## ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their helpful comments.

## REFERENCES

[1] C. Cafforio and F. Rocca, "Methods for measuring small displacements of television images," *IEEE Trans. Inform. Theory*, vol. 22, pp. 573–579, 1976.
[2] ISO/IEC JTS 11172, "Coding of moving pictures and associated audio for storage at up to about 1.5 Mbit/s," Nov. 1992.
[3] ISI/IEC JTS 11172-2, "Coding of moving pictures and associated audio," Nov. 1993.
[4] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commum.*, vol. 29, pp. 1799–1808, 1981.
[5] T. Koga *et al.*, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, Nov. 29–Dec. 3, pp. G5.3.1–G5.3.5.
[6] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.*, vol. 38, pp. 950–953, 1990.
[7] K. W. Chun and J. B. Ra, "Fast block matching algorithm by successive refinement of matching criterion," in *Proc. SPIE: Visual Communications and Image Processing'92*, vol. 1818, pp. 552–560.
[8] B. Liu and A. Zaccarin, "New fast algorithm for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 148–187, 1993.
[9] C. Hsieh, P. Lu, J. Shyn, and E. Lu, "Motion estimation algorithm using interblock correlation," *Electron. Lett.*, vol. 26, pp. 276–277, 1990.
[10] M. Bierling and R. Thoma, "Motion compensating field interpolation using a hierarchically structureed displacement estimator," *Signal Process.*, vol. 11, pp. 387–404, 1986.
[11] R. Li, B. Zeng, and M.-L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 438–442, 1994.
[12] C. L. Huang and C. Y. Hsu, "A new motion compensation method for image sequence coding using hierarchical grid interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 72–85, 1994.
[13] D. Hepper, "Efficiency analysis application of uncovered background prediction in a low bit rate image coder," *IEEE Trans. Commun.*, vol. 36, pp. 1578–1584, 1990.
[14] B. Girod, "The information theoretical significance of spatial and temporal masking in video signals," in *Proc. SPIE Conf. Human Vision, Visual Processing, Digital Display*, 1989, vol. 1077, pp. 178–187.
[15] J. O. Limb and J. A. Murphy, "Measuring the speed of moving objects from television signal," *IEEE Trans. Commun.*, pp. 474–478, Apr. 1975.

# A Fast Motion Estimation Algorithm Based on the Block Sum Pyramid

Chang-Hsing Lee and Ling-Hwei Chen

***Abstract*—In this correspondence, a fast approach to motion estimation is presented. The algorithm uses the block sum pyramid to eliminate unnecessary search positions. It first constructs the sum pyramid structure of a block. Successive elimination is then performed hierarchically from the top level to the bottom level of the pyramid. Many search positions can be skipped from being considered as the best motion vector and, thus, the search complexity can be reduced. The algorithm can achieve the same estimation accuracy as the full search block matching algorithm with much less computation time.**

***Index Terms*—Block matching, motion estimation, pyramid.**

## I. INTRODUCTION

In image sequence coding, the correlation between consecutive frames can be reduced by the motion estimation/motion compensation technique [1]. Motion estimation plays an important role in reducing the bit rates for transmission or storage of video signals. The block matching algorithm (BMA) [2], which estimates the amount of motion on a block-by-block basis, is the most popular motion estimation method. In BMA, for each template block in the present frame, the best matching block within the search area in the previous frame will be determined by evaluating some matching criterion. The displacement vector of the best matching block relative to the template block is taken as the motion vector of the template block. In general, the search area is limited in the window of size $(2W+1) \times (2W+1)$ centering around the position of the template block. The mean absolute difference (MAD) is the most popular matching criterion because it requires no multiplication operations and is defined as

$$\text{MAD}_{(u,\,v)} = \sum_{i=1}^{N} \sum_{j=1}^{N} |f_t(i,\,j) - f_{t-1}(i+u,\,j+v)|, \quad -W \le u,\, v \le W \tag{1}$$

where $(u,\,v)$ is the displacement vector of the candidate block relative to the template block, $f_t(\cdot,\,\cdot)$ is the gray value of a pixel in the present frame and $f_{t-1}(\cdot,\,\cdot)$ is the gray value of a pixel in the previous frame. The displacement vector $(u,\,v)$ of a candidate block associated with minimal $\text{MAD}(u,\,v)$ is selected as the motion vector. The full-search algorithm (FSA) evaluates the MAD for all of the $(2W+1)^2$ search positions. Therefore, the computation of FSA is very intensive. Several fast-search algorithms, such as the three-step search (TSS) [3] and its improvements [4]–[6], two-dimensional (2-D) logarithmic search [7], one-at-a-time search [8], [9], orthogonal search [10], cross search [11], hierarchical search [12]–[14], genetic search [15], and one-dimensional (1-D) full search [16], etc., have

