REPRESENTATION AND ESTIMATION OF MOTION USING A DICTIONARY OF MODELS

Daniel Lauzon and Eric Dubois

INRS-Télécommunications, 16 Place du Commerce
Ile-des-Soeurs (Verdun), Québec H3E 1H6, Canada
lauzon@inrs-telecom.uquebec.ca, eric@inrs-telecom.uquebec.ca

ABSTRACT

This paper presents a novel method for representing motion information based on a *Dictionary of Motion Models* and a *Tag Image* which indicates which motion model is used at any given image position. Each model is composed of low-order polynomial-based motion fields. The motion in most sequences can be adequately represented by a very small number of such motion models. We further present an efficient way of estimating and coding this representation. Comparative results are presented which indicate a performance superior to that of motion representations found in classical block-based codecs.

1. INTRODUCTION

In video coding standards widely in use today there is a need to represent the motion information that is used to perform motion compensated coding. This paper addresses the efficient representation of motion information. The efficiency of the representation is measured in terms of its capacity to perform good motion compensation at a low coding cost. Known representation techniques such as those in use in MPEG-2 are not flexible enough and do not represent the motion in a compact enough way.

The motion present in typical video sequences can be represented by a few simple motion models. To exploit this fact, we represent these few motion models in a dictionary (or codebook) and assign a tag to each image position to determine which motion model should be used at that position. The tag information is spatially very redundant and entropy coding techniques can be used to encode it very compactly.

In this paper, we wish to introduce the approach of representing motion by a dictionary of models. We then demonstrate its feasibility by reporting the results of a first implementation of both the estimation and coding aspects of our approach.

Countless others have addressed many different aspects of motion representation. Some authors [7, 8] have addressed the representation of motion fields in parametric fashion using polynomial models of varying degree: these are however typically applied to blocks (sometimes of varying size) independently. Others [6, 1, 2, 9] have addressed the need to incorporate a rate constraint in the estimation of the motion information to be sent.

However none of these methods exploits the fact that a typically very small number of motion models can be used to represent the motion information effectively.

2. REPRESENTATION OF MOTION

For our purposes motion is mapping which relates spatial coordinates (p) in one image to spatial coordinates in another (p'):

$$M_i : p \mapsto p' \quad p = (x, y).$$

A model M_i is defined for all x, y in an image. This kind of motion is often used to form a motion compensated predictor (P) from another image (I) as in:

$$P(p) = I(p - M_i(p)) \quad \forall p \in R_i.$$

where the R_i partition the image into disjoint regions and a different motion M_i is applied to each region. In most approaches the regions R_i are simply blocks of possibly varying size which tile the image.

2.1. Motion Models

In many current video coding contexts motion is simply modeled as a pure translation:

$$M(p) = \begin{bmatrix} u_x^0 \\ u_y^0 \end{bmatrix}$$

But some [7, 8] have extended motion compensation to higher order models such as: Affine Transformations,

$$M(x, y) = \begin{bmatrix} u_x^0 + u_x^1 x + u_x^2 y \\ u_y^0 + u_y^1 x + u_x^2 y \end{bmatrix}$$

Bilinear Transformations,

$$M(x, y) = \begin{bmatrix} u_x^0 + u_x^1 x + u_x^2 y + u_x^3 xy \\ u_y^0 + u_y^1 x + u_x^2 y + u_y^3 xy \end{bmatrix}$$

and even higher order models. All of these generalize to:

$$M(p) = \begin{bmatrix} \sum_m u_x^m g_m(p) \\ \sum_m u_y^m g_m(p) \end{bmatrix},$$

where the $g_m(p)$ are low-order polynomials as functions of spatial position $p = (x, y)$.

A motion model is then completely determined by the basis functions g_m and associated model constants u_x^m, u_y^m. We also define the model order as the maximum polynomial degree of the $g_m(p)$, for example the *Bilinear Transformations* defined above are models of order 2.

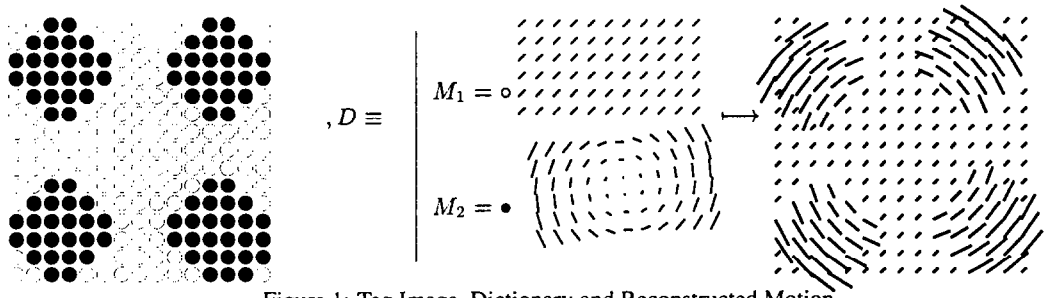

Figure 1: Tag Image, Dictionary and Reconstructed Motion

2.2. Dictionary and Tags

We define a Dictionary D as a collection of N motion models M_i as defined above:

