Motion Estimation Via Cluster Matching

Article *in* IEEE Transactions on Pattern Analysis and Machine Intelligence · December 1994

DOI: 10.1109/34.334394 · Source: IEEE Xplore

Ying Sun
University of Rhode Island

have applied the hierarchical model to astronomical image problems with very encouraging results.

ACKNOWLEDGMENT

The author wishes to thank B. D. Ripley for his constructive and also destructive (of an early version of the correspondence) comments. Thanks are also due to F. Murtagh for introducing me to the problem of deconvolving HST images and for comments on the correspondence. J. Mateos and J. Abad provided programming assistance.

REFERENCES

[1] H. M. Adorf, "HST image restoration-status and prospects," in *Data Analysis in Astronomy*, vol IV, V. Di Gesù, L. Scarsi, R. Buccheri, P. Crane, M. C. Maccarone, and H. U. Zimmermann, eds. New York: Plenum Press, pp. 83–96, 1992.

[2] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*. New York: Springer Verlag, 1985, chaps. 3 and 4.

[3] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987, pp. 13–15.

[4] W. L. Buntine and A. Weigend, "Bayesian back-propagation," *Complex Systems*, vol. 5, pp. 603–643, 1991.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data," *J. Royal Statist. Soc. B*, vol. 39, pp. 1–38, 1972.

[6] D. Geman, "Random fields and inverse problems in imaging," in *École d'Été de Probabilités de Saint-Flour XVIII-1988*, P. L Hennequin, Ed. New York: Springer-Verlag, pp. 172–175, 1990.

[7] S. F. Gull, "Developments in maximum entropy data analysis," in *Maximum Entropy and Bayesian Methods*, J. Skilling, Ed. Norwell, MA: Kluwer Academic Press, pp. 53–71, 1989.

[8] S. F. Gull and J. Skilling, *MEMSYS 5 Users's Manual*. Maximum Entropy Consultants Ltd., 1991.

[9] M. G. Kang and A. K. Katsaggelos, "Simultaneous iterative image restoration and evaluation of the regularization parameter," *IEEE Trans. Signal Processing*, vol. 40, pp. 2329–2344, 1992.

[10] A. K. Katsaggelos and M. G. Kang, "Iterative evaluation of the regularization parameter in regularized image restoration," *J. Visual Communication and Image Representation*, vol. 3, pp 446–455, 1992.

[11] J. Llacer and J. Núñez, "Iterative maximum likelihood estimator and Bayesian algorithms for image reconstruction in astronomy," in *The Restoration of HST Images and Spectra*, R. L. White and R. J. Allen, Eds. Baltimore, MD: STScI, pp. 62–70, 1990.

[12] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astron. J.*, vol. 79, 745–754, 1974.

[13] D. J. C. MacKay, "A practical Bayesian framework for backprop networks," *Neural Computation.*, vol. 4, pp. 448–472, 1992.

[14] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, pp. 415–447, 1992.

[15] R. Molina, N. Pérez de la Blanca, and B. D. Ripley, "Statistical restoration of astronomical images," in *Data Analysis in Astronomy*, vol III, V. Di Gesu, L. Scarsi, P. Crane, J. H. Friedman, S. Levialdi, and M. C. Maccarone, Eds. New York: Plenum Press, pp. 75–82, 1992.

[16] R. Molina and B. D. Ripley, "Using spatial models as priors in astronomical images analysis," *J. Appl. Statist.*, vol. 16, pp. 193–206, 1989.

[17] R. Molina, A. del Olmo, J. Perea, and B. D. Ripley, "Bayesian deconvolution in optical astronomy," *Astrom. J.*, vol. 103, pp. 666–675, 1992.

[18] R. Molina, B. D. Ripley, A. Molina, F. Moreno, and J. L. Ortiz, "Bayesian deconvolution with prior knowledge of object location. applications to ground-based planetary images," *Astrom. J.*, vol. 104, pp. 1662–1668, 1992.

[19] R. Molina and A. K. Katsaggelos, "On the iterative evaluation of the regularization parameter and the hierarchical Bayesian approach to image restoration," submitted to the Conference on Visual Communications and Image Processing '94, 1994.

