EMBEDDED OPTICAL FLOW MOTION COMPENSATION AND FINITE STATE HIERARCHICAL VECTOR QUANTIZATION

Kunal Mukherjee and Amar Mukherjee

School of Computer Science
University of Central Florida
Orlando, Fl 32816

ABSTRACT

We propose a video coding and delivery scheme which is geared towards low bit-rate and real-time performance requirements. We use a finite state wavelet-based hierarchical lookup vector quantization (FSWHVQ) scheme, which embeds the optical flow calculations in table-lookups. This video coding scheme is both fast (table-lookups) and accurate (dense motion field), and avoids the blocking artifacts and poor prediction which plagues block coding schemes at low bit rates. For restricted image compression/transmission scenarios like teleconferencing, for which a good training set may be available, the FSWHVQ scheme may be viewed as storing as an internal representation in its lookup tables, a valid and complete model of the problem domain.

1. INTRODUCTION

Interest in very low bit rate image/video coding has received a great impetus in recent times ([5],[8]) due to the accessibility of very low bit-rate channels, that were originally designed to transmit speech or text, e.g. radio channels, public switched telephone networks (PSTN) and computer networks for delivering e-mail. The target rate for these channels is typically less than 100 kb/s.

To get this kind of compression, we need to squeeze out the last drops of spatial and temporal correlation, by using a combination of motion compensation (temporal) and block coding or vector quantization (spatial) techniques. MPEG (BMA/DCT) seemed to be a good solution, by estimating the motion of each image block by a single vector ([8]). But at very low bit rates, this coarse approximation suffers from very visible and unpleasant artifacts. Therefore, we are faced with the following design criteria for any practical low bit-rate compression scheme: a) The motion estimation should be accurate. This automatically means that the residual errors will be small, and can be compactly coded; b) Rate/distortion (R/D) should be optimal, and ideally this should be adaptable (i.e. dynamically tunable) according the bandwidth availability; and c) Both encoding and decoding should be fast, possible in real time, and/or realizable in VLSI.

In this paper we are motivated to satisfy the above design criteria as closely as possible. To this end, we propose the following solutions: a) Use optical flow instead of block matching, to obtain a dense and accurate motion field estimation ([2],[4],[5]); b) Use vector quantization, which uses an overall rate-distortion optimization scheme in the codebook design (i.e. training) phase ([6]); and c) use FSHVQ (finite-state hierarchical table-lookup vector quantization, see [7]), which “embeds” the optical flow computations, by storing state information.

This gives a fast encoder and decoder which proceed through table-lookups rather than complex computations. The operational bit-rate can then be dynamically tuned to the available bandwidth, by using successive stages of table-lookups ([1],[3],[7]). HVQ also gives us the added “bonus” of cheap trans-coding to lower bit-rates, by distributing the codebooks over a heterogeneous network (see also [1]).

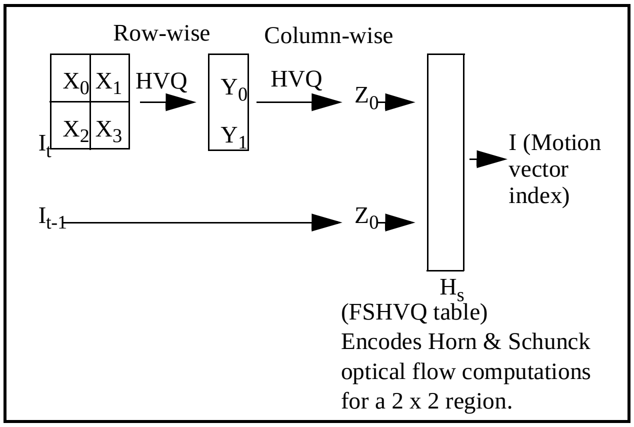
The “building blocks” of our method, i.e. the Horn/Schunck optical flow method, WHVQ (wavelet-based hierarchical table-lookup vector quantization) and FSHVQ (finite state HVQ) are well described in [9], [1], and [7] respectively. Optical flow has been used in other video coding work, e.g. [2],[3],[5], but not in a fast coding scheme like table-lookups which is practical for a real-time encoder, and which can be implemented in hardware ([10]). The idea of embedding higher-level, e.g. classification tasks in the HVQ frame-work was introduced in [3]. Finite state table-lookup hierarchical vector quantization (FSHVQ) is used in [7] for still image coding, using side intensities as the “state”. We extend the FSHVQ idea for image sequences, by using codewords used to encode the previous frame, and previously estimated motion fields as the “state”. The new technique introduced here is the embedding of optical flow in FSHVQ for video coding.

