the image plane is given. The hand moves from the bottom line towards the top right corner of the image plane, while at the same time, the fingers spread out slightly.

Both schemes recover the flow vectors at the finger tips, where the motions are parallel to the edges (aperture problem). But notice that Hildreth's scheme propagates the flow field across the edge that joins the second finger and a texture line in the background (indicated by a rectangle). On the other hand, our scheme does not, and is more desirable for tasks such as flow based object segmentation. Again, Hildreth's scheme computes erroneous flow at points such as that indicated by a rectangle at the top right corner in (g).

In the third example, two static coffee mugs whose surfaces were marked with features were taken by a moving camera that moved from the left to the middle in the image plane coordinate (see figure 3).

Both schemes demonstrate an ability to recover smoothed flow fields, but Hildreth's scheme smooths the flow field more than ours. As a whole, the optic flows computed by both schemes are similar. Our scheme is, however, considerably faster than Hildreth's.

Finally, the optic flow of a hand moving in depth is estimated (see figure 4). In addition, the moving object is brightly lit, so that the image is much noisier. This is shown in the locally

computed normal and tangential flow in (b) and (d) of figure 4. Without median filtering, some parts of the flow field computed by Hildreth's scheme are unstable and render the entire flow field almost useless.

For comparatively simple scenes giving rise to few edges, our scheme is about 3 times faster than Hildreth's (two cars moving in the image plane, a hand moving in the image plane, and a hand moving against the distinguished background). However, as the scene becomes more complex (more and longer edges in the image), the relative advantage of our scheme increases. This is strongly demonstrated by the results from coffee mug's ego-motion, an approaching hand, and the hand rotating in depth.

References

S.G. Gong, 1988 (September). Improved Local Flow. In *Alvey Vision Conference*, pages 129–134, University of Manchester, Manchester, England.

S.G. Gong, 1989a (March). Curve Motion Constraint Equation and its Applications in the Parallel Visual Motion Estimation. In *IEEE Workshop on Visual Motion*, University of California, Irvine, California.

S.G. Gong, 1989b (September). *Parallel Computation of Visual Motion.* PhD thesis, Departmant of Engineering Science, Oxford University.

E.C. Hildreth, 1984. *The Measurement of Visual Motion.* MIT Press, Cambridge, Massachusetts.

D.W. Murray and B.F. Buxton, 1989. *Experiments in the Machine Interpretation of Visual Motion.* MIT Press, Cambridge, Massachusetts.

G.L Scott, 1986. *Local and Global Interpretation of Moving Images.* PhD thesis, Cognitive Studies Program, University of Sussex, England.

G.L. Scott, S. Turner and A. Zisserman, 1988 (September). Using a Mixed Wave/Diffusion Process to Elicit the Symmetry Set. In *Alvey Vision Conference*, University of Manchester, Manchester, England.

A. Yuille, 1988 (December). A Motion Coherence Theory. In *IEEE International Conference on Computer Vision*, Tampa, Florida.

