A Pattern Matching Algorithm Applied to the Estimation of Motion Vectors in Video Compression

Renê Togni Del Pietro and Max H. M. Costa
Department of Communications - FEEC - Unicamp
P.O. Box 6101– 13083-970–Campinas, SP
togni, max@decom.fee.unicamp.br

Abstract - **In this article, new results are presented of the application of the modified successive elimination (SE) algorithm [1] in the estimation of motion vectors as used in video compression. The modified SE algorithm represents an evolution of the SE algorithm proposed by Li and Salari [2]. SE algorithm works by pre-selecting blocks in the research area before the block matching operations. The modified SE algorithm improves this pre-selection, by adding new discard and preordering criteria of the research area. Results of simulations are shown that indicate a significant improvement in the computational cost of the motion estimation process, while guaranteeing that global optimal motion vectors are obtained for a given research region.**

I. INTRODUCTION

In the coding of video sequences, a considerable part of the temporal redundancy contained in successive frames of the sequence can be eliminated by motion estimation and compensation techniques. The efficiency of temporal redundancy extraction and the resulting rate-distortion performance depend directly on the adopted motion estimation technique. There are, basically, two classes of motion estimation algorithms: the ones that use the individual pixels as the estimate units (e.g., pel-recursive algorithm - PRA [3]) and those that use groups of pixels as units of estimation (e.g., block matching algorithm – BMA [4], [5]). Given their implementation simplicity and smaller computational cost, the algorithms of the BMA class are adopted by several video coding standards, as CCITT H.261 [6], ITU-T H.263 [7] and MPEG [8].

In the algorithms of the BMA class, each video frame is divided up in blocks of fixed size. These blocks constitute the estimate units. Then, for each block, motion vectors are estimated to indicate the position, within a given research area of another frame in the video sequence, of the most similar block according to a certain distortion measure or matching criterion.

For this purpose, the exhaustive search (ES) algorithm, guarantees that the chosen block yields the minimum distortion, since all possible distortions within the research area are calculated. The drawback, of course, is the high computational cost of this comprehensive approach. Alternative algorithms, such as 2D-log search [5] and cross-search [9] reduce the computational cost by making certain simplifying assumptions, but lead to sub-optimal solutions. The successive elimination (SE) algorithm of Li and Salari [2], reduces the computational cost of exhaustive search, while maintaining its optimal solution. This is achieved by the application of a clever discard criterion prior to the computation of distortion between blocks. Thus, the domain of candidate blocks is reduced and so is the computational burden of the motion estimation process. In [1], a modified version of the SE algorithm was proposed, that further reduces computational cost by adding new discard and preordering criteria.

In this article, new results of the modified SE algorithm [1], when applied to limited search areas are presented. Simulations indicate an improvement in the estimation process with respect to computational cost and rate×distortion performance, while still giving the optimal motion vector estimate within the search area.

II. THE MODIFIED SEA ALGORITHM

Let **X** be a block in the current frame for which an estimate is sought of the corresponding motion vector to a different frame, and let **Y**(i,j) be the block in coordinates (i,j) in the latter frame. The estimated motion vector of **X** is given by the difference between the coordinates of **X** and the coordinates (i,j) of the block that minimizes the mean absolute error (MAE) norm, defined by

$$MAE = \left\| \mathrm{X} - \mathrm{Y}_{(i,j)} \right\| \quad (1)$$

where $\| X \| = \Sigma_k x_k$, and k includes all the elements of the argument. In many applications, the blocks have dimensions of 16 × 16 pixels, with the coordinates (i,j) located within the interval [-16, 16] pixels with respect to the coordinates of **X** in the current frame.

The modified SE algorithm reduces the cost of computation of the minimum *MAE* by the incorporation of block discard criteria and by pre-ordering the blocks in the research area before the *MAE* computation is performed.

A. Successive Block Discard Criteria

Using block discard criteria, we can eliminate a portion of the competitive blocks for the minimum *MAE* location, without loss of generality. As the discard criteria evaluation consume less mathematical operations than the MAE computation itself, a computational gain is obtained in the process.

