A Sum Square Error based Successive Elimination Algorithm for Block Motion Estimation

J.J. Francis and G. de Jager
Department of Electrical Engineering, University of Cape Town
Rondebosch, 7700, South Africa
{jfrancis,gdj}@dip.ee.uct.ac.za

Abstract
Block Motion Compensation is used for finding and compressing temporal differences between frames and estimating the motion between frames. Optimal temporal compression demands minimizing the error between blocks and this is best accomplished by having as large a search area as possible. Motion estimation, on the other hand, demands a small search area to preserve the coherence of the motion vector field. This paper develops a lower bound on the Sum of Squared Errors (SSE) in terms of the energies of the candidate best match and the search blocks and describes a fast method for calculating the energies of overlapping blocks. The search for a best matching block can then be truncated by eliminating search positions which do not satisfy this bound. The proposed algorithm is compared to an exhaustive search. The results show a reduction in search time and identical motion fields.

1 Introduction
Block motion estimation (BME) is used to compress temporal differences between frames in several video compression algorithms, notably MPEG-2 [7]. BME involves dividing a frame (called the reference frame) into a series of blocks (the search blocks) and for each block finding the position of the best match block, via some criterion, within a region (the search window) of the next frame (called the motion frame) [15]. This paper considers an optimised algorithm for block motion estimation in the context of froth imaging.

The performance achieved by block motion compensation depends on the search space. Examining every possible point in the search space leads to best performance at the expense of speed. This is the exhaustive search algorithm against which other block matching algorithms are compared [15, 12]. Several algorithms have been proposed in the literature which examine only a small neighbourhood around the origin of the search, these include, Two Dimensional Logarithmic Search[17], Block Based Gradient Descent Search [14], Conjugate Search [15], Cross Search [6], Dynamic Search Window Adjustment [10], Four Step Search [16], Improved Three Step Search [4], Modified Three Step Search [11], One Dimensional Full Search [5], and Three Step Search [15]. These algorithms are sub-optimal in that they trade performance for speed.

Until the development of the Successive Elimination Algorithm (SEA) by Li and Salari [12] a full search involved examining every point within a search window (or for temporal compression, the entire frame). The SEA, on the other hand, generates upper and lower bounds on the sum of pixels in the best match block and, having pre-calculated the sum of pixels for every possible best match block, reduces the size of the search space for every improvement in the match. Li and Salari’s SEA is tied to the use of the Minimum Absolute Difference (MAD) error measure to measure the match. The algorithm described in this paper uses a similar process of calculating upper and lower bounds, but uses the SSE error estimate. The SSE has the advantage of being directly related to the correlation of the reference and motion blocks.

This paper begins by briefly overviewing the SEA algorithm and block motion estimation and then derives the bounds on SSE. An experimental comparison of the SSE based algorithm and a full-search is made and finally the results are discussed.

1.1 Some definitions
Denoting the search block by vector $g$ and the candidate best match block by vector $f$, the following norms may be defined [9]:

$$ l_p(g) = \|g\|_p = \sqrt[p]{\sum_k |g_k|^p} \quad (1) $$

The sum of pixels within the search and best match
blocks can be seen to be $l_1(g)$ and $l_1(f)$ respectively. The $l_2$ norm is defined as the Euclidean, or inner norm, and may be derived from the inner product as follows:

$$\|f\|_2 = \sqrt{\langle f, f \rangle}$$

(2)

The inner product may be defined as:

$$\langle f, g \rangle = \sum_k f_k g_k$$

(3)

The Sum of absolute differences (SAD) may be defined as [15]:

$$\text{SAD}(f, g) = N \text{MAD}(f, g) = \|f - g\|_1$$

(4)

The SSE and Mean Square Error (MSE) may be defined as [15]:

$$\text{SSE}(f, g) = N \text{MSE}(f, g) = \|f - g\|_2^2,$$

(5)

where $N$ is the number of elements within vectors $f$ or $g$.

## 2 An overview of the Successive Elimination Algorithm

The Successive Elimination Algorithm generates bounds on the error measure. Only if a candidate best match block lies within these bounds is the computationally expensive error measure computed. This naturally leads to a faster algorithm. The initial algorithm of Li and Salari as well as some previous attempts to impose bounds on the SSE error measure are examined.

