Robust Computation Of Optical Flow

Ee-Ping Ong and Michael Spann
School of Electronic and Electrical Engineering
University of Birmingham
Birmingham B15 2TT, U.K.

Abstract

This paper presents an algorithm to compute optical flow accurately at motion discontinuities and occlusion regions based on a robust estimator (the Least-Median-Squares estimator). The motion constraint equation and the 2-D affine motion model are used to compute the optical flow in a local neighbourhood. The use of the least-median-squares robust estimator enables points where optical flow cannot be computed to be rejected as outliers rather than assigning erroneous flow to such points. In addition, the use of an overlapping neighbourhood strategy eliminates the block-effects that are commonly faced in local differential methods for computing optical flow. The algorithm is also able to deal with cases of the local neighbourhood straddling regions of three motions. Results for both synthetic and real image sequences are presented.

1. Introduction

Optical flow is the dense velocity field in the image plane due to the projection of moving patterns in the perceived scene onto the image plane. These moving patterns arise because of the relative motion between moving objects in the scene and the observer. The moving patterns will appear as instantaneous changes in brightness values in the image. Optical flow has already found potential in many different applications, such as motion segmentation, object tracking, image compression, structure from motion and robot navigation.

Optical flow cannot be computed locally at a point in the image independent of neighbouring points since the flow at any single point has two motion components, whereas the instantaneous changes in the brightness at a point yields only one measurement. A number of different methods for computing optical flow have been proposed: those based on differential, correlation, energy, and phased-based techniques [1]. Methods based on the differential technique for computing optical flow can be classified into local and global methods. In local methods, optical flow is computed locally in a neighbourhood using some forms of regression minimisations, such as least-squares minimisation [8], weighted least-squares [1], and the M-estimator [11]. In global methods, optical flow is computed based on global smoothness constraints by minimising a regularisation function defined over the entire image [7]. To overcome the problems faced in computing optical flow at motion discontinuities and occlusion regions, different approaches have been proposed. Heitz and Bouthemy [6] use both gradient-based and feature-based motion constraints based on Markov random fields. Schunck [13] uses clustering of local gradient-based constraints and surface-based smoothing. Odobez and Bouthemy [11] use an M-estimator based on an affine motion model of the gradient-based motion constraint equation.

BMVC 1995 doi:10.5244/C.9.57

In this paper, a local method for computing optical flow is adopted and the flow vectors have been computed using an affine model of the flow based on the motion constraint equation and least-median-squares regression. Overlapping neighbourhoods are used and each neighbourhood is allowed to be shifted from its nominal location by a few pixels to maximise the number of inliers that each of the neighbourhood can cover. The optimum motion parameters at a point is selected to be the parameters which corresponds to the smallest robust scale estimate of the model-fitting among all the overlapping neighbourhoods. Points which are outliers to these overlapping neighbourhoods are rejected and no optical flow is computed. The use of overlapping neighbourhoods eliminates the block-effects commonly seen in local computation of the optical flow. The shifting of the neighbourhood to maximise the number of inliers also makes the motion estimates more robust when the neighbourhood straddles regions of three motions or when it straddles regions of two motions with equal numbers of supporting data points.

The organisation of this paper is as follows: Section 2 gives an introduction to the computation of optical flow and the regression method. Section 3 gives a detailed description of the algorithm employed in this paper and section 4 presents the results for both synthetic and real image sequences. Section 5 gives conclusions for the presented algorithm and some possible extensions to the current work.

2. Computation of optical flow

The local differential method computes the motion parameters of the optical flow by minimising the sum of the model errors using some means of statistical regression within a local neighbourhood in the image. The regression methods that have been used for computing optical flow include classical least-squares [8], weighted least-squares [1], and M-estimators [11].

2.1. Regression methods

Regression methods are used to fit a model to given data. The classical linear model assumes a relationship of the form:

$$y_i = \mathbf{x}_i^T \Theta + e_i, \quad \text{for } i = 1,..., n, \tag{1}$$

where $\mathbf{x}_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{ip})^T$ are the explanatory variables, $\Theta = (\theta_1, \theta_2, \ldots, \theta_p)^T$ are the parameters of the postulated model, y_i is the observation, and e_i is the error term. The aim is to estimate the unknown parameters, Θ. Applying a regression estimator to the data yields the estimated parameters, $\hat{\Theta} = (\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_p)^T$. The estimated value of y_i is then given by: $\hat{y}_i = \mathbf{x}_i^T \hat{\Theta}$. The residual of the i-th observation is: $r_i = y_i - \hat{y}_i$.

