MOTION COMPENSATION OF MOTION VECTORS

Joseph Yeh†, *Martin Vetterli*‡, *and Masoud Khansari*

231 Cory Hall
Department of EECS, University of California
Berkeley, CA 94704
jyeh@po.eecs.berkeley.edu

ABSTRACT

Motion compensation of image data has long been used for video compression. "Meta-motion" compensation, or motion compensation of motion vectors is presented as a new approach to data compression useful in very low bitrate (VLB) applications. Whereas present motion coding simply takes advantage of temporal redundancies among images, this new, computationally simple approach takes advantage of temporal redundancies between subsequent vector fields to efficiently code motion. Taking advantage of both the temporal redundancy of motion vectors and the "mapping" property of motion vectors, this method is shown to decrease the bitrate for vector coding in sequences with complex motion.

1. INTRODUCTION

It is a well-known fact that for video signals, there is a large amount of redundancy along the temporal dimension. Therefore, a considerable amount of compression can be achieved through removal of these redundancies. This is indeed the approach taken in various video compression standards (e.g., H.261, MPEG-1,MPEG-2) and other proposed coding algorithms. A popular approach is to detect and compensate for image motion to predict the present frame from one or more previous frames. The value of these motion vectors and the prediction residuals are then transmitted.

In very low bitrate video coding, the bitrate used to represent motion vectors can become a non-negligible portion of the bit budget. Thus, efficient methods to code the motion vector field are necessary. So far, only intraframe dependency has been used to code this vector field and the inter-dependencies among these consecutive vector fields are not taken into consideration.

Currently, MPEG-1 and 2 use first-order intraframe differential coding of motion vectors to compress motion data. Although intraframe differential coding will be effective when the motion field is smooth, it does not achieve satisfactory results when the motion vectors vary greatly from one block to the next. Differential coding is the most common strategy used to compress image information, although work has been done to achieve compression with vector quantization of motion vectors [2] and layered decomposition of motion vectors [3].

This paper presents a new method of motion coding different from the methods stated above. It explains an algorithm that takes advantage of the temporal redundancy between motion vector fields to reduce the cost of sending motion information. Note that algorithm is decoupled from the actual process of estimating motion from images, and can be used on vectors computed by any motion estimation scheme.

The following section will explain and illustrate our method of motion coding using a simple, three-frame image sequence. This paper then discusses potential problems of our method, presenting simulations and results afterward.

2. INTERFRAME CODING OF MOTION VECTORS

Consider three successive frames of a video sequence: $I[1]$, $I[2]$, and $I[3]$. From this sequence, two motion fields can be defined; $\mathbf{V}[2]$, which allows the decoder to compose $I[2]$ from $I[1]$, and $\mathbf{V}[3]$, which allows the decoder to compose $I[3]$ from $I[2]$. Figure 1 shows the three underlying images, and the second row of Figure 1 shows quiver plots of the motion vector fields extracted from the three images.

Essentially, the sequence of motion vector fields has few distinguished differences from the sequence of time-varying images. First, a two dimensional vector, namely x and y velocity components, is assigned to each block of pixels. Whereas an image is a two dimensional array of scalar values, a motion vector field is a two dimensional array of vector values. In most video coding algorithms, motion vector fields are used to reconstruct images in a video sequence from other images in the same sequence. In that sense, a motion vector field can be seen as an *operation* or mapping, transforming an image $I[\mathrm{n}]$ into another $I[\mathrm{n}+1]$, where n refers to an instance in time. In the same sense, a motion vector field can be applied to itself, transforming into another motion vector field.

An example can be given with pixel-accuracy motion vectors. If there is a vector $< 3,4 >$ at pixel location $(1,2)$, a guess could be made that in the next frame, pixel location $(1+3, 2+4) = (4,6)$ has the vector $< 3,4 >$.

†This work was supported in part by an undergraduate research stipend from the College of Engineering at the University of California at Berkeley.
‡This work was also supported in part by NSF grant MIP-93-21302.

