ADAPTIVE FEATURE-BASED MOTION ANALYSIS AND ITS APPLICATION TO INTERFRAME CODING

N W Garnham and M K Ibrahim

University of Nottingham, UK

INTRODUCTION

Contemporary methods of interframe coding use motion estimation and compensation as a means of reducing the prediction error in algorithms that provide a large reduction in data rate. These methods can broadly be divided into two categories - pel recursive and block matching algorithms. Further work has investigated model-based coding, using the displacement and deformation of image objects as the criteria for interframe coding.

Block-based methods are well documented and whilst simple, they normally require a large computational overhead, which may be incompatible with the real-time nature of a video codec system. Such techniques range from a simple full-search method [1], to the segmentation of an image into squares of different sizes using quadtree decomposition [2][3]. Both techniques use statistical relationships in order to deduce interframe motion between blocks.

Consideration of a typical audiovisual scene (figure 1) indicates that a search of all block in the image is unnecessary, since motion is mainly concentrated in the detailed area around the subject's face. The technique described in this paper can be considered as a step between the simple full-search block matching method and the more involved approach of model-based coding. It will be demonstrated that, using a low-resolution image, objects can be segmented into features of consistent texture, small enough to provide interframe detail and large enough to provide the basis of an algorithm of low complexity and real-time processing potential.

FEATURE CLASSIFICATION

Low-resolution pre-processing

The first step in the feature-based algorithm is the decomposition of image subregions into a classified set of different shaped features. Initial tests have shown that the majority of shapes over two video frames retain their fundamental outline, irrespective of motion they may have undergone. The classification of these shapes and subsequently searching for their origin in adjacent frames allows a speedy derivation of the nature and magnitude of their displacement.

To increase the speed of the classification and searching algorithm, pre-processing is used to quantise the image and then set 2×2 blocks of pixels equal to the value of their group mode. The effect of this process is an image of reduced resolution and spatial quality, however the groups of pixel values which constitute mainly stationary areas are much larger and the spatial contrast of foreground objects, likely to undergo some type of displacement, is increased. This reduced set of clearly-defined pixel clusters allows implementation of the algorithm, making use of the perceptive tendency to notice high-contrast objects undergoing motion [4].

Figure 2 shows the result of quantisation and mode blocking used in pre-processing, where the quantisation step interval, Δ, is set to eight levels of luminance. It can be seen that the overall effect is not significantly detrimental to the spatial quality of the image. When viewed as part of a sequence at thirty frames per second, the reduced resolution is hardly noticeable.

The effect of pre-processing is to remove completely single-pixel features, which add little to the spatial quality of the image, but would require separate accounting in any scheme that classified all constituent features. Use of the group mode value ensures that large areas of the image will be unaffected by the blocking effect, as no new values are being introduced (as would be the case if mean values were used).

Subregion classification

Classification uses region-oriented segmentation [5], considering pixel relationships rather than the boundaries between groups of pixels. Consider that an image, R, may be comprised of up to n subregions:

$$\bigcup_{i=1}^{n} R_i = R$$

where R_i is a connected subregion, $i = 1, 2, \ldots, n$

and $R_i \cap R_j = \varnothing$ for all i and j, $i \neq j$

where $\varnothing$ is the null set, demonstrating that adjacent subregions are disjoint.

For all the constituent pixels of R, we can define a subregion using a predicate $P(R_i)$ = TRUE for $i = 1, 2, ..., n$ which shows that the pixel values (in this case) will be the same and predicate $P(R_i \cup R_j)$ = FALSE for $i \neq j$, indicating that pixels in adjacent regions are different.

The nature of the segmented subregions is shown in figure 3, taken from the image sequence *Salesman*. In the enlarged area of figure 3(b), two distinct shapes are highlighted - subregion 1 is part of the background and retains its outline and position in the next frame, figure 3(c). However, subregion 2 is part of the moving box and its morphology changes due to motion. Subregion 2 will require reclassification, whilst subregion 1 has not changed or been displaced and so can be represented by a null vector.