Classical least-squares (LS) estimates the regression parameters by solving the following minimisation problem [4]:

$$\hat{\Theta} = \arg \min_{\Theta} \sum_i r_i(\Theta)^2 \tag{2}$$

The LS method achieves optimum results when the underlying error follows a Gaussian distribution. The method becomes unreliable when the error distribution is non-Gaussian or if outliers (data with values far from the local trend) are present [12]. Thus, LS has a breakdown point of 0%, which means that algorithms that are based on LS cannot tolerate the presence of a single outlier.

The M-estimators estimate the regression parameters by solving [12]:

$$\hat{\Theta} = \arg\min_{\Theta} \sum_i \rho(r_i(\Theta)) \tag{3}$$

where $\rho(r_i(\Theta))$ is a symmetric positive-definite function of the residuals $r_i(\Theta)$ with a unique minimum at zero. Different ρ functions have been proposed in the statistical literature in order to reduce the influence of large residual values on the estimated fit using the postulated model. However, the reliability of the initial guess is usually important. The least-squares method can be regarded as a special case of the M-estimators with $\rho(r_i(\Theta)) = r_i(\Theta)^2$. M-estimators are more robust to outliers than LS estimator with a breakdown point of less than $1/(p+1)$ [9], where p is the number of parameters in the regression model.

The least-median-squares (LMS) robust regression method estimates the regression parameters by solving the following non-linear minimisation problem [12]:

$$\hat{\Theta} = \arg\min_{\Theta} \underset{i}{med}\, r_i(\Theta)^2 \tag{4}$$

Thus, this estimator must yield the smallest value for the median of the squared residuals computed for the entire data set. The LMS method achieves a 50% breakdown point, which means that it can tolerate up to 50% of outliers in the data. The LMS only has a relative efficiency[1] of 0.637, and the time complexity is very high [12]. The relative efficiency of the LMS can be improved by combining it with a weighted least-squares step and the time complexity can be reduced by utilising a Monte-Carlo type speed-up technique. The LMS regression has recently been applied in surface reconstruction [14] and the recovery of unknown epipolar geometry [15].

3. Description Of Algorithm

3.1. Least-Median-Squares (LMS) Motion Estimator

The computation of the optical flow is based on the motion constraint equation [7]:

$$\nabla \mathbf{I}(\mathbf{x},t)^T \mathbf{u}(\mathbf{x},t) + I_t(\mathbf{x},t) = 0 \tag{5}$$

where $I(\mathbf{x},t)$ is the brightness function, $\nabla \mathbf{I}(\mathbf{x},t) = (I_x(\mathbf{x},t), I_y(\mathbf{x},t))^T$, $I_x(\mathbf{x},t)$, $I_y(\mathbf{x},t)$ and $I_t(\mathbf{x},t)$ are the partial derivatives of $I(\mathbf{x},t)$ with respect to space and time at the point $(\mathbf{x},t)$; $\mathbf{u} = (u,v)^T$ is the flow vector with flow velocities, u and v, in both x and y directions, and $\mathbf{x} = (x,y)$. The constraint equation models the interaction between the image velocities (u,v) and the image brightness $I(\mathbf{x},t)$.

The flow field, $\mathbf{u}$, can be modelled by a 2-D polynomial motion model which may be a constant model, an affine model, or a quadratic model [11]. An affine motion model of the flow field is defined as:

$$\mathbf{u}(\mathbf{x}) = \mathbf{A}\mathbf{x}^T + \mathbf{b} \tag{6}$$

where $\mathbf{A} = \begin{pmatrix} a_0 & a_1 \\ a_2 & a_3 \end{pmatrix}$, $\mathbf{b} = (u_0, v_0)^T$. The time dependency has been dropped for simplicity of notation where no confusion will arise. An affine motion model, which is a

[1]Relative efficiency is defined as the ratio between the lowest achievable variance for the estimated parameters and the actual variance provided by the given regression method.

first order approximation of the motion field, has been shown to be sufficient to describe complex motions such as expansion, rotation and shear [3]. The affine motion model has been adopted by Bouthemy and Francois [2], and Goh and Martin [5] for motion segmentation. Meyer and Bouthemy [10] have also employed this model for tracking regions in long image sequences.