A block of 16 × 16 pixels can be looked at as a point in a vector space of 16 x 16 = 256 dimensions. The norm in Eq. (1) defines a distance between the points **X** and **Y**(i,j) in this vector space. As the distance between two points in a vector space is always larger than the distance between the same points when projected in a subspace of the original space, we can discard, given the calculated MAE_0 between **X** and a given block **Y**(u,v), all the blocks **Y**(i,j) whose distances to **X** projected onto a chosen subspace is superior to this particular MAE_0 distance. Fig. 1 gives a graphical illustration of the idea.

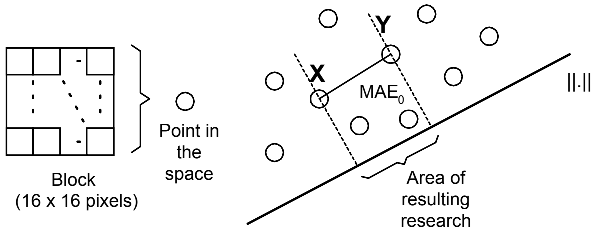


Fig. 1. Blocks, as points in a vector space, and their projections onto a subspace.

Mathematically, the discard criterion can be derived from the following Theorem.

Theorem: Let a, b, c and d be positive integers. Then, the following inequality applies:

$$\left|a-b\right|+\left|c-d\right| \geq \left|a+c\right|-\left|b+d\right| \tag{2}$$

Proof: Two manifestations of the triangular inequality can be expressed by inequalities |a| + |b| ≥ |a+b| and |a-b| ≥ |a| - |b| [11]. Therefore

$$\begin{aligned}&\left|a-b\right|+\left|c-d\right| \geq \left|a-b+c-d\right| = \left|a+c-(b+d)\right| \\ &\geq \left|a+c\right|-\left|b+d\right|.\end{aligned} \tag{3}$$

Expanding the definition of the *MAE* norm in (1), and considering the standard dimensions of 16 × 16 elements for the blocks, we have

$$MAE = \sum_{l=1}^{16}\sum_{k=1}^{16}\left|x_{l,k} - y_{i+l,j+k}\right| \tag{4}$$

Rearranging the sum and recursively applying inequality (2), we have

$$\begin{aligned}&\sum_{l=1}^{16}\sum_{k=1}^{16}\left|x_{l,k} - y_{i+l,j+k}\right| \geq \sum_{l=1}^{8}\sum_{k=1}^{8}\left|a_{(l,k)}^{(2)} - b_{(l,k,i,j)}^{(2)}\right| \\ &\geq \sum_{l=1}^{4}\sum_{k=1}^{4}\left|a_{(l,k)}^{(4)} - b_{(l,k,i,j)}^{(4)}\right| \geq \sum_{l=1}^{2}\sum_{k=1}^{2}\left|a_{(l,k)}^{(8)} - b_{(l,k,i,j)}^{(8)}\right| \\ &\geq \left|a_{(1,1)}^{(16)} - b_{(1,1,i,j)}^{(16)}\right|\end{aligned} \tag{5}$$

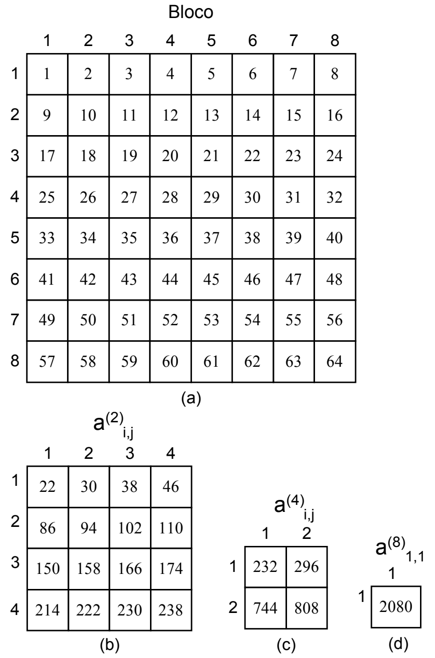
where the terms $a_{(p,q)}^{(M)}$ and $b_{(p,q,r,s)}^{(M)}$ are defined by

$$a_{(p,q)}^{(M)} = \sum_{m=1}^{M}\sum_{n=1}^{M} x_{m+M(p-1),n+M(q-1)} \tag{6}$$

$$b_{(p,q,r,s)}^{(M)} = \sum_{m=1}^{M}\sum_{n=1}^{M} y_{m+M(p-1)+r-1,n+M(q-1)+s-1} \tag{7}$$