[20] L. Reginald and J. Biemond, *Iterative Identification and Restoration of Images*.Norwell, MA: Kluwer Academic Press, pp. 3 and 124–129, 1991.

[21] W. H. Richardson, "Bayesian-based iterative methods of image restoration," *J. Opt. Soc. Am.*, vol. 62, pp. 52–55, 1972.

[22] B. D. Ripley, *Spatial Statistics*. New York: John Wiley, pp. 88–90, 1981.

[23] B. D. Ripley, *Statistical Inference for Spatial Processes*, New York: John Wiley, pp. 88–94, 1988.

[24] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterington, "A study of methods of choosing the smoothing parameters in image restoration by regularization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 326–339, 1991.

[25] Y. Vardi and D. Lee, "From image debluring to optimal investments: Maximum likelihood solutions for positive linear inverse problems (with discussion)," *J. R. Statist. Soc. B*, vol. 55, pp. 569–612, 1993.

[26] N. Weir, "Recent developments in maximum entropy based image restoration," *ST-ECF Newsletter*, vol. 16, pp. 8–11, 1991.

[27] R. L. White and R. J. Allen, eds, *Proc. Workshop The Restoration of HST Images and Spectra*, Space Telescope Science Institute, Baltimore, 21–22 August, 1990.

Motion Estimation Via Cluster Matching

Dane P. Kottke and Ying Sun

***Abstract*— A new method for estimating displacements in computer imagery through cluster matching is presented. Without reliance on any object model, the algorithm clusters two successive frames of an image sequence based on position and intensity. After clustering, displacement estimates are obtained by matching the cluster centers between the two frames using cluster features such as position, intensity, shape and average gray-scale difference. The performance of the algorithm was compared to that of a gradient method and a block matching method. The cluster matching approach showed the best performance over a broad range of motion, illumination change and object deformation.**

***Index Terms*— Displacement estimation, cluster matching, affine motion model, clustering, variable object brightness.**

I. INTRODUCTION

Estimation of the two-dimensional motion field from an image sequence is a common problem in computer vision applications such as autonomous navigation, satellite imagery analysis and medical imaging [1]. In the past, many researchers contributed to the extraction of optical flow through image sequences based on models of the object and the illumination condition. In this study, clustering is used as a means to group pixels with similar features in each image frame; the displacements are estimated by matching clusters between two frames in the image sequence.

Clustering techniques were used for eliminating non-plausible matches in the velocity space before identifying the correspondence of feature points [2] and for dealing with discontinuities in the velocity field [4]. The approach proposed here is different from

Manuscript received June 10, 1992; revised January 25, 1994. Recommended for acceptance by Associate Editor Y. Aloimonos.

D. P. Kottke was with the Department of Electrical Engineering, University of Rhode Island, Kingston, RI 02881 USA. He is now with Lockheed Sanders Inc., Nashua, NH 03061 USA.

Y. Sun is with the Department of Electrical Engineering, University of Rhode Island, Kingston, RI 02881 USA.

IEEE Log Number 9405244.

the previous approaches in the sense that clustering is performed directly in the image space before the velocity field is identified. The purpose of this study is to develop a general framework for 2-D motion field estimation that requires very little *a priori* knowledge of the content of the image scene or the type of motion. Specifically, no assumptions regarding object rigidity, brightness constancy and scene illumination are needed. The method should also be applicable to motion estimation for images containing deformable objects with varying brightness.

II. Methodology

First, pixels in each image frame are clustered; the two successive frames are treated independently. For the present study, the feature space is three dimensional: position and intensity. Each of the resulting clusters tends to be spatially connected because of the position constraint and contains pixels of similar gray-scale values because of the intensity constraint. Second, clusters are matched between the two consecutive frames. In addition to the position and the intensity, the shape of each cluster and the average gray-scale difference between two clusters are used as matching criteria. The displacement between the two matching cluster centers is assigned to every pixel in the cluster. This results in a dense set of displacement estimates.