To perform subregion segmentation, a raster-type scan is employed, working from the top left-hand corner of the image to the bottom-right. Each pixel found to be different from its predecessors is taken as the seed pixel. Starting at this seed location and working across and down, a subregion of pixels is grouped together so that they all have spatial adjacency and are equal to the seed pixel value. The algorithm uses the four-neighbour principle, where pixels are considered spatially adjacent if they are neighbours in horizontal and vertical directions only, which can be depicted:

```
  q
t p r
  s
```

where q is vertically unit-distance adjacent to **p** and thus a member of the four-neighbour set $N_4(\mathbf{p})$

When no further pixels can be added to the subregion in this way, it is classified and the algorithm moves back to the next pixel in the scanning process, not already marked as being part of a subregion.

The classification of a subregion comprises the origin (x, y), a perimeter runlength descriptor, the pixel mode value and area. Thus we have a simple means for the comparison of subregions between frames, using look-up tables to seek similar values for each of the four feature parameters, as shown in figure 4. Whilst intensive for smaller areas of motion detail, the effect of pre-processing ensures that larger features of constant mode value, such as would be found in a simple stationary background, use no greater degree of description. As will be mentioned later, static features can easily be noted, allowing the algorithm to concentrate on local feature classification for areas of regular motion.

As a further refinement of feature classification, a shorthand notation may be developed for frequently-occurring primary shapes. In place of a runlength descriptor, a simple code can be used to describe the outline, together with a coefficient showing enlargement from primitive unit-length values. Figure 5 demonstrates how a t-shape, expressed as unit-distance values, can simply be multiplied to any larger shape of the same perspective outline. The nature of the algorithm is adaptive, ensuring that whilst some primitives, particularly simple rectangles, may be held in permanent look-up directories, others found to be popular in a given sequence may subsequently be added, with conversion made between the runlength descriptor and the primitive code. This technique has proved most useful, since it reduces the extent to which runlengths need to be re-constructed in order to deduce interframe relationships.

MOTION VECTORS

The feature descriptor is stored in a look-up directory, accessible for subsequent motion vector generation. The algorithm has been developed to be completely adaptive, allowing directory amendment and extendible local searching. The comparison of look-up directories for temporally adjacent frames results in a set of vectors. Where no motion has been found, the origin (x, y), followed by a null vector denotes no change. For subregions displaying evidence of motion, the data sent to the decoder may consist of a vector, together with all or part of the classification code where some parameter of the feature has altered. Figure 6 shows the result of employing the displacement information to reconstruct a frame. It can be seen that whilst most of the image has been successfully reconstructed, some features have been distorted, causing overlaps or voids. Where the feature-based motion analysis algorithm has been applied as motion compensation in a predictive hybrid codec scheme [6], it has been shown that such errors would be interpolated by the inclusion of DPCM data at the decoder.

IMPLEMENTATION

The feature analysis algorithm was implemented in the laboratory using 150 frames of the CIF sequence *Miss America* and 100 frames of the sequence *Salesman*. Both sequences, whilst essentially typical of audiovisual scenes, demonstrate the extent to which the algorithm can be applied. *Miss America* is fairly simple and interframe displacements are not significant, whereas the *Salesman* sequence has an irregular background intensity that will lead to more subregions being created when exposed or covered by foreground activity.

Quantisation was applied throughout the sequences, with Δ=8 and 2×2 pixel blocks set to their mode value, reducing the spatial resolution to 25%. It is easily seen that the size of subregions is governed by the level of quantisation employed. Figure 7 shows the frequency distribution of features, based on the quantity of constituent 2×2 mode-value blocks, for a single frame of the *Miss America* sequence. It is seen that the increase in Δ reduces the overall size of the classification sourcebook. This is of considerable benefit as an overall reduction in the time required for local searching.

Vectors produced by comparing interframe subregion classifications are received by the decoder as either single or vector pairs. Uniform displacements, with no change in form, are easy to map, as the origin and feature runlength may simply be translated on the reconstructed image. However, it has been found that the process of rotation normally leads to some other change in the subregion topography, suggesting that other components of the classification will be required, in addition to a displacement vector pair, showing the destination of the first origin and the source of the second origin (figure 8). Although this is an obvious limitation, in practice no significant decoding effects have been observed in the test sequences, suggesting either that subregion rotation in general is not a normal feature in these sequences, or that the overall rotation of larger objects is practically compensated as a series of uniform displacements for each of the smaller constituent features. Indeed, it was noted that vector pairs accounted for less than 2% of all vectors produced for the *Miss America* sequence.