Manuscript received December 20, 1995; revised January 20, 1997. This research was supported in part by the National Science Council of the R.O.C. under Contract NSC-84-0404-E009-038. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Eric Dubois.

C.H. Lee is with the Department of Information Science, Chinese Culture University, Taipei 1114, Taiwan, R.O.C. (e-mail: chlee@ccu016.pccu.edu.tw).

L.-H. Chen is with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu 30050, Taiwan, R.O.C. (e-mail: lhchen@cc.nctu.edu.tw).

Publisher Item Identifier S 1057-7149(97)07894-9.

been proposed. These algorithms reduce the search complexity by limiting the number of search positions based on the assumption that the matching error monotonically increases as the search position moves away from the location of the optimal motion vector. The assumption is not always true in reality and they may converge to a local minimum on the error surface instead of the global minimum as in the FSA.

Recently, Li and Salari [17] proposed a fast motion estimation method called the successive elimination algorithm (SEA), which can achieve the same estimation accuracy as the FSA while requiring less computation time. In SEA, the displacement vector of the corresponding block in the previous frame is used as the initial motion vector for the present template block [18]. The SEA uses the sum norm of a block as a feature to eliminate unnecessary search positions. The sum norm of a block $B$ of size $N \times N$ is defined as

$$S_B = \sum_{i=1}^{N} \sum_{j=1}^{N} |B(i, j)| \quad (2)$$

where $B(i, j)$ is the gray level of the $(i, j)$th pixel of block $B$. Let $S_T$ be the sum norm of the template block $T$, $S_X$ be the sum norm of a candidate matching block $X$, and $\text{MAD}_{\min}$ be the current minimal MAD during the search process. Let $\text{MAD}(T, X)$ be the MAD between $T$ and $X$ and is defined as

$$\text{MAD}(T, X) = \sum_{i=1}^{N} \sum_{j=1}^{N} |T(i, j) - X(i, j)|$$

where $T(i, j)$ and $X(i, j)$ represent the gray values of the $(i, j)$th pixels of $T$ and $X$. The authors had shown that the following inequality is true:

$$\text{MAD}(T, X) \geq |S_T - S_X|. \quad (3)$$

Based on the above inequality, the SEA discards each candidate matching block $X$ with $|S_T - S_X| \geq \text{MAD}_{\min}$, which can save a lot of search time.

In this correspondence, a fast approach to motion estimation called the *block sum pyramid algorithm* (BSPA) is introduced. The BSPA can achieve the same estimation accuracy as FSA and SEA while needing much less computation requirement than these two algorithms.

## II. The Block Sum Pyramid Algorithm (BSPA)

### A. Principles of the Algorithm

As mentioned previously, the SEA uses the sum norm of a block as a feature to eliminate unnecessary block matches. Here, a more efficient search algorithm, the BSPA, will be proposed by exploiting the sum pyramid structure of a block to eliminate those impossible matching blocks. An image pyramid is a hierarchical data structure originally developed for image coding [19]. In the following, we will introduce the sum pyramid data structure first.