### 2.1 Li and Salari

This is a refinement of the BME process [12]. BME involves dividing a frame (the search frame) into a series of blocks or search blocks. The search blocks are individually correlated with another frame (called the reference frame) to find the best match position. The difference between the best match position and the corresponding coordinates of the search block on the reference frame, gives the motion vector.

The SEA algorithm uses the SAD to derive bounds on the sum of pixels within the best match block. The lower bound on the SEA is can be shown to be [12, 19, 2]:

$$\text{SAD}(f, g) \geq \|f\|_1 - \|g\|_1,$$

(6)

where $\|f\|_1$ denotes the sum of pixels in the search block, $\|g\|_1$ denotes the sum of pixels in the current best match block under consideration, and SAD($f$, $g$) is the SAD of the best match thus far. Equation 7 and 8 shows the bounds.

$$\|f\|_1 - \text{SAD}(f, g) \leq \|g\|_1 \leq \|f\|_1 + \text{SAD}(f, g)$$

(7)

Dividing by the number of pixels within a block,

$$\frac{\|f\|_1}{N} - \frac{\text{MAD}(f, g)}{N} \leq \frac{\|g\|_1}{N} \leq \frac{\|f\|_1}{N} + \frac{\text{MAD}(f, g)}{N}$$

(8)

### 2.2 Attempts to bound the SSE

Wang and Mercereau [19] describe a lower bound on the SSE in terms of the difference of the averages of the search and candidate best match block. Brunig and Niehsen make use of a similar bound [2].

Given the Cauchy-Swartz inequality for vectors $a$ and $b$ [9]:

$$|a^T b| \leq \|a\|_2 \|b\|_2$$

(9)

If one sets $a$ to be a column vector of ones, and allows $b$ to be the pixel differences for the error measure, equation 9 becomes:

$$\|f - g\|_1 \leq \sqrt{N} \|f - g\|_2$$

and squaring,

$$\text{SAD}^2 \leq N \text{SSE}$$

(10)

$$\text{MAD}^2 \leq \text{MSE}$$

(11)

Applying the SEA lower bound (equation 6) leads to:

$$\frac{(\|f\|_1 - \|g\|_1)^2}{N^2} \leq \text{MAD} \leq \text{MSE}$$

(12)

This allows Li and Salaris’s algorithm to be used with only one modification. The Li and Salari bounds are replaced by the give inequality and only pixels which satisfy this inequality have the computationally expensive MSE (or SSE) evaluated.

Wang and Mercereau also apply Partial Distortion Elimination (PDE) to speed the computation of the MSE. Brunig and Niehsen further explore splitting blocks into smaller regions in order to generate a tighter bound on the error measure used. A form of PDE is suggested to speed the computation of the error measure. These optimisations are neglected here, since they do not reduce the number of times the SAD error measure is evaluated, but simply optimise that evaluation.
2.3 Fast computation of block sums

Clearly computing the sum of pixels of overlapping blocks can be computationally expensive. Li and Salari [12] proposed a fast method for calculating the sum of pixels (or sum-norms).

The calculation proceeds as follows for block size \( N \times N \) on an image \( I \) (neglecting blocks which touch or overlap the image borders), with the result stored in image \( S \):

1. Row sums are generated. The sum at a pixel is:

\[
S(x, y) = S(x-1, y) + I(x + \left\lfloor \frac{N}{2} \right\rfloor, y) - I(x - \left\lfloor \frac{N}{2} \right\rfloor - 1, y)
\]  

(13)

2. The column sums are generated. The sum at a pixel is:

\[
S(x, y) = S(x, y-1) + S(x, y + \left\lfloor \frac{N}{2} \right\rfloor) - S(x, y - \left\lfloor \frac{N}{2} \right\rfloor - 1)
\]  

(14)

This requires \( 2N + 2N \) arithmetic operations per block (neglecting blocks at the borders of the image and initial calculations to initialise the row and column sums).

