A FAST BLOCK MATCHING MOTION ESTIMATION ALGORITHM BASED ON SIMPLEX MINIMISATION

Mohammed E. Al-Mualla, Nishan Canagarajah, David R. Bull

Image Communications Group, Centre for Communications Research, University of Bristol,
Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, U.K.
Tel: +44-117-954 5126, Fax: +44-117-954 5206, e-mail: M.Almualla@bris.ac.uk

ABSTRACT

A fast block matching motion estimation algorithm is presented. The algorithm is based on a generic unconstrained optimisation technique called *simplex minimisation* (SM). In order to apply this method to the constrained minimisation problem of block matching motion estimation, a suitable initialisation procedure, termination criterion, and constraints on the independent variables of the search, are proposed. The algorithm is demonstrated to outperform other fast block matching motion estimation techniques providing better reconstruction quality, a smoother motion field and reduced computational complexity.

1. INTRODUCTION

In video coding, the high correlation between successive frames can be exploited to improve coding efficiency. This is usually achieved using motion compensated prediction. Among existing methods for motion estimation, the *block matching motion estimation* (BMME) algorithm has, due to its simplicity, received considerable attention and has been incorporated into various video coding standards (*e.g.* MPEG 1-2 [1][2], H.261 [3], and H.263 [4]). In BMME, the current frame is divided into blocks. For each block, the best match block within a search window in a reference frame is estimated according to a distortion measure.

The simplest BMME algorithm is the *full search* (FS) algorithm. This gives an optimum solution by exhaustively searching over all possible blocks within the search window. The main drawback of this algorithm is its high computational complexity. Such computational power may not be available in many applications, *e.g.* wireless video communication, especially if real-time video coding is required.

Many fast algorithms [6,7,9,10] have, therefore, been developed to alleviate this complexity by limiting the number of search locations. Most of these algorithms are, implicitly or explicitly, based on the *unimodal error surface assumption* which states that *the distortion measure increases monotonically as the search location moves away from the global minimum*. In many cases, this assumption does not hold true and such algorithms can easily get trapped in a local minimum.

Another approach to reduce complexity is as follows. The BMME algorithm can be formulated as a constrained two-dimensional optimisation problem. This problem can, therefore, be solved with reduced complexity using well-known optimisation techniques. Srinivasan and Rao [10] proposed two fast block matching techniques, *conjugate directions search* (CDS) and *one-at-a-time search* (OTS), based on the well-known *conjugate directions* (CD) optimisation method [8]. In this paper we solve the same problem using *simplex minimisation* (SM) [5].

Simplex Minimisation is a generic method for search and optimisation and, as such, certain aspects of the method must be defined according to the application. Taking into account the nature and properties of the BMME problem, we propose a suitable initialisation procedure, termination criterion, and constraints on the independent variables of the search.

This paper is organised as follows. Section 2 describes the process of BMME and formulates it as a constrained two-dimensional optimisation problem. Section 3 briefly describes the SM method. Section 4 describes why the SM method is an attractive choice for solving the BMME problem. It then shows how SM can be used for BMME. Section 5 presents the results of testing the algorithm and compares its performance to other fast algorithms. Finally, section 6 gives some concluding remarks.

2. BLOCK MATCHING MOTION ESTIMATION (BMME)

In BMME, the current frame is divided into blocks of $M \times L$ pels. Each block is then matched against a corresponding block within a search window of $M+2p \times L+2p$ in a reference frame, where p is the maximum allowed motion displacement. The best match on the basis of a *block distortion measure* (BDM) yields the motion vector (d_x, d_y) which is assigned to all pels within the block. Various BDMs, such as the *sum of squared differences* (SSD) or the *sum of absolute differences* (SAD), can be used. In general, the matching process can be expressed as follows:

$$BDM(i, j) = \sum_{x=1}^{M} \sum_{y=1}^{L} g\left[f_c(x, y) - f_r(x+i, y+j) \right], -p \le i, j \le p \qquad (1)$$

$$(d_x, d_y) = \arg [\min_{i,j} BDM(i, j)] \qquad (2)$$

where f_c refers to the intensity values of the block in the current frame and f_r refers to the intensity values of the block in the reference frame. For the SSD, $g[\ .\ ] = (\ .\ )^2$, and for the SAD, $g[\ .\ ] = |\ .\ |$

Equations 1 and 2 clearly indicate that BMME is a constrained two-dimensional minimisation problem. The two dimensions are the vertical, j, and horizontal, i, motion displacements, the function to be minimised is the BDM, and the independent variables are constrained within a limited range, $-p \le i, j \le p$, and are usually evaluated to a certain accuracy, *e.g.* full pel accuracy.