A. Clustering

The clustering algorithm can be thought of as a variation of the K-means clustering algorithm [5] and incorporates three features for each pixel, $\mathbf{p}^i = [xy\ I(x,y)]^t$, $(i = 1, 2, \cdots, N)$. They are the x and y coordinate locations, and the pixel's intensity, $I(x,y)$. Each cluster, $\mathbf{k}^j$, $(j = 1, 2, \cdots, K)$ is defined by the means, $\mathbf{m}^j = [m_x^j\ m_y^j\ m_z^j]^t$, of the pixels within that cluster. Initially, clusters are created by partitioning the image into K square regions. Each cluster is centered at m_x^j and m_y^j $(j = 1, 2, \cdots, K)$; m_z^j is the sample mean of gray-scale in each cluster.

After the initial clusters are determined, an iteration sequence commences wherein each pixel is assigned to a cluster based on a weighted squared Euclidean distance criterion. The iteration stops when the maximum shift of the cluster centers drops below a specified value. Pixel $\mathbf{p}^i$ is assigned to cluster k^j if

$$\epsilon^{ij} = (\mathbf{p}^i - \mathbf{m}^j)^t \mathbf{W}^j (\mathbf{p}^i - \mathbf{m}^j)$$

is a minimum for $j = 1, 2, \cdots, K$.

The clustering criterion is a weighted measure to reflect the fact that position and intensity cannot be directly compared. The weights

$$\mathbf{W}^j = \begin{bmatrix} w_x^j & 0 & 0 \\ 0 & w_y^j & 0 \\ 0 & 0 & w_z^j \end{bmatrix},$$

for $j = 1, 2, \cdots K$, are set so that features that have small variances give greater emphasis to the clustering decision. $\mathbf{W}^j$ is determined by minimizing the intraset distance between the pixels in cluster k^j, subject to the constraint that $w_x^j w_y^j w_z^j = 1$. These weights are given by [5]

$$w_x^j = \frac{c^j}{\sigma_x^j}, w_y^j = \frac{c^j}{\sigma_y^j}, w_z^j = \frac{c^j}{\sigma_z^j}. \tag{1}$$

$c^j = (\sigma_x^j \sigma_y^j \sigma_z^j)^{\frac{1}{3}}$. In a noise free region, it is possible that $\sigma_z^j = 0$. In this case we set $w_x^j = w_y^j = 0$ and $w_z^j = 1$. $(\sigma_x^j)^2$ is defined as the unbiased variance estimate of the pixels in cluster k_j in the x direction:

$$(\sigma_x^j)^2 = \frac{1}{(N_j - 1)} \sum_{x \epsilon k^j}^{N_j} (x - m_x^j)^2 \tag{2}$$

where N_j is the number of pixels in cluster k^j. Similarly, $(\sigma_y^j)^2$ and (σ_z^j) are defined. Note that a unique set of weights is calculated for each cluster rather than assigning one set of weights for all clusters.

As the iterations continue, the clusters change to reflect the contents of the image. The initial square regions adapt to whatever homogeneous regions exist in the image. The purpose of the weighting matrices is to balance the emphasis of position and intensity in the cluster. Therefore, it is important to recalculate $\mathbf{W^j}$ periodically to adapt to changes in the cluster contents. Accordingly, the weights of each cluster are recalculated every T_w iterations. We have found from our experiments that recalculating the weighting matrices every 10–30 iterations produces the best results.

B. Cluster Matching and Displacement Estimation

Once both images are clustered, cluster matching is performed. At this stage, we have two sets of clusters, one for each image. Let the set of clusters at time t be k^j, $\{j = 1, \cdots, K_t\}$ and at time $t + \Delta t$ be k^l, $\{l = 1, \ldots, K_{t+\Delta t}\}$. Let $\mathbf{b}^j$ be the vector of matching features for cluster k^j and, $\mathbf{c}^l$ denote the vector of matching features for k^l. To find accurate matches, two features are added to the means of the position and intensity of the clusters:

$$\mathbf{b}^j = \begin{bmatrix} \mathbf{m}^j \\ m_m^j \\ m_s^j \end{bmatrix} \text{ and } \mathbf{c}^l = \begin{bmatrix} \mathbf{m}^l \\ m_m^l \\ -m_s^l \end{bmatrix}$$

where

$$m_s^j = \frac{1}{N_j} \sum_{x,y \in k_j} I(x,y,t) - I(x,y,t+\Delta t) \tag{3}$$

is the average gray-scale difference in cluster k^j. The negative sign is needed for m_s^l to reverse the negation inherent in the gray-scale subtraction. The sum of the squared central moments in the x and y direction is defined by

$$m_m^j = \sum_{x,y \in k_j} \left[(x - \overline{x}^j)^2 + (y - \overline{y}^j)^2\right] I(x,y) \tag{4}$$

with

$$\overline{x}^j = \frac{\sum_{x,y \in k_j} x I(x,y)}{\sum_{x,y \in k_j} I(x,y)}$$
$$\overline{y}^j = \frac{\sum_{x,y \in k_j} y I(x,y)}{\sum_{x,y \in k_j} I(x,y)}. \tag{5}$$

