World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:11, 2009

Efficient Block Matching Algorithm for Motion Estimation

Zong Chen

Open Science Index, Computer and Information Engineering Vol:3, No:11, 2009 publications.waset.org/1581.pdf

***Abstract*—**Motion estimation is a key problem in video processing and computer vision. Optical flow motion estimation can achieve high estimation accuracy when motion vector is small. Three-step search algorithm can handle large motion vector but not very accurate. A joint algorithm was proposed in this paper to achieve high estimation accuracy disregarding whether the motion vector is small or large, and keep the computation cost much lower than full search.

***Keywords*—** Motion estimation, Block Matching, Optical flow, Three step search.

I. INTRODUCTION

MOTION is a prominent source of temporal variations in image sequences. Motion in image sequences acquired by a video camera is induced by movements of objects in a 3D scene and by camera motion. Thus, camera parameters, such as its 3D motion (rotation, translation) or focal length, play an important role in image motion modeling. If these parameters are known precisely, only object motion needs to be recovered. However, this scenario is rather rare and both object and camera motions usually need to be computed. The 3D motion of objects and camera induces 2D motion on the image plane *via* a suitable projection system. It is this 2D motion, also called *optical flow*, which needs to be recovered from intensity and color information of a video sequence.

2D motion finds diverse applications in video processing and compression as well as in computer vision. In video processing, motion information is used for standards conversion (motion-compensated 3D sampling structure conversion), noise suppression (motion-compensated filtering) or deblurring (motion-compensated restoration). In computer vision, 2D motion usually serves as an intermediary in the recovery of camera motion or scene structure. In video compression, the knowledge of motion helps remove temporal data redundancy and therefore attain high compression ratios.

In video compression, consecutive video frames are similar except for changes induced by objects moving within the frames. In the trivial case of zero motion between frames (and no other differences caused by noise, etc.), it is easy for the encoder to efficiently predict the current frame as a duplicate of the prediction frame. When this is done, the only information necessary to transmit to the decoder becomes the syntactic overhead necessary to reconstruct the picture from the original reference frame. When there is motion in the images, the displacement of moving objects between successive frames will be estimated (motion estimation) first. The resulting motion information is then exploited in efficient inter-frame predictive coding (motion compensation). Consequently, the prediction error is transmitted. The motion information also has to be transmitted, unless the decoder is able to estimate the motion field. An efficient representation of the motion is thus critical tin order to reach high performance in video compression.

In the field of motion estimation, many techniques have been applied [1]-[5]. Basically, these techniques can be divided into four main groups: gradient techniques, pixel-recursive techniques, block matching techniques, and frequency-domain techniques. Among those four groups, block matching is particularly suitable in video compression schemes based on discrete cosine transform (DCT) such as those adopted by the recent standards H.261, H.263 and MPEG family [6].

Optical flow estimation has been extensively researched. Although most optical flow estimation algorithms are gradient (for each pixel) based algorithms, they also can be applied to blocks. Optical flow methods such as Lucas-Kanade's [7] achieve high accuracy for scenes with small displacements but fail when the displacements are large.

In order to achieve high accurate motion estimation without fail when displacements are large, a joint algorithm is proposed in this paper. The rest of the paper is organized as follows. Optical flow algorithm and block matching algorithms will be described in section II and III. The joint algorithm will be presented and the performance will be evaluated in section IV. The simulation results demonstrate the significant achievement of the proposed joint algorithm.

II. OPTICAL FLOW

The hypothesis of optical flow estimation is that the image luminance is invariant between frames.

$$E(X,t) = E(X + \Delta X, t + \Delta t) \tag{1}$$

where X=[x, y]^T. The Taylor series expansion of the right hand side of (1) gives

$$E(X + \Delta X, t + \Delta t) = E(X,t) + \nabla E \cdot \Delta X + E_t \cdot \Delta t \tag{2}$$

where ∇E = [(∂E/∂x), (∂E/∂y)]^T is the gradient operator.

Zong Chen is with School of Computer Sciences and Engineering, Fairleigh Dickinson University, Teaneck, NJ 07666, USA (email: zchen@fdu.edu).

Assuming Δ*t*→0, and defining the motion vector v = $(v_x, v_y)^T = \Delta X/\Delta t$, we obtain

$$\nabla E \cdot v + \partial E / \partial t = 0 \qquad (3)$$

Equation (3) is known as the spatio-temporal constraint equation or the optical flow constraint equation.