Assume that each block is of size $N \times N$ with $N = 2^n$. Then, for each block $X$, a pyramid of $X$ (see Fig. 1) can be defined as a sequence of blocks $\{X^0, \cdots, X^{m-1}, X^m, X^{m+1}, \cdots, X^n\}$ with $X^{m-1}$ having size $2^{m-1} \times 2^{m-1}$ and being a reduced-resolution version of $X^m$. Note that $X^0$ has only one pixel. A pyramid data structure can be formed by successively operating over $2 \times 2$ neighboring pixels on the higher levels. That is, the value of a pixel $X^{m-1}(i, j)$ on level $m-1$ can be obtained from the values of the corresponding $2 \times 2$ neighboring pixels $X^m(2i-1, 2j-1)$, $X^m(2i-1, 2j)$, $X^m(2i, 2j-1)$, and $X^m(2i, 2j)$ on level $m$. In other words, $X^{m-1}(i, j)$ can be obtained by $X^{m-1}(i, j) = \boldsymbol{f}[X^m(2i-1, 2j-1), X^m(2i-1, 2j), X^m(2i, 2j-1), X^m(2i, 2j)]$, where $\boldsymbol{f}$ is an operating function. In the sum pyramid structure, the operating function $\boldsymbol{f}$ is a summation function, i.e., the value of each pixel is obtained by summing the values of its corresponding $2 \times 2$ neighboring pixels on the next level. For example

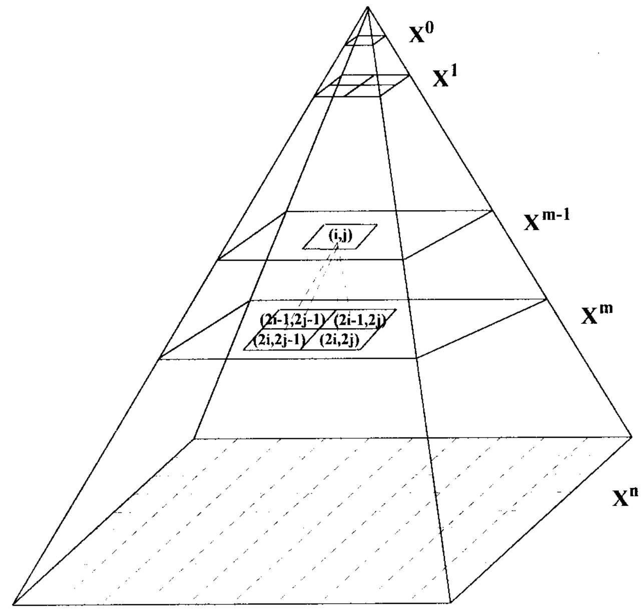


Fig. 1. A pyramid data structure.

$$\begin{aligned} X^{m-1}(i, j) = X^m(2i-1, 2j-1) + X^m(2i-1, 2j) \\ + X^m(2i, 2j-1) + X^m(2i, 2j). \end{aligned} \quad (4)$$

For two blocks $X$ and $Y$, let $\text{MAD}^m(X, Y)$ be $\text{MAD}(X^m, Y^m)$, i.e.,

$$\text{MAD}^m(X, Y) = \sum_{j=1}^{2^m} \sum_{h=1}^{2^m} |X^m(j, h) - Y^m(j, h)|$$

where $X^m(j, h)$ and $Y^m(j, h)$ represent the values of the $(j, h)$th pixels on $X^m$ and $Y^m$, respectively. Thus, on the top level, $\text{MAD}^0(X, Y) = |S_X - S_Y|$. From the above definition, we have the following theorem.

*Theorem 1:*

$$\begin{aligned} \text{MAD}(X, Y) \geq \text{MAD}^{n-1}(X, Y) \geq \text{MAD}^{n-2}(X, Y) \\ \geq \cdots \geq \text{MAD}^0(X, Y). \end{aligned} \quad (5)$$

*Proof:* Since

$$\begin{aligned} &\text{MAD}^{m+1}(X, Y) \\ &= \sum_{a=1}^{2^{m+1}} \sum_{b=1}^{2^{m+1}} |X^{m+1}(a, b) - Y^{m+1}(a, b)| \\ &= \sum_{j=1}^{2^m} \sum_{h=1}^{2^m} \{|X^{m+1}(2j-1, 2h-1) \\ &\quad - Y^{m+1}(2j-1, 2h-1)| \\ &\quad + |X^{m+1}(2j-1, 2h) - Y^{m+1}(2j-1, 2h)| \\ &\quad + |X^{m+1}(2j, 2h-1) - Y^{m+1}(2j, 2h-1)| \\ &\quad + |X^{m+1}(2j, 2h) - Y^{m+1}(2j, 2h)|\}. \end{aligned}$$

