MATCHING PURSUIT FOR COMPRESSION AND APPLICATION TO MOTION COMPENSATED VIDEO CODING

Martin Vetterli and Ton Kalker[1]

Department of EECS, UC Berkeley, CA 94720,
martin@eecs.berkeley.edu, kalker@eecs.berkeley.edu

ABSTRACT

In many hybrid video compression algorithms there is a clear distinction between motion estimation and compensation (MEMC) on the one hand, and transform coding of the residue on the other hand. By translating MEMC into the framework of Matching Pursuit (MP), we will show that unification of both steps is possible. Moreover, adapting the MP approach to the small size dictionaries involved in MEMC, it is shown that unification may in principle lead to a better performance of hybrid coding schemes. We close with a discussion on the practical obstacles of the technique, and how to avoid them.

1. INTRODUCTION

A technique used in statistics and time-frequency analysis called **matching pursuit** is proposed for signal compression. It is a successive approximation technique with a redundant dictionary of prototype waveforms. We derive an orthogonalized version that is well suited for compression applications, and propose a greedy algorithm that searches for an approximation in a rate-distortion sense. We discuss dictionaries that are optimized in a rate-distortion sense, and indicate differences and generalizations with respect to the usual matching pursuit algorithms.

We then indicate that traditional block based motion compensated video coding is a particular case of matching pursuit, with a dictionary based on past frames. The matching pursuit framework allows analysis of current algorithms, as well extensions thereof. In particular, we discuss the inclusion of illumination changes, scaling and subpixel accuracy using our framework. Moreover, we show that due to the small size of the dictionaries involved, the recursive nature of the general MP algorithm can be replaced by the direct computation of the orthogonal projection on the span of the dictionaries involved.

[1]The author is an employee of Philips Research, currently a visiting scholar at UC Berkeley.

2. MATCHING PURSUIT AND ITS ORTHOGONALIZED VERSION

We will concentrate on the finite dimensional case, that is, consider vectors from the Hilbert space $H = R^N$. The inner product of $x, y \in H$, defined as $\langle x, y \rangle = \sum_n x[n] \cdot y[n]$, defines a norm $\|x\| = \langle x, x \rangle^{1/2}$. Choose a dictionary $D = \{\phi_k\}$ in H, with $\|\phi_k\| = 1$ and $\text{span}(D) = H$. Typically, $|D| = M$ is larger than N, and thus $\{\phi_k\}$ is a redundant or over complete set of vectors in H. Matching pursuit [2] is a greedy algorithm to match an input signal $f \in H$ by a linear combination of ϕ_k's. Start by searching for ϕ_{k_0} such that

$$|\langle \phi_{k_0}, f \rangle| \geq \alpha \, |\langle \phi_i, f \rangle|, \quad i \neq k_0, \quad 0 < \alpha \leq 1 \qquad (1)$$

(α is typically close or equal to 1) and write f as its projection onto ϕ_{k_0} and a residue $R_1 f$,

$$f = \langle \phi_{k_0}, f \rangle \, \phi_{k_0} + R_1 f. \qquad (2)$$

The algorithm is then iterated on $R_1 f$ and so on, until some convergence criterion $\|R_n f\| < \epsilon \|f\|$, where $\epsilon > 0$, is met. We can thus write (calling $f = R_0 f$)

$$f = \sum_{i=0}^{n-1} \langle \phi_{k_i}, R_i f \rangle \, \phi_{k_i} + R_n f. \qquad (3)$$

While $R_n f \perp \phi_{k_{n-1}}$, the residue is in general not orthogonal to the other vectors $\phi_{k_i}, i = 0 \ldots n-2$. Thus, even in finite dimensional spaces, the matching pursuit will usually converge slowly, while an orthogonalized version will converge in N steps [?]. We derived an orthogonal matching pursuit based on Gram-Schmitt orthogonalization that simply keeps an orthogonal set of best matches. The idea is to successively project the remaining vectors onto the orthogonal complement with respect to the current (orthogonal) set of chosen vectors. The resulting orthogonalized set is called $\{\psi_i\}$.

3. MATCHING PURSUIT FOR COMPRESSION

In the context of compression, we have to consider both the approximation quality (including quantization of

0-8186-6950-0/94 $4.00 © 1994 IEEE

the inner products involved in the expansion) and the rate associated to selecting a particular vector and the related quantized inner product. Designing a good dictionary in this rate-distortion sense is a difficult problem, since it amounts to a vector quantization (VQ) codebook design problem. Actually, it can be seen that matching pursuit compression is a cascade VQ scheme, where the VQ is of the gain-shape type [1]. However, the size of vectors we consider is usually much larger than what is used in VQ (e.g. 64 in the example considered below).

