Global Motion Estimation for Image Sequence Coding Applications[1]

Zvi Eisips and David Malah

Department of Electrical Engineering
Technion I.I.T.
Haifa 32000, ISRAEL

Abstract

This paper describes two algorithms for the estimation of global motion in image sequences. Global motion compensation improves the image prediction while adding only a small amount of side-information in an image sequence coder, thus saving bitrate and/or improving quality. A 3-parameter model is considered for global motion representation, including zoom ratio and horizontal and vertical pan. Both algorithms are based on block displacement estimates and take into account superimposed local motion. The first algorithm selects a set of most reliable block displacements and applies a Least Squares (LS) scheme to obtain an initial estimate of the model parameters out of these displacements. Then a Verification Stage is used to discard blocks whose displacements are not compatible to this estimate. Finally a LS scheme, using only the verified block displacements, is applied to obtain a finer estimate of the model parameters. The second algorithm uses a Hough Transform to separate background blocks whose block displacements are processed by a LS scheme to obtain the model parameters. Simulation results are given for the "Table-Tennis" image sequence containing zooming and show a rate reduction of over 20% for this sequence.

I. Introduction

Image sequence coding has attracted a great deal of attention in recent years. The development of fast, low-cost digital processing devices permits the realization of real-time image processing systems capable of encoding Videophone signals at low rates down to 64 kbps [7]. While Videophone produces a low complexity image sequence, general TV scenes are much more complex. The requirement to encode TV images, with satisfactory quality, at rates of about 1 Mbps is dictated by the throughput of CD-ROM systems, which are one of the most promising storage devices for future video applications. CD-ROM video storage, combined with an appropriate algorithm can also provide features like random access, fast search and reverse playback [1].

Image sequence compression exploits both spatial and temporal redundancy. Spatial redundancy is mostly exploited by transforming the image block and quantizing the transform coefficients. The Discrete Cosine Transform (DCT) was shown to be particularly effective in image coding [2] and is widely used. Temporal redundancy is exploited by interframe prediction using motion compensation algorithms which estimate motion in the scene and predict the coded image based on the previous reconstructed image and motion estimates. Motion estimation is a complicated task by itself, but when the objective is image compression, coding constraints must also be taken into account. Every motion field or pattern can be approximated by translatory motion of sufficiently small blocks or image pieces but the coding cost of this translatory motion of many small blocks may be prohibitively high. This is why translatory motion is generally estimated on 8x8 or 16x16 blocks, using Block Matching (BM) [3]. A prediction image is then created from blocks in the previous image using this translatory motion estimation. As motion in image sequences does not generally follow a block translation scheme, and as new objects may enter the image and therefore cannot be predicted, it is necessary to add a correcting step to cope with image areas where the prediction error is high. The BM technique may also cause visually disturbing effects such as blocking and "sticking" noise.

Sometimes, most of the image changes in the image sequence are caused by camera motion or zooming. These affect the whole image, and the motion they induce in the image is called *global motion*, in distinction from local motion caused by moving objects. With proper estimation of the global motion, a good prediction image can be obtained based on the previous image and a small amount of side information needed to represent the global motion parameters. The global transformation applied to the previous image gives smoothly changing displacement values thus alleviating the blocking effect. Following the compensation of the global motion, local motion can be dealt with as before.

Global motion estimation is being investigated as it appears to have the potential of providing an important contribution to image sequence coding algorithms [4]. We propose here two global motion estimation algorithms which are presented below as algorithms A and B. Both algorithms are based on BM. Three parameters of the global motion are estimated and used in the prediction step to improve the coder performance in terms of rate and/or quality. The three parameters are the *zoom ratio* (expansion or contraction of the image) and the *pan* in vertical and horizontal directions (global translation of the whole image). Superimposed local motion can hinder the estimation of the global motion parameters and is taken into consideration by the proposed algorithms. In [5] a differential, iterative approach is used to estimate zoom and pan parameters but is complex.

Sections II and III describe the two proposed algorithms. Section IV shows simulation results and section V concludes the paper.