From (3) and the definition of the sum pyramid, for any $m$, $0 \leq$

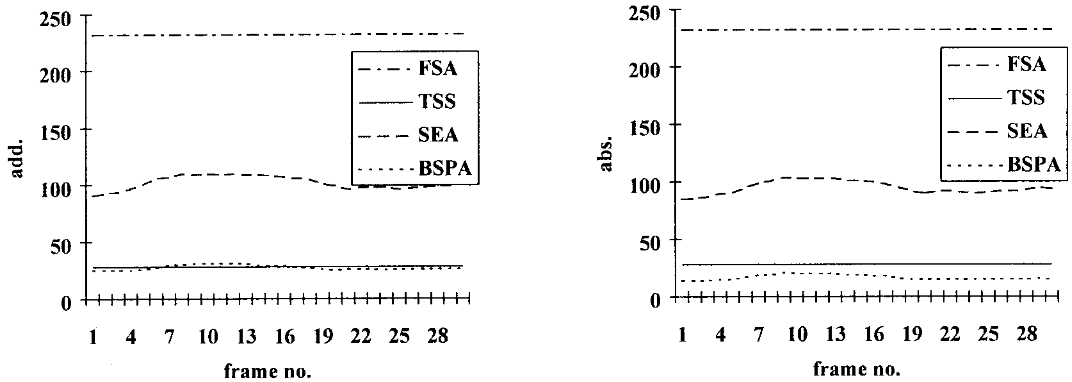


Fig. 2. Comparison of addition/absolute operations for the salesman image sequence.

TABLE I
PERFORMANCE EVALUATION OF VARIOUS BLOCK-MATCHING ALGORITHMS

| | Salesman | | Missa | | Claire | | Swing | |
|---|---|---|---|---|---|---|---|---|
| Algorithm | DFD | PSNR | DFD | PSNR | DFD | PSNR | DFD | PSNR |
| FSA | 2.91 | 34.97 | 1.93 | 39.03 | 0.83 | 42.56 | 3.07 | 35.67 |
| TSS | 2.97 | 34.75 | 2.01 | 38.74 | 0.83 | 42.56 | 3.11 | 35.30 |
| SEA | 2.91 | 34.97 | 1.93 | 39.03 | 0.83 | 42.56 | 3.07 | 35.67 |
| BSPA | 2.91 | 34.97 | 1.93 | 39.03 | 0.83 | 42.56 | 3.07 | 35.67 |

$m < n$, we can get

$$\begin{aligned} \mathrm{MAD}^{m+1}(X, Y) &\geq \sum_{j=1}^{2^m} \sum_{h=1}^{2^m} |X^m(j, h) - Y^m(j, h)| \\ &= \mathrm{MAD}^m(X, Y). \end{aligned}$$

Since $\mathrm{MAD}(X, Y) = \mathrm{MAD}^n(X, Y)$, we can easily obtain

$$\begin{aligned} \mathrm{MAD}(X, Y) &\geq \mathrm{MAD}^{n-1}(X, Y) \geq \mathrm{MAD}^{n-2}(X, Y) \\ &\geq \cdots \geq \mathrm{MAD}^0(X, Y). \end{aligned}$$

With the above theorem in hand, we will begin describing the BSPA. The BSPA first constructs the sum pyramid of every block that corresponds to a search position in the previous frame. To search for the best matching block of a template block $T$, the sum pyramid of $T$ is established. Then, the MAD between $T$ and the block with displacement vector (0, 0) is evaluated, and this value is considered as the current minimum MAD, $\mathrm{MAD}_{\min}$. For any other search block $X$, the algorithm first checks the MAD on the top level, $\mathrm{MAD}^0(T, X)$. If $\mathrm{MAD}^0(T, X)$ is greater than $\mathrm{MAD}_{\min}$, this block can be eliminated from being considered as the best matching block. Otherwise, the MAD on the first level is checked. If $\mathrm{MAD}^1(T, X)$ is greater than $\mathrm{MAD}_{\min}$, for the same reason as above, this block can be eliminated. If it is not, the second level is tested. The process is repeated until this block is eliminated or the bottom level is reached. If the bottom level is reached, then $\mathrm{MAD}(T, X)$ is calculated and checked. If $\mathrm{MAD}(T, X) < \mathrm{MAD}_{\min}$, the current minimum distortion $\mathrm{MAD}_{\min}$ is replaced with $\mathrm{MAD}(T, X)$.