Assume we have a reasonable dictionary (to be discussed below) and rate measures $r(k_i)$ and $r(Q[\langle \phi_{k_i}, f \rangle])$. In the above, $Q[\ .\ ]$ is an appropriate (scalar) quantization function and $r(\ .\ )$ is the rate or number of bits used to represent the discrete variable (typically, of the order of $\log_2 L$ where L is the number of elements, but note that entropy coding is assumed). Then we can use the following greedy algorithm. At step n, we have the current approximation error $R_{n-1}f$ and are searching for the next best matching vector $\psi_{k_{n-1}}$. Call

$$r_\Delta(\psi_i) = r(i) + r(\ Q[\langle \psi_i, R_{n-1}f \rangle]\ )\ , \qquad (4)$$

the increase in rate if the orthogonalized vector ψ_i is picked. In the quantized case, the new approximation error, given that ψ_i is used, amounts to

$$R_n f(\psi_i) = R_{n-1}f - Q[\ \langle \psi_i, R_{n-1}f \rangle\ ]\psi_i\ . \qquad (5)$$

Note that $R_n f$ is not orthogonal to $Q[\ \langle \psi_i, R_{n-1}f \rangle\ ]\psi_i$ in general because of the quantization. Call

$$\begin{aligned} d_\Delta(\psi_i) &= \\ &= \|R_{n-1}f\|^2 - \|R_n f(\psi_i)\|^2 \\ &\simeq |Q[\ \langle \psi_i, R_{n-1}f \rangle\ ]|^2, \end{aligned}$$

the decrease in distortion if ψ_i is chosen. Then, pick the best vector ψ_{k_n} such that

$$\frac{d_\Delta(\psi_{k_n})}{r_\Delta(\psi_{k_n})} \geq \beta \frac{d_\Delta(\psi_i)}{r_\Delta(\psi_i)}, \quad i \neq k_n,\ 0 < \beta \leq 1. \qquad (6)$$

This gives a rate-distortion optimized matching pursuit. We indicate a few generalizations that can be helpful.

First, one can change the dictionary as the approximation progresses. This is typically done in cascade VQ. Thus, we have now dictionaries D_n, where possibly D_n depends on previous choices $\psi_l, l < n$.

Second, we have only considered fitting by subspaces of dimension 1 so far. However, especially in the compression case, it can be of interest to fit larger subspaces at once (to be more precise, since we quantize the coefficients of the expansion, we are not generating a subspace, but only a discrete set of points on the subspace corresponding to the basis vectors). A typical example is approximation by transform coding (e.g. DCT), which is actually a full space fitting but with coarse quantization. This means that we can consider, instead of ϕ_i and ψ_i, subspaces V_i and W_i (where W_i is an orthogonalized version of V_i). These subspaces have both an approximation (which includes quantization) and a rate attached to them, and thus can be used in the above algorithm.

4. MOTION COMPENSATED VIDEO CODING AS MATCHING PURSUIT

As hinted earlier, the difficulty with matching pursuit for compression is the design of a suitable dictionary. There is however a very important case where such a dictionary is available, namely motion compensated video coding. While at first we will simply rephrase classical motion estimation/compensation as matching pursuit, it will become apparent that a number of generalizations and possible improvements become easy.

Let us briefly recall the classic block motion estimation/compensation followed by DCT compression. Given a block of size N by N in the current frame, find a block in the past frame (or the past reconstructed frame) that best matches the current block and take the difference of the two, to obtain a prediction error. This prediction error is then coded using transform coding. Let us rephrase this in terms of matching pursuits. A video sequence is a sequence of frames $I(l, m, n)$ (l, m and n denote horizontal, vertical and time dimensions, respectively). A block of N by N pixels from frame n, with upper left corner at (i, j), is denoted by $B_N(i, j, n) = [I(i, j, n) \ldots I(i + N - 1, j + N - 1, n)]$. The signal we want to code is a current block in frame n, or $f_{kl} = B_N(kN, lN, n)$. In classic motion estimation/compensation, the dictionary for the first search is made up of the set $\{B_N(kN - i, lN - j, n - 1)\}, i, j \in [-m \ldots m]$. Then a motion vector (i_m, j_m) is obtained by minimizing $\|f_{kl} - B_N(kN - i, lN - j, n-1)\|$ over i and j. This leads to the prediction error signal $e_{kl} = f_{kl} - B_N(kN - i_m, lN - j_m, n-1)$. This prediction error is expressed in the DCT basis and quantized.