This paper is organized as follows. In Section 2 we show how to use state information to embed optical flow in the FSHVQ framework, for the basic coding scheme. In Section 3 we will explain the rate/distortion optimization in the training phase, and the extension of the basic scheme to perform multi-scale optical flow on smooth wavelet co-efficients, which handles large displacements (see also [2]). We give some results in Section 4, and conclude in Section 5.

2. EMBEDDED OPTICAL FLOW AND FSHVQ

We will illustrate the method with a specific example (Figure 1). In normal HVQ, blocks of pixels are vector quantized (using table-lookups) by quantizing, or “merging” two vectors (at the first step these are a pair of pixel values), alternatively, along rows and columns (see [1]). So a 2x2 block of pixels (Figure 1) is quantized into one table index in two steps. First, two row-wise lookups quantize the row-wise pixel values to give two table indices, Y0 and Y1, followed by a column-wise second-level table lookup, which quantizes the column-wise indices into one index, Z0, which represents a 2x2 code-book vector.

FIGURE 1. Two steps of HVQ lookup followed by 1 step of FSHVQ table lookup to give the index of the optical flow vector



Horn and Schunck’s optical flow method ([9]) proceeds by applying spatial and temporal derivative masks, e.g. of size 2x2, at corresponding spatial locations of the current and previous images. So if the quantized codebook indices of the previous frame (at the 2x2 block quantization level) are stored, along with the motion field averages u_{av} and v_{av} as the *state* (see [7] for a description of state in the FSHVQ method), then the optical flow vector at time t and at location x,y can be conveniently expressed as a function of the state at time t, s_t, and the previous and current images, i_{t-1} and i_t in equation 1.

$$(u, v)_t = f(s_t, i_{t-1}, i_t) \qquad \textbf{(EQ 1)}$$

This simply means that feeding the (2x2 level) code-book indices $Z0_t$ and $Z0_{t-1}$ into H_s, the FSHVQ table for state s, we can look up the index of the motion vector, (u,v). The FSHVQ code books and tables are, of course, populated in the training phase, by using the Horn and Schunck formulation ([9]). This embeds the optical flow computation in table-lookups, and gives an extremely fast encoder.

Subsequently, the usual HVQ strategy of table lookups can be continued for the motion vector indices, to build up “super blocks” of motion vectors (see also [6]). The residual errors for each block can be sent using a lower bit-rate HVQ scheme (since the residual errors are usually small and well clustered).

To complete the encoding, the state is updated by storing a matrix of the local motion field vectors (u_{av}, v_{av}), and the output or reconstructed image is stored as the “previous image”, i_t for the next frame at t+1, as shown in equation 2.

$$i_t \leftarrow \hat{i}_t \qquad \textbf{(EQ 2)}$$

This allows the encoder to track the decoder, and prevent error accumulation - it represents the “closed loop” of the generic FSHVQ system ([7]), as the previous image, i_t, and the local motion field averages, (u_{av},v_{av}), together comprise the state of this encoder.

A further departure from the single-frame FSHVQ method ([7]) is that overlapping blocks are coded, up until the motion vector lookup stage from the FSHVQ table H_s (Figure 1), using “sliding windows”. This means that there will be no blocking artifacts in the motion compensation.

3. EXTENSIONS AND OPTIMIZATIONS

3.1 Static Regions

The biggest performance per cost optimization is also perhaps the simplest - send compact codes or escape sequences for static regions (see also [4]). Typically, in video teleconferencing applications a large percentage of the image is static, and we can use this fact, by applying suitable thresholds.

3.2 Rate-distortion optimization

Many optimizations are possible during codebook design, in the training phase. In our view, one important optimization is to pick those codebook vectors which give the smallest increase in overall distortion, in coding at a lower rate. To do this we need to minimize the true R/D cost associated with the motion codebook. Here, R represents the total number of bits needed for compression, which includes both the rate for encoding the motion vectors and the rate spent on quantizing (using lower bit rate HVQ) the prediction error. The optimization problem can be formulated as an unconstrained minimization of the Lagrangian, in equation 3.