Let the error (or residual) in modelling the motion field using an affine motion model at a point $\mathbf{x}$ be $r(\mathbf{x})$, where :

$$r(\mathbf{x}) = \nabla \mathbf{I}(\mathbf{x},t)^T \mathbf{u}(\mathbf{x},t) + I_t(\mathbf{x},t) \tag{7}$$

The above equation can be rewritten as:

$$r(\mathbf{x}) = \mathbf{a}(\mathbf{x})^T \Theta - y(\mathbf{x}) \tag{8}$$

where $\mathbf{a}(\mathbf{x}) = (I_x, xI_x, yI_x, I_y, xI_y, yI_y)^T$, $\Theta = (u_o, a_0, a_1, v_0, a_2, a_3)^T$, $y(\mathbf{x}) = -I_t(\mathbf{x})$.

The motion parameters are computed by using the least-median-squares (LMS) robust regression method [12], which minimises the median of the squared-residuals:

$$\hat{\Theta} = \arg\min_{\Theta} \underset{\mathbf{x}}{med}\, r(\mathbf{x},\Theta)^2 \tag{9}$$

where $\mathbf{x} \in R$, R being the neighbourhood for computing the optical flow.
Since it is impossible to write down a single formula for the LMS estimator, the algorithm is presented as follows.

Given a data set of n observations (i.e. n points in the neighbourhood), the algorithm repeatedly draws m sub-samples each of p different observations from the data set using a Monte-Carlo type technique, where p is the number of parameters in the model (p = 6 for the affine motion model). For each sub-sample, indexed by J, the corresponding parameters (denoted by $\hat{\Theta}_J$) are estimated from the p observations. In addition, the median of the squared residuals, denoted by M_J, with respect to the entire data set is also determined, where:

$$M_J = \underset{\mathbf{x}}{med}\, r(\mathbf{x},\hat{\Theta}_J)^2 \tag{10}$$

The LMS solution is the $\hat{\Theta}$ for which the corresponding M_J is the minimum among all the m different M_Js. Although the maximum number of sub-samples that can be chosen is C_p^n in order to get a globally optimum solution, it is computationally infeasible when n and p are large. In such cases, m can be chosen such that the probability that at least one of the m sub-samples will consist of p good observations is almost 1. The probability that at least one of the m sub-samples consist of p good observations is given by:

$$P = 1 - [1 - (1-\varepsilon)^p]^m \tag{11}$$

where ε is the fraction of outliers that may be present in the data.

Since the LMS relative efficiency is poor in the presence of Gaussian noise, a single-step weighted least-squares procedure is performed based on the estimates of the LMS step. First, an initial scale estimate is computed by:

$$\sigma^0 = 1.4826\left(1 + \frac{5}{n-p}\right)\sqrt{\underset{\mathbf{x}}{med}\, r(\mathbf{x},\hat{\Theta})^2} \tag{12}$$

where the factor 1.4826 is introduced for consistent estimation of σ^0 in the presence of Gaussian noise, and the term $5/(n-p)$ is introduced as a finite sample correction to improve the estimate when the sample size is small [12]. The initial scale estimate is used to determine an initial weight $w(\mathbf{x})$ for each of the n observations, where:

$$w(\mathbf{x}) = \begin{cases} 1 & \text{if} \left| r(\mathbf{x}, \hat{\Theta}) / \sigma^0 \right| \le 2.5 \\ 0 & \text{if} \left| r(\mathbf{x}, \hat{\Theta}) / \sigma^0 \right| > 2.5 \end{cases} \tag{13}$$

These initial weights are then used to compute the final robust scale estimate :

$$\hat{\sigma} = \sqrt{(\sum_{\mathbf{x}} w(\mathbf{x}) r(\mathbf{x}, \hat{\Theta})^2) / (\sum_{\mathbf{x}} w(\mathbf{x}) - p)} \tag{14}$$

The final weights are computed using equation (13) by replacing the initial scale estimate, σ^0, by the final scale estimate, $\hat{\sigma}$. The motion parameters are then estimated using the weighted least-squares procedure:

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{\mathbf{x}} w(\mathbf{x}) r(\mathbf{x}, \Theta)^2 \tag{15}$$

The final parameters, $\hat{\Theta}$, are obtained by solving [4]:

$$\hat{\Theta} = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{y} \tag{16}$$

where $\mathbf{X} = (\mathbf{a}(\mathbf{x}_1), \mathbf{a}(\mathbf{x}_2), \ldots, \mathbf{a}(\mathbf{x}_n))^T$, $\quad \mathbf{x}_i = (x_i, y_i)$, for $i = 1, 2, \ldots, n$,

$\mathbf{W} = \text{diag}\,(w(\mathbf{x}_1), w(\mathbf{x}_2), \ldots, w(\mathbf{x}_n))$, $\quad \mathbf{y} = (y(\mathbf{x}_1), y(\mathbf{x}_2), \ldots, y(\mathbf{x}_n))^T$.