$$D = \{M_1, \ldots, M_N\}.$$

We further define a tag map $T(p)$ which associates an index into our dictionary to each spatial position in the image:

$$T(p) \in \{1, \ldots, N\} \quad \forall p \in \text{Image}.$$

Figure 1 illustrates how the motion representation we introduce is composed of the Dictionary D and the tag map $T(p)$ which together define motion over an entire image:

$$M(p) = M_{T(p)}(p) \quad \forall p \in \text{Image}.$$

In a slight refinement of the model, we may force the tag value $T(p)$ to be constant inside a small block (say of 4x4 or 8x8 pixels) which allows us to represent the tag map at a lower resolution. Note also, that even if the tag value $T(p)$ is made constant inside a small block, the reconstructed motion $M(p)$ at the different positions in the block may still vary.

3. ESTIMATION

The estimation process produces the dictionary and the associated tag map. We may wish to find a representation which best approximates a known or independently estimated motion field, or alternatively we might want to estimate the representation which directly minimizes an objective function, such as minimizing the magnitude of the prediction error induced by the motion representation.

$$\min_{D,T(p)} \text{Cost}(D, T(p))$$

We may view our motion representation as a form of quantizer where the Dictionary defines a number of representation levels for motion vectors. This view is not typical inasmuch as the quantizers' output levels vary with the position of the quantized motion vector, but otherwise the analogy holds. With this in mind we can define our estimation process as a Generalized Lloyd Algorithm [5] for quantizer design.

With this approach the dictionary defines a set of representative motion vectors at a given position, and the tag for that position identifies which representation level was chosen.

3.1. Quantizer Design

To define a quantizer we need to specify a number of representation levels, and also define how any admissible value will be mapped to one of those levels. With the GLA, a *locally* optimal quantizer is designed by iterated improvement by applying the two following rules successively.

Nearest Neighbor Partitioning In our case this amounts to assigning a tag to each position associated to the Dictionary entry which minimizes some distortion criterion for the motion vector at that position.

$$T(p) = \min_{t \in \{1,\ldots,N\}} d(M_t(p))$$

where $d(M_t(p))$ is the distortion measure applied for the t^{th} model at position p. E.g.

$$d(M_t(p)) = |M_t(p) - M'(p)|^2$$

for a known motion field M' or alternatively

$$d(M_t(p)) = |I(p) - \hat{I}(p - M_t(p))|^2$$

for an Image I and a previous reconstructed frame $\hat{I}$ (Motion compensated Prediction error).

Centroid Computation In our case this amounts to reestimating the model parameters for each model in the dictionary based on all positions which are quantized to that model.

$$M_i \equiv \min_{u} \sum_{\{p|T(p)=i\}} d(M_i(p)) \qquad \forall i$$

This is accomplished using a least squares fit of the model parameters u and the appropriate distortion measure.

3.2. Incorporating an Entropy Constraint

In the process of designing a quantizer it is possible to incorporate coding cost into the distortion measure of the quantizer. This is called *entropy constrained quantizer design* [4]. Its purpose is to find a quantizer which minimizes a distortion measure that takes into account the total rate-distortion performance of the encoding system which includes a quantization step followed by a coding step.

As will be described in section 4, we can calculate the coding cost for the tag information at each position. The rate constraint modifies the distortion measure to bias the choice of tags toward those that are coded at a lower cost. This induces a modified *nearest neighbor* rule:

$$d'(M_t(p)) = d(M_t(p)) + \lambda Rate(T(p)) \quad \lambda > 0.$$

By further refinement it is possible to include a rate constraint which takes into account a conditional coding cost. This is referred to as *conditional entropy constrained quantizer design* [3].