II. Algorithm A

Displacement estimates for all the blocks in the current input image are first obtained by a full search BM algorithm which minimizes the MAD (Mean Absolute Difference) between a block in the current original image and a block in the previous original image, using a search window of ±7 pixels and 8x8 pixel blocks. The displacement estimates obtained do not represent the real motion of the scene. Some areas of the image cannot be well matched due to several reasons, such as motion greater than ±7 pixels, or a new part of the scene entering the image area because of camera motion or because of an object moving into the camera view. Other blocks are not well matched as a result of rotational motion or non-rigid body motion. This is why we should

[1]This work was partially supported by RAFAEL (050-704).

use only some of the blocks in the image and their displacement estimates for further processing to obtain good estimates of the global motion parameters.

A block is well matched when the obtained MAD is lower than a threshold. The match will be most reliable when this MAD is significantly better than the MAD without motion compensation. This may be seen as a way of avoiding the use of displacement estimates obtained for smooth image areas, as well as for noise-like textured areas, where real motion cannot be properly estimated by a local matching algorithm as BM.

After selecting the blocks with most reliable displacements we use them to estimate the global motion parameters. We consider the center of a block as moving from the matched position in the previous frame to its present position in the current frame. To explain this motion by global motion we write for a block :

$$\Delta_x = (X - X_o)(1-\xi) + P_x \qquad (1a)$$

$$\Delta_y = (Y - Y_o)(1-\xi) + P_y \qquad (1b)$$

where ξ is the zoom ratio; P_x,P_y are the horizontal and vertical pan parameters; X, Y are the coordinates of the center of the matched block in the previous image; Δ_x,Δ_y are the horizontal and vertical displacements (given from the BM algorithm); and X_o, Y_o are the coordinates of the center of the image (focus of expansion/contraction).

For each selected block we have two equations, having a total of 2L equations, where L is the number of selected blocks, but only 3 global motion parameters to estimate, so a Least Squares (LS) scheme is applied to find the best fit parameters. In matrix notation the equations are given by:

$$\underbrace{\begin{bmatrix} X^1 - X_o & 1 & 0 \\ Y^1 - Y_o & 0 & 1 \\ X^2 - X_o & 1 & 0 \\ Y^2 - Y_o & 0 & 1 \\ \vdots & \vdots & \vdots \\ X^L - X_o & 1 & 0 \\ Y^L - Y_o & 0 & 1 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} (1-\xi) \\ P_x \\ P_y \end{bmatrix}}_{\underline{a}} = \underbrace{\begin{bmatrix} \Delta_x^1 \\ \Delta_y^1 \\ \Delta_x^2 \\ \Delta_y^2 \\ \vdots \\ \Delta_x^L \\ \Delta_y^L \end{bmatrix}}_{\underline{d}} \qquad (2)$$

and the Least Squares solution is given by :

$$\underline{a} = \left(M^T M\right)^{-1} M^T \underline{d} \qquad (3)$$

An improvement of the above estimation of ξ,P_x,P_y is done by adding a Verification Stage which compares, for each block, the displacement estimates Δ_x, Δ_y with the displacements computed from the estimated global motion parameters. Those which are close enough define a reduced set of blocks whose displacement estimates are used again in a LS estimation scheme. The point in this Verification Step is to separate blocks that satisfied the above criteria, as having reliable displacement estimates, but do not follow the global motion. These blocks are mostly situated on locally moving objects. Therefore, by separating them out we alleviate the effect of locally moving objects on the estimation of the global motion parameters.

Algorithm A was found to perform well when moving objects occupy only a sufficiently small portion of the scene (e.g., less than 30% of the selected blocks). When this is not the case, locally moving objects can affect the initial estimate of the global motion parameters in a way that only a few, or even no block displacement estimates, pass the verification stage. In this case blocks affected solely by global motion (usually called background blocks) must be found before estimating the model parameters. For this situation the following algorithm is proposed :

III. Algorithm B

This algorithm uses a 3-D Hough Transform [6]. The purpose of this transform is to find an initial estimate of the global motion parameters, based only on blocks which have motion coinciding with the global motion. The transform is three dimentional, where each dimention corresponds to one of the model parameters, namely, zoom ratio, pan in horizontal direction and pan in vertical direction.