3.2. Implementation Strategy

The above-mentioned LMS motion estimator is used to compute the optical flow in each local neighbourhood from 2 successive frames of images. P is chosen to be 0.95, and ε is 0.5 such that a 50% of outliers can be tolerated in the data set. For an affine model, $p = 6$, and hence the minimum number of sub-samples required is 191 in order to satisfy equation (11). Each sub-sample of 6 points is chosen as follows:

(1) Pick a point randomly in order to centre a window for picking the 6 points.

(2) Open a window of s by s pixels (6x6 is used throughout the experiments) with respect to the point initially picked and then randomly pick 6 points from within the window to compute the affine motion parameters. The 6 points chosen must have spatial-temporal gradient whose magnitudes are in the top three-quarter of the region.

Note that for each sub-sample, 6 points are randomly picked from a smaller support region rather than from the entire neighbourhood because it is believed that the points for computing the optical flow should be close together, yet from a large enough area to tolerate the presence of outliers such that there is a higher chance of obtaining a sub-sample which contains only good points. In addition, points with higher gradient magnitudes are used because points with zero or low gradient magnitudes do not convey motion information. The partial derivatives are computed by first differences in a 2x2 spatial neighbourhood and temporally over two frames as in Horn and Schunck [7].

The neighbourhoods are derived by dividing the image into overlapping blocks, each block overlapping its immediate neighbours by a certain amount (at least 50%). The overlapping neighbourhood strategy eliminates the block-effects commonly faced in

local methods for computing the optical flow. The block-effects prevents accurate estimation of the flow around the boundary regions of the moving objects.

The motion parameters are also computed for each block shifted from its nominal location by half of the distance between adjacent neighbourhoods in both x and y directions. The final motion parameters for each block is the parameters which contains the maximum number of inliers. This step will prevent inaccurate parameter estimates when the block straddles regions containing three motions and none of the motions make up 50% of the points in the block. This effect can be seen in Figure 1 where the original neighbourhood location encounters such a situation and the shifted neighbourhood location causes more than 50% of the points in the neighbourhood to correspond to the motion on the right sphere and hence solves the ambiguous problem. In addition, this step also prevents inaccurate parameter estimates when the block straddles regions containing 2 motions with the same numbers of supporting points for each motion.

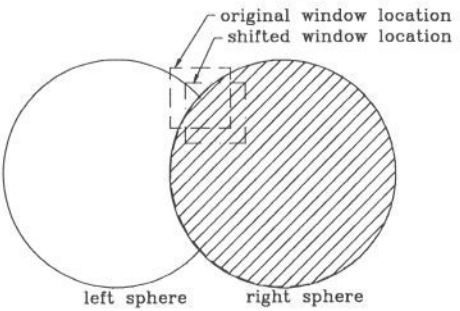

Figure 1: Effect of shifting the location of the neighbourhood for computing the optical flow

With the overlapping neighbourhood strategy, each point in the image may have no associated motion parameters when it is considered as an outlier to all the overlapping blocks (this is a point where optical flow cannot be computed, normally corresponding to regions of occlusion), or it may have more than one set of estimated motion parameters. The motion parameters corresponding to the smallest robust scale estimate will then be assigned to each point. These motion parameters are used to compute the flow velocities.