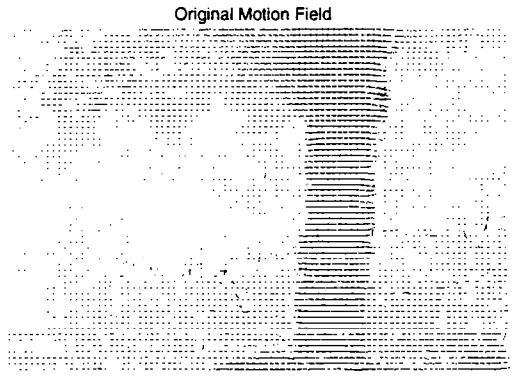

Figure 2: Original Motion Field and Image

4. CODING

To encode the representation we need to encode the dictionary D and the tag map $T(p)$. Encoding the dictionary is straightforward, we simply transmit the coefficients u_x^j, u_y^j for each model using a fixed length representation. The tag map however is encoded in a more sophisticated manner.

The tag for each position in the image is encoded using a conditional arithmetic code. Using an optimal arithmetic code allows us to code a tag value $T(p)$ with approximately $-\log_2(\Pr(T(p))$ bits. The optimal coding cost therefore depends on our modeling of these probabilities. For the bitstream to be decodable, the decoder must also have access to identical values for these probabilities.

4.1. 2-Neighbor Model

We chose to model the probability of the tag $T(p)$ at position p, as conditioned on values of two of its causal neighbors (tags having already been encoded). In our implementation we condition on the tags in the positions immediately preceding p in the horizontal (p_h), and vertical (p_v) directions.

Defining for convenience $t_0 = T(p)$, $t_h = T(p_h)$ and $t_v = T(p_v)$, and the three estimated quantities:

$$\begin{aligned} P_t &= \Pr(t_0 = t) \\ P_{both} &= \Pr(t_0 = t_h = t_v | t_h = t_v) \\ P_{either} &= \Pr(t_0 = t_h \text{ or } t_0 = t_v | t_h \neq t_v) \end{aligned}$$

We define the conditional probability: $\Pr(t_0 \mid t_h, t_v) = P_{\text{cond}}$

$$P_{\text{cond}} = \begin{cases} P_{both} & \text{if } t_h = t_v, t_0 \in \{t_h, t_v\}, \\ (1 - P_{both}) \frac{P_{t_0}}{1 - P_{t_h}} & \text{if } t_h = t_v, t_0 \notin \{t_h, t_v\}, \\ P_{either}/2 & \text{if } t_h \neq t_v, t_0 \in \{t_h, t_v\}, \\ (1 - P_{either}) \frac{P_{t_0}}{1 - P_{t_h} - P_{t_v}} & \text{if } t_h \neq t_v, t_0 \notin \{t_h, t_v\}. \end{cases}$$

This determines a probability model for $T(p)$ as described in section 4. The estimated constants P_t, P_{both} and P_{either} are transmitted as side information; the decoder can then evaluate the conditional probability P_{cond} from known quantities.

5. IMPLEMENTATION AND RESULTS

The described estimation process and encoding scheme were implemented in software and tested under various conditions. The motion from various sequences was used as input to our simulations. Some parameters need to be specified to operate the encoder: the number N of entries in the dictionary, the order O of the model (see end of section 2.1), and the block size b of the tag information which a single entry in the transmitted tag image represents (see end of section 2.2).

Figure 2 shows an original image and an independently estimated fine representation for the motion in the Flower sequence. Figure 3 shows a typical simulation result. We can see that a large portion of the image is represented by a single motion model as represented by a single grey level in the tag image. We can also see that the motion within a region of identical tags is not constant but rather is varying smoothly according to the associated motion model. It is important to note that a single model can represent arbitrarily shaped and possibly disconnected regions.

In table 1 we report on simulations using two measures to quantify the performance of the encoder: the total cost of encoding the motion information (measured in bits per image pixel), and the *peak signal to prediction error ratio* (PSPR). The results are compared to MPEG-2's method of encoding motion information, both at full pixel, and at half pixel resolutions. Our

Seq.	Coding Method	Rate	PSPR	Δ
Flower	MPEG-2 1 pixel	0.006	22.94	
	Dict N=4 O=2 b=16	0.004	27.15	4.21
	MPEG-2 $\frac{1}{2}$ pixel	0.010	24.68	
	Dict N=4 O=2 b=8	0.010	27.89	3.21
Foreman	MPEG-2 1 pixel	0.007	30.00	
	Dict N=4 O=2 b=16	0.005	32.40	2.40
	MPEG-2 $\frac{1}{2}$ pixel	0.013	31.40	
	Dict N=4 O=1 b=8	0.009	33.37	1.97

Table 1: typical simulation results: Rate vs. PSPR

For each MPEG-2 result, we present the result for the best simulation parameters of our method with a rate less than that of the MPEG-2 result. Thus, holding the motion encoding rate constant we can see a net gain in prediction quality in all cases on the order of 2-4 dB.