0-8186-7310-9/95 $4.00 © 1995 IEEE

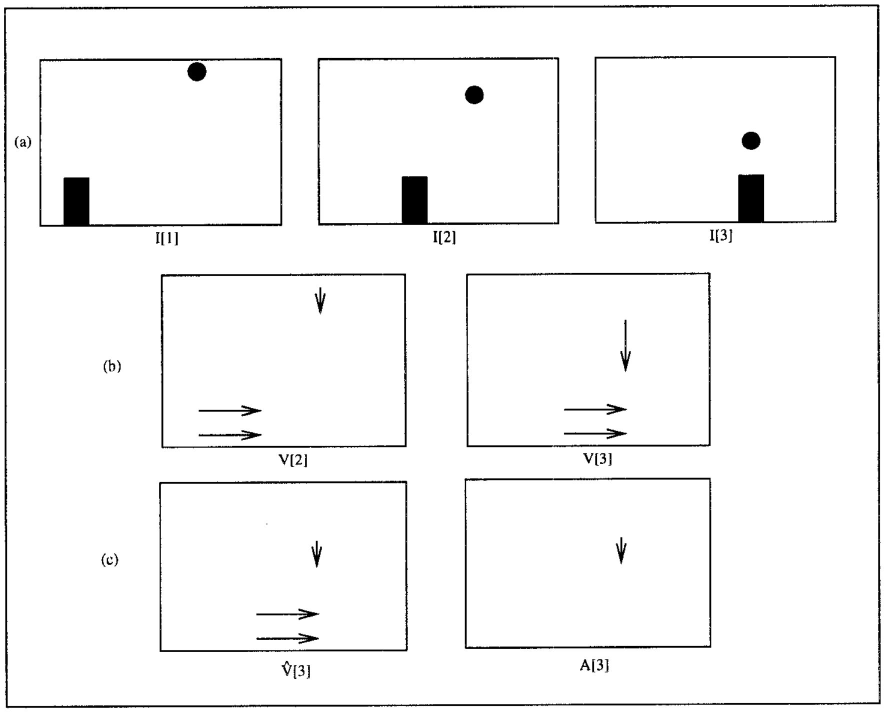

Figure 1: (a) Three hypothetical images of a video sequence. The rectangular object is moving at constant velocity, while the falling circle is accelerating. (b) The extracted motion sequence. (c) The compensated field $\hat{\mathbf{V}}[3]$ and its difference from $\mathbf{V}[3]$.

To generalize, let $\mathbf{V}$ be a motion vector field where $\mathbf{V}_{(\mathbf{z})}$ is the value of the motion vector field at location $\mathbf{z}$. Then one can find a new motion vector field through an operation called *autocompensation*:

$$\hat{\mathbf{V}}[\mathrm{n}+1]_{(\mathbf{z}+\mathbf{k})} = \mathbf{V}[\mathrm{n}]_{(\mathbf{z})} \qquad (1)$$

where

$$\mathbf{k} = \mathbf{V}[\mathrm{n}]_{(\mathbf{z})} \qquad (2)$$

First, consider the case where the objects in the scene are moving with constant velocity (both the magnitude and the direction of the speed are constant). Then, the motion vector field at time $n+1$ can be found by autocompensating the field at time n using Eqn. (1). Similarly, by autocompensating $n+1$, one can find $n+2$, etc. Therefore, the knowledge of $\mathbf{V}[0]$ (the first motion vector field) is sufficient to generate the entire sequence.

However, in most cases, the velocity of objects is not constant, and a way must be found to account for this acceleration. Hence, let us now define

$$\mathbf{A}[\mathrm{n}] = \mathbf{V}[\mathrm{n}] - \hat{\mathbf{V}}[\mathrm{n}] \qquad (3)$$

Note that $\mathbf{A}[\mathrm{n}]$ corresponds to the acceleration of the objects in the scene and also that knowing $\mathbf{V}[\mathrm{n}]$ and $\hat{\mathbf{V}}[\mathrm{n}]$ is sufficient to characterize $\mathbf{A}[\mathrm{n}]$. If we now assume that the direction of the motion does not change with time and only its magnitude is time-varying, then the direction of each element of $\mathbf{A}[\mathrm{n}]$ is the same as that of the corresponding motion vector in $\mathbf{V}[\mathrm{n}]$ and hence a scalar component would be sufficient to describe each component of $\mathbf{A}[\mathrm{n}]$. In general, however, both the magnitude and the direction of the velocity change with time and a vector component is necessary to characterize the acceleration.

The energy of the components of $\mathbf{A}[\mathrm{n}]$ are not necessarily smaller than that of $\mathbf{V}[\mathrm{n}]$ since the acceleration can be due to the change in the direction and not the magnitude of the motion vector field. As a result, for coding applications, it may not be desirable to find the acceleration field from Eqn. (2) and further refinement of $\hat{\mathbf{V}}[\mathrm{n}]$ is necessary.

The third row of Figure 1 shows the field $\hat{\mathbf{V}}[3]$ extracted from $\mathbf{V}[2]$ and also the acceleration field $\mathbf{A}[3]$ resulting from subtracting $\hat{\mathbf{V}}[3]$ from $\mathbf{V}[3]$. As can be seen, the vector field $\mathbf{A}[3]$ will be much easier to code than $\mathbf{V}[3]$, while providing just as much information to the receiver. Figure 2 shows a block diagram of the entire vector coding system.