Squared central moments are invariant to translation and rotation and represent the shape information of each cluster.

Each matching feature contributes to the correct match. For clusters with different intensities, but the same shape and equal displacements, the mean intensity values (m_z) influence the matching problem. If the moving clusters have the same intensities and equal displacements but different shapes, the shape information held in the values of the squared central moments (m_m) contributes strongly to the correct match. Finally, when clusters have the same shape and intensity, but one is stationary, while the other moves, the difference averages (m_s) help determine the correct match.

As in clustering, the matching criteria is a weighted squared Euclidean distance measure:

$$D^{jl} = (\mathbf{b}^j - \mathbf{c}^l)^t \mathbf{U} (\mathbf{b}^j - \mathbf{c}^l). \tag{6}$$

The index l that results in the minimum value of D^{jl} defines the match of cluster k^j and k^l. The 5×5 diagonal matrix $\mathbf{U}$ has u_x, u_y, u_z, u_m, and u_s on the diagonal and zero otherwise.

The matching procedure is a two-step iteration process: weight determination followed by cluster matching. The weights are initially set so that they generate matches corresponding to the closest clusters on the image plane. Next, cluster matching proceeds, wherein the matches yielding the minimum values of D^{jl} are found. For every iteration, the weights are recalculated so that the total displacement of the cluster centers is minimized, subject to the constraint that the product of the weights is unity. That is,

$$\min_{u_x, u_y, u_z, u_m, u_s} D_T = \sum_{j=1}^{K^j} D^{jl},$$

subject to $u_x u_y u_z u_m u_s = 1$. Applying Lagrange Multipliers, we have

$$D_T = \sum_{j=1}^{K^j} D^{jl} + \lambda(u_x u_y u_z u_m u_s - 1) \tag{7}$$

$$\lambda = -\frac{\sum_{j=1}^{K^j}(m_x^j - m_x^l)^2}{(u_y u_z u_m u_s)}. \tag{8}$$

From which we obtain,

$$u_y = \frac{u_x \sum_{j=1}^{K^j}(m_x^j - m_x^l)^2}{\sum_{j=1}^{K^j}(m_y^j - m_y^l)^2}. \tag{9}$$

Similarly,

$$u_z = \frac{u_x \sum_{j=1}^{K^j}(m_x^j - m_x^l)^2}{\sum_{j=1}^{K^j}(m_z^j - m_z^l)^2} \tag{10}$$

$$u_m = \frac{u_x \sum_{j=1}^{K^j}(m_x^j - m_x^l)^2}{\sum_{j=1}^{K^j}(m_m^j - m_m^l)^2} \tag{11}$$

$$u_s = \frac{u_x \sum_{j=1}^{K^j}(m_x^j - m_x^l)^2}{\sum_{j=1}^{K^j}(m_s^j - m_s^l)^2} \tag{12}$$

where

$$u_x = \left[\sum_{j=1}^{K^j}(m_y^j - m_y^l)^2\right]^{\frac{1}{5}} \left[\sum_{j=1}^{K^j}(m_z^j - m_z^l)^2\right]^{\frac{1}{5}} \left[\sum_{j=1}^{K^j}(m_m^j - m_m^l)^2\right]^{\frac{1}{5}} \left[\sum_{j=1}^{K^j}(m_s^j - m_s^l)^2\right]^{\frac{1}{5}} \left[\sum_{j=1}^{K^j}(m_x^j - m_x^l)^2\right]^{-\frac{4}{5}}. \tag{13}$$

The iteration sequence of weight calculation-cluster matching continues until the total displacement, D_t, does not change between iterations.

