A SEQUENTIAL VECTOR SELECTION ALGORITHM FOR CONTROLLABLE BANDWIDTH MOTION DESCRIPTION ENCODING

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

This paper presents a method for controlling the bandwidth given to the motion description in motion-compensated video coding by restricting the number of motion vectors used. A new algorithm, called the metric method, for selecting the vectors is presented. This approach produces predicted images that are a close approximation of the optimal that can be achieved for a fixed number of vectors with relatively little additional computational cost. Results show that, in a video codec, the metric method produces lower overall errors than the standard approach of sending one vector per block, and is better than the histogram method previously used to select the vectors.

1. INTRODUCTION

In motion compensated video codecs the bandwidth for each motion-compensated image is distributed between the motion description and the residual image. Most of the current video coding standards (MPEG [1], H.263 [2]) provide no mechanism for controlling the bandwidth allocated to the motion description and hence their performance can be less than optimal. This is particularly true at low bit rates where the motion description constitutes a significant proportion of the overall bandwidth.

It is therefore advantageous to be able to have some degree of control over the motion description bandwidth. To this end, several techniques have been proposed including rate-constrained motion estimation [3, 4] and an embedded quad-tree motion description [5].

We have recently proposed a technique that controls the motion bandwidth by varying the number of motion objects in each image N, where the motion of each object is given by a unique motion vector [6]. In this approach the motion description consists of two parts: a list of the N vectors and a mapping describing which of the vectors in the list is assigned to each image block. As the number of blocks in each image is much greater than the number of motion objects the overall coding performance is improved in comparison with the traditional approach of sending one vector for each block, hereafter termed Simple Vector Motion (SVM).

In [6], the mechanism for selecting the vectors for the list was to place the best vector for each block (as chosen by some metric) into a histogram and to select the N most frequent vectors. This approach will be referred to as the histogram method. By varying N the bandwidth assigned to the motion description can be controlled to some degree. The histogram method is straightforward and easy to implement but cannot guarantee the predicted image, as measured by a given error metric, is the best that can be produced using N vectors.

This paper presents the metric method, an alternative algorithm for selecting the N vectors that constitute the motion field, that aims to produce a predicted image that is close to optimal for a given number of vectors, N, with relatively little additional computational cost over that of SVM. The vectors are picked *sequentially* (according to an error metric), with each additional vector ($n < N$) giving the most improvement to the predicted image obtained using $n-1$ vectors, as measured by the chosen comparison metric.

The layout of this paper is as follows. Section 2 describes the metric method and results and discussion are presented in Section 3. Finally conclusions are drawn in Section 4.

2. THE METRIC METHOD

The algorithm derives its motion description from two input images: the *current image* and the *reference image*. The current image is divided into equally sized blocks, which are the smallest unit of the image that can be assigned a motion vector. These blocks are labelled, in raster order, with a unique *block ID*, β, that ranges from 0 to β_{m}.

The maximum extent of a motion vector is limited by both the horizontal and vertical search ranges, $(x_{\mathrm{SR}}, y_{\mathrm{SR}})$. As with the blocks, each of these possible vectors is as-

This work was supported by Tandberg TV and EPSRC

signed a unique *vector ID*, ν, such that there is a one-to-one mapping between vectors and their IDs. The range of ν is from 0 to $\nu_\mathrm{m} = (2x_\mathrm{SR}+1)(2y_\mathrm{SR}+1)-1$. Although the method of assigning the vector IDs is arbitrary, they are typically indexed by scanning the search area in a raster fashion. Equations 1 and 2 provide a convenient means of converting between a vector (v_x, v_y) and its vector ID ν.

$$\begin{aligned} k &= 2x_\mathrm{SR}+1 \\ V_\mathrm{ID} &= (v_x + x_\mathrm{SR}) + k(v_y + y_\mathrm{SR}) \end{aligned} \tag{1}$$

$$\begin{aligned} v_x &= \mathrm{quot}(V_\mathrm{ID}, k) - x_\mathrm{SR} \\ v_y &= \mathrm{rem}(V_\mathrm{ID}, k) - y_\mathrm{SR} \end{aligned} \tag{2}$$

