IMPROVED MOTION DESCRIPTION CODING USING THE LIST MAPPING MOTION DESCRIPTION (LMMD)

Simon Tredwell and Adrian N. Evans*

Department of Electronic and Electrical Engineering
University of Bath
Bath, BA2 7AY, UK
S.Tredwell@bath.ac.uk A.N.Evans@bath.ac.uk

ABSTRACT

A novel technique for motion description encoding is proposed. The method is based on a vector list and a mapping, each of which is losslessly encoded using variety of algorithms. Results show that in comparison to H.263, the best performing algorithms offer a significant reduction in the number of motion bits required, for motion fields with relatively few unique vectors. The bits saving are particularly important for low bit rate coding applications.

1. INTRODUCTION

In video coding, temporal redundancy can be reduced by predicting subsequent frames from a reference frame. In many practical video codecs this is achieved by using a block matching technique to generate a motion description which is then used by the decoder to form a prediction of the current frame from a reference frame. As the predicted frame is only an approximation of the of the true current frame a residual or error image is used to improve the quality of the predicted image to some degree. The bit allocation for each predicted frame is therefore distributed between the motion description and the residual image.

While the number of bits used for the residual is made scalable by using a lossy compression scheme, the bit allocation for the motion description is fixed. The bit allocation between the motion description and the residual image is therefore sub-optimal, reducing the overall coding efficiency. This is particularly true for low bit rate coding where the motion description constitutes a significant proportion of the overall bandwidth.

One attempt to overcome this problem is the so-called rate-distortion method of motion estimation [1, 2]. The idea behind this approach is that instead of encoding the optimal motion vector in terms of some error metric, a sub-optimal vector with a reducing coding cost is selected. As the vectors are normally encoded by differencing with a vector predicted from a local causal neighbourhood, this approach tends to favour locally similar vectors. Rate-distortion techniques are focused on vector selection on a block-by-block basis and as such it is difficult to incorporate a global criteria for the motion bit allocation. Furthermore, although the motion bandwidth is reduced, is is impossible to know by how much until the final vector has been chosen.

Post-filtering has also been used to reduce the number of bits used for the motion description [3]. In this work a conditional vector median filter provides replacement vectors only if the increase in the residual error is below a threshold. Again, this decision is evaluated locally and the threshold value is only an approximate means of control.

We have previously proposed a global technique for selecting the motion vectors in a rate-distortion sense [4, 5]. In this approach, rather than locally selecting appropriate vectors the predicted image is produced from a fixed number of unique vectors, with the advantage that the number vectors can be used to explicitly control the motion bandwidth. The problem of producing the best possible predicted image from a fixed number of unique motion vectors has also been considered [6]. This approach has the further advantage that when the number of unique motion vectors is small, alternative more efficient motion description encoding techniques can be considered. One such new approach, the List Mapping Motion Description (LMMD) is the subject of this paper.

*Mr. S. Tredwell supported by Tandberg Television Ltd. and EPSRC

2. LIST MAPPING MOTION DESCRIPTION

Vector field motion descriptions are conventionally encoded be differencing and entropy coding. For example, in the H.263 standard [7] each vector is subtracted from a prediction (based on neighbouring, previously transmitted vectors) before being encoded with a variable length code.

The LMMD is a novel technique, designed to offer increased compression when there are few unique vectors in the motion field. Rather than store the motion description as a simple vector field, the LMMD splits it into two components (see Figure 1):

1. A list of all the vectors used in the motion description.
2. A mapping showing which vector from the list is associated with each motion block.

The real advantage of this method is to reduce the two arrays which make up the vector field into a single array. However, extra cost is incurred when encoding the list, which is why this approach is more appropriate when the motion description contains a smaller number of unique vectors. Also, although the original correlation in the motion description is retained, there is no relationship between different indices in the mapping.

3. VECTOR LIST ENCODING

The vector list often contains a certain amount of redundancy which can be exploited in the encoding. For example, during a fast pan, the vectors $(7,0)$, $(7,-1)$ and $(7,1)$ may all appear in the mapping. Rather than encode the absolute value of the vector, it is

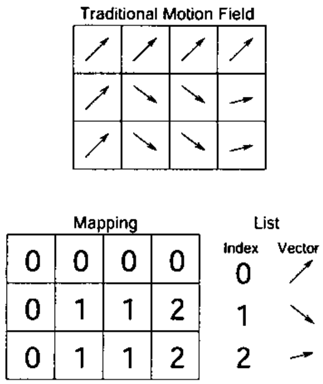


Fig. 1. Comparing tradition field based motion descriptions with the LMMD. The same motion description is shown both as a traditional motion field (top) and using the LMMD (bottom).

usual to only encode the differences between successive vectors, for example in the standard H.263 encoding scheme. Consequently, if the vector list is sorted into an appropriate order before encoding the bit cost can be reduced. Five methods are proposed to sort the list, each with their own strengths and weaknesses in terms of complexity, speed and accuracy.