From the set of cluster center matches, the displacement estimates for each pixel in the cluster are obtained via

$$\begin{aligned} \hat{d}_x &= m_x^l - m_x^j \\ \hat{d}_y &= m_y^l - m_y^j. \end{aligned} \tag{14}$$

To separate the clusters belonging to moving object(s) from the non-moving background clusters, a confidence level for cluster matching is established. This confidence level arises from the fact that large magnitudes of m_s^j are associated with cluster motion. The confidence level, α, quantifies the certainty in the belief that a nonzero value of m_s^j indicates motion. For a clean, noise-free image, the confidence level can be very low. A threshold is set based on α. Let

$$\tau = \alpha \max_j |m_s^j|. \tag{15}$$

For each cluster in the image, the comparison between τ and $|m_s|$ is made.

$$\begin{cases} \hat{d}_x = m_x^l - m_x^j, \quad \hat{d}_y = m_y^l - m_y^j, & \text{if } \tau < |m_s^j|, \\ \hat{d}_x = \hat{d}_y = 0, & \text{otherwise.} \end{cases}$$

III. Performance Evaluation

The accuracy of the cluster matching approach was tested both quantitatively and qualitatively. Two other algorithms, the stationary spatial gradient [6] and the affine-model matching [3] were tested on the same image sets to provide a performance reference. The stationary spatial gradient method obtains motion estimates from spatial and temporal derivatives of the image brightness. A Gaussian smoothing filter, $G(\sigma_s)$ is convolved with the image brightness before finding the spatial derivatives. The affine model matching algorithm incorporates affine models for motion and illumination changes into block matching approach. The matching is performed over a 4-D parameter space. The model parameters are s (scale), θ (rotation) and δ_x, δ_y, (displacement). Overlapping regions of size $B \times B$ with the regions centered at distances smaller than $B/2$ pixels in each direction are used.

To post-process the displacement estimates, the gradient algorithm uses a Gaussian smoothing filter and the affine-model algorithm uses a vector median filter. In order to assess the robustness and the intrinsic accuracy of each algorithm, we compared the displacement estimates without incorporating the post-processing stage.

To evaluate the displacement estimates quantitatively, the average displacement error is determined according to

$$E = \frac{1}{N} \sum_{\text{image}} \sqrt{(d_x - \hat{d}_x)^2 + (d_y - \hat{d}_y)^2}, \tag{16}$$

where d_x and d_y are the true displacements for each pixel.

Three different sets of images were used for evaluation purposes. First, a quantitative study was conducted with a synthetic image sequence containing a moving sphere illuminated by a fixed-position light source. Second, a model car was imaged by using a video camera. This image sequence represented the case of a rigid object with varying brightness caused by light reflection. Third, a satellite image sequence of a hurricane was used to represent the case of a deformable object.

IV. Results

A. Synthetic-Sphere Images

The $256 \times 256 \times 8$-bit synthetic-sphere image created by a ray-tracing algorithm is shown in Fig. 1 (upper-left). To cluster this image, we used $K = 36$ initial regions and recalculated the weights every $T_w = 30$ iterations. The image took 29 iterations until the maximum shift of the cluster centers was less than 1.0 pixel. The initial weights for cluster matching were set to $u_x = u_y = u_z = u_s = 10$ and $u_m = 10^{-4}$. The sphere was moved seven pixels up and seven pixels to the right; the result of displacement estimates by the cluster matching algorithm is shown in Fig. 1 (upper-right). The average error was $E = 0.27$ pixel. The combined time to cluster and match both images was 17 minutes and 45 seconds on a DECstation 5000 Model 200 (Digital Equipment Corporation, Maynard MA).

The result of the gradient algorithm is shown in Fig. 1 (lower-left). The Gaussian smoothing filter parameter σ_s was equal to 4.0. The average error was $E = 5.0$ pixels. The algorithm took 2 minutes and 56 seconds. The general directions of the displacement estimates were correct. However, the region of the highlight on the sphere presented a problem for the gradient method.