Inspection of the reconstructed image reveals a number of spatial defects, particularly where the absence of interpolation leaves voids or subregion overlaps. However, the ability of the algorithm to use classification detail to rectify errors on subsequent frames makes error propagation an insignificant issue. A plot of the signal to noise ratio for the reconstructed sequence is shown in figure 9.

Further work has developed mode-value spatial filtering to remove these displaced subregion defects, analogous to normal lowpass filtering. As such defects are rarely more than a group of a few pixels, a filter template of 3×3 is sufficient. The results are most satisfactory and are shown in figure 10. It can be seen from the plots that the overall effect is to reduce the signal to noise ratio. To allow a comparison with conventional full-search block matching motion compensation, figure 9 shows both signal to noise ratio plots. It can be seen that whilst both result sets are around the same mean value, the classified feature approach results in less overall variation.

Conclusion

This paper has introduced a novel method of interframe coding using displacement vectors generated by the analysis of classified feature motion. The technique can compensate for both uniform and complex motion in varying degrees of efficiency. Whilst errors are introduced by way of quantisation and mode-value blocks, their effect is to lower image resolution and it has been shown that the net propagation of errors in a sequence is not evident. The use of spatial filtering allows reconstruction errors to be removed, improving the perceived spatial and temporal quality of the video sequence.

References

[1] Jain J R and Jain A K, "Displacement measurement and its application to interframe coding", IEEE Trans, COM-29 (12), 1799-1808

[2] Chan M H, Yu Y B and Constantinides A G, "Variable size block matching motion compensation with applications to video coding", Proc IEE, 137, 205-212

[3] Seferidis V and Ghanbari M, "Generalised block-matching motion estimation using quad-tree structured spatial decomposition", Proc IEE, 141, 446-452

[4] Treisman A, "Features and objects in visual processing", Scientific American, 255, 115-125

[5] Gonzalez R C and Woods R E, "Digital Image Processing", Addison-Wesley, 1992

[6] Garnham N W and Ibrahim M K, "Adaptive sub-region variable shape motion compensated prediction", Proc SPIE, 2298, 35-44

Figure 1: Frame 1 of the *Miss America* sequence

Figure 2: The effect of spatial pre-processing

Figure 3(a): Frame 58 from the *Salesman* sequence

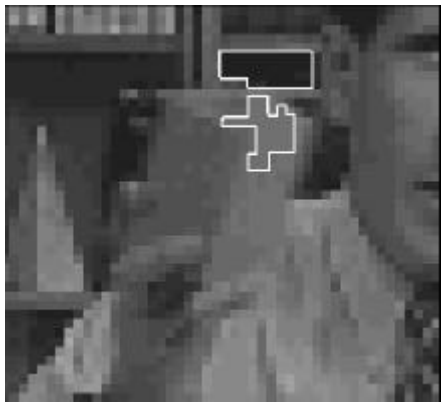

Figure 3(b): Enlarged area showing the subregions

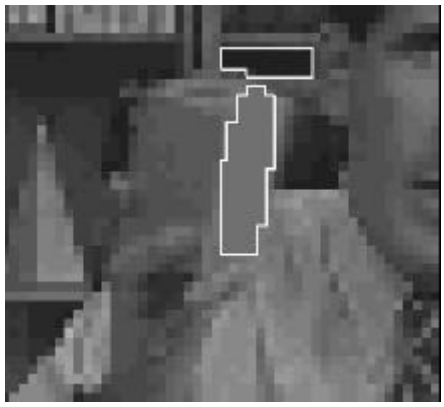

Figure 3(c): Subregion displacement (next frame)