Instead, we create a dictionary of normalized past blocks, or $\phi_{ij} = B_N(kN - i, lN - j, n-1)/\|B_N(kN - i, lN - j, n-1)\|$, and add the DCT basis $\{d_1, d_2, \ldots d_{N^2}\}$ together with quantization as an "approximation" subspace V_{dct}. That is, our dictionary is $D = \{\phi_{-m,-m}, \ldots \phi_{m,m}, V_{dct}\}$ and we have appropriate quantization of the inner products. We can start a matching pursuit. Typically, a past block will give a best match, thus

$$f_{kl} = \langle \phi_{i_0,j_0}, f \rangle \phi_{i_0,j_0} + R_1 f. \qquad (7)$$

The motion vector is usually different from the solution obtained in classic block motion estimation. But

$$\|R_1 f\| \leq \|f_{kl} - B_N(kN - i_m, lN - j_m, n-1)\|, \qquad (8)$$

that is, the prediction error is in general reduced in a matching pursuit approach. However, we have to

transmit the value of the inner product $\langle \phi_{i_0,j_0}, f \rangle$. Its value can be well predicted from the past block since:

$$\langle \phi_{i_0,j_0}, f \rangle = \alpha \cdot \cos\beta \cdot \| B_N(kN - i_0, lN - j_0, n-1) \|, \quad (9)$$

where α represents illumination change, and β the angle between $B_N(kN-i_0, lN-j_0, n-1)$ and $B_N(kN, lN, n)$. Note that both α and $\cos\beta$ are close to 1.

5. MATCHING PURSUIT AND PROJECTIONS

The Matching Pursuit framework allowed us to translate the notion of motion estimation and compensation (MEMC) into the terminology of vector spaces. Whereas in general matching pursuit one has to find approximations by a recursive procedure, this is not necessarily so in the case of MEMC. In general matching pursuit, the recursive procedure of Eq. 3 is forced by the computational infeasibility of finding in one step the best linear combination of dictionary elements approximating the target vector. For the same reason, subspace fitting, as outlined in the previous section, is not a feasible option.

In the MEMC case however, the size of the (local) dictionaries can be very limited. In a number of relevant cases the dictionaries are not even complete. In such cases, instead of relying on the recursive procedure of Eq. 3, we can actually compute the *projection* of the target vector on the span of the dictionary. In the following sections we will consider 3 scenarios, differing from each other in the structure of the local dictionaries and the approximation method.

1. $(2m+1)^2$ previous blocks and MP, or
2. n previous blocks and projection, or
3. n previous blocks, l DCT basis functions and projection.

The first scenario corresponds to pixel accuracy MEMC with a search range of m pixels horizontally and vertically, the second scenario corresponds to subpixel refinement MEMC, and the third to combined subpixel refinement MEMC and DCT coding of the residue.

5.1. Scenario 1: Pixel Accuracy MEMC

This section describes scenario 1 in more detail. For each block in the current frame a collection of $(2m+1)^2$ blocks in the previous (or future frame, depending on the direction) frame serves as the local dictionary.

In vector space terminology, searching for a pixel accuracy vector field in the MP sense, amounts to projecting onto 1 dimensional subspaces. The block B_p that matches best in the MP sense satisfies that a scaled version αB_p is closest to the current block B_c for all possible scale factors and other blocks in the local dictionary. Note that block matching MC using an MP motion field performs suboptimal. The better performance of MPME only becomes apparent when scaled MC is performed. A typical situation is presented in Fig. 5.1. The figure shows that spending a few bits on a scale factor improves the prediction quality, and the MP field performs better than the BM field, though the difference can be very slight, depending on the image sequence.

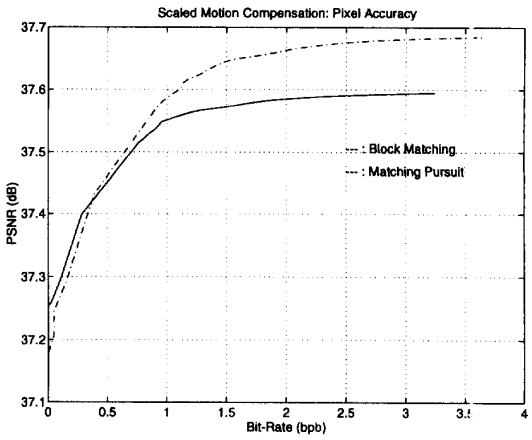

Figure 1: Scaled motion compensation: Block Matching vs Matching Pursuit: solid = block matching, dashed = matching pursuit.