The proposed algorithm is a "coarse to fine" technique, which can eliminate many search blocks without evaluating their MAD's. Since evaluating the MAD between two blocks needs more time than evaluating the MAD on the top levels of the sum pyramid, the elimination of many blocks before their MAD's are evaluated can save a great deal of time.

### B. Calculation of the Sum Pyramid

In BSPA, the block sum pyramid of each candidate block in the search area must be known. Assume that the size of the image frame is $W \times H$. For each level of the pyramid, calculation of the sum of $2 \times 2$ neighboring pixels requires $3(W-1)(H-1)$ additions. However, using the idea for fast calculation of the sum norm developed in [17], the complexity can be reduced to be $(2W-1)(H-1)$ additions for each level. If the block size is $16 \times 16$, i.e., $N = 16$, the overhead for constructing the sum pyramid is $4(2W-1)(H-1)$. Since there are $(W/N)(H/N)$ template blocks in an image frame, the computation overhead for each template block is

$$\begin{aligned} &4(2W-1)(H-1)/[(W/N)(H/N)] \\ &\quad = N^2(8 - 4/W - 8/H + 4/WH) \approx 8N^2. \end{aligned}$$

Since each block matching requires $N^2$ operations, the overhead is approximately equivalent to eight search positions.

## III. EXPERIMENTAL RESULTS

We have investigated the performance of the proposed algorithm by comparing it with the FSA, TSS, and SEA on a Sun SparcStation 20. The first 30 frames in four image sequences (salesman, Missa, Claire, and swing) are used in the simulation. Each image frame is

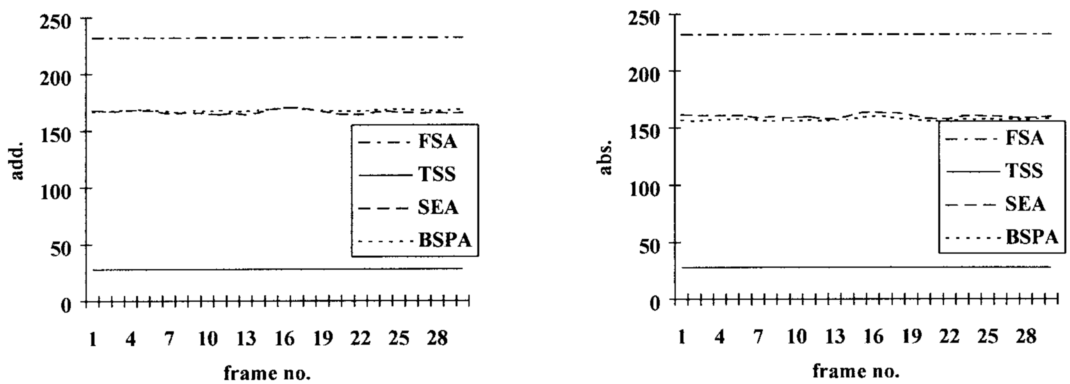


Fig. 3. Comparison of addition/absolute operations for the Missa image sequence.