Although the metric method works with any comparison metric, in the interest of clarity we will illustrate its operation using the Sum of Absolute Differences (SAD), given by

$$\mathrm{SAD} = \sum_{i=0}^{W-1}\sum_{j=0}^{H-1} |f(x_1+i, y_1+j) - g(x_2+i, y_2+j)| \tag{3}$$

where (x_1, y_1) and (x_2, y_2) are the co-ordinates of the blocks being compared and W and H are the width and height of the blocks respectively.

The first step of the algorithm is to calculate the SAD obtained by applying each possible vector ν to every block β and to store the results in a *metric table*, see figure 1. The procedure for generating the table is computationally equivalent to the exhaustive search block matching algorithm.

	Block ID					
Vector ID	0	1	2	...	$\beta_\mathrm{m}-1$	β_m
0	383	886	777	...	915	793
1	335	386	492	...	649	421
2	362	27	690	...	59	763
⋮	⋮	⋮	⋮	⋱	⋮	⋮
$\nu_\mathrm{m}-2$	211	368	567	...	429	782
$\nu_\mathrm{m}-1$	530	862	123	...	67	135
ν_m	929	802	22	...	58	69

Figure 1: Example of a metric table.

To ensure that valid entries in the metric table are obtained at every position, the cases where the search range requires an image block that lies wholly or partly outside the reference image must be accommodated. This problem can be solved by *extending* (or *padding*) the reference image with extra pixels so that all vectors are valid for every block. The extent of the padding required depends on the search range $(x_\mathrm{SR}, y_\mathrm{SR})$.

There are many ways of choosing how to pad the image. In this implementation an extruding padder was used in which the value of the edge pixel is copied out to the edge of the extension, see figure 2. After padding, a full metric table can be found by exhaustive search.

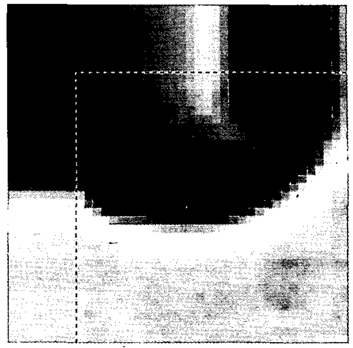

Figure 2: Example of extruding padding on the top left corner of an image.

Once the metric table has been generated, the metric method finds a list of N vectors, $\{\nu_1, \nu_2, \ldots, \nu_N\}$, such that each vector in the list ν_n gives the greatest improvement in the total SAD for the predicted image produced using vectors ν_1 to ν_{n-1}. We use the notation $\mathrm{SAD}_{n,\nu}$ to denote the total image SAD that would be obtained if the vector ν were to be added to the list of the $n-1$ vectors previously found.

Using the metric table, the total SAD for any single vector ν, $\mathrm{SAD}_{1,\nu}$, can simply be found by summing the SADs for each block in the image, such that

$$\mathrm{SAD}_{1,\nu} = \sum_{\beta=0}^{\beta_\mathrm{m}} M(\beta, \nu) \tag{4}$$

where $M(\beta, \nu)$ is table entry for block β and vector ν. The vector that produces the minimum value in (4) is the best vector for the whole image, denoted ν_1. The SAD values for ν_1 are copied from the table into into an array $\boldsymbol{\nu}_{\mathbf{best}_1}$ of length $\beta_\mathrm{m}+1$.

For the second vector, ν_2, we wish to choose the vector that results in the lowest total SAD for two vectors, $\mathrm{SAD}_{2,\nu}$, given that vector ν_1 has previously been selected. Once again we can iterate over ν but this time in the summation we choose the minimum SAD from the corresponding positions in $M(\beta, \nu)$ and $\boldsymbol{\nu}_{\mathbf{best}_1}$, giving

$$\mathrm{SAD}_{2,\nu} = \sum_{\beta=0}^{\beta_\mathrm{m}} \min(M(\beta, \nu), \nu_{\mathrm{best}_1}(\beta)) \tag{5}$$

where $\nu_{\text{best}_1}(\beta)$ is the entry for block ID β in the array $\boldsymbol{\nu}_{\mathbf{best_1}}$.