5.2. Scenario 2: subpixel accuracy MEMC

This approach of scaling can be extended to subpixel accuracy vector fields. The usual approach to subpixel motion estimation consists of two steps: in the first step the previous frame is interpolated horizontally and vertically by a factor 2 using a fixed set of filter coefficients. In the second step a current block is matched against interpolated blocks in the previous frame. In terms of vector spaces this means that the current block is approximated by a linear combination of blocks in the previous frame, where the coefficients are chosen from a small fixed set. For reasons of computational complexity, the procedure for finding the subpixel accuracy motion field is divided in two stages: first finding a pixel accuracy approximation, followed by a subpixel refinement.

We can mimic this approach in the matching pursuit context. First computing a pixel accuracy motion field, we choose a local dictionary $\mathcal{D} = \{\psi_i\}$ which consist of 9 blocks in the previous frame. These are the blocks less than 1 pixel away from the central block pointed to by the pixel accuracy vector field. As the size of the dictionary is very small, we apply projection to find the best approximation in the span of the dictionary.

The computation of the projection takes as basic ingredients (1) the 9x9 matrix M of inner products $\langle \psi_i, \psi_j \rangle$, and (2) the column vector $P = \langle \psi_i, B_c \rangle$ of inner products of B_c with the functions ψ_i. Using the technique of singular value decomposition we decompose M as $M = F^t F$ such that $D = F F^t$ is a non-singular, positive definite, diagonal matrix. Defining E as $E = D^{-1} F$, the coefficients C defining the orthogonal projection $\hat{B}_c$ are given by $C = E^t E P$. Defining Y as $Y = EC$, the distance $d^2(B_c, \hat{B}_c)$ between B_c and $\hat{B}_c$ is given by $\langle B_c, B_c \rangle - \langle Y, Y \rangle$. The coordinate transform E has the nice property that for any block $\tilde{B}$ in the span of $\mathcal{D}$, specified by coordinates $\tilde{C} = E^t \tilde{Y}$, the distance $d(B_c, \tilde{B})$ between B_c and $\tilde{B}$ satisfies

$$d^2(B_c, \tilde{B}) = d^2(B, \hat{B}_c) + d^2(Y, \tilde{Y}).$$

In particular this means that the preferred domain of quantization is not the domain of coefficients C (with respect to $\mathcal{D}$), but the domain of Y coordinates (or any orthogonal transform there of). The error introduced by the quantization $Q(Y)$ of Y is the same as the overall distortion after reconstructing.

We tested these ideas on the MPEG-4 test sequence MOTHER. In the experiments performed, we took consecutive frames from this sequence, and computed pixel and subpixel accuracy motion fields. The pixel accuracy motion field was used to determine the local dictionaries. The subpixel motion field was used to determine an initial guess B_i. The coordinates of the difference between $\hat{B}_c$ and B_i were uniformly quantized over a large range of quantizer step sizes. Performance of coding was measured by computing the first order entropy of quantized coefficients. Distortion was measured in dB (PSNR).

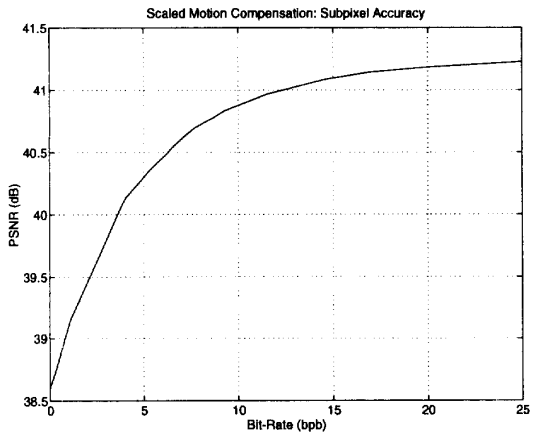

Figure 2: Scaled subpixel accuracy motion compensation.

A typical curve is given in Fig. ??. In general we found that the initial part of the curve is very steep. One easily gains 1 DB in PSNR by spending less than 3 bits per block.

5.3. Scenario 3: Combined MEMC and residue coding

In the previous section the local dictionaries consisted only of motion blocks. In this section we enlarge the dictionary with a small number of DCT basis functions. The reason for not including a large number of DCT basis functions lies in the fact that projections on high dimensional spaces are computational very expensive. In a typical example, as used in our experiments, a local dictionary will consist of 9 motion blocks and 6 low frequency DCT basis functions.