The Hough Transform involves two steps. The first one is a voting step in which each data mesurement (in our case block displacement measurements) "votes" for each of the values in the parameter space that are compatible with this measurement. The parameter space is represented by a 3-D accumulator array, where each array cell represents a 3-D segment (or cube) of the parameter space and accumulates "votes" given to it. In the classical Hough Transform [6] (used to find lines in an image) each point in the input image "votes" for all the lines passing through it. In our case each block displacement measurement votes for all parameter sets (a triplet made of a zoom value, an horizontal pan value and a vertical pan value) which could have caused such a displacement for this specific block. The voting is done in the following way:

Let $\xi^i = \xi^{min} + i \cdot \xi^{step}$, $i = 1,2,...,N_\xi$, be the values represented by the zoom dimention in the Hough parameter space where ξ^{min} is the minimum possible value for the parameter ξ; ξ^{step} is the difference between consecutive accumulator center values of ξ; and N_ξ is the number of accumulator cells in the ξ dimention. Then, for each ξ^i, we calculate the pan values P_x^i and P_y^i which would result if the zoom ratio is ξ^i as:

$$P_x^i = -(X - X_o)\left(1-\xi^i\right) + \Delta_x \qquad (4a)$$

$$P_y^i = -(Y - Y_o)\left(1-\xi^i\right) + \Delta_y \qquad (4b)$$

where X, Y are the coordinates of the center of the matched block in the previous image; Δ_x,Δ_y are the horizontal and vertical displacements (given from the BM algorithm) and X_o, Y_o are the coordinates of the center of the image. We then give a vote to the accumulator cell (i,j,k), where :

$$j = \left[\frac{P_x^i - P_x^{min}}{P_x^{step}}\right] \qquad (5a)$$

$$k = \left[\frac{P_y^i - P_y^{min}}{P_y^{step}}\right] \qquad (5b)$$

and $[u]$ denotes here rounding of u to the closest integer; P_x^{min},P_y^{min} are the minimum values for the parameters P_x,P_y; and P_x^{step},P_y^{step} are the differences between consecutive accumulator center values of P_x,P_y, respectively.

The second step in the Hough Transform is the interpretation of the accumulators content obtained in the first step. The accumulator cell containing the largest peak is searched for and denoted $(i_{max}, j_{max}, k_{max})$. The estimated values of the global motion parameters ξ,P_x and P_y are given by:

$$\xi = \xi^{min} + i_{max} \cdot \xi^{step} \quad (6a)$$

$$P_x = P_x^{min} + j_{max} \cdot P_x^{step} \quad (6b)$$

$$P_y = P_y^{min} + k_{max} \cdot P_y^{step} \quad (6c)$$

Because of the coarse resolution used in the Hough Transform this result is only an approximate one. The main role of the Hough Transform is the separation of background blocks from which a better estimate of the global motion parameters can be computed. All blocks that "voted" for the cell $(i_{max}, j_{max}, k_{max})$ are used by a LS scheme as the one in Algorithm A to obtain a better estimate of the global motion parameters.

IV. Simulation Results

Algorithms A and B were implemented and embedded in a hybrid/DCT image sequence coder similar to the one described in [7]. The image sequence used in our simulation was the Table-Tennis ISO/MPEG sequence [1], and the algorithms were applied to frames 21 to 70 because these frames contain zooming.

The global motion parameters estimated by the above algorithms were used to create a global-motion-compensated image based on the previous reconstructed image at the decoder. This compensation may need the application of spatial interpolation when performing the global motion compensation for zooming and/or pan with sub-pixel displacements. This may cause distortions in the interpolated image due to excessive edge smoothing if, for example, a bi-linear interpolation scheme is used, or disappearance or duplication of some rows and columns – if a zero-order interpolation approach is used. This is why some blocks can still be better matched to blocks in the previous reconstructed image (using local motion compensation) than to blocks in the global-motion-compensated image, even if the exact parameters are known.

In the image sequence coder used in the simulations the BM algorithm (with block size 16x16) is applied twice: once between the original image and the previous reconstructed image, and once between the original image and the global-motion compensated image (besides the BM between original images used for global motion estimation as explained before). The prediction image is built by choosing the best matched block from these two images. This adds approximately one bit of side-information per block (to inform the receiver from which image the block was selected). The simulations showed that an average of 72% of the blocks were chosen from the global-motion-compensated image. Image blocks whose prediction MAD error is higher than a threshold (called correcting threshold) are either intra-frame or inter-frame coded by a DCT following CCITT Recommendation H.261 [7].