As the image intensity change at a point due to motion gives only one constraint, while the motion vector at the same point has two components, the motion field cannot be computed without an additional constraint. Lucas and Kanade [7] implemented a least square fit of local first-order constraints (3) to a constant model for *v* by minimizing

$$L(v) = \sum [\nabla E(p_i) \cdot v + \partial E(p_i) / \partial t]^2 \qquad (4)$$

The solution of (4) is given by

$$(A^T A) v = A^T b \qquad (5)$$

where, for a block with $n \times n$ pixels at time *t*,

$$A = [\nabla E(p_1), \nabla E(p_2), ..., \nabla E(p_{n^2})]^T \qquad (6)$$

$$b = -[\partial E(p_1)/\partial t, \partial E(p_2)/\partial t, ..., \partial E(p_{n^2})/\partial t]^T \qquad (7)$$

If the position of p_i is (*x*, *y*) in frame *t*, then

$$\nabla E(p_i) = \begin{bmatrix} E(x+1, y, t) - E(x, y, t) \\ E(x, y+1, t) - E(x, y, t) \end{bmatrix} \qquad (8)$$

$$\partial E(p_i) / \partial t = E(x, y, t+1) - E(x, y, t) \qquad (9)$$

For a block with $n \times n$ pixels, it requires $2n^2$ addition to calculate the matrix A (E*x* and E*y*), and n^2 addition to calculate the vector b ($-E_t$).

The solution of (5) is $v = (A^TA)^{-1}A^T b$, which is solved in closed form when (A^TA) is nonsingular. Otherwise, $v = (A^TA)^+A^T b$, where $(A^TA)^+$ is the generalized inverse matrix of (A^TA).

In general, for a matrix multiplication $A_{m\times n} \cdot B_{n\times k} = C_{m\times k}$ requires $m\times k\times n$ multiplication and $m\times k\times(n-1)$ addition. Therefore, it needs $2\times2\times n^2$ multiplication and $2\times2\times(n^2-1)$ addition to calculate $(A^T_{2\times n^2} \cdot A_{2\times n^2})$, $2\times1\times n^2$ multiplication and $2\times1\times(n^2-1)$ addition to calculate $(A^T_{2\times n^2} \cdot b_{n^2\times1})$. Also, it needs one 2×2 matrix inverse for $(A^TA)^{-1}_{2\times2}$, plus 2×1×2 multiplication and 2×1×1 addition to calculate $(A^TA)^{-1}_{2\times2} \cdot (A^T b)_{2\times1}$. Therefore, it requires total $6n^2$ multiplication and about $9n^2$ addition, plus one 2×2 matrix inverse.

III. Block matching

In block matching algorithms, each macroblock (8×8 pixels) in the new frame is compared with shifted regions of the same size from the previous frame, and the shift which results in the minimum error is selected as the best motion vector for that macro-block. The motion compensated prediction frame is then formed from all the shifted regions from the previous decoded frame.

In most systems, the “best” match is found using the sum of absolute error (SAE) criterion. The SAE of two blocks, X and Y, with $n \times n$ pixels is defined as

$$SAE(X, Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} | X(i, j) - Y(i, j) | \qquad (10)$$

The computation cost for each SAE calculation is $2n^2$ additions.

There are several well-known algorithms that perform the block matching motion estimation. The full search algorithm (FSA) [3]-[5] that determines the motion vector of a macroblock by computing the SAE at each location in the search area (basically ±7 pixels). This is the simplest method, it provides the best performance, but at a very high computational cost: 15×15=225 SAE calculation, i.e. $225\times2n^2=450n^2$ additions.

Significant computation savings can be made with hierarchical search algorithm, also called three-step search algorithm [8]. In three-step search algorithm, eight positions around a center are tested and the position of minimum distortion becomes the center of the next stage. For a center [*cx, cy*] and step size *s*, the positions [*cx-s, cy-s*], [*cx-s, cy*], [*cx-s, cy+s*], [*cx, cy-s*], [*cx, cy*], [*cx, cy+s*], [*cx+s, cy-s*], [*cx+s, cy*], [*cx+s, cy+s*] are examined. After each stage the step size is reduced.

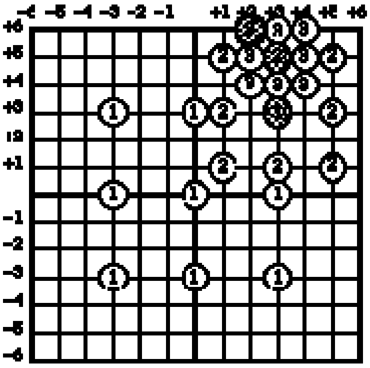


Fig 1. An example of three-step search algorithm.

The maximum displacement is ±6 pixels in both directions (as in Figure 1) with step size 3, 2, 1. This allows the algorithm to halt in three steps and hence its name. There are also variants allow the maximum displacement to be ±7 pixels with step size 4, 2, 1.

Three-step search algorithm needs to calculate total 9×3=27 SAE, i.e., its computation cost is $27\times2n^2=54n^2$ additions.