In the application of the proposed discard criteria, we begin with the criteria that demand less computation, reducing the amount of blocks to be tested with the more complex criteria. The last discard criterion to be used corresponds to the *MAE*'s calculation between blocks **X** and **Y**(i,j). If this is smaller than MAE_0 (initially taken as reference), this value of *MAE* becomes the new reference to be used in the application of the discard criteria to the remaining blocks **Y**(i,j). The smaller the *MAE* value used as reference, the more effective is the discard criteria, as we can observed from Fig. 1.

For the application of the proposed discard criteria there is a need to compute the terms $a^{(M)}(.,.)$ and $b^{(M)}(.,.,.,.)$ defined, respectively, for the block **X** and for the blocks **Y**(i,j). These terms correspond to the sums of the pixels contained in sub-blocks of M × M elements (M = 16, 8, 4 and 2). The partial sums corresponding to block **Y**(i,j) are not changed in the estimation process corresponding to different blocks **X**, so that the terms $b^{(M)}(.,.,.,.)$ are not modified, and can be calculated in the pre-processing stage. To compute the $b^{(M)}(.,.,.,.)$ terms, we add the pixels in **Y**(i,j) in groupings of 2×2 elements (M=2). To calculate the following stage (adding the pixels in groupings of 4 x 4 elements), we need not to apply Eqs. (6) and (7) directly, since we can use the results of the previous groupings (M=2) that already contains partial sums. We just need to compute the sums of 2 × 2 elements in these groupings. Proceeding this way, we can have efficient computation of the cumulative terms $b^{(M)}(.,.,.,.)$ needed for the application of the discard criteria. Fig. 2 exemplifies the process of calculation of cumulative terms $a^{(M)}(.)$ for a block **X** with 8 × 8 pixels.



Bloco

	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23	24
4	25	26	27	28	29	30	31	32
5	33	34	35	36	37	38	39	40
6	41	42	43	44	45	46	47	48
7	49	50	51	52	53	54	55	56
8	57	58	59	60	61	62	63	64

(a)

$a^{(2)}_{i,j}$

	1	2	3	4
1	22	30	38	46
2	86	94	102	110
3	150	158	166	174
4	214	222	230	238

(b)

$a^{(4)}_{i,j}$

	1	2
1	232	296
2	744	808

(c)

$a^{(8)}_{1,1}$

	1
1	2080

(d)

Fig. 2. Example of computation of cumulative terms defined by Eq. (6): (a) block X with 8 × 8 pixels; (b) groupings of 2 × 2 elements M=2); (c) groupings of 4 × 4 elements (M=4); (d) grouping of 8 × 8 elements (M=8).

B. Pre-ordering of Blocks

We have already pointed out that the smaller the *MAE* value taken as reference for the application of the discard criteria, the more effective these criteria become, since the reference *MAE* value is used to delimit the subspace region corresponding to blocks still under research (Fig. 1).

From the inequalities in (5), we can write

$$MAE = \left\| \mathbf{X} - \mathbf{Y}_{(i,j)} \right\| \geq \left| \left\| \mathbf{X} \right\| - \left\| \mathbf{Y}_{(i,j)} \right\| \right| . \qquad (8)$$