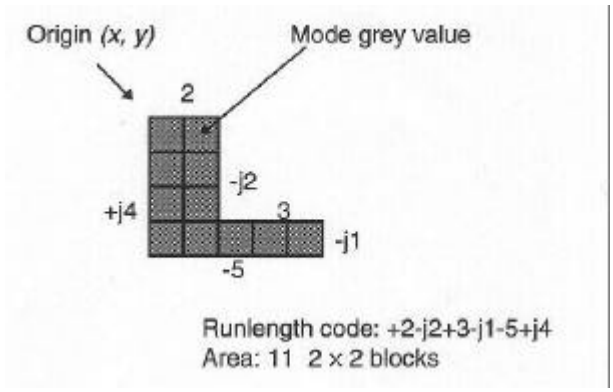

Figure 4: Feature parameters for classification

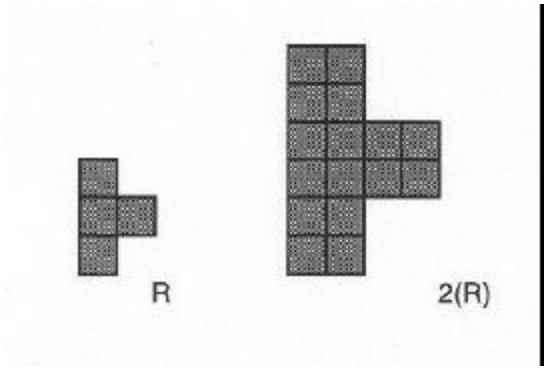

Figure 5: Description as a multiple of a unit-length feature

Figure 6: Reconstructed frame

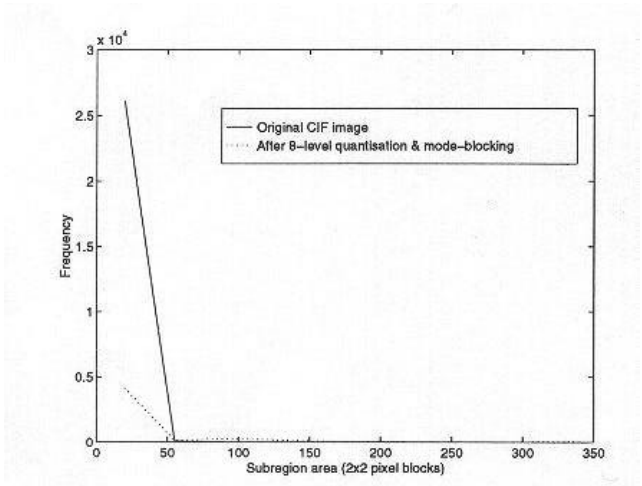

Figure 7: Subregion area frequency distribution

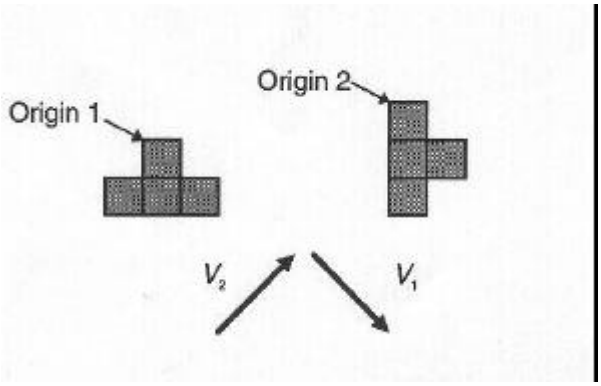

Figure 8: Generating a vector pair describes rotation

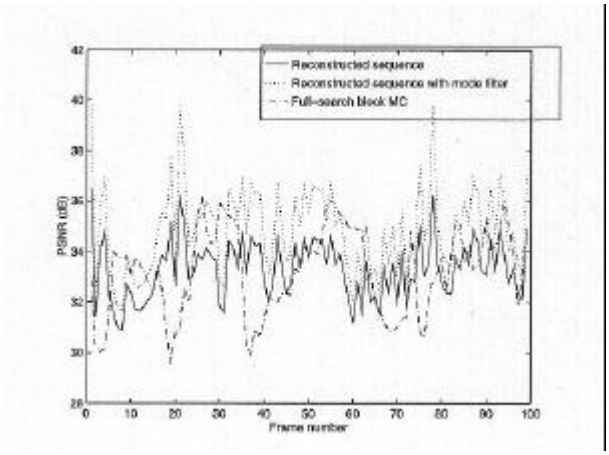

Figure 9: Reconstructed sequence signal to noise ratio

Figure 10: The effect of spatial filtering