IV. Simulation and the Joint Algorithm

In order to evaluate the performance of optical flow, full block search and three-step search algorithm, two video sequences were used. One is the "Claire" video sequence, which is composed of one woman talking with small motions before a flat background. Another is the "Table Tennis" video sequence, which is composed of a high speed moving objects. Figure 2 and 3 show two scenes from these two video sequences.

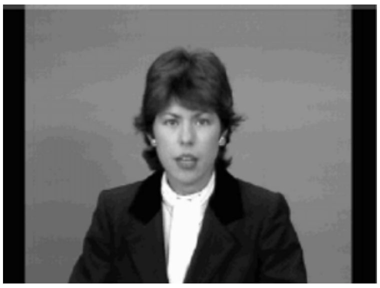


Fig 2. "Claire" video sequence

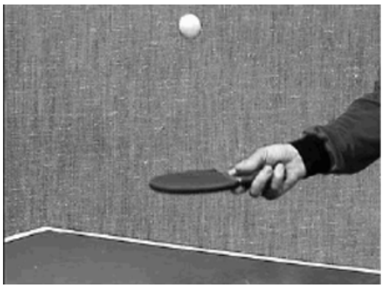
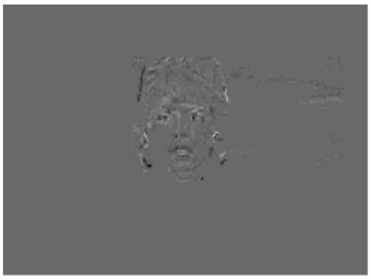
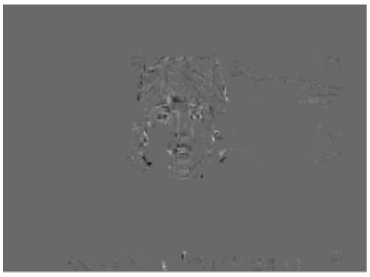


Fig 3. "Table tennis" video sequence

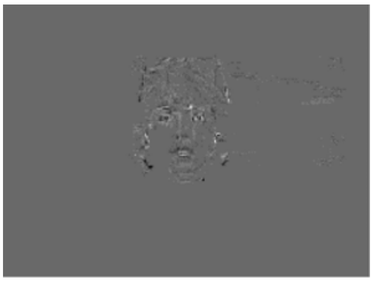
The SAE of optical flow, three-step search, and full search algorithms for "Claire" video sequence are 80.28, 87.94 and 72.09. The prediction errors of optical flow, three-step search, and full search algorithms for "Claire" video sequence are shown in figure 4.



(a) Prediction errors of optical flow



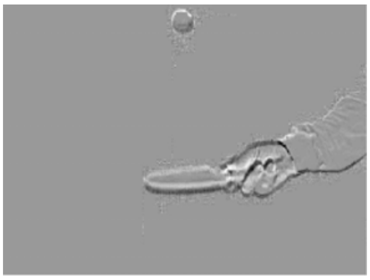
(b) Prediction errors of three step search



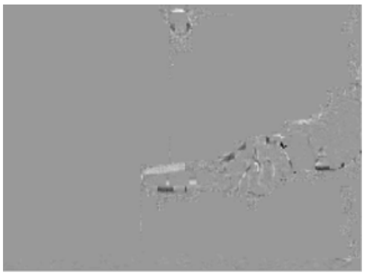
(c) Prediction errors of full search

Fig 4. Prediction errors for "Claire"

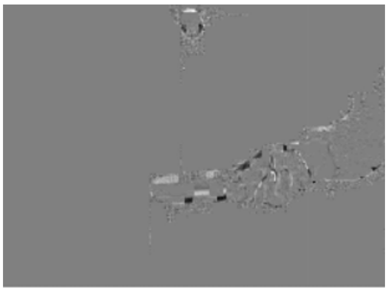
The SAE of optical flow, three-step search, and full search algorithms for "Table Tennis" video sequence are 685.33, 293.58, and 220.67. The prediction errors of optical flow, three-step search, and full search algorithms for "Table Tennis" video sequence are shown in figure 5.



(a) Prediction errors of optical flow



(b) Prediction errors of three step search



(c) Prediction errors of full search

Fig 5. Prediction errors for “Table tennis”

Optical flow method achieves high accuracy for scenes with small motion vector and small displacements (< 1-2 pixels/frame) but fail when the motion vector and displacements are large [9]. It is understandable; because the invariant luminance hypothesis of optical flow is invalid when there is a large motion vector and displacement like the example in figure 6.

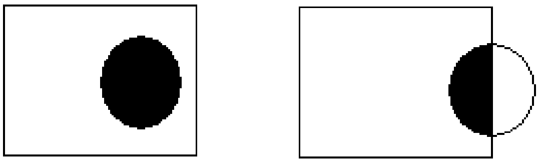


Fig 6. Example of luminance change between frames.