From (8) we see that the *MAE* value corresponding to blocks **X** and **Y**(i,j) is always greater than the magnitude of the difference between the norms of **X** and **Y**(i,j). To perform motion estimation from frame A to frame B, we initially compute the norms of all blocks **Y**(i,j), ||**Y**(i,j) ||, in frame B. Then we place these norms in increasing order. These are the $b^{(16)}(.)$ terms. This task may be cumbersome but it will be useful for the motion estimation of all blocks in frame A. In typical video sequences, it is worth the computational cost. Now consider the steps needed to find the estimate of the motion vector corresponding to a particular block **X**. The norm of **X** is taken as a constant. The search is initiated with the **Y**(i,j) block whose norm ||**Y**(i,j) || minimizes | ||**X** || - ||**Y**(i,j) || |. Call it block $\mathbf{Y}_1$. The *MAE* between **X** and $\mathbf{Y}_1$ is computed and taken as the initial reference *MAE* (denoted by MAE_{ref}).

We note that the initial ordering operation has a twofold purpose: it establishes a particular order for the search process, and it automatically performs the first discard criterion on the candidate blocks. To proceed with the search only those blocks **Y**(i,j) whose norms are in the interval delimited by

$$\left| MAE_{ref} - \left\| \mathbf{X} \right\| \right| \leq \left\| \mathbf{Y}_{(i,j)} \right\| \leq MAE_{ref} + \left\| \mathbf{X} \right\| \qquad (9)$$

need to be considered.

To continue the search with the next discard criterion (based on the $b^{(8)}(.)$ terms), we pick the block that is immediately above $\mathbf{Y}_1$ in the ordered norm list. Call it $\mathbf{Y}_2$. Then we compute the terms involving $a^{(8)}$ and $b^{(8)}$ for **X** and $\mathbf{Y}_2$, respectively, and calculate the projected distance between **X** and $\mathbf{Y}_2$ that corresponds to the second criterion (the 2×2 sum in the series of inequalities (5)). If this distance is greater than MAE_{ref}, then $\mathbf{Y}_2$ can be discarded. Otherwise, we can proceed to apply the third criterion to $\mathbf{Y}_2$, which is that based on the $b^{(4)}(.)$ terms. If $\mathbf{Y}_2$ passes the third criterion we move on to the fourth, and so on, until the *MAE* of $\mathbf{Y}_2$ is calculated. If that, too, is smaller than MAE_{ref}, then it becomes the new MAE_{ref}. When this happens the set of candidate blocks is reduced according to (9).

The algorithm continues with the block immediately below $\mathbf{Y}_1$ in the ordered norm list. Call it $\mathbf{Y}_3$. Then criteria 2, 3 and so on are successively applied to $\mathbf{Y}_3$. If $\mathbf{Y}_3$ passes all criteria, then its *MAE* will also replace MAE_{ref}.

We proceed by taking blocks alternately above and below $\mathbf{Y}_1$ in the ordered norm list, and updating MAE_{ref} accordingly, until no more blocks are contained in the candidate list given by (9). At that point, the current MAE_{ref} is the global minimum value of *MAE*, and the differences in coordinates between **X** and the corresponding **Y** yields the estimated motion vector for **X**.

This empirical scheme for the order in which blocks are submitted to the sequence of discard criteria is aimed at increasing the probability that the block with the minimum *MAE* will be evaluated early in the process.

The above description corresponds to the application of the algorithm when the research area is the entire picture frame. The algorithm can also be applied to smaller areas of research, but in this case the pre-ordering of the blocks may prove to be exceedingly time consuming, since the research areas vary from block to block, and a different ordering is required for each block **X**. Thus, to apply the algorithm to restricted research areas of different sizes, we have opted to eliminate the pre-ordering stage in these cases. Accordingly, we apply the successive discard criteria to the candidate blocks within the research area in a standard raster scan sequence, from left to right and top to bottom.

To give a simple example of the algorithm, let us consider a set of 5 blocks of dimensions 4×4 indexed from (a) to (e), from which the closest to block (f), according to the *MAE* criterion, is to be selected. The example is depicted in Fig. 3, where the corresponding norms are indicated just above the blocks. As block (f) shows a norm of 27, the closest block in norm is block (c), whose norm equals 30. This will be equivalent of $\mathbf{Y}_1$. Computing the *MAE* between these blocks we find the value 17. Therefore, only blocks whose norms are in the interval [30-17, 30+17] = [10, 44] need to be considered further. These include blocks (b), (c) and (d) as shown in Fig. 3. Following the described algorithm, block (d) is discarded, and block (b) is found to give the minimum *MAE*.