3 Derivation of the Bounds on the SSE

The process of computing upper and lower bounds on the Sum Squared Error (SSE) is explored here. The Cauchy-Swartz inequality \( \langle f, g \rangle \leq \|f\|_2 \|g\|_2 \) may be applied to the SSE set an upper bound on the cross-correlation peak [18]:

\[
\text{SSE} = \|f - g\|_2^2 \quad \text{from equation 5},
\]

\[
= \langle f, f \rangle - 2 \langle f, g \rangle + \langle g, g \rangle 
\]

\[
\geq \|f\|_2^2 - 2 \|f\|_2 \|g\|_2 + \|g\|_2^2
\]

\[
= (\|f\|_2 - \|g\|_2)^2
\]  

(15)

Where \( \|f\|_2 \) and \( \|g\|_2 \) denote the square root of the energy within the search and candidate best match blocks. Equation 15 has roots:

\[
(\|f\|_2 - \|g\|_2 + \sqrt{\text{SSE}})(\|f\|_2 - \|g\|_2 - \sqrt{\text{SSE}}) \leq 0
\]

\[
\Rightarrow \|g\|_2 - \sqrt{\text{SSE}} \geq \|f\|_2 \geq \|g\|_2 + \sqrt{\text{SSE}}
\]  

(16)

or.

\[
\Rightarrow \|g\|_2 + \sqrt{\text{SSE}} \geq \|f\|_2 \geq \|g\|_2 - \sqrt{\text{SSE}}
\]  

(17)

Unlike equation 16, only equation 17 is consistent for real numbers larger than, or equal to, one. This equation then gives a upper and lower bound on the square root of the energy of the candidate best match block. The SSE, which is expensive to compute, is only computed for pixels which satisfy the lower bound in equation 15, or alternatively, equation 17.

It should be noted that the same method employed by the SEA to pre-calculate the sum-norms, or sum of pixels within a block, can be used by the proposed algorithm. The only modification is before the sum-norms are calculated, the pixels are replaced by their squared values. A further optimisation for motion estimation involves pre-calculating the energies for both frames since the search blocks now overlap (the original SEA only computed sum-norms for the motion frame [12]).

4 Experimental results

An experimental comparison of the proposed SSE bound, an exhaustive search using SSE, and Wang and Mercereau’s bound is made. A border ten pixels wide was neglected to account for edge effects in the motion estimation. Both algorithms employ a block size of \( 7 \times 7 \) pixels and a search window of \( 11 \times 11 \) pixels centered on the position of the search block in the reference frame. These parameters are selected arbitrarily and are expected to be data dependent.

Figures 1(a) and 1(b) show two frames of a froth video sequence used to compare the performance of the algorithms. The reference frame is figure 1(a) and the motion frame is figure 1(b).

Figure 2 shows the motion vector field, a registered image and an error image. The registered image is computed by predicting the reference frame from the motion frame using the motion field. The motion field shows the predicted position of a reference pixel in the motion frame. An average of the pixels in the motion frame can be computed around this position (the averaging is Gaussian with a sigma of 0.8 and a neighbourhood of \( 7 \times 7 \)). Again these parameters were arbitrarily selected. The error image is the absolute difference between the registered image and the reference image. For display purposes the image is inverted (each pixel is subtracted from the maximum pixel value).

The peaks in the difference image and the corresponding regions in the registered image show regions where the block matching algorithm did not describe the motion well. Comparison with the motion field show these regions which were not predicted well, had discontinuities in the motion field.

The registration error for all algorithms is \( 2.20 \times 10^6 \). The sum of the MSE for the motion estimate at each pixel examined is \( 6.99 \times 10^6 \) for all the algorithms.
The motion fields are shown by a pixel by pixel comparison to be identical for the algorithms considered. This is expected.

The proposed algorithm evaluates the computationally expensive MSE at 6,736,502 points. The exhaustive search evaluates the error measure at 16,846,248 points. The proposed algorithm shows an improvement of a factor of 2.5 over the exhaustive search. Wang and Mercereau’s algorithm evaluates the MSE 16,471,974 times, a marginal improvement of a factor of 1.02 over the exhaustive search.

Note that the computational savings of all the considered algorithms are likely to be data dependent.

5 Conclusions

The proposed algorithm provides the same results as the exhaustive search algorithm but require fewer computations. The algorithm places bounds on the energy of the candidate best match blocks. This allows the search space to be reduced each time a better match is found without sacrificing accuracy. The algorithm is suitable for motion estimation (and its use of small search areas) and searching the entire frame, if needed, for temporal compression. The expected savings in computations can be expected to increase as the search window and size of the blocks increases.

Comparing the number of MSE evaluations as generated by Wang and Mercereau bound with the proposed bound, show the proposed bound is tighter. This is shown to increase the speed of the algorithm.

This method implies a fast algorithm for finding cross-correlation peaks for general use.

References


Figure 1: The data set used.


(a) Motion Field.

(b) Registered Image.

(c) Difference (inverted) between original and registered image.

FIGURE 2: Applying the algorithms.