3. SIMPLEX MINIMISATION (SM)

Simplex minimisation, as introduced by Nedler and Mead [5], is a multidimensional unconstrained optimisation method. A simplex is a geometrical figure which consists, in N dimensions, of N+1 vertices and all their interconnecting line segments, polygonal faces, *etc*. A nondegenerate simplex is one that encloses a finite inner N-dimensional volume.

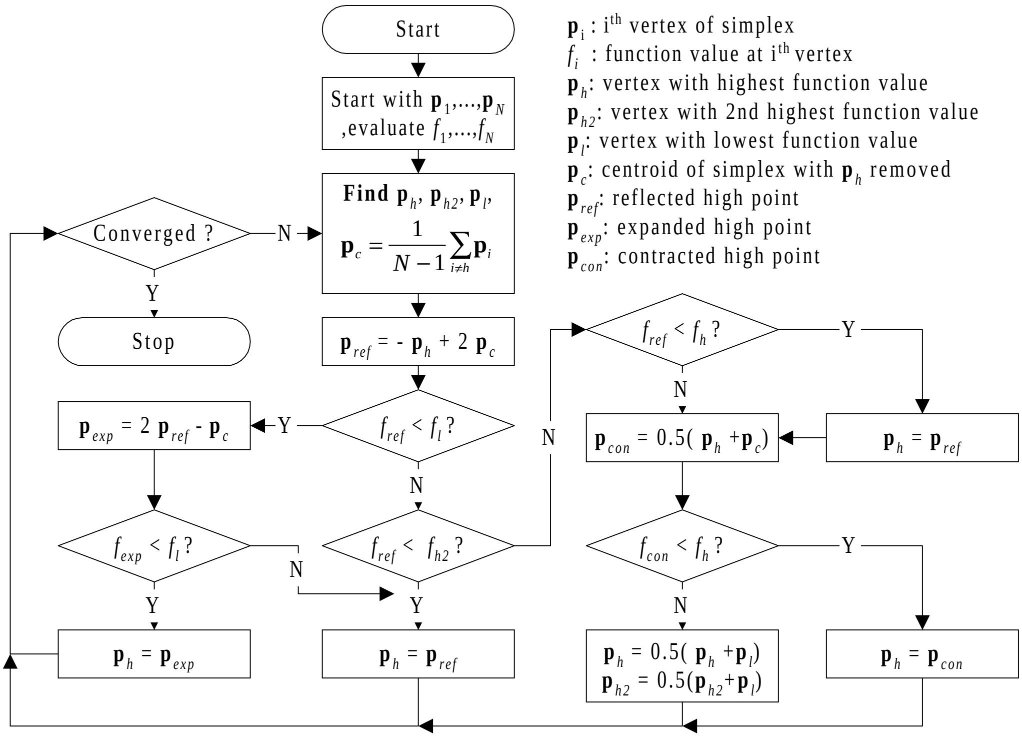


Figure 1. Simplex method for function minimisation.

The SM method is initialised with N+1 points defining an initial nondegenerate simplex in the search space. The method then takes a series of steps *reflecting*, *expanding* and *contracting* the simplex, from the point where the function is largest, in an attempt to find a better point. This is repeated until a termination criterion is satisfied. Figure 1 shows a flow chart of the SM method.

4. THE APPLICATION OF SM TO BMME

The SM method is an attractive choice for solving the BMME problem for two reasons. Firstly, the reflection and expansion steps reduce the possibility of getting trapped in a local minimum. As already mentioned, the local minimum problem is the main drawback of most existing fast search algorithms. Secondly, the set of searched locations is obtained using very simple computations (refer to figure 1 and note the simple equations). Since the main aim, here, is to reduce computational complexity, then the SM method is more suitable than other better but more complex optimisation techniques, *e.g.* gradient methods.