For each of the methods described a figure is given showing the path traced when an example set of 14 vectors are traversed in the order into which they are sorted, see Figure 2. The length of this path is an indication of the cost of encoding the list.

3.1. Travelling Salesman Approach

There is obviously an optimal list for which the differential coding is most efficient. The Travelling Salesman Approach (TSA) can be used to find this list. In the travelling salesman problem a hypothetical salesman is required to visit a number of interconnected cities. Given that he should visit each city once, and only once, what is his shortest route? A travelling salesman scenario is set up by treating the points as cities. However, rather than using the geometrical distance between the vectors as the cost of moving from one city to the next, the cost in bits of differentially encoding the vectors is used instead (as this is the quantity which is to be minimised).

As the first vector in the list is differenced with $(0,0)$ the starting point for the tour is important. It should also be noted that complete solutions to the TSA can be very computationally expensive as the number of vectors in the list increases.

3.2. Sorting by Component

The most straightforward method of sorting the list is to arrange the vectors in order of increasing x or y component, with any ties resolved by looking at the other component.

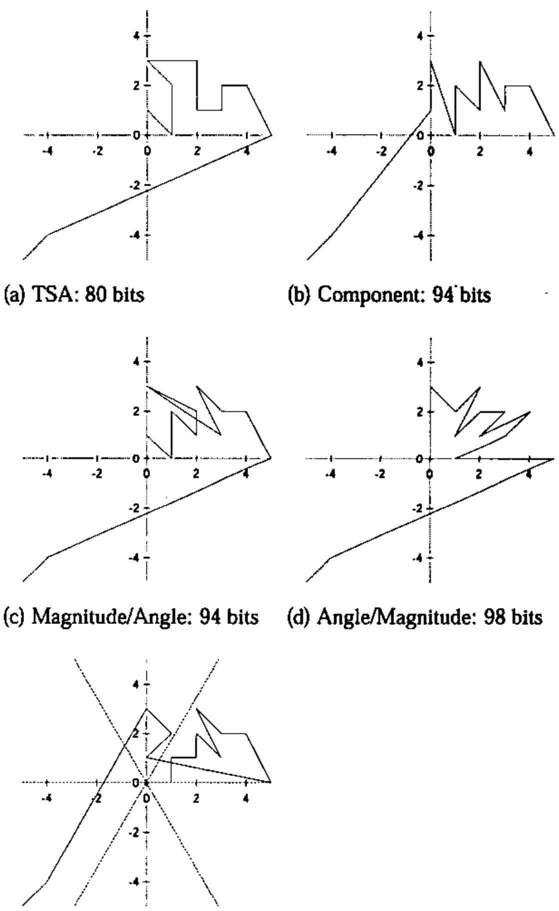
3.3. Sorting by Magnitude and Angle

Each vector is converted from Cartesian form (x, y) to polar form (r, θ) and these values are used to sort the vectors. There are two approaches which can be taken in this case;

- The list can be sorted by magnitude, r, and any ties resolved by sorting by angle, θ.
- The list can be sorted by angle, θ, and any ties resolved by sorting by magnitude, r.

3.4. Sector Sort

The vector space is broken up into a number of sectors of equal angle, the number of sectors being determined by the total number of vectors. Typically each sector should contain four to six vectors. Within each sector, the vectors are sorted by magnitude. The sector with the vector nearest the origin is identified, and the vectors are added to the list in order of increasing magnitude. The neighbouring sector is then chosen and its vectors are added to the list in either increasing or decreasing magnitude depending on whether the vector with the highest or lowest magnitude is closest to the last vector in the previous sector. This then continues around the sectors until all the vectors have been encoded.



(a) TSA: 80 bits

(b) Component: 94 bits

(c) Magnitude/Angle: 94 bits

(d) Angle/Magnitude: 98 bits

(e) Sector: 90 bits

Fig. 2. Comparing different methods for sorting the vector list. For each method the path traced by the sorted list is shown, along with the cost of encoding the list.