The same image sequence coder was run without the global motion estimation and compensation while keeping the same coding parameters, namely the correcting threshold and the quantization step for the DCT quantized coefficients. Fixing these parameters keeps the image quality approximately constant while allowing rate fluctuations. Results obtained with and without Global Motion Compensation (GMC) are shown in Table 1 where the global motion estimation was done using Algorithm B. Results obtained using Algorithm A were practically the same (837 Kbits/sec instead of 822 Kbits/sec). It is seen from the table that a saving of more than 20% in rate is achieved by using global motion compensation.

Table 1: Image sequence coder performance.

	Peak SNR (dB)	Rate (Kbits/sec)
GMC	32.80	822
No GMC	32.33	1,068

Picture 1 shows one image (no. 44) of the image sequence, where the blocks selected as having the most reliable displacement estimates are shown lighted (the thresholds used are: MAD lower than 3 gray levels per pixel and MAD being 3 times better than the MAD without motion compensation). Block displacements are also shown by small white arrows (motion vectors) on each block. The radial displacement pattern generated by the zoom is clearly seen. It can also be seen that many selected blocks lie on the moving player, thus following local player's motion *combined* with global motion. When all these block displacements are used by the LS scheme in Algorithm A, the resulting motion parameters are ξ=0.9744, P_x=-1.0 and P_y=-0.6 pixels, which are obviously incorrect pan values, since the actual pan is zero – as background blocks in the center of the image are seen to have zero displacement. The Verification Step of Algorithm A verifies here only 14 blocks, as shown in picture 2, which are not located on the background as they should. The verification is done allowing a disparity, between the actual displacements and the displacements calculated from the above estimated global parameters, of 0.0025 in zoom and 0.5 pixel in pan. A second LS estimation using these 14 blocks gives the result of ξ=0.9745, P_x=-1.0 and P_y=-0.7 which again, of course, are not the correct pan values. This demonstrates the problem caused by a relatively large object moving in the scene, as explained earlier.

The same blocks of Picture 1 are now used in a coarse Hough Transform (Algorithm B) having a resolution of ξ^{step}=0.005 and $P_x^{step}=P_y^{step}$=1.0, (with $P_x^{min}=P_y^{min}$=-5 and ξ^{min}=0.95) which results in the largest peak being at ξ=0.975, P_x=0.0 and P_y=0.0. This peak, having a value of 110, (i.e. 110 block displacements voted for this accumulator cell) is produced by the highlighted blocks in picture 3, which are all on the background, as desired. The LS scheme applied to these blocks is therefore expected to give a better estimate of the global motion parameters than the estimate given by Algorithm A. The result of the LS computation using these 110 blocks is ξ=0.9758, P_x=0.2 and P_y=-0.1, which is certainly a much better result than the one obtained by Algorithm A. For demonstration, these parameter values were used to create a global motion compensated image out of the original image no. 43. Displacement estimates given by a BM algorithm between this global motion compensated image and image no. 44 are shown in Picture 4 where it is seen that almost all the background blocks are given a zero displacement, while blocks on the player receive displacements according to local player's motion.

V. Conclusions

Two algorithms for the estimation of global motion in image sequences were presented. The use of these algorithms to improve image sequence coder performance was examined by computer simulations showing promising results.

Both algorithms are based on motion estimates given from a Block Matching algorithm and use simple calculations on these motion estimates, making the

algorithms very attractive even for real-time applications.

While Algorithm B is more accurate than Algorithm A as it overcomes better the difficulty of local superimposed motion (as demonstrated in the example given in the previous section), the overall performance of both algorithms is similar for the image sequence used. This can be explained by the fact that the zoom ratio values obtained by the two algorithms are very close whereas if the pan values are not correctly estimated, they can be partially corrected by local block displacements when BM is applied to the global-motion compensated image. The fact that the image sequence used in the simulations does not contain objects moving in the camera direction (which would induce motion similar to zoom) helps Algorithm A to obtain good zoom parameters while the local motion of the player affects mostly the estimation of the pan values.

Acknowledgement

The authors wish to thank Dr. A. Saad and Mr. I. Florentin, both of RAFAEL, for fruitful discussions held during the course of this work. The partial support given to this research by RAFAEL is greatfully acknowledged.

References

[1] A. Nagata, I. Inoue, A. Tanaka and N. Takeguchi, "Moving Picture Coding System for Digital Storage Media Using Hybrid Coding", Image Communication, Vol. 2, pp. 109-116, 1990.