Finding the vector that gives the minimum value in (5) allows the second best vector ν_2 to be identified. The array $\boldsymbol{\nu}_{\mathbf{best_1}}$ can then be updated to give $\boldsymbol{\nu}_{\mathbf{best_2}}$ using,

$$\begin{aligned}\nu_{\text{best}_{n+1}}(\beta) &= \min(M(\beta, \nu_{n+1}), \nu_{\text{best}_n}(\beta)) \\ &\quad \mid \beta \in \{0, 1, 2, \ldots, \beta_{\text{m}}\} \qquad (6)\end{aligned}$$

So, in general, given a list of n vectors and the array $\boldsymbol{\nu}_{\mathbf{best_n}}$ (which holds the minimum SAD for each block taken from the n vectors) it is possible to select the next best vector ν_{n+1} by finding the value of ν which minimises,

$$\text{SAD}_{n+1,\nu} = \sum_{\beta=0}^{\beta_{\text{m}}} \min(M(\beta, \nu), \nu_{\text{best}_n}(\beta)) \qquad (7)$$

Obviously there is no need to consider the n vectors already contained in the list when determining $\text{SAD}_{n+1,\nu}$ as they will not provide any improvement to the total SAD. Likewise, when calculating the sum over β in (7), if $\text{SAD}_{n+1,\nu}$ becomes greater than the current minimum sum, the summation can be abandoned and the next vector considered.

3. RESULTS AND DISCUSSION

The metric method of section 2 was evaluated within a wavelet based video codec using the "flower" (53 frames) and "foreman" (300 frames) sequences with the following parameters,

- image size: 160×128
- block size: 8×8
- frame rate: 25 fps
- x_{SR}: 16
- y_{SR}: 16
- bit rate: 131,072 bits/s (128 kbits/s)

The flower sequence contains relatively simple motion, mostly introduced by the parallax. In comparison, the motion in the foreman sequence is quite complex. Much of this results from the fact that it has been shot with a hand held camera, which introduces camera shake and slight rotations to the sequence.

The compression performance achieved using vectors selected by the metric and histogram methods was evaluated inside a wavelet based video codec test-bed. Also evaluated was the SVM field, which was compressed using differential adaptive arithmetic encoding.

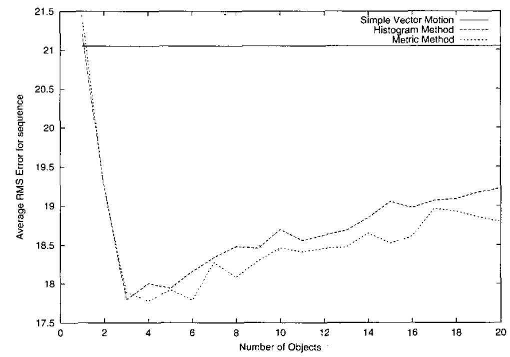

Figure 3: Average RMS error over the flower sequence for a varying number of objects.

Figure 3 presents the average RMS error for the flower sequence for different values of N using the histogram and metric methods for selecting the vectors. Also shown is the average error produced with the SVM method; this line is horizontal as number of objects is not a parameter, giving a constant value.

The motion in this sequence is relatively simple and both methods for selecting the top N vectors produce better results than the SVM method for all cases except $N = 1$. However, it can be seen that the N vectors are chosen using the metric method the average error is less than that for the histogram method for all $N > 3$.