4. ENCODING THE MAPPING

After the vector list has been encoded it is necessary to encode the array which maps the vectors to each individual image block. This mapping contains a list of indices to the vectors in the list and often contains correlation which can be exploited during coding.

However, due to the fact that only indices exist in the mapping and not vectors themselves, the correlation is restricted to blocks which share the same index. It does not make sense to difference indices the same way as one would difference the corresponding vectors.

A number of techniques for encoding the mapping are proposed. As with the vector list techniques, each have their strengths and weaknesses.

4.1. Raw Encoding

This method simply encodes each element of the mapping with a fixed length code. As the number of vectors in the list is already known, length of code needed to uniquely identify any index in the mapping can be deduced.

For a mapping containing indices between $0 \leq n \leq N$ the cost, in bits, to encode each index is given by;

$$N_{bits} = \text{floor}(\log_2(N)) + 1 : N > 0 \quad (1)$$

Apart from the fact that no attempt is made to take advantage of any correlation between indices, this method is rather inefficient for many values of N. However, it is fast and the cost of encoding can be determined without the need to iterate over the mapping (the cost simply being N_{bits} times the number of blocks in the mapping).

4.2. Linearised Map Encoding

The remaining methods operate on a linearised version of the mapping. To transform the two-dimensional mapping into this form alternative raster and anti-raster scanning is used for each row, as this preserves any correlation between adjacent vectors in the leftmost and rightmost columns. Alternatively, the scanning direction can be vertical in alternating up/down direction. The resulting vector can be encoded by the following techniques.

Run Length Encoding Runs of symbols in the vectors are encoded using a standard run length scheme. The symbol is sent raw (see Equation 1), and then length of run encoded. For QCIF size images (with 99 indices in the mapping) the the maximum length of run permitted is 8.

Predictive Encoding The first symbol is sent raw. Each subsequent symbol is compared to the previous one, and a bit flag is used to indicate whether or not they are the same. If they differ the symbol needs to be sent raw.

Both of the above methods can be used with either the horizontal or vertical scanning giving a total of five techniques, with the addition of the raw encoding.

5. EXPERIMENTAL RESULTS

A collection of 'typical' motion descriptions was created using an H.263 video codec and standard test sequences. The sequences were QCIF sized (176×144) and were encoded at 128 kbits/second, 30 frames/second.

For each frame-pair the optimum motion description (in terms of the Sum of Absolute Differences (SAD)) was found using the Exhaustive Search Algorithm (ESA) to half pixel accuracy, which contains the maximum number of unique vectors, V_{max}. The Metric Method [5] was then used to generate descriptions with a reduced number of vectors N, with $2 \leq N \leq V_{max}$. Thus for each frame-pair up to $V_{max} - 1$ motion descriptions are created of varying complexity and accuracy. Table 1 gives details of the sequences used.

Sequence	No. Frames	No. Motion Descriptions
Foreman	400	12395
Carphone	382	12740
Container	300	3263
News	300	2350

Table 1. The number of frames and the number of generated motion descriptions for each sequence used.

5.1. Evaluating the List and Mapping Encodings

The TSA provides the optimal solution for the encoding of the vector list, however alternate methods need to be considered due to its prohibitive computational cost. Table 2 shows the average number of bits required to encode the motion descriptions from the foreman sequence (with an upper limit of 15 placed upon the number of vectors in the list to reduce the computational load of the TSA).

Technique	Average cost in bits
Magnitude/Angle	57.52
Angle/Magnitude	56.86
Sector	54.82
Component	57.19
TSA	49.50

Table 2. Comparison of the different vector list encoding techniques.

In terms of the average cost in bits the sector sort is closest to the optimal result given by the TSA. However, due to the many different types of list which can be produced all the methods can prove to be useful.

The same is true for the mapping encoders. They each perform well under circumstances dictated by the statistics of the mapping. When there is very little correlation in the mapping (which can be the case when there are many vectors in the list) the runlengthing and predictive approaches fail, giving a higher cost in bits than if the mappings were encoded raw.

5.2. Comparison with H.263 Motion Description Encoding

For each motion description in Table 1 the vector list was encoded using each of the methods described in section 3. The TSA which was only used for $2 \leq N_v \leq 15$ vectors as it is too computationally intensive for higher values of N_v, (where N_v is the number of vectors in the list of the LMMD). Likewise, the mapping was encoded using each of the proposed methods from section 4. The best list and mapping method were then found and a 3 bit flag used to communicate the encoding method used for the mapping. The LMMD was then converted into a vector field and encoded using the motion encoder from H.263. It should be noted that *all* the blocks had their motion vectors encoded; none of the blocks were treated as INTRA blocks.