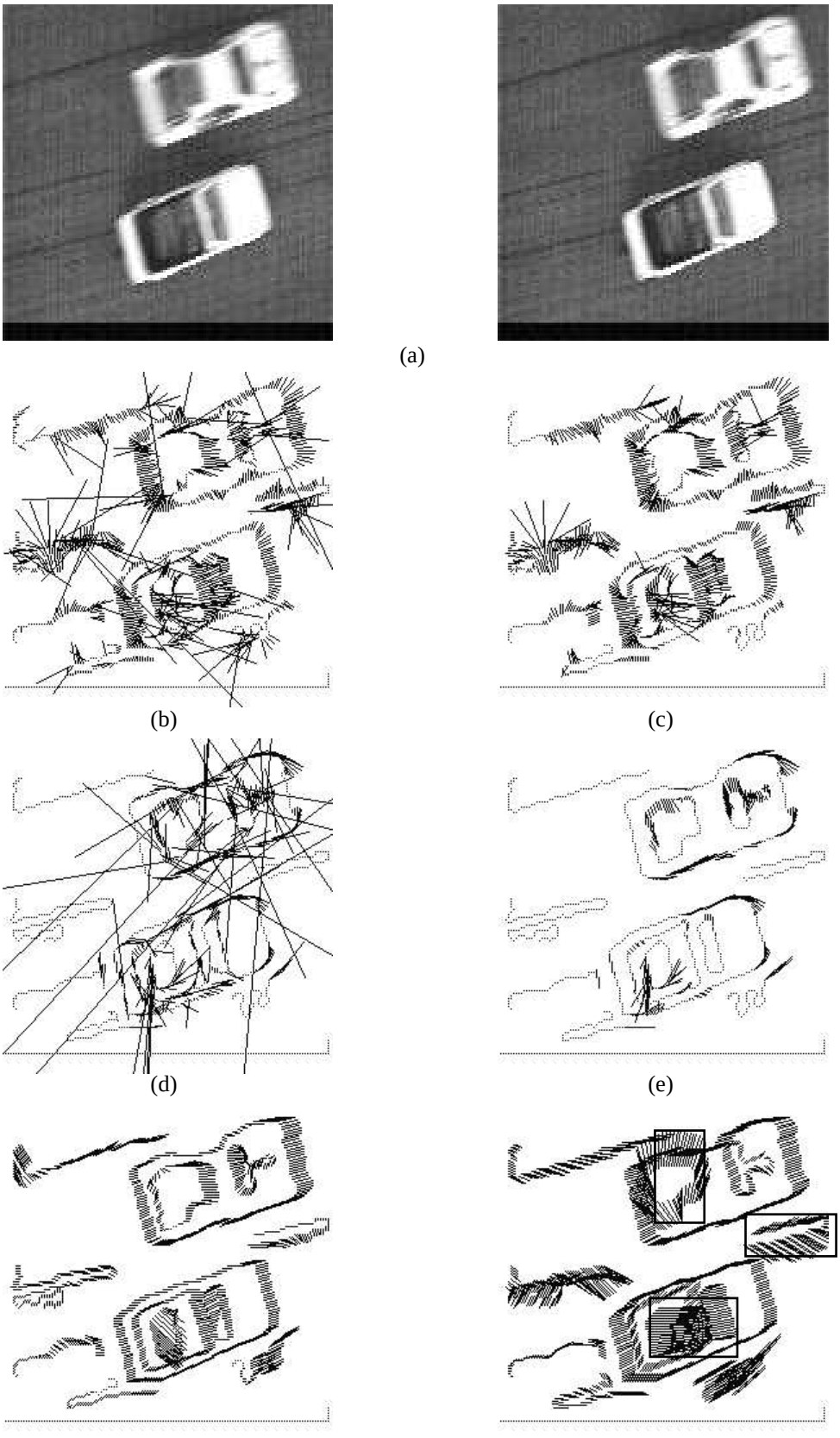
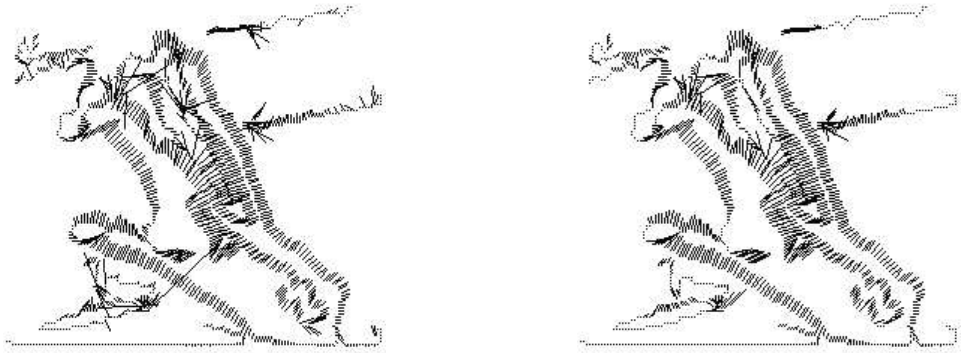
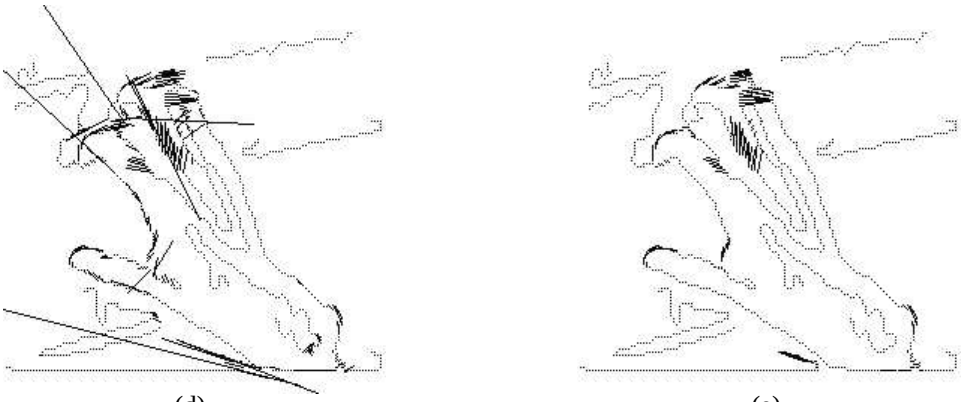