Figure 4 shows similar results for the foreman sequence. The motion in this sequence is quite complex and consequently the SVM method performs best for low number of

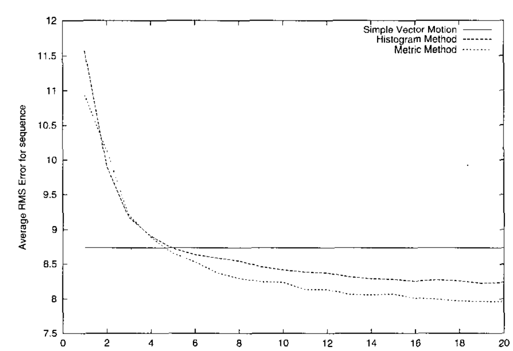

Figure 4: Average RMS error over the foreman sequence for a varying number of objects.

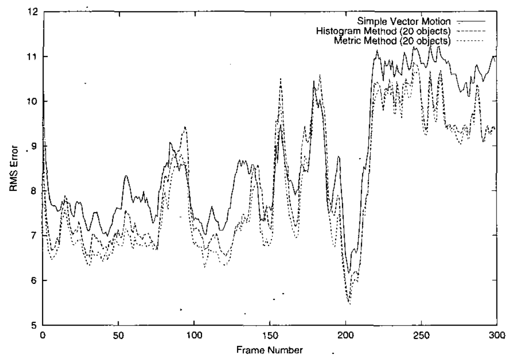

Figure 5: Frame-by-frame RMS error for the foreman sequence, ($N = 20$).

objects ($N < 5$). However, for $N \geq 5$ both new methods out-perform SVM and again the error for the metric method is less than that for the histogram method.

Figure 5 shows the frame-by-frame results from the foreman sequence. In this case the maximum number of motion objects allowed was $N = 20$. It can be seen that for most of the sequence the both the histogram method and the metric method out-perform SVM. However, there are still occasions in the sequence where the SVM gives the best results, though the improvement in for these frames is marginal.

Finally, the effect that the choice of padding has on the results was investigated. In comparison with the extrusion method, the algorithm was also tested with null padding, where pixels outside the reference image were assumed to have a value of zero. The results of this experiment are given in figure 6 which shows that the choice of padding technique can have a large effect on the overall error.

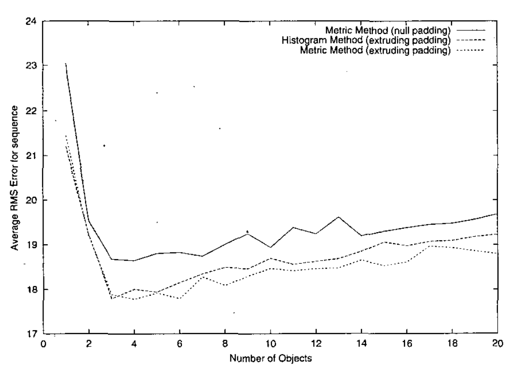

Figure 6: The effect of the choice of padding.

4. CONCLUSION

The metric method proposed in this paper has been shown to improve upon the results obtained when the histogram method is used to pick the list of N vectors. It can also be seen that both methods can give better results than SVM which always sends one vector per block.

The results have demonstrated that the minimum error position for different sequences occurs at different values of N and therefore for different motion description bandwidths; the dynamic choice of N is an area of current work. The importance of choosing a good padding method has also been demonstrated.

It is interesting to note that sequential picking of N vectors does not guarantee the optimal vectors (in terms of minimising the SAD). Finding the optimal N vectors is a highly computationally expensive process, due to the combinatorial nature of the problem. Ongoing work is investigating methods for further improving the vector selection process to give a closer approximation to the optimal set.

5. REFERENCES

[1] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard.* Chapman & Hall, 1996.

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

[3] 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*, vol. 5, pp. 2613–2616, 1998.

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

[5] S. Tredwell and A. N. Evans, "Embedded quad-tree motion estimation for low bit rate video coding," in *IEE European Workshop on Distributed Imaging, Ref. 1999/109*, pp. 3/1–5, November 1999.

[6] S. Tredwell and A. N. Evans, "Block grouping algorithm for motion description encoding." To appear in Proceedings IEEE International Symposium on Circuits and Systems, Sydney May 2001.