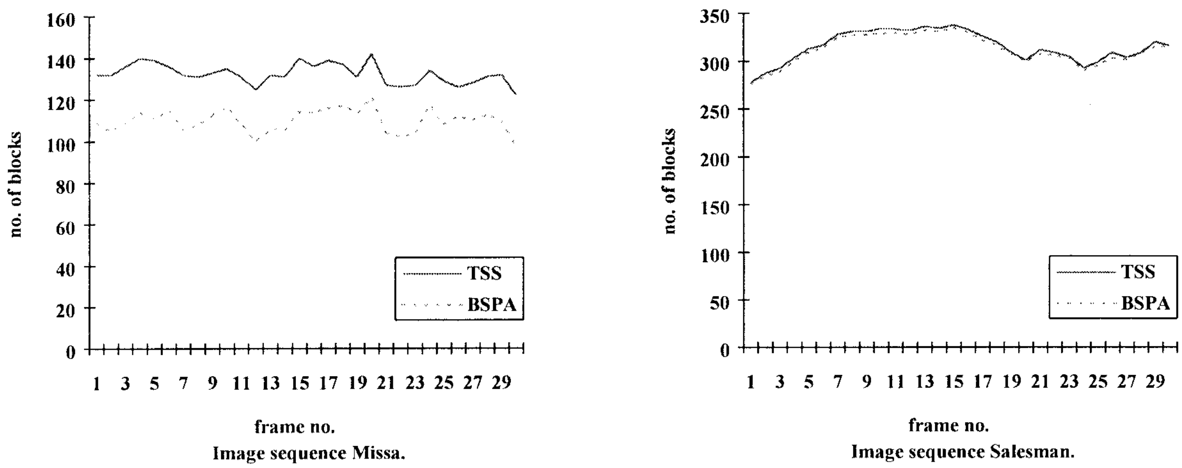


Fig. 4. Comparison of the number of blocks with prediction error greater than a threshold $T_m$ for each 16 × 16 block ($T_m$ is 512 for MAE).

of size 352 × 288. The block size for motion estimation is 16 × 16. The search area is of size 15 × 15. The displaced frame difference (DFD) [4] is used to measure the performance of the algorithms, and is defined as

$$\frac{1}{352 \times 288} \sum_{i=1}^{352} \sum_{j=1}^{288} |f_t(i, j) - f_{t-1}[i - u(i, j), j - v(i, j)]|$$

where $[u(i, j), v(i, j)]$ is the motion vector obtained at each point $(i, j)$.

Table I shows the comparison results for the image sequences in terms of DFD and PSNR. From this table, we can see that BSPA, SEA, and FSA have the same estimation error, which is lower than that of the TSS algorithm.

Figs. 2 and 3 compare addition and absolute operations for the image sequences salesman and Missa, respectively. As can be seen from Fig. 2, BSPA outperforms SEA and TSS algorithms. In Fig. 3, the required operation of BSPA is almost the same as that of SEA, but is more than that of the TSS algorithm. From these two figures, we can see that for image sequences with more complex backgrounds (e.g., salesman), the reduction is much more than that for an image sequence with a simple background (e.g., Missa).

In video coding, the prediction error between each block and its best matching block will be coded and transmitted if the prediction error is greater than a certain threshold $T_m$. Therefore, the more the number of blocks with prediction error greater than $T_m$, the more bits and time complexity is needed. Fig. 4 shows the number of blocks with prediction error greater than $T_m$ for the BSPA and TSS algorithms. From this figure, we can see that the number of blocks with prediction error greater than $T_m$ for the TSS algorithm is more than that of the BSPA.

## IV. Conclusions

A new approach to motion estimation, BSPA, has been presented in the correspondence. The algorithm uses the block sum pyramid to eliminate unnecessary search positions. The BSPA can find the global optimal solution and outperforms the SEA. The reduction of computational requirements depends on the characteristic of the image sequences. For an image sequence with a more complex background, the reduction is much more than that for an image sequence with a simple background.

## References

[1] H. G. Musmann, P. Pirsch, and H. J. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, Apr. 1985.

[2] F. Giorda and A. Racciu, "Bandwidth reduction of video signals via shift vector transmission," *IEEE Trans. Commun.*, vol. COMM-23, pp. 1002–1004, 1975.

[3] T. Koga *et al.*, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommunications Conf.*, 1981, pp. G5.3.1–G5.3.5.

[4] K. W. Chun and J. B. Ra, "An improved block matching algorithm based on successive refinement of motion vector candidates," *Signal Process.: Image Commun.*, vol. 6, pp. 115–122, 1994.

[5] L. W. Lee, J. F. Wang, J. Y. Lee, and J. D. Shie, "Dynamic search-window adjustment and interlaced search for block-matching algorithm," *IEEE Trans. Circ. Syst. Video Technol.*, vol. 3, pp. 85–87, Feb. 1993.