Figure 1: *(a) Two cars move in the image plane. (b) Normal flow before median filtering. (c) Median filtered normal flow. (d) Tangential flow before median filtering. (e) Median filtered tangential flow. (f) Diffusion smoothed full flow. (g) The associated full flow from Hildreth's scheme.*

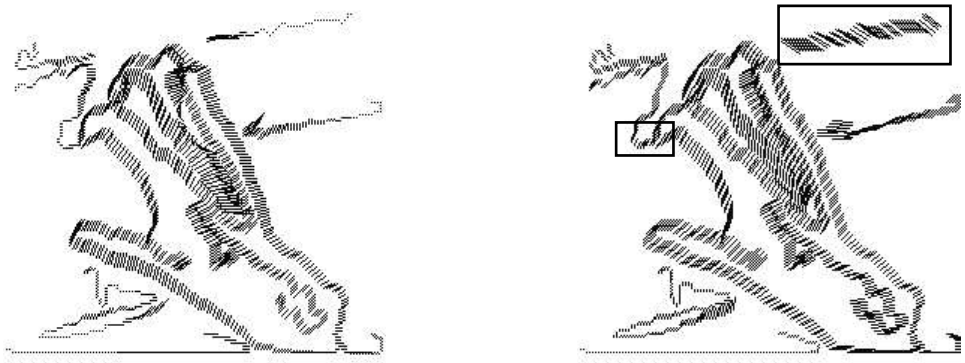
(a)



(b) (c)



(d) (e)



(f) (g)

Figure 2: *(a) A hand moves in the image plane. (b) Normal flow before median filtering. (c) Median filtered normal flow. (d) Tangential flow before median filtering. (e) Median filtered tangential flow. (f) Diffusion smoothed full flow. (g) The associated full flow from Hildreth's scheme.*