Since SM is a generic approach for search and optimisation, when applied to block matching motion estimation, certain aspects of the method need to be investigated, defined and parameterised.

4.1. Choosing the Initial Three Points

In two dimensions (N=2), a simplex is a triangle (N+1=3). Thus, 3 points need to be chosen to define the initial non-degenerate simplex. The performance of the SM search is highly dependent on the choice of these points [5]. We propose the following initialisation procedure.

The BDM is evaluated at three predictions of the motion vector. The choice of predictions takes into account two important properties of the block motion field of a typical video sequence. The first property is that the *distribution of the global minimum is centre-biased* [9]. In other words, the motion vector **0** has the highest probability of occurrence. To exploit this property we include the motion vector **0** as one of the three predictions. The second property is that the field is *smooth and varies slowly* [9]. As a result, it is not uncommon to find neighbouring blocks with identical or nearly identical motion vectors. In fact, most video coding standards take advantage of this property by coding the motion vectors differentially. To exploit this property, and to match the motion estimation process to the motion coding process, we propose to use the motion vectors of the blocks to the *left* and *above* the current block as the other two predictions. If such neighbouring vectors are not available, as in border blocks, they are set to **0**.

If $\mathbf{p_m} = (dx_m, dy_m)$ is the prediction that yields the smallest BDM, then the BDM is also evaluated at its 8 nearest neighbours, $(dx_m, dy_m \pm 1)$ $(dx_m \pm 1, dy_m)$ $(dx_m \pm 1, dy_m \pm 1)$. If the minimum BDM occurs at $\mathbf{p_m}$, then no further displacements are considered. This takes into account the assumption that the BDM *is monotonic in a small neighbourhood around a minimum* [9]. If, however, the minimum occurs at one of the 8 neighbours, then 8 further displacements are examined, $(dx_m, dy_m \pm s)$ $(dx_m \pm s, dy_m)$ $(dx_m \pm s, dy_m \pm s)$. Experimental results show that a step size of $s = p/3$ gives good results for most video sequences.

At this stage, all displacements are arranged in ascending order according to their BDMs and the best three are chosen as the initial vertices of the simplex.

The SM search then proceeds as shown in figure 1, subject to the constraints outlined in 4.2. and is terminated when the criterion proposed in 4.3. is satisfied.

4.2. Constraints on the Independent Variables

As already mentioned, SM is a generic approach and as such, it assumes continuous unconstrained independent variables. When applied to block matching motion estimation, two constraints have to be imposed. Firstly, the vertices of the simplex must be within the search window. Any point produced by reflection, expansion or contraction must be set to the closest point within the range $-p \le i, j \le +p$, before any BDM evaluation can take place. Secondly, the vertices of the simplex must be set to the required search accuracy. For example, if full-pel accuracy is assumed, then any non-integer point produced by reflection, expansion or contraction must be rounded to the nearest integer value before the BDM evaluation can take place.

4.3. Termination Criterion

There are many possible ways to terminate the SM algorithm. The most widely used approach is to terminate the search if the fractional range from the highest, in terms of function value, to the lowest vertices of the simplex is below some threshold [11]. In BMME, the behaviour of the BDM function is different from one sequence to another, from one frame to another, and even from one block to another. This means that the threshold must also be different in each case making such a criterion unsuitable for block matching. We propose a more suitable criterion where the search is terminated if *the vertices of the simplex are neighbours*. This termination criterion has two advantages. Firstly, it does not depend on a threshold. Secondly, it conserves the nondegeneracy of the simplex.

4.4. Motion Vector Refinement

One problem with the proposed termination criterion is that it is not based on the function to be minimised, *i.e.* the BDM. The search, therefore, may converge to a suboptimal point. Experimental results show that, in most cases where this happens, the optimum point is in the neighbourhood of the point produced by SM. An extra step is, therefore, added to the search where the displacement produced by SM is refined by searching its 8 nearest neighbours. This step does not add significantly to the complexity of the algorithm because the BDMs of most of those neighbours are often available from the SM search.