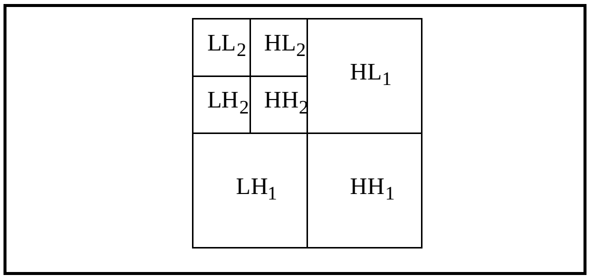
$$E = D + \lambda R \qquad \textbf{(EQ 3)}$$

This translates into modifying the basic LBG (Linde-Buzo-Gray) training algorithm to simultaneously pick those motion and spatial vectors which will give the smallest reconstruction error for each book.

3.3 Multiscale Extension of the Optical Algorithm

To keep the discussion simple, we used pixel values in the algorithm description in Section 2. In actual practice, we use short wavelet filters, e.g. Haar or Daubechies 4-tap filters to vector-quantize the wavelet coefficients, instead of grey levels (see also [1],[2],[8]), to improve the quality of the compressed images.

FIGURE 2. Two scale wavelet decomposition. Optical flow can be computed on the band LL_2 to handle larger displacements



Thus, it would be simple to apply Horn and Schunck's optical flow method on a low-pass version of the image (i.e. smooth coefficients, see Figure 2) in order to handle large displacements or motions. Then the decoder would perform the motion compensation or interpolation in a coarse-to-fine manner. Inter-level smoothness constraints could also be used (see [2]).

4. CODING RESULTS

We show some experimental results of coding the "Claire" news caster sequence at 100 KB/s (sustained) after the initial frame was sent (intracoded) in Figure 3. The optical flow technique gave an average PSNR of 36.0dB, whereas BMA gave 34.8dB. Also our algorithm performed much better at the edges, whereas BMA gave severe blocking artifacts.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a finite state table-lookup HVQ with embedded motion estimation. The encoder is implemented using only table lookups, and is thus practical for real time low bit rate systems. We have used finite states to encode the local motion field vectors, embedding the Horn and Schunck optical flow algorithm in a table-lookup. We have also shown how to extend this algorithm to compute larger displacements, by using the optical flow FSHVQ lookup on smooth (low-pass) coefficients.

The most important contributions of this work are: 1) it combines an accurate method like optical flow with a fast method like hierarchical table-lookup vector quantization to achieve speed and performance, all computationally expensive operations being incorporated into the off-line training phase; and 2) it uses the concept of finite states to encode information about the previous frames in order to compute and utilize motion compensation.

As further work we are interested in using the EZW scheme to encode the motion vectors and residual errors instead of HVQ. This might allow better control and fine tuning of thresholds. In our work we found that thresholding had quite large impact on the R/D performance (see also [4]). Furthermore, as many of the motion vectors and residual errors are close to zero, EZW may represent the positional information (significance map) more economically.

6. REFERENCES

[1] Vishwanath, M. and Chou, P., "An Efficient Algorithm for Hierarchical Compression of Video", *Proceedings of the International Conference on Image Processing* - 1994, Austin, Texas, vol. 111, pp. 275-279, November 1994.

[2] Moulin, P., Krishnamurthy, R., and Woods, J.W., "Multiscale Modeling and Estimation of Motion Fields for Video Coding", *IEEE Transactions on Image Processing*, Vol. 6, No. 12, pp. 1606-1620 December 1997.

[3] Chaddha, N., Perlmutter, K., and Gray, R.M., "Joint Image Classification and Compression using Hierarchical Table-Lookup Vector Quantization", *Data Compression Conference*, pp. 23-32, 1996.

[4] Diab, Z., and Cohen, P., "Motion Compensated Video Compression using Adaptive Transforms", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Part 4, pp. 2881-2884, Munich, Germany, 1997.

[5] Lin, S. and Shi, Y.Q., "An Optical Flow Based Motion Compensation Algorithm For Very Low Bit-Rate Video Coding", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, V4, pp. 2869-2872, 1997.

[6] Regunathan, S.L. and Rose, K., "Motion Vector Quantization in a Rate-Distortion Framework", *Proceedings of IEEE International Conference on Image Processing*, part 2, pp.21-24, 1997.