[2] W. Chen and C. Smith, "Adaptive Coding of Monochrome and Color Images", IEEE Trans. Commun., Vol 92, Nov. 1977.

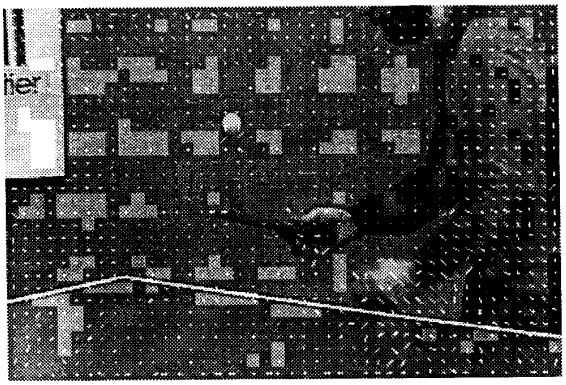
[3] J. R. Jain and A. K. Jain, "Displacement Measurement and its Applications in Interframe Image Coding", IEEE Trans. Commun., Vol. 29, pp. 1799-1808, Dec. 1981.

[4] C. Herpel, D. Hepper and D. Westerkamp, "Adaptation and Improvement of CCITT Reference Model 8 Video Coding for Digital Storage Media Applications", Image Communication, Vol. 2, pp. 171-185, 1990.

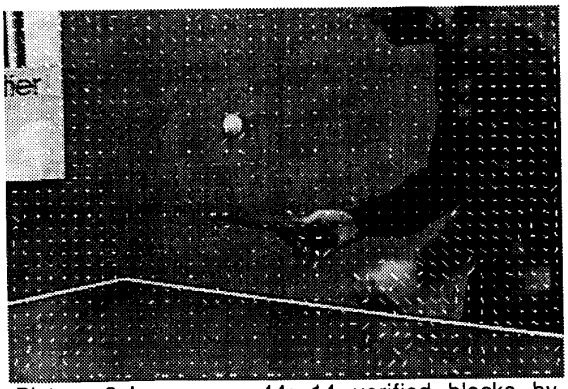
[5] M. Hoetter, "Differential estimation of the global motion parameters zoom and pan", Signal Processing 16, pp. 249-265, 1990.

[6] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to detect lines and curves in pictures", Comm. of ACM, Vol. 15, pp. 11-15 , 1972.

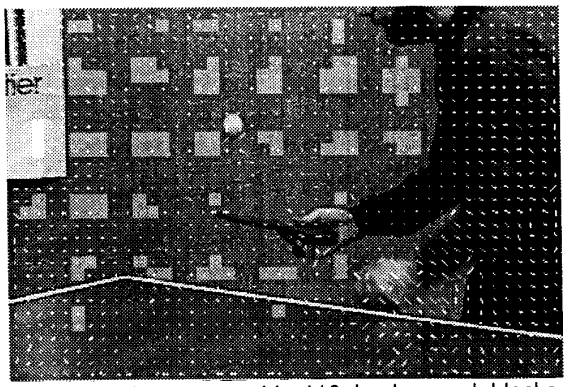
[7] "Draft Revision of Recommendation H.261: Video Codec for Audiovisual Services at p_x64 kbits/s", Image Communication, Vol. 2, pp. 221-239, 1990.



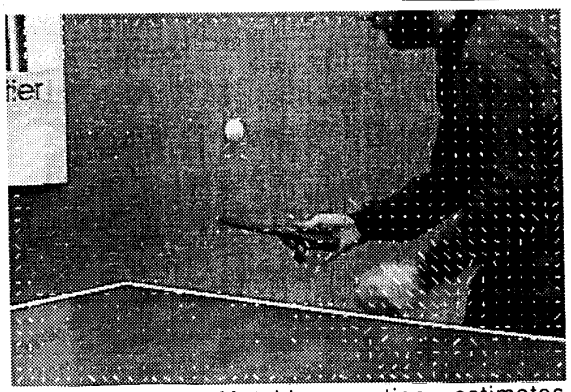
Picture 1: Image no. 44: Most reliable block displacements (lighted)



Picture 2: Image no. 44: 14 verified blocks by Algorithm A Verification Stage (lighted)



Picture 3: Image no. 44: 110 background blocks separated by the Hough Transform of Algorithm B (lighted).



Picture 4: Block Matching motion estimates (white arrows) between global motion compensated image and original image.