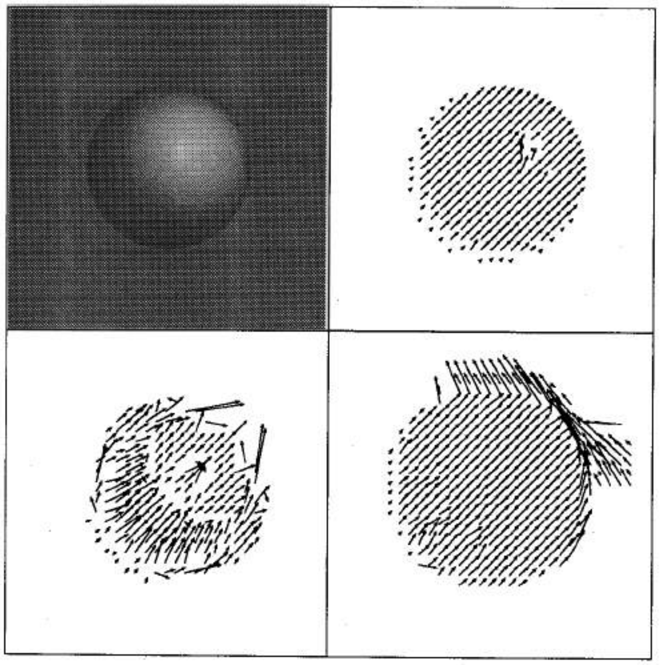

Fig. 1. Synthetic-sphere image (upper-left) and the displacement estimates by cluster matching (upper-right), spatial gradient (lower-left) and affine model (lower-right).

TABLE I
EFFECT OF NUMBER OF CLUSTERS ON DISPLACEMENT ESTIMATES

K	Iterations	Time	E (pixels)
16	13	364s	1.6
25	57	1575s	1.6
36	29	1085s	0.27
49	59	2303s	0.36
64	59	2783s	0.35

The result of the affine-model algorithm is shown in Fig. 1 (lower-right). The search space used was $\delta_x, \delta_y = 0, 14$ pixels by 1 pixel increments, $\theta = -6°, 6°$ by 2° increments and $s = 0.8$ to 1.2 by 0.1 increments. The regions, each containing 21×21 pixels, were centered at 8 pixels intervals throughout the image. The average error was $E = 2.7$ pixels. Total processing time was 7 hours and 53 minutes. Large errors were observed near the edge of the sphere. This was caused by the block-matching strategy employed by the algorithm.

The sensitivity of the cluster matching algorithm with respect to the choice of the initial cluster number K was evaluated. The number of regions was varied from 16 to 64, with the other clustering parameters held constant: $T_w = 30$, and $\alpha = 0.03$. The results are summarized in Table I. Note that once the number of clusters reached 36, the displacement estimation error dropped dramatically. The choice of K is important in ensuring that matchable clusters are obtained. In general, it is best to pick the smallest value of K that produces clusters accurately representing the structure of the image.

The performance of all three algorithms with respect to the magnitude of the displacement was also evaluated. The displacement of the sphere was varied over a range of $x = 0$ to 10 pixels and $y = 0$ to 10 pixels by 1-pixel increments. The parameters for the three algorithms were set the same as in the previous test with the exception that the maximum displacements δ_x and δ_y in the affine model were set at twice the displacement of the sphere. The average error in the displacement estimates (E) was plotted versus the magnitude of the actual displacement as shown in Fig. 2. While the errors for the gradient and affine algorithms increased with increasing displacement magnitude, the cluster matching method showed a much smaller error (<0.5 pixel) over the entire range of displacement.

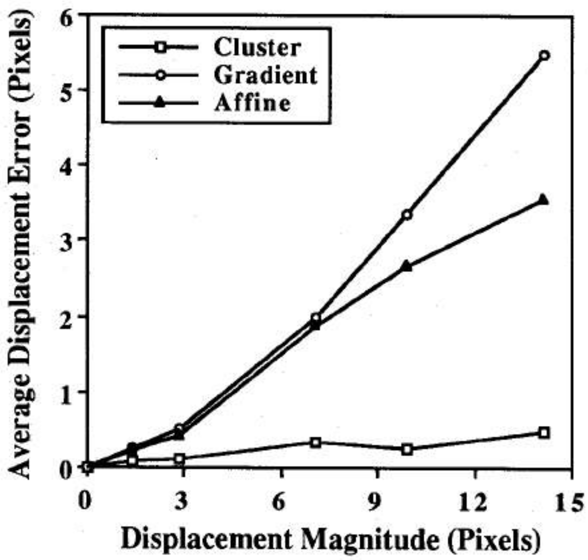

Fig. 2. Comparison of the three algorithms: average displacement error as a function of the displacement magnitude.