4. Results

4.1. Synthetic Images

Figure 2(a) shows the first frame of the computer-generated spheres sequence (Each frame of the sequence is 128x128 pixels). The spheres and the background are textured by adding sinusoidally-varying greyscale of different frequencies. The left sphere is translating to the left, and the right sphere is diverging outwards. Only two frames are used for computing the optical flow, and no smoothing is performed prior to computing the flow. Figure 2(b) shows the needle diagram (sub-sampled at an interval of 8 pixels) of the optical flow computed using the LMS motion estimator. An 8x8 pixel local neighbourhood is used to compute the optical flow, and the neighbourhood is allowed a maximum movement of 2 pixels in both x and y directions. Each of the local neighbourhood overlaps its neighbours by 50%. Figure 2(c) shows the confidence image of the optical flow computed using the LMS motion estimator. A greyscale of zero

(black) indicates that no optical flow can be computed. It can be seen in Figure 2(b) that the flow has been computed accurately at the motion boundaries of the two occluding spheres, while Figure 2(c) shows that the motion discontinuities and occlusion regions (points in black) have been successively isolated. Note that at the top and bottom of the regions where the right sphere occludes the left sphere, the local neighbourhood for computing the optical flow encounters the situations where it straddles three types of motions (one translating to the left, one diverging, and one stationary). However, the algorithm has successfully computed the flow for such regions. For comparison, the optical flow computed using the least-squares (LS) motion estimator over 8x8 pixel non-overlapping neighbourhoods is shown in Figure 2(d) (this needle diagram is shown at the same scale as the needle diagram for the optical flow computed using the LMS estimator). It is obvious that errors have occurred at the motion boundaries of the spheres. Figure 2(e) shows the magnitude (represented as greyscale) of the flow computed using the LMS motion estimator (This image has been pre-processed with histogram-stretching to provide better visual effects). It can be seen that block-effects, commonly faced in local methods for computing the optical flow, has been eliminated. Figure 2(f) shows the corresponding magnitude (represented as greyscale) of the flow computed using the LS motion estimator (This image has also been histogram-stretched). The block-effects can be distinctly observed.

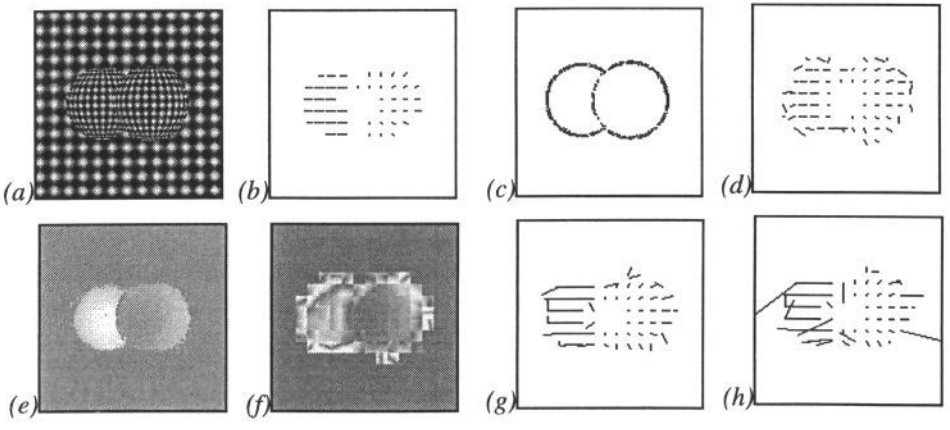

Figure 2: (a) First frame of the spheres sequence. (b) Needle diagram of the optical flow computed using LMS motion estimator. (c) Confidence image of the LMS motion estimator (A greyscale of zero (black) indicates that optical flow cannot be computed). (d) Needle diagram of the optical flow computed using LS motion estimator. (e) Magnitude (represented as histogram-stretched greyscale) of the flow computed using LMS motion estimator. (f) Magnitude (represented as histogram-stretched greyscale) of the flow computed using LS motion estimator. (g) Needle diagram of the optical flow computed using Lucas and Kanade method without pre-smoothing and thresholding. (h) Needle diagram of the optical flow computed using modified Lucas and Kanade method without pre-smoothing and thresholding.

In [1], Barron et. al. have examined the performance of several different methods for computing the optical flow and they have found that the local differential method of

Lucas and Kanade and the phase-based method of Fleet and Jepson have the best performance. The optical flow computed using Lucas and Kanade method [1] will be used for comparison with the method presented in this paper. Figure 2(g) shows the needle diagram of the optical flow computed using the Lucas and Kanade method without pre-smoothing of the images before the computation of the optical flow and also without thresholding based on the eigenvalues of the moment matrix, while Figure 2(h) shows the corresponding needle diagram of the optical flow computed using the modified Lucas and Kanade method without pre-smoothing and also without thresholding. It can be seen that errors have occurred at the motion boundaries of the spheres for the two needle diagrams. The optical flow for the case with pre-smoothing and thresholding is similar to that without pre-smoothing and thresholding. (See [1] for details on both the original and modified Lucas and Kanade method).