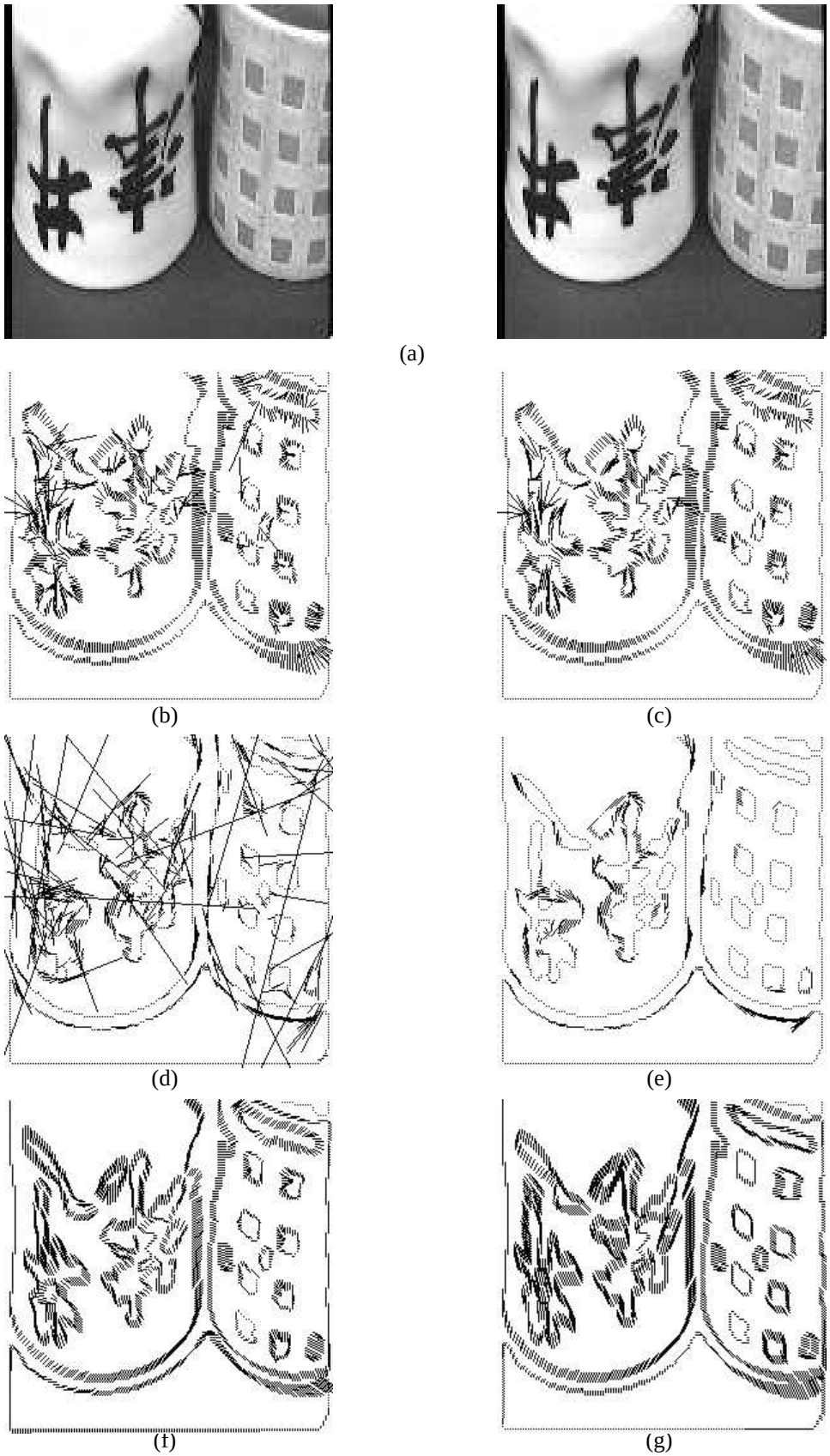


Figure 3: *(a) Ego-motion of cups. (b) Normal flow before median filtering. (c) Median filtered normal flow. (d) Tangential flow before median filtering. (e) Median filtered tangential flow. (f) Diffusion smoothed full flow. (g) The associated full flow from Hildreth's scheme.*