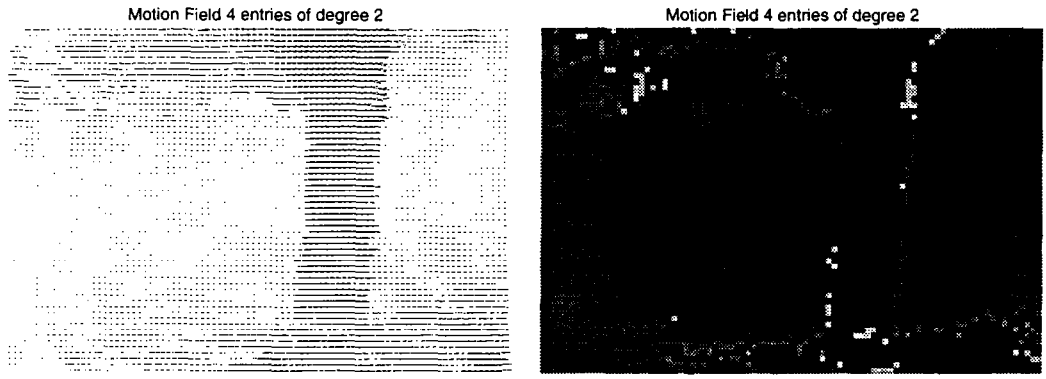

Figure 3: Reconstructed Motion Field and Tag Image,N=4 Order 2

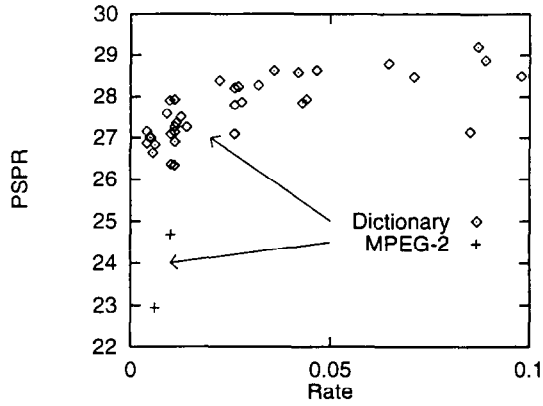

Figure 4: Simulation results for Flower sequence

This is also shown in figure 4, where all simulation results for the Flower sequence are shown, including all combinations of the parameters $N \in \{4, 16\}, O \in \{1, 2\}$ and $b \in \{4, 8, 16\}$. Figure 4 demonstrates the flexibility of the approach, where it would possible to select different encoding parameters N, O and b to attain a desired encoding rate or prediction quality over quite a wide range of values: bit rates from .004 to 0.1 bits per image pixel yielding PSPR values between 26.5 and 29 dB. This flexibility could be exploited in a coder where the rates allocated to motion information and prediction error information could be determined dynamically to optimize a rate-distortion criterion.

6. ACKNOWLEDGMENT

This research was supported by a joint grant from the Canadian Institute for Telecommunications Research under the NCE program of the Government of Canada, the Natural Sciences and Engineering Research Council of Canada and General Datacomm.

7. CONCLUSION

We have introduced a new representation for motion information based on a dictionary of models. Our approach may apply a single motion model to an arbitrarily shaped region without explicit recourse either to segmentation or contour representation techniques. We have also demonstrated that our first implementation is significantly more efficient than MPEG-2 in terms of *prediction quality* and *coding cost*. Other methods of coding the tag image still need to be investigated. We believe the method will show its full benefit when it is integrated into a complete rate-constrained coder.

8. REFERENCES

[1] H. Bi and W.-Y. Chan, "Rate-constrained hierarchical motion estimation using BFOS tree pruning," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. 2315–2318, May 1996.

[2] M. C. Chen and A. N. Wilson Jr., "Rate-distortion optimal motion estimation algorithm for video coding," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 4, pp. 2096–2099, May 1996.

[3] P. A. Chou and T. Lookabaugh, "Conditional entropy-constrained vector quantization of linear predictive coefficients," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, pp. 197–200, 1990.

[4] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, pp. 31–42, Jan. 1989.

[5] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1991.

[6] B. Girod, "Rate-constrained motion estimation," in *Proc. SPIE Visual Communications and Image Process.*, vol. 2308, pp. 1026–1034, Sept. 1994.

[7] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 339–356, June 1994.

[8] C. Papadopoulos and T. G. Clarkson, "Motion compensation using second-order geometric transformations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 319–331, Aug. 1995.

[9] J. Ribas-Corbera and D. L. Neuhoff, "On the optimal motion vector accuracy for block-based motion-compensated video coders," in *Proc. SPIE*, vol. 2668, pp. 302–314, Mar. 1996.