4.2. Real Images

The first frame of the real image sequence (rubic-cube sequence) used in this experiment is shown in Figure 3(a). Each frame of the sequence is 256x240 pixels. Unless specified, no smoothing is performed on the images prior to the computation of the optical flow. Figure 3(b) shows the needle diagram of the optical flow (sub-sampled at an interval of 8 pixels) using the LMS motion estimator. A local neighbourhood of 16x16 pixels is used to compute the optical flow, and the neighbourhood is allowed a maximum shift of 4 pixels in both x and y directions. Each local neighbourhood overlaps its neighbours by 50%. It can be seen from Figure 3(b) that reasonable qualitative estimation of the optical flow has been obtained on the rotating cube and the side of the turntable. However, some motion vectors have occurred in the stationary background. This is because this image sequence is very noisy, whereas the motion constraint equation is inherently sensitive to noise. Figure 3(c) shows the confidence image of the LMS motion estimator. Points with greyscale of zero (black) indicates that the optical flow cannot be computed. Note that the white surface of the turntable has been represented by black region in the confidence image. This is due to the greyscale of this region being constant, causing the moment matrix for computing the optical flow to become singular. Hence, the numerical computations breaks down when trying to compute the motion parameters. For comparison, the needle diagram of the optical flow computed using the LS motion estimator (with non-overlapping 16x16 pixel local neighbourhood) is shown in Figure 3(d). This needle diagram is shown at the same scale as the needle diagram for the LMS estimator. Although not distinctly noticeable in the needle diagram at this scale, non-zero motion vectors have also occurred in the stationary background. The background motion vectors that have occurred using the LS motion estimator is smaller in magnitude compared to that computed using the LMS motion estimator. This is because the LS has a higher relative efficiency in terms of Gaussian noise compared to the LMS. To drastically reduce or even eliminate the motion vectors in the stationary background, smoothing can be performed on the images prior to computing the optical flow (as used by Barron et. al. [1] and Schunck [13]). Figure 3(e) shows the needle diagram of the optical flow computed using the Lucas and Kanade method [1] without any pre-smoothing of the images before computation of the flow and also without any thresholding based on the eigenvalues of the moment matrix, while Figure 3(f) shows the corresponding needle diagram with spatial-temporal pre-smoothing over 15 frames using standard deviation of 1.5 and threshold of 1.0 as recommended in [1]. The modified

Lucas and Kanade method [1] produces optical flow very similar to that shown in Figure 3(e) and 3(f) except that the motion vectors are more sparse. Finally, Figure 3(g) shows the needle diagram of the flow computed using the LMS motion estimator with 11x11 pixel Gaussian pre-smoothing (using standard deviation of 1.0) over 2 frames. The result compares favourably with that of Lucas and Kanade method.

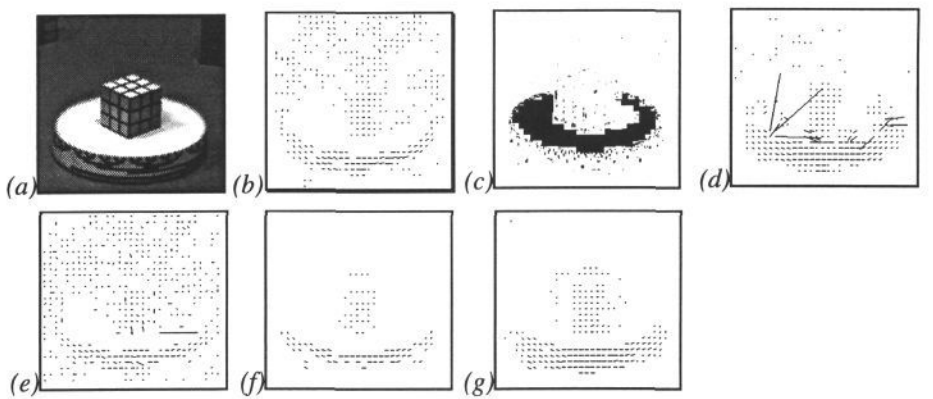

Figure 3: (a) First frame of the rubic-cube sequence. (b) Needle diagram of the optical flow computed using LMS motion estimator. (c) Confidence image of the LMS motion estimator. (d) Needle diagram of the optical flow computed using LS motion estimator. (e) Needle diagram of the optical flow computed using Lucas and Kanade method without pre-smoothing and thresholding. (f) Needle diagram of the optical flow computed using Lucas and Kanade method with pre-smoothing (standard deviation = 1.5) and thresholding (threshold = 1.0). (g) Needle diagram of the optical flow computed using LMS motion estimator with 11x11 pixel Gaussian pre-smoothing (standard deviation = 1.0).