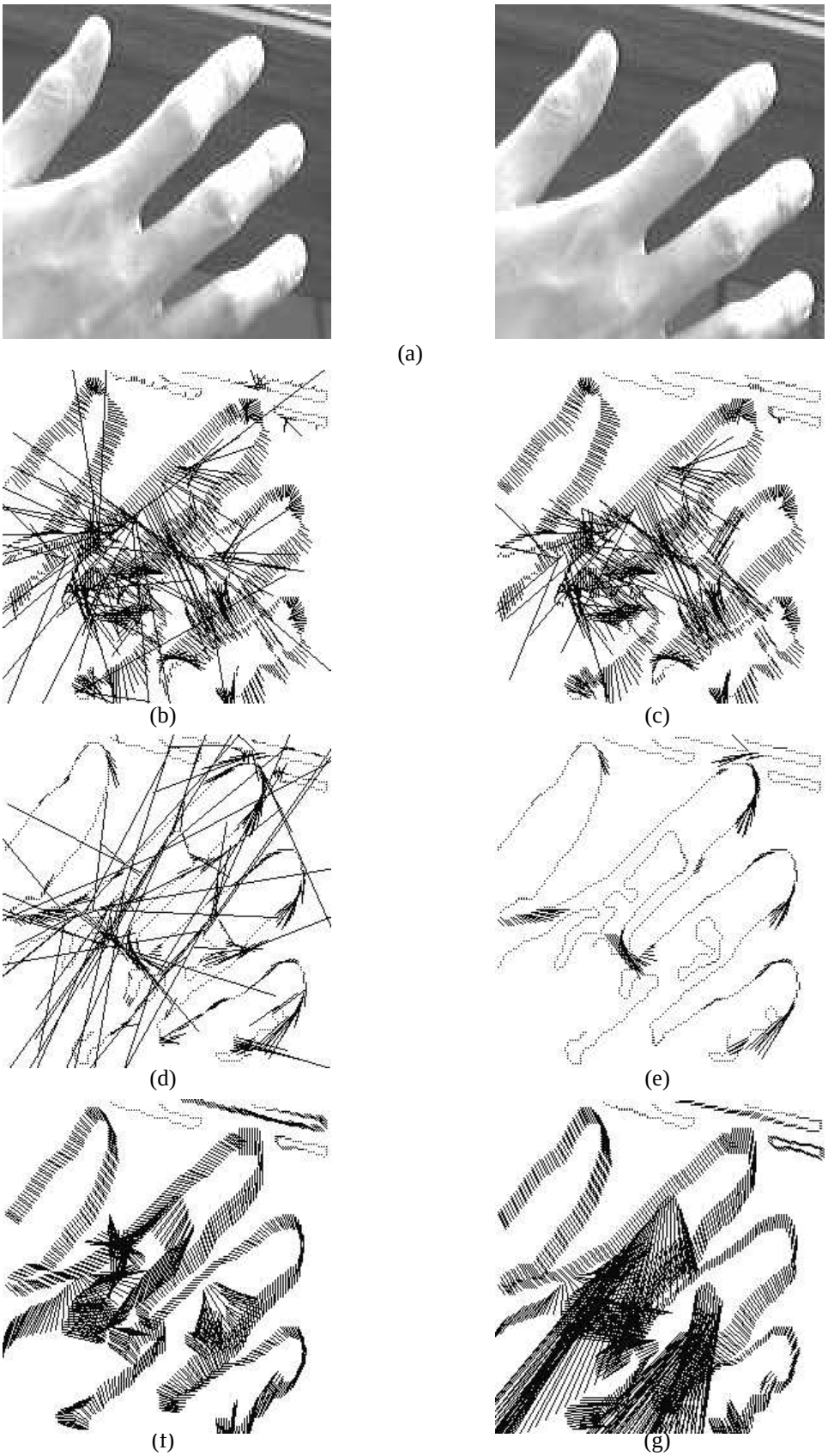


Figure 4: *(a) A hand approaches the viewer. (b) Normal flow before median filtering. (c) Median filtered normal flow. (d) Tangential flow before median filtering. (e) Median filtered tangential flow. (f) Diffusion smoothed full flow. (g) The associated full flow from Hildreth's scheme.*