B. Model-Car Images

A frame from the moving model-car sequence is shown in Fig. 3 (upper-left). Note that in this sequence the reflectance on the hood, trunk and roof changed due to motion. This sequence contains rigid motion with large displacements and a change in object brightness. The clustering result from the cluster matching algorithm is shown in Fig. 3 (upper-right). The clustering parameters were $K = 25$ and $T_w = 10$. It took 24 iterations with a total processing time of 14 minutes and 38 seconds. The displacement results are shown in Fig. 3 (lower-left), with a certainty factor of $\alpha = 0.10$ for removing the small nonzero values in the background. The affine-model took 31 hours and 51 minutes; its result is shown in Fig. 3 (lower-right). The parameters were set at: $s = 0.8, 1.2$ by 0.1 increments, $\theta = -10°, 10°$, by 4° increments, $\delta_x, \delta_y = (0, 50), (0, 50)$ by 2-pixel increments. Due to the large displacement the gradient algorithm performed poorly; its result is not shown here.

C. Hurricane Satellite Images

In Fig. 4 (upper-left) a satellite image of a hurricane is shown. This sequence was used as a test for nonrigid motion. The result of the cluster matching algorithm is shown in Fig. 4 (upper-right). The clustering parameters were $K_0 = 25$ and $T_w = 15$. It took 28 iterations with a total processing time of 16 minutes and 32 seconds. The gradient algorithm took 8 minutes and 23 seconds; its result is shown in Fig. 4 (lower-left). The Gaussian smoothing parameter σ_s was 6.0. The affine-model took 30 hours and 45 minutes; its result is shown in Fig. 4 (lower-right). The parameters were set at: $s = 0.8, 1.2$ by 0.1 increments, $\theta = -10°, 10°$, by 2° increments, $\delta_x, \delta_y = -10, 10$ by 1-pixel increments.

V. DISCUSSION

In this paper, a cluster matching method for displacement estimation in computer imagery was described. A clustering process

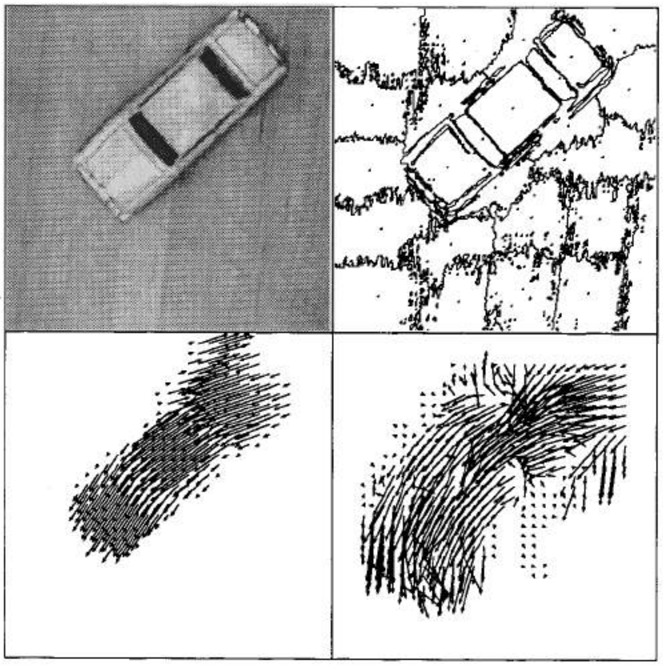

Fig. 3. A frame from the model-car image sequence (upper-left), the clustering result (upper-right), the displacement estimates by cluster matching (lower-left), and displacement estimates by affine model (lower-right).