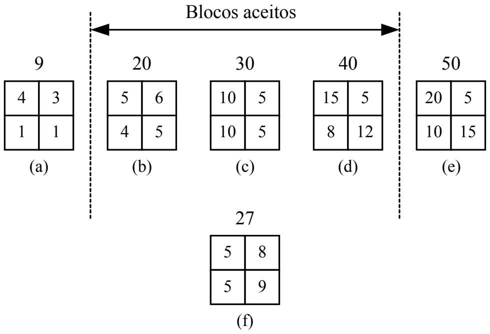


Fig. 3. Example with 5 candidate blocks.

III. SIMULATIONS

To verify the effectiveness of the algorithm, we compare the results obtained with the exhaustive search algorithm (ESA), the successive elimination algorithm (SEA), and the modified SEA. These algorithms were applied to research areas varying through 16×16, 32×32, 64×64, 128×128 pixels and the whole 720×480 frame. For the smaller research areas, we observe a significant improvement in the speed of the estimation process without any noticeable degradation in the quality of the sequences.

The simulations were performed with monochrome frame sequences in CCIR 601 format, of dimensions 720 x 480 pixels, and depth of 8bpp. The sequences were extracted from the standard video sequences known as "Flower Garden", "Mobile Calendar", "Kiel" and 'Tennis." The motion estimation is considered from frame 1 to frame 2. A PC Pentium III computer 500MHz and with 128Mbytes of RAM memory was used for the accomplishment of the simulations. The results in terms of processing time are shown in Tables I through IV.

TABLE I

Processing Time (s) – "Flower"

	Width of the Area of Research				
Algorithm	**16**	**32**	**64**	**128**	**All**
ESA	11	43	160	566	3700
SEA	2.8	6.4	15.2	36	144
SEA MOD	1.4	2.1	3.8	8.1	13

TABLE II

Processing Time (s) – "Kiel"

	Width of the Area of Research				
Algorithm	**16**	**32**	**64**	**128**	**All**
ESA	12	43	160	565	3696
SEA	6.2	18	52	147	678
SEA MOD	2.3	4.7	10.8	27	56

TABLE III

Processing Time (s) – "Mobile"

	Width of the Area of Research				
Algorithm	**16**	**32**	**64**	**128**	**All**
ESA	11	43	160	566	3699
SEA	5.6	16.4	48.2	131	558
SEA MOD	2.1	4.3	10	22.9	41

TABLE IV

Processing Time (s) – "Tennis"

	Width of the Area of Research				
Algorithm	**16**	**32**	**64**	**128**	**All**
ESA	11	43	160	566	3690
SEA	7.4	23	66.4	168	645
SEA MOD	3.9	10	24.7	54.5	119

As can be observed, the modified SE algorithm is able to reduce the processing time for motion estimation in all the analysed sequences and all the considered research-area sizes. The reduction becomes more accentuated as the width of the research area increases, so that the pixels become less correlated, which tends to improve the efficiency of the discard criteria. On the other hand, we note that, in general, the larger the research area, the better the accomplished motion estimation, and the better the rate × distortion performance of the encoded sequence.

The smallest processing time was found for the sequence “Flower”, since the motion of its scene objects tends to follow a linear, appropriately horizontal translation, adequate for motion estimation with pixel blocks. We have obtained gains in computation speed varying from approximately 8 times, for a 16×16 research area, to approximately 284 times, when the algorithm is applied to the entire frame. For the other sequences, where the model of motion of the scene objects does not follow a linear translation (“Kiel with motion caused by zoom, and “Mobile” and “Tennis” with motion characterized by vertical and horizontal translations), the speed gains were smaller. Nevertheless, the performance of MOD SEA was, in all cases, superior to those of SEA and ESA.

With respect to the quality of the reconstructed frames, Table V shows the corresponding peak signal-to-noise ratio (PSNR) in dB.