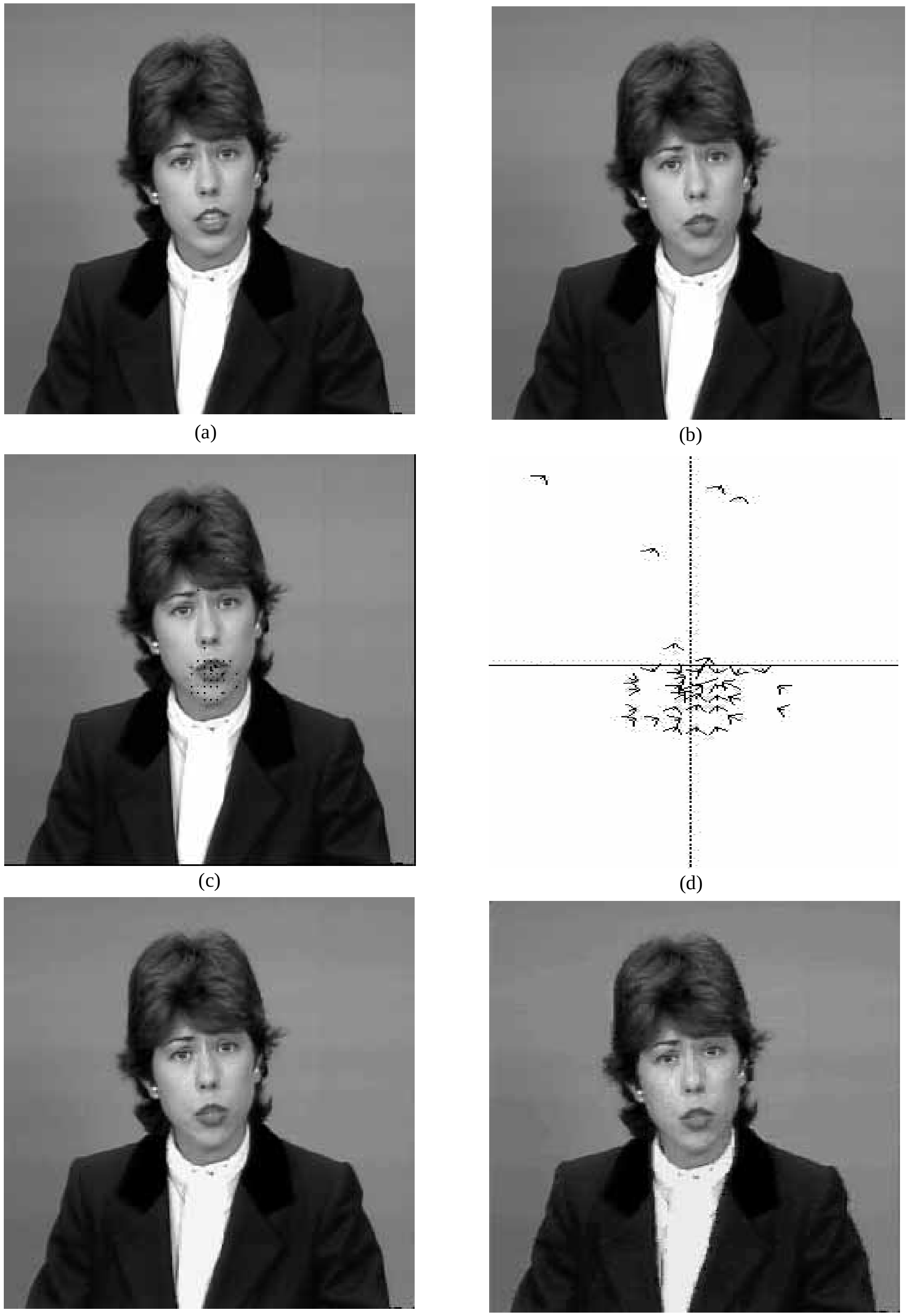
[7] Chaddha, N., Mehrotra, S., and Gray, R.M, "Finite State Hierarchical Table-Lookup Vector Quantization for Images", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, part 4, pp.2024-2027, 1996.

[8] Podilchuk,C.I. and Safranek,R.J. "Image and video compression: A review", *International Journal of High Speed Electronics and Systems*, Vol. 8, No.1, pp. 119-177, 1997.

[9] Ballard,D.H. and Brown,C.M., *Computer Vision*, Prentice Hall, 1982.

[10] Chaddha, N. and Vishwanath, M., "A Low Power Video Encoder with Power, Memory and Bandwidth Scalability", *9th International Conference on VLSI Design*, 358-363, 1996.

FIGURE 3. Results from the “Claire” news caster sequence a,b-Original frames; c,d-Optical flow vectors computed (d is not on the same scale as the images); e-FSHVQ reconstruction; f-BMA reconstruction



(a)

(b)

(c)

(d)

(e)

(f)