[6] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 4, pp. 438–442, Aug. 1994.
[7] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.,* vol. COMM-29, pp. 1799–1808, Dec. 1981.
[8] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.,* vol. COMM-33, pp. 888–896, Aug. 1985.
[9] S. Kappagantula and K. R. Rao, "Motion compensated interframe image prediction," *IEEE Trans. Commun.,* vol. COMM-33, pp. 1011–1015, Sept. 1985.
[10] A. Puri, H. M. Hang, and D. L. Schilling, "An efficient block-matching algorithm for motion compensated coding," *Proc. ICASSP,* pp. 1063–1066, 1987.
[11] M. Chanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.,* vol. 38, pp. 950–953, July 1990.
[12] M. Bierling, "Displacement estimation by hierarchical block-matching," in *Proc. SPIE Conf. Visual Communications and Image Processing,* 1988, vol. 1001, pp. 942–951.
[13] Q. Wang and R. J. Clarke, "Motion estimation and compensation for image sequence coding," *Signal Process.: Image Commun.,* vol. 4, pp. 161–174, 1992.
[14] H. M. Jong, L. G. Chen, and T. D. Chiueh, "Accuracy improvement and cost reduction of 3-step search block matching algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 4, pp. 88–90, Feb. 1994.
[15] K. H. K. Chow and M. L. Liou, "Genetic motion search algorithm for video compression," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 3, pp. 440–445, Dec. 1993.
[16] M. J. Chen, L. G. Chen, and T. D. Chiueh, "One-dimensional full search motion estimation algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 4, pp. 504–509, Oct. 1994.
[17] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing,* vol. 4, pp. 105–107, Jan. 1995.
[18] Y. Q. Zhang and S. Zafar, "Predictive block-matching motion estimation for TV coding—Part II: Interframe prediction," *IEEE Trans. Broadcast.,* vol. 37, pp. 102–105, Sept. 1991.
[19] P. J. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.,* vol. COMM-31, pp. 523–540, Apr. 1983.

# Motion Segmentation by Multistage Affine Classification

Georgi D. Borshukov, Gozde Bozdagi,
Yucel Altunbasak, and A. Murat Tekalp

***Abstract*—We present a multistage affine motion segmentation method that combines the benefits of the dominant motion and block-based affine modeling approaches. In particular, we propose two key modifications to a recent motion segmentation algorithm developed by Wang and Adelson. 1) The adaptive k-means clustering step is replaced by a merging step, whereby the affine parameters of a block which has the smallest representation error, rather than the respective cluster center, is used to represent each layer; and 2) we implement it in multiple stages, where pixels belonging to a single motion model are labeled at each stage. Performance improvement due to the proposed modifications is demonstrated on real video frames.**

## I. Introduction

Motion segmentation refers to grouping together pixels that undergo a common motion. Various methods for motion segmentation can be classified as those belonging to affine clustering approach [1], dominant motion approach [2], and simultaneous motion estimation and segmentation approach [3].

Recently, Wang and Adelson [1] proposed an affine-clustering-based motion segmentation method for layered representation of video. First, a dense flow field is estimated. This flow field is, then, divided into nonoverlapping rectangular blocks, and affine motion parameters are estimated for each block. Next, the affine parameters are subjected to a reliability test, based on how well the affine motion vectors within each block fit the estimated dense flow field. Those affine parameters (hypothesis) that pass the test are clustered into a small number of classes using an adaptive k-means algorithm. Finally, each flow vector is assigned to one of the resulting classes, represented by their cluster centers, using a minimum residual classifier. Motion vectors where the minimum residual is above a prespecified threshold are handled separately. Our experimentation with this method has led to the following observations.

1) The adaptive $k$-means clustering in essence computes an "average" affine model (hypothesis) for each layer. We show that better results can be obtained by replacing the clustering step with a "merge" step, which in effect picks the model having the smallest residual for each layer.
2) Labeling the class memberships of all motion vectors in a single stage generally produces unsatisfactory results. That is, it may result in either segmentation of a single motion into multiple

Manuscript received January 24, 1996; revised February 14, 1997. This work was supported in part by an NSF IUCRC grant and a New York State Science and Technology Foundation grant to the Center for Electronic Imaging Systems at the University of Rochester, and by a grant from Eastman Kodak Company. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Michael T. Orchard.

G. D. Borshukov is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA.

G. Bozdagi is with Xerox Corporation, Webster, NY 14580 USA (e-mail: bozdagi@wrc.xerox.com).

Y. Altunbasak is with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA.

A. M. Tekalp is with the Department of Electrical Engineering and Center for Electronic Imaging Systems, University of Rochester, Rochester, NY 14627 USA.

Publisher Item Identifier S 1057-7149(97)07962-1.