A MULTiresolution BLOCK MATCHing TECHNique FOR IMAGE 
MOTION ESTIMATION

I Rhee, G R Martin and R A Packwood

1 Introduction

Motion estimation forms an integral part of many inter-frame coding schemes, and its effectiveness is partly responsible for either the achieved compression ratio or the quality of the reconstructed image sequence. This is particularly true for low bit-rate systems.

The idea of performing block matching motion estimation within a multiresolution context is not new, but existing schemes [WC90, CYC90] adopt a so-called ‘top-down’ approach. The ‘top’ is a low resolution image which is searched first to generally identify areas of motion. This information is refined by moving ‘down’ the image pyramid using progressively higher resolution, to more precisely identify areas of motion and the associated vectors.

However, these top-down approaches are likely to suffer from a ‘majority’ effect: a potentially differing motion in a small area within the larger region may have only a small effect on the motion vector for the whole block. The resulting motion information tends to be biased toward the motion of the majority of the region, possibly indicating invalid motion for the small independent area.

For example, an algorithm by Wang and Clarke [WC90] generates approximate motion vectors from a low resolution copy of an image by block matching. Further searches, based on these initial estimated vectors, are made using an image of higher resolution. However, the initial estimates of motion vectors may exclude a different motion of a small area, which can only be detected at a higher resolution. As the technique searches only the areas adjacent to the initial estimates, the true motion vector for the small area will not be found. A similar effect can be observed within Chan et al’s work [CYC90]. Their technique starts from a motion vector derived by matching a large (but not decreased resolution) block. The error obtained by matching the large block is used to determine whether or not the block should be divided to provide multiple motion vectors instead of one. In their algorithm, a large block may not split further because the majority of the block has a good match with an existing vector. However, a minority region of the block may require a different vector; one motion vector for the entire block may not be enough to describe the blocks’ motion properly.

It is necessary to better utilise the information in the high-resolution/small-blocks of an image to generate accurate estimates of the motion. In this paper, we present a ‘bottom-up’ multiresolution block matching technique. We first compute a set of ‘candidate’ motion vectors for each comparatively small block. Then, we take an intersection of these sets of neighbouring blocks. The resulting set will be the set of candidate motion vectors for the larger block that encompasses the neighbouring blocks. This way, it preserves the motion information extracted at a high resolution and uses the information at low resolutions only selectively to improve the motion estimates.

Authors are with the Dept. of Computer Science, University of Warwick, Coventry CV4 7AL.
Performance evaluation indicates that our technique works better than the existing ones especially when the image is complex and involves many different translational motions. Also, application of the technique on real image sequences to which random noise is added showed that our technique works consistently better under various noise levels than the other techniques.

Section 2 describes our algorithm in detail and Section 3 presents the results of the performance evaluation. In Section 4, we summarise the work.

2 Multiresolution Block Matching Algorithm

Given two consecutive image frames $f_{i-1}$ and $f_i$ in an image sequence, image $f_i$ is divided into fixed-sized small square blocks (we used 4x4 blocks in our experiments). We denote each block by a tuple $(x, y, s)$ where $(x, y)$ is the coordinate of the upper-leftmost pixel of the block and $s$ is the length of a side of the square block.

Motion information for each of these small blocks is gathered by block matching. Given a block $(x, y, s)$ and a predefined search window of motion vectors $V$, we calculate the mean absolute error or mean squared error between blocks $(x, y, s)$ in $f_i$ and $(x + x', y + y', s)$ in $f_{i-1}$ for each $(x', y')$ in $V$. For each block $b$ in $f_i$, we obtain a set of motion vectors $I_b$, called the initial set of $b$, whose matching error is less than a predefined threshold.

Then we merge those adjacent blocks that have at least one motion vector in common in their initial set of motion vectors. This merging process helps solve the aperture problem [MU81] because ambiguous motion at small blocks is refined as the blocks grow into a larger region. This also reduces the number of regions in the image and therefore the number of motion vectors.

To simplify the merging process, we impose a complete quad-tree structure \(^1\) onto these small blocks and merge the small square blocks into larger square blocks. In the tree, there is a one-to-one correspondence between leaves and the initial square blocks. The parent block (or node) of a block is defined to be $(2s \lfloor \frac{x}{2s} \rfloor, 2s \lfloor \frac{y}{2s} \rfloor, 2s)$. We define sibling blocks to be those blocks that share the same parent block. The parent and children block relationship is illustrated in Figure 1.

Let us define a set of motion vectors $I_Sb$ for each block $b$ in the tree, called the intersection set of $b$ as follows.

\(^{1}\)For brevity of presentation, we assume that the number of the initial blocks is a power of 4.
If $b$ is a leaf,

$$IS_b = I_b$$

Otherwise

$$IS_b = IS_i \cap IS_j \cap IS_k \cap IS_l$$

where $i$, $j$, $k$ and $l$ are the children of $b$.

We merge four sibling blocks of the tree into a parent block if and only if the intersection set of each sibling block is not empty and the intersection set of the parent block is not empty. The reasoning behind this process is as follows. When $IS_b$ is empty, there is no vector that is common to the four children of block $b$. This means that there exists at least one child/descendent block $b'$ that undergoes a different motion from the other sub-blocks in block $b$. Merging those blocks into $b$ will provide an incorrect motion vector for $b'$.