5. Conclusions

This paper has presented an algorithm to compute optical flow accurately in regions of motion discontinuities and occlusion. The algorithm is based on the application of a least-median-squares robust estimator to compute the optical flow using the motion constraint equation and a 2-D affine model of the flow. It has shown that block-effects commonly faced in local differential methods for computing the optical flow can be eliminated using the proposed approach. This property makes it suitable as an efficient preliminary motion estimation step for motion segmentation. In addition, optical flow can also be computed accurately from any regions containing three motions as long as the shifting of the block is able to enable one of the motions to have a supporting region of at least 50% of the block's size. The major shortcoming of the algorithm is the high computational cost (The current algorithm takes about an hour to process a 128x128 pixel image sequence on a Sparc-10 workstation whereas both the Lucas and Kanade method and the Least-Squares method take less than 5 minutes). However, this is not a problem because the algorithm can be implemented easily on parallel processors.

There are a number of ways to improve the present algorithm. The first consideration is to implement the algorithm in a multiresolution scheme such that optical flow can be computed for objects with large motions. In addition, the inherent properties

of the algorithm makes it suitable for extension into an efficient motion segmentation scheme. Work along these lines is currently in progress.

Acknowledgements

The authors thank J.L. Barron for providing the Lucas and Kanade programs and the rubic-cube sequence (created by R. Szeliski). This research work was funded by the School of Electronic and Electrical Engineering, University of Birmingham, and an Overseas Research Student Award granted to the first author.

Reference

[1] Barron JL, Fleet DJ, and Beauchemin SS, "Performance of optical flow techniques", *Int. Journal of Computer Vision*, **12**, 1, 1994, pp. 43-77.

[2] Bouthemy P, and Francois E, "Motion segmentation and qualitative dynamic scene analysis from an image sequence", *Int. Journal of Computer Vision*, **10**, 2, 1993, pp. 157-182.

[3] Campani M, and Verri A, "Motion analysis from first-order properties of optical flow, *CVGIP: Image Understanding*, **56**, 1, July 1992, pp. 90-107.

[4] Draper NR, and Smith H, *Applied regression analysis*, 2nd Ed., John Wiley & Sons, New York, 1981.

[5] Goh WB, and Martin GR, "Model-based multiresolution motion estimation in noisy images", *CVGIP: Image Understanding*, **59**, 3, May 1994, pp. 307-319.

[6] Heitz F, and Bouthemy P, "Multimodal estimation of discontinuous optical flow using markov random fields", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **15**, 12, December 1993, pp. 1217-1232.

[7] Horn BKP, and Schunck BG, "Determining optical flow", *Artificial Intelligence*, **17**, 1981, pp. 185-203.

[8] Kearney JK, Thompson WB, and Boley DL, "Optical flow estimation: An error analysis of gradient-based methods with local optimisation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **9**, 2, March 1987, pp. 229-244.

[9] Meer P, Mintz D, and Rosenfeld A, "Robust regression methods for computer vision: A review", *Int. Journal of Computer Vision*, **6**, 1, 1991, pp. 59-70.

[10] Meyer FG, and Bouthemy P, "Region-based tracking using affine motion models in long range sequences", *CVGIP: Image Understanding*, **60**, 2, September 1994, pp. 119-140.

[11] Odobez JM, and Bouthemy P, "Robust multiresolution estimation of parametric motion models in complex image sequences", Proc. of 7th European Conf. on Signal Processing, Edinburgh, September 1994.

[12] Rousseeuw PJ, and Leroy AM, *Robust regression and outlier detection*, John Wiley & Sons, New York, 1987.

[13] Schunck BG, "Image flow segmentation and estimation by constraint line clustering", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **11**, 10, October 1989, pp. 1010-1027.

[14] Sinha SS, and Schunck BG, "A two-stage algorithm for discontinuity-preserving surface reconstruction", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **14**, 1, January 1992, pp. 36-55.

[15] Zhang ZY, Deriche R, Faugeras O, and Luong QT, "A Robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", Technical Report No. 2273, INRIA, France, May 1994.