3. IMPLEMENTATION OF METHOD

Uncovered boundaries, uncovered objects, and fading have presented problems in the whole area of motion estimation from images. In turn, motion estimation of motion vectors presents even more problems. In the following paragraph, this paper explains two major problems of motion estimation: ill-posedness, and vector location of non-integer val-

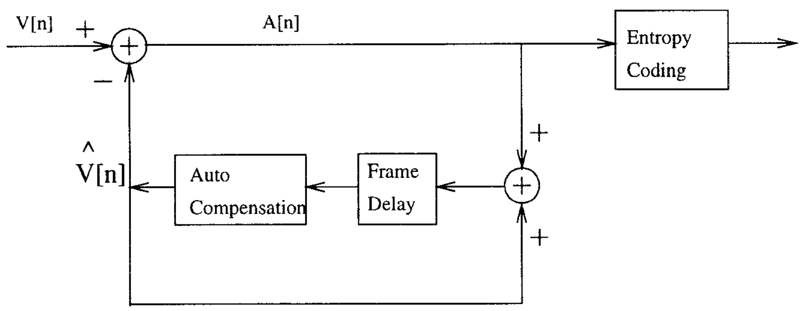

Figure 2: Block diagram of a whole "motion compensated" motion vector coding system.

ued vectors. The last paragraph of this section will discuss a method of solving another problem, motion compensation on non-intensity based vectors.

In luminance-based motion estimation from images, a portion of an image $\mathbf{I}[n+1]$ may not be found in $\mathbf{I}[n]$, causing an ill-posed problem in the estimation. When estimating a subsequent motion field with (1), similar ill-posed problems can occur. For example, if two or more motion vectors in a motion field $\mathbf{V}[n]$ point to the same location z, then there is a conflict as to which vector will be represented in $\hat{\mathbf{V}}[n+1]$. In the same way, there will be a problem if all of the vectors in the field point away from a location z. Therefore, there will be no motion vector representing z in the subsequent field.

A problem related to and perhaps even more important than the above is vector location. In order to be useful, vectors in fields must correspond to a certain pixel (for pixel motion vectors) or block (for block motion vectors) location. Equation (1) does not guarantee this condition. Therefore, a re-interpolation must be done around pixel locations to compute the new motion field. A weighted average of vectors surrounding the location is used; this equation summarizes the method:

$$\hat{\mathbf{V}}[n+1]_{(i,j)} = \Big(\sum_{a} \frac{1}{d_a^2}\Big)^{-1} \sum_{a} \frac{1}{d_a^2}\mathbf{v}_a \qquad (4)$$

Where $\mathbf{v}_a$'s are all of the vectors in the field $\mathbf{V}[n]_{(\mathbf{z}+\mathbf{k})}$ which are located within a certain distance from location (i, j), and the d_a's used for the weighing averages correspond to the distance of these vectors from location (i, j). Note that this method gets around the case of no vectors being located at (i, j), however, when a vector is actually located at (i, j), it takes precedence over all other vectors. Currently, when more than one vector is located at (i, j), our algorithm chooses one at random.

Another problem of motion compensation of motion vectors in general is that the motion compensation does not take into account the underlying luminance/chrominance values of the images. Therefore, motion compensation of motion vectors may actually increase the transmission bandwidth needed for image residuals when the calculation of motion vectors is not based on luminance. Hence, we need a correction factor to $\hat{\mathbf{V}}[n]$. A vector field $\mathbf{D}[n]$ is found through an exhaustive search algorithm which minimizes the image residual energy caused by $\tilde{\mathbf{V}}[n]$, which is defined by:

$$\tilde{\mathbf{V}}[n] = \hat{\mathbf{V}}[n] + \mathbf{D}[n] \qquad (5)$$

In this situation, Equation (1) becomes

$$\mathbf{A}[n] = \mathbf{V}[n] - \tilde{\mathbf{V}}[n] \qquad (6)$$

4. SIMULATIONS AND RESULTS

The goal of our simulations was to compare intra-frame differential vector coding, the most common method used currently, with our method of inter-frame differential vector coding. We used the 512x512 "MIT" sequence for our simulations, taking the upper left 256x256 quadrant and lower right 256x256 quadrant as separate "sequences". The upper-left quadrant, with computer generated images combined with television sitcom samples, represented a sequence with complex motion. The lower-right quadrant was the result of a camera panning and zooming on a stationary photograph, and represented simple motion.