TABLE V

Quality of the reconstructed frame (PSNR (dB))

	Width of the research area				
Sequence	**16**	**32**	**64**	**128**	**All**
Flower	34.91	34.91	34.91	34.91	34.91
Kiel	27.03	27.04	27.04	27.04	27.04
Mobile	24.9	25.00	25.06	25.12	25.17
Tennis	29.35	29.62	30.03	30.08	30.14

The values of PSNR obtained for the three algorithms were identical. This is expected since the three algorithms achieve the optimal solution for the motion vectors within each research area.

We note that the PSNR values are essentially unchanged for the different research area values. In the case of the “Tennis” sequence, an improvement of 0.79 dB is observed when the research area varies from 16×16 to the entire frame.

The processing overhead for the computation of the various $b^{(M)}(.)$ terms corresponding to all blocks in the frame (for M=16, 8, 4, and 2) and the pre-ordering stage (used when the research area is the full frame) is shown in Table VI.

TABLE VI

Processing Overhead

	Sequence			
Algorithm	**Flower**	**Kiel**	**Mobile**	**Tennis**
SEA MOD	12.09%	2.51%	3.41	1.23%

The pre-processing phase represents a fixed computational cost of the modified SE algorithm. For this reason, sequences with shorter processing times tend to present larger values of overhead. In practice, the pre-processing overhead imposes a limit on the maximum speed that can be achieved by the algorithm

IV. CONCLUSIONS

In this article, the performance of the modified successive elimination algorithm (MOD-SEA) is compared with respect to those of the exhaustive search algorithm (ESA) and of the successive elimination algorithm (SEA), for various sizes of research area. The modified SE algorithm constitutes an evolution of the SE algorithm that incorporates an increased number of discard criteria and a pre-ordering phase (for the case of unrestricted research area). The use of more discard criteria reduces the computational cost of the estimation process as it eliminates blocks with less computer operations than those required by the *MAE* distortion measure. Also, the pre-ordering of the blocks increases the efficiency of the discard criteria.

We find that the modified SE algorithm achieves significant reductions in the computational cost of motion estimation in comparison with ESA and SEA, while still obtaining the optimal motion vector within the considered research area.

REFERENCES

[1] R. T. Del Pietro and M. H. M. Costa “Fast algorithm for the estimation of motion vectors,” (in Portuguese), *Proceedings of the 19th Brazilian Telecomm. Symposium*, Fortaleza, CE, Sept. 2001.

[2] W. Li and E. Salari, “Successive elimination algorithm for motion estimation,” *IEEE Trans. Image Processing, vol.* 4, pp 105-07, Jan. 1995.

[3] N. Natravali and J. D. Robbins, “Motion compensated television coding: Part I,” *Bell Syst. Tech. J.*, vol. 58, pp. 631-670, Mar. 1979.

[4] H. G. Musmann, P. Pirsh, and H. J. Gilbert, “Advances in picture coding,” *IEEE Trans. Image Processing*, vol. 4, pp. 105-107, Jan. 1995.

[5] J. R. Jain and A. K. Jain, “Displacement measurement and its application in interframe

coding," *IEEE Trans. Commun.,* vol. 29, pp. 1799-1808, Dec. 1981.

[6] CCITT Standard H.261, "Video coding for audiovisual services at px64kbits/s," ITU, 1990.

[7] ITU-T Draft Standard H.263, "Video coding for narrow telecommunication channel at (below) 64kbit/s," ITU, Apr. 1995.

[8] ISO-IEC JTC1/SC2/WG11, "Preliminary text for MPEG video standard coding," ISO, Aug. 1990.

[9] M. Ghanbari, "The cross-search algorithm for motion estimation," *IEEE Trans. Commun.,* vol. 38, pp. 950-953, July 1990.

[10] K. H. K. Chow and M. L. Liou, "Generic motion search algorithm for video compression," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 3, pp.440-446, Dec. 1993.

[11] T. M. Apostol, Mathematical Analysis, Reading, MA: Addison-Wesley, 1975.