5. SIMULATION RESULTS

The algorithm was tested using the luminance component of three QCIF sequences: FOREMAN (176×144 @ 25 fps), TUNNEL (176×144 @ 25 fps) and TABLE TENNIS (176×120 @ 30 fps). For FOREMAN and TABLE TENNIS, 300 frames were used, and for TUNNEL, 250 frames were included. The BDM was defined to be the SAD, the block size was set to 16×16 pels, and the maximum allowed motion displacement, p, was assumed to be 15 pels in both directions. The search was performed to full pel accuracy and motion vectors were restricted so that they do not point outside the frame. A lossless *displaced frame difference* (DFD) coding was assumed. That is, for each frame, motion was estimated and compensated using the original previous frame. Motion vectors were coded differentially using the variable length coding table of the H.261 [3].

Tables 1 – 3 summarise the results of applying the SM algorithm to the three test sequences, and compare its performance to *full search* (FS), *two dimensional logarithmic search* (2DL) [6], the *cross search algorithm* (CSA) [7], and the *one at a time search* (OTS) [10].

Table 1. Average PSNR (dB) for various algorithms.

	FS	**SM**	2DL	CSA	OTS
FOREMAN	32.20	**32.04**	31.81	30.96	31.21
TUNNEL	31.08	**30.93**	30.80	30.14	30.57
TENNIS	32.22	**31.75**	31.68	30.96	31.27

Table 2. Average searched locations/frame for various algorithms

	FS	**SM**	2DL	CSA	OTS
FOREMAN	77439	**1100**	1639	920	604
TUNNEL	77439	**903**	1533	527	497
TENNIS	65621	**837**	1362	461	448

Table 3. Average motion bits/frame for various algorithms.

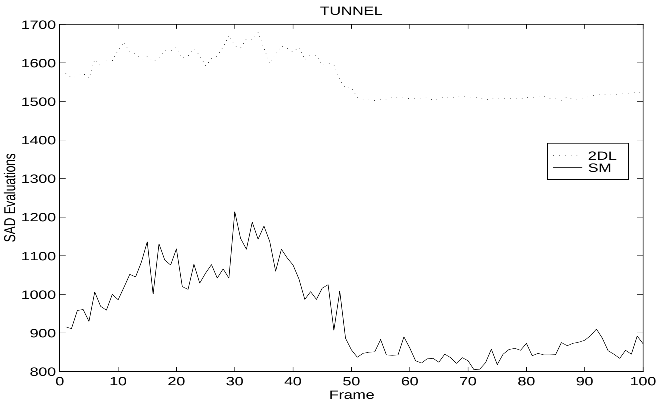
	FS	**SM**	2DL	CSA	OTS
FOREMAN	388	**361**	394	461	388
TUNNEL	276	**265**	275	283	266
TENNIS	279	**247**	269	281	246

Compared to FS, the SM algorithm provides substantial savings in both computational complexity and motion overhead with only a small loss in the reconstruction quality. For the FOREMAN sequence, the SM algorithm provides savings of about 98.6% in the number of searched locations/frame, and savings of about 27 motion bits/frame, with only 0.16 dB loss in the reconstruction quality.

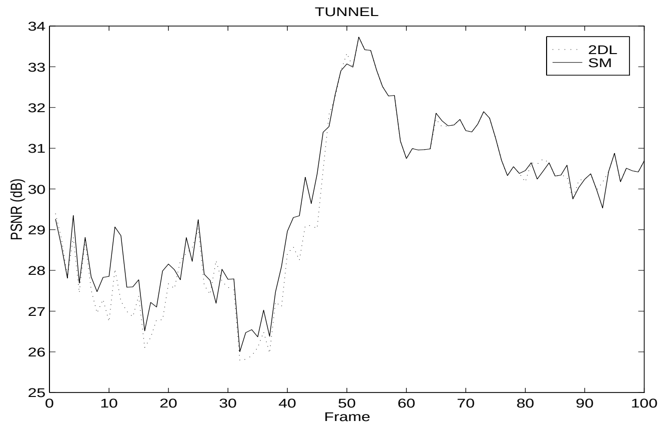
The SM algorithm outperforms all other fast algorithms considered in this simulation. In general it provides better reconstruction quality, a smoother (and hence easier to code) motion field, and reduced computational complexity. Compared to the well known 2DL search, the SM search produces 0.23 dB improvement in the reconstruction quality of the FOREMAN sequence, 33 bits/frame reduction in the motion information, and about 33% reduction in the computational complexity. This superior performance can also be seen in figure 2 for the TUNNEL sequence.