We computed 8x8 block motion vectors for both sequences using an exhaustive search algorithm to minimize (this equation is adapted from [1])

$$\text{error} = \sum_{\text{block}} (I[n]_{(x,y)} - I[n-1]_{(x-V_x[n],\, y-V_y[n])})^2 \qquad (7)$$

We calculated $\mathbf{A}[n]$ vector fields for the two sequences, and scanned all the $\mathbf{A}[n]$ fields into a "inter-frame" vector matrix. Figure 3 shows a original vector frame from the complex motion sequence, the frame after autocompensation, and the difference from the next frame.

For comparison, we also scanned each of the $\mathbf{V}[n]$ vector fields into matrices, took the first order difference of the matrix, and and then combined all the fields into a "intra-frame" vector matrix. This was done to simulate present vector coding methods.

The x and y dimensions of each the vector matrices were separated, and for each vector matrix, we measured the zeroth-order entropy of the x and y dimensions, afterwards adding the two entropies to calculate a cost estimate of the bitrate per vector.

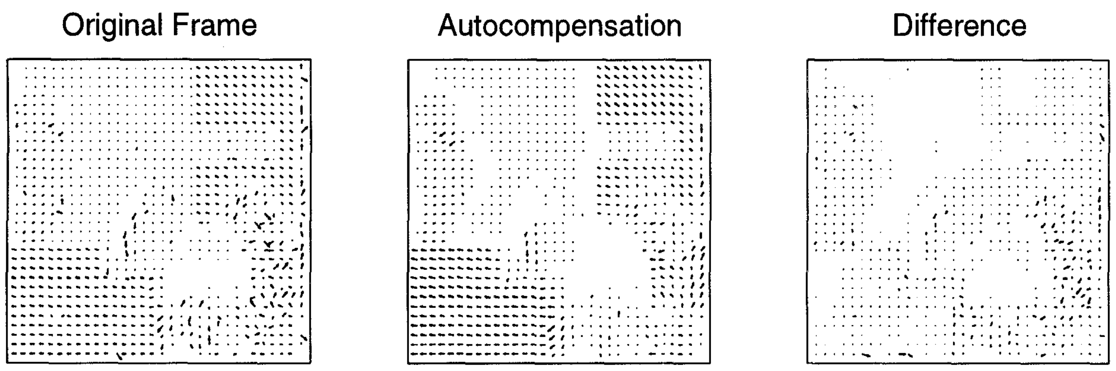

Figure 3: The vector field on the left is a $\mathbf{V}[n]$ motion field, which is autocompensated to give the $\hat{\mathbf{V}}[n+1]$ field in the center. The vector field on the right is the difference $\mathbf{A}[n]$ from the next calculated field $\mathbf{V}[n+1]$. Empty spaces in the displayed fields represent zero vectors.

Because groups of vectors in the matrices often have the same value, we also used run-length coding. We evaluated an equivalent bitrate by multiplying the entropy of the run-lengths by the total number of run-lengths, and then dividing by the total matrix size to obtain the equivalent motion vector bitrate.

The following table summarizes our results. Note that without compression, the vectors would take 8 bits (4 bits in each direction) to code.

Sequence	Estimated Bitrate			
	Vectors		Run Lengths	
	Inter frame	Intra frame	Inter frame	Intra frame
Simple Motion	3.09	2.50	2.54	2.30
Complex Motion	4.91	4.91	4.49	4.82

Run-length coding improves the bitrate in our inter-frame coding scheme. Our results show that autocompensation can reduce the bitrate for complex motion sequences, but additional work is required to improve th performance in simple motion sequences.

5. CONCLUSION

Autocompensation of motion vectors has been presented as a new approach to coding motion for very low bitrate applications. Taking advantage of both the temporal redundancy and the "mapping" property of motion vectors, this method predicts a subsequent motion vector field from a previous vector field without any side information. A simple version of this method based on interpolation has been discussed in this paper. Used with run-length coding, it has decreased the bitrate for vector coding in sequences with complex motion. Further investigation into different methods of compensating vectors , including both lossy and lossless methods, is still needed in order to demonstrate full applicability to real-time video coding.

6. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Michael Orchard of the University of Illinois at Urbana-Champaign and Dr. Janusz Konrad of INRS-Telecom/Québec for their insightful suggestions and discussion.

7. REFERENCES

[1] Barry G. Haskell and Arun N. Netravali. *Digital Pictures: Representation and Compression.* Plenum Press, 1988.

[2] Yoon Yung Lee and John W. Woods. Motion vector quantization for video coding. *IEEE Transactions on Image Processing,* March 1995.

[3] John Y. A. Wang and Edward H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing,* September 1994.