The full search algorithm has the most accurate estimation result. However, the computation cost is much higher than the others. Three-step search algorithm can handle large displacements efficiently but are not very accurate [10] [11], because it only search sub-optimal result in each step, instead of searching optimal result from global area.

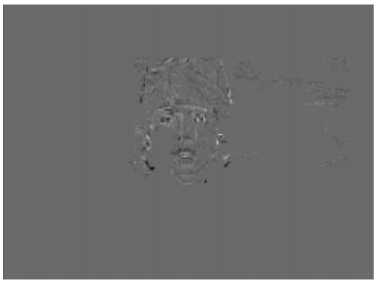
The best scenario would have high estimation accuracy disregarding whether the motion vector is small or large, and the computation cost is not too much. A straightforward idea is using three-step search algorithm and optical flow estimation at the same time. After compute the prediction error, the motion vector which produces smaller error would be used as the estimation result. The algorithm can be written as follows:

1) Use optical flow algorithm to estimate the motion vector V_{opt}, and calculate the prediction error Err_{opt}.
2) Use three-step search algorithm to estimate the motion vector V_{3step}, and calculate the prediction error Err_{3step}.
3) If $Err_{opt} < Err_{3step}$, $V = V_{opt}$, otherwise, $V = V_{3step}$.
4) Return V as the final estimation result.

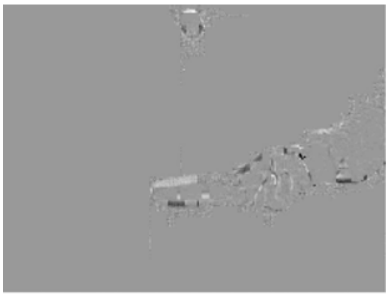
“Claire” and “Table Tennis” video sequences also were used to evaluate the performance of this joint algorithm. Table I shows the simulation results of optical flow, three-step search, full search, and the joint algorithms. The prediction error images are shown in Figure 7.

TABLE I PREDICTION ERRORS OF OPTICAL FLOW, THREE-STEP SEARCH, FULL SEARCH, AND THE JOINT ALGORITHMS FOR VIDEO “CLAIRE” AND “TABLE TENNIS”

	Optical Flow	Three-Step Search	Full Search	Joint Algorithm
“Claire”	80.28	87.94	72.09	73.32
“Table Tennis”	685.33	293.58	220.67	238.40



(a) Prediction errors of "Claire"



(b) Prediction errors of "Table tennis"

Fig 7. Prediction errors of the Joint Algorithm

From Table1, it shows clearly that the prediction error of the proposed joint algorithm is nearly as small as the full search algorithm. At the same time, the computation cost of the joint algorithm is only the sum of optical flow and three-step search, which is $(54+9)\times n^2=63n^2$ addition, $6n^2$ multiplication, plus one 2×2 matrix inverse, for a block with $n \times n$ pixels. It is still much lower than the computation of full search, which requires $450n^2$ addition for a block with $n \times n$ pixels.

V. CONCLUSION

An efficient algorithm with joint of optical flow and three-step search is presented in this paper. The joint algorithm can achieve high accuracy disregarding whether the motion vector is small or large, and the computation cost is not too much. Two video sequences, “Claire” which only contains slow moving objects, and “Table Tennis” which contains high speed moving objects, were used to test the performance of optical flow, three-step search, full search, and the joint algorithm. The result shows that the joint algorithm works nearly as well as the full search algorithm, while cost much lower computation.

REFERENCES

[1] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., Vol. COM-29, pp. 1799-1808, Dec. 1981.

[2] H. G. Musmann, P. Pirsch and H.-J. Grallert, "Advances in picture coding," Proc. IEEE, Vol. 73, No. 4, pp. 523-548, 1985.

[3] V. Bashkaran and K. Konstantinides, "Image and video compression standards: algorithms and architectures," Kluwer Academic Publishers, 1995.

[4] A. Murat Tekalp, "Digital video processing," Prentice-Hall, 1995.

[5] K. R. Rao and J. J. Hwang, "Techniques and standards for image, video and audio Coding," Prentice-Hall, 1996.

[6] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," Proc. IEEE, Vol. 83, No. 6, June 1995.

[7] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proceedings of DARPA Image Understanding, pp. 121-130, 1981.

[8] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, T. Ishiguro, "Motion-compensated interframe coding for video conferencing," Proceedings NTC'81 (IEEE), pp.5.3.1 - G.5.3.4

[9] A. Singh, "Optical flow: a unified perspective," IEEE Computer Society Press, Loas Alamitos, CA, 1991.

[10] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, "Performance of Optical Flow Techniques," in International Journal of Computer Vision, February 1994, vol. 12(1), pp. 43–77.

[11] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," in International Journal of Computer Vision, Vol. 2, pp. 283-310, January 1989.