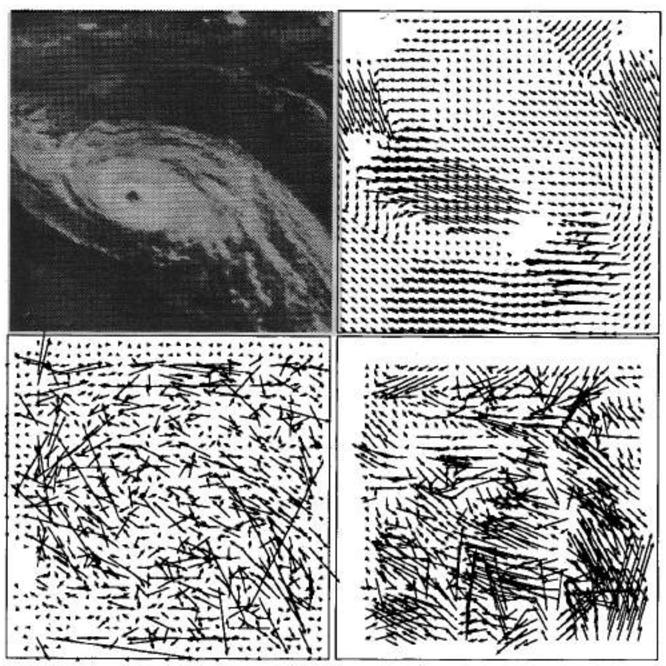

Fig. 4. Satellite image of hurricane (upper-left) and the displacement estimates by cluster matching (upper-right), spatial gradient (lower-left), and affine model (lower-right).

followed by cluster center matching provided the displacement estimates. The performance of the cluster matching method was compared to other displacement estimation schemes both quantitatively on synthetic imagery and qualitatively on real-world scenes. For the images tested, the cluster matching method performed better than the stationary spatial gradient and the affine-model matching methods.

The cluster matching method has two unique features relating to the loose constraints regarding object brightness, illumination conditions, and scene contents. First, the 3-D clustering space and the 5-D matching space give flexibility to accommodate changes in object brightness or illumination. The information contained in the clusters' matching vectors provides five components to influence the displacement estimation. In comparison, techniques like the gradient and block matching methods operate on intensity values at the pixel or group of pixel levels. The amount of available information is limited unless specific motion or illumination models are incorporated.

The second feature is that clustering adapts to scene contents with little need for *a priori* knowledge. In contrast to traditional feature tracking approaches, the clustering method does not search for specific trackable features. Rather, the structure of the image evolves over the course of the clustering iterations. This structure is then matched to obtain displacement estimates. Because the pixels in the image are classified into distinct cluster membership, the resulting displacement estimates are denser than other feature tracking methods. That is, the other approaches need either to state the explicit relationship between the features through rigidity assumptions or to accept a sparse set of estimates.

There are three areas under continuing study: the questions of the initial cluster partition and number selection, the *automatic* separation of cluster displacement caused by changes in illumination versus motion and the problems presented by object rotation and deformation. Considering the initial number of clusters, the correct cluster resolution is very important to obtain valid clustering results and accurate matches. If too few clusters are used, the matching will not be accurate since the cluster features will be quite different between the images. A more flexible scheme might include cluster growing/splitting to adjust the number of clusters automatically to the image contents. Also, the problem of automatic separation of motion versus background deformation might be solved by incorporating cluster growing/splitting to combine small homogeneous clusters into larger ones and a more sophisticated matching procedure that does not accept absurd matches. Finally, the cluster matching method does not include rotation, scaling or deformation models into its approach. As a result, there was a noticeable coarseness in the displacement estimates of the car sequence. Work is under way to incorporate intra-cluster rotation and deformable cluster motion models without loosing the appealing generality of the approach.

References

[1] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—A review," *Proc. IEEE*, vol. 76, pp. 917–935, 1988.

[2] A. Bandopadhay and J. Aloimonos, "Image motion estimation by clustering," *Int. J. Imaging Systems and Technology*, vol. 2, pp. 345–355, 1990.

[3] C. S. Fu and P. Maragos, "Motion displacement estimation using an affine model for image matching," *Optical Eng.*, vol. 30, no. 7, pp. 881–886, 1991.

[4] B. G. Schunck, "Image flow segmentation and estimation by constraint line clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 10, pp. 1010–1027, 1989.

[5] J. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Reading, MA: Addison-Wesley, 1974.

[6] A. Verri, F. Girosi, and V. Torre, "Differential techniques for optical flow," *J. Opt. Soc. Am. A*, vol. 7, pp. 912–922, 1990.