We repeat this process at each level from the bottom level to the top level. The motion vector for a block which cannot be merged into its parent is chosen to be the one giving the minimum error for the block among those in its intersection set.

### 3 Performance Evaluation

We evaluate the performance of our technique in terms of the mean square error (MSE) and compare it with that of existing techniques. Two existing block matching techniques are used for comparison. One is the well-known fixed-size block matching [JJ81] and the other is Chan et al.'s variable block matching [CYC90]. The exhaustive search with search window size 15x15 is adopted for all the tested techniques. Image sequences on which the tests are performed are 256x256 ‘Miss America’ and ‘Split Screen’ image sequences. Figures 3 and 4 show motion compensated frames on which a block structure is superimposed. These frames are obtained using our technique.

Note that Wang and Clarke’s technique is also a fixed-size block matching, and the test image sequences generally do not involve motion with distances further than 8 pixels per frame. Thus, we assume that the performance of Wang and Clarke’s method should be similar to the exhaustive search fixed-sized block matching.
Since our technique relies more on motion information from a high resolution, it is presumably sensitive to the noise present in the images. We also measure the sensitivity through experimentation with noisy image sequences. Figure 2 illustrates the test procedure. Given two consecutive image frames $i$ and $i+1$ in an image sequence, we first add a prescribed degree of random Gaussian noise to frame $i+1$ to obtain a noisy image frame. Then, the noisy frame and frame $i$ are used to estimate motion in frame $i+1$. The resulting motion compensated frame is then compared with the original image frame (i.e., frame $i+1$) to compute the MSE. We perform this procedure over the entire image sequence under a given noise level, and repeat the whole procedure for different noise levels.

Real image sequences contain varying amounts of noise, and an ideal technique should accurately estimate the true motion in an image regardless of the level of noise present. The motion information estimated by the ideal technique, since it estimates the true motion, should give a close match to the original noiseless image frame, but not to the noisy image frame used as input to the estimation. Therefore, in our test procedure, we measure the MSE of the motion compensated image frame and the original noiseless image frame, and believe that this test should give at least a rough indication of the performance of a technique under various noise levels.

Figure 5 shows the results of our experiment in terms of the MSE in the ‘Miss America’ image sequence. FIXED, CYC and RMP denote the fixed-size block matching technique, Chan et al’s algorithm and our algorithm respectively. The number of blocks$^2$ used for the results in the figure is 256, and the error criterion applied for block matching is the mean absolute error. For FIXED, we use a $16 \times 16$ block size, resulting in 256 blocks. For RMP, the starting block size is $4 \times 4$ and the merge threshold of the technique is adjusted to obtain 256 blocks. Similarly, for CYC, the initial block size is $64 \times 64$ and the split threshold is adjusted to obtain 256 blocks.

Figure 6 shows the same type of performance figure for a noisy ‘Miss America’ image sequence. The average MSE and signal-to-noise ratio (SNR) of the noisy image sequence over the original ‘Miss America’ image sequence are 36 and 19dB respectively. Similarly, Figures 7 and 8 show the same type of performance results for the (noiseless) ‘Split Screen’ image sequence and a noisy ‘Split Screen’ image sequence. The average MSE and SNR of the noisy image sequence over the original ‘Split Screen’ noiseless image sequence are 281 and 12dB respectively.

Our experiment indicates that RMP performs consistently better than FIXED and CYC over various noise levels. In the noiseless ‘Miss America’ sequence, RMP shows about a 10% improvement over CYC and an 8% improvement over FIXED on average (see Figure 5). In the noisy ‘Miss America’ sequence, RMP shows about a 6% improvement over CYC and a 9% improvement over FIXED. In the noiseless ‘Split Screen’ sequence, RMP shows about a 33% improvement over CYC and a 35% improvement over FIXED. In the noisy ‘Split Screen’ sequence, RMP shows about a 18% improvement over CYC and a 26% improvement over FIXED.

The more dramatic improvement with the ‘Split Screen’ sequence is related to the content of the images. The ‘Miss America’ sequence is relatively amenable to motion estimation because only a small number of areas in each image undergo some translational motion. Thus, all the techniques perform well with this sequence. In the ‘Split Screen’ sequence, many small regions of each frame in the sequence undergo different motions. In this type of sequence, it is expected that the two ‘top-down’ methods (FIXED and CYC) should fail while RMP performs better.

Generally, RMP shows a higher sensitivity to noise than CYC and FIXED because RMP relies more on the motion information from a high resolution. However, RMP performs comparably well even in the noisy images. This experiment indicates clearly that random noise does not have so significant an impact on RMP as to leave it unusable when its performance is compared with those of CYC and FIXED.

$^2$Similar results are reported for other numbers of blocks.
4 Conclusion

In this paper, we discussed a new multiresolution block matching technique. Unlike previously known techniques, this method estimates motion bottom-up so that approximate motion information from a high resolution is refined by the information from lower resolutions. Our experiment on real image sequences to which various levels of noise were added showed that this method performs better than other existing top-down techniques.

It has been tested as part of a simple DCT codec, not described in this paper because of space constraints. A similar performance improvement over other techniques was observed.
5 Acknowledgement

The authors are pleased to acknowledge the support for this work from the Science and Engineering Research Council, Grant No. GR/J79997.

References