For each value of N_v the percentage of times that the LMMD encoding cost less bits than H.263 was found. To give a more fair comparison, values of N_v for which there were less then 50 motion descriptions were ignored. The results for each of the sequences used are summarised in Figure 3.

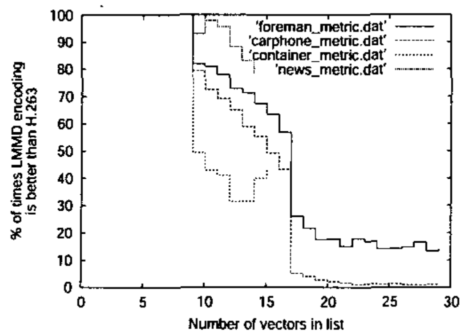


Fig. 3. Comparing H.263 motion description encoding with LMMD based encoding

5.3. Frame by Frame Comparison

Figure 4 shows a frame-by-frame comparison of the LMMD based encoding with that used by H.263 for the first 50 frames of the foreman sequence. For each frame a half pixel motion field was found using the ESA. The Metric Method[5] was then used to reduce the number of vectors in the field. Sufficient vectors were chosen so that the error in the residual (in terms of the SAD) for each frame was within 100 of that of the ESA motion description. The resulting motion description was then encoded as described in Section 5.2.

It can be seen that that the LMMD based encoding is more effective for a large majority of the time. There are occasions when it performs much worse, for example at frame 47 where the LMMD performs badly mainly due to the large number of vectors in the list (21). However, on average the LMMD is 48 bits better.

6. CONCLUSIONS

A new technique, the LMMD, has been proposed for encoding motion descriptions with relatively few unique vectors. The technique can be seen to perform well when compared to traditional methods of encoding motion descriptions, such as that used by H.263, for frames with less than 16 unique vectors to be encoded. The bit savings are particularly significant for low bit rate coding where the motion bandwidth is a relatively large proportion of the overall bandwidth.

The steep drop in performance of the LMMD method after 16 (which is clearly visible in Figure 3) is mainly due to the rise in the

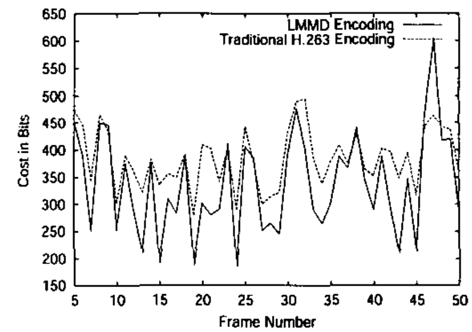


Fig. 4. A frame by frame comparison of the LMMD based encoding with H.263 using 50 frames of the foreman sequence. The mean cost per frame is 337 bits for the LMMD based technique compared with 385 bits for H.263. The mean number of vectors used per frame is 8.8.

cost of encoding the mapping. All the techniques used to encode the mapping rely in some way on the cost of encoding an index raw. As this cost rises from 4 bits per symbol for mappings containing 16 different indices, to 5 bits for 17 indices. Future work will look at other more efficient ways for encoding the mapping, which consumes the majority of the bits in the LMMD encoding.

7. REFERENCES

[1] M. Z. Coban and R. M. Mersereau, "Fast rate-constrained N-step search algorithm for motion estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 1998, vol. 5, pp. 2613–2616.

[2] Y. Yang and S. S. Hemami, "Rate-constrained motion estimation and perceptual coding," in *Proceedings of the 1997 International Conference on Image Processing*, Oct 1997, vol. 1, pp. 81–84.

[3] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini, "Adaptively weighted vector-median filters for motion-fields smoothing," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, May 1996, pp. 2267–2270.

[4] S. Tredwell and A.N. Evans, "Block grouping algorithm for motion description encoding," in *Proceedings of IEEE International Symposium on Circuits and Systems*, May 2001, vol. 5, pp. 211–214.

[5] S. Tredwell and A.N. Evans, "A sequential vector selection algorithm for controllable bandwidth motion description encoding," in *Proceedings of IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing*, May 2001, pp. 209–212.

[6] S. Tredwell and A.N. Evans, "Selecting motion vectors for bandwidth controllable motion description coding," in *Proceedings of Irish Machine Vision and Image Processing Conference*, September 2001.

[7] ITU, "Recommendation H.263 (02/98) – video coding for low bit rate communication," ITU-T H.263, International Telecommunication Union, 1998.