6. CONCLUSIONS

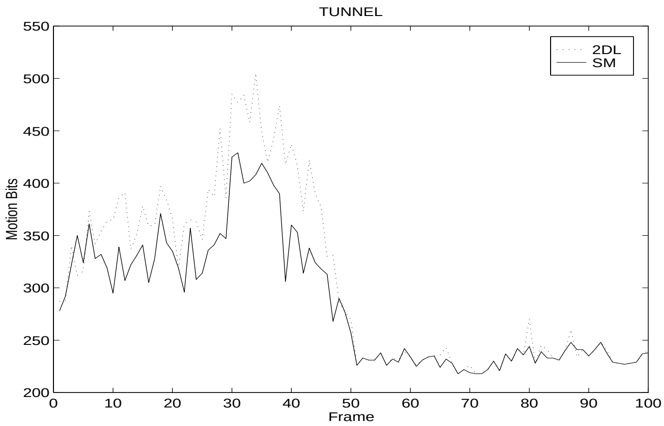
Block matching motion estimation can be formulated as a two-dimensional constrained minimisation problem. This problem can, therefore, be solved with reduced complexity using well-known optimisation techniques. In this paper, a generic unconstrained optimisation technique called *simplex minimisation* (SM) [5] was utilised. In order to apply this technique to block matching, a suitable initialisation procedure, termination criterion, and constraints on the independent variables of the search, were proposed. Simulation results showed that the proposed algorithm outperforms other fast block matching techniques. In general it provides better reconstruction quality, a smoother (and hence easier to code) motion field, and reduced computational complexity. Compared to FS, the algorithm provides substantial savings in both computational complexity and motion overhead with only a small loss in the reconstruction quality.



(a) Computational complexity



(b) Reconstruction quality



(c) Motion overhead

Figure 2. Comparison between SM and 2DL when applied to the first 100 frames of the TUNNEL sequence.

7. ACKNOWLEDGMENT

This work was supported by Etisalat College of Engineering, a division of Emirates Telecommunications Corporation (ETISALAT), United Arab Emirates.

8. REFERENCES

[1] ISO/IEC JTC1/SC29/WG11, "ISO/IEC 11172-2 Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s - Part 2: Video," MPEG1 Draft, December 1991.

[2] ISO/IEC JTC1/SC29/WG11, "ISO/IEC 13818-2 Information Technology - Generic Coding of Moving Pictures and Associated Audio - Part 2: Video," MPEG2 Draft, November 1994.

[3] ITU, "Video Codec for Audio-visual Services at p×64 Kbits/s," ITU-T Recommendation H.261, March 1993.

[4] ITU, "Video Coding for Low Bitrate Communication," Draft ITU-T Recommendation H.263, May 1996.

[5] J. A. Nedler, and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, Vol. 7, 1965, pp. 308 - 313.

[6] J. R. Jain, and A. K. Jain, "Displacement Measurement and Its Application in Interframe Image Coding." *IEEE Transactions on Communications*, Vol. COM-29, No. 12, December 1981, pp. 1799 - 1808.

[7] M. Ghanbari, "The Cross-Search Algorithm for Motion Estimation," *IEEE Transactions on Communications*, Vol. 38, No. 7, July 1990, pp. 950 - 953.

[8] M. R. Hestenes, *Conjugate Direction Methods in Optimisation*, New York: Springer-Verlag, 1980.

[9] R. Li, B. Zeng, and M. L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 4, August 1994, pp. 438 - 442.

[10] R. Srinivasan, and K. R. Rao, "Predictive Coding Based on Efficient Motion Estimation," *IEEE Transactions on Communications*, Vol. COM-33, No. 8, August 1985, pp. 888 - 896.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Edition, New York: Cambridge University Press, 1992, pp. 408-412.