The purpose of this section is to compare the projection method (with small, mixed local dictionaries) with the classical method of separate MC and DCT coding of the residue. In the experiments performed, we took consecutive frames from the video sequence MOTHER, and computed pixel and subpixel accuracy motion fields. Performance of coding was measured by computing the first order entropy of quantized coefficients. Distortion was measured in dB (PSNR). When comparing bit-rates, the rates associated with the vector fields were not taken into consideration, as these are equal for both methods.

The computation of the projection onto the span of the dictionary proceeds in the same manner as in the previous section. There are two minor differences: (1) the size of the inner product matrix M and the column vector P are greater and (2) the inner product matrix has some structure due to the fact that the DCT basis functions are orthonormal. To be precise, the inner product matrix has the form

$$\begin{pmatrix} M & T^t \\ T & Id \end{pmatrix},$$

where the matrix T contains the relevant DCT coefficients.

Following the same procedure as the previous section, again using the subpixel motion field for an initial guess we found the following results (see also Fig. 5.3):

1. Performing residue coding using only the first 15 DCT basis functions, and performing no quantization of coefficients in either of the two methods, the projection method performs considerably better than DCT coding.

2. Any good initial prediction of the current block can be incorporated in the projection method scheme. We simply apply the transformation E to this initial guess and use the difference with the projection block as the data to be quantized. In order to perform at least as well as DCT coding of the residue, subpixel prediction seems to be the best initial guess.

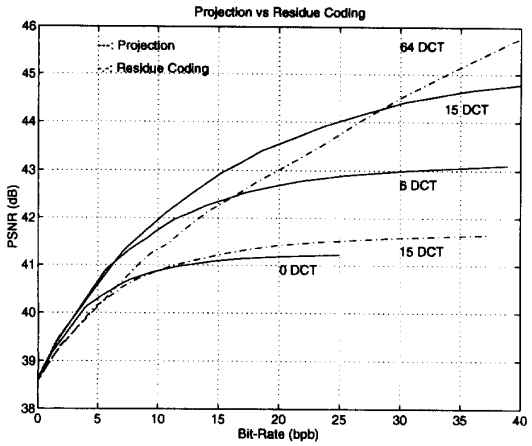

Figure 3: Distortion-Rate curves for projection and residue coding: solid = projection, dashed = residue.

3. When using an equal number of basis functions in both methods, i.e. keeping only 15 DCT components in residue coding, and quantizing the appropriate coefficients, we find that the projection methods performs better over all possible bit-rates.

4. Increasing the number of coefficients kept in residue coding up to all 64, there exists a threshold in bit-rate beyond which residue coding performs better than projection coding (with only 15 basis functions). For bit-rates below 15 bits per block, projection performs better. The improvement over residue coding can be as large as 1 dB.

Taking the subpixel prediction as an initial guess, and working in E-coordinates, leaves a residue of which the components are uncorrelated and approximately Gaussian. Numerically computing the distortion-rate curve D(R), we find that rate R_r and distortion D_r for component r of the residue are to a very good approximation related by $R_r = \frac{1}{2}\log(\frac{\sigma_r}{D_r})$, where σ_r is the variance of the r^{th} component.

This observation allows us (at least in principle) to find the optimal coding scheme for a given bit-rate budget. In the optimal coding scheme all the Y-coordinate coders are operating at points of the same slope on their respective $R(D)$ curves.

6. CONCLUSIONS

We have shown that the MP framework allowed us to translate MEMC techniques into the framework of orthogonal projections on low ($\leq$ 20) dimensional subspaces. We have also shown that in a clean context, i.e. prediction of frames from non coded, original frames, the projection method with a small dictionary (9 ME / 6 DCT) gives a better SNR than separate motion compensation and full blown DCT coding of the residue. There are of course many remaining issues, three of which we shall address now.

Firstly, we have not yet applied the techniques as described above in a *real* environment, i.e. prediction of frames from coded frames. The real value of the proposed techniques will only become apparent when we have done experimental verification in a real environment.

Secondly, there is a considerable computational requirement on the decoder receiving the data $\mathcal{Q}(Y)$. It cannot immediately reconstruct the image, but it needs first to recompute the matrix E. Assuming that no errors occurred in transmitting previous data, this is in principle possible as E only depends on inner products of previous data. The decoder needs to perform the following actions:

- Using the motion field, the appropriate 9 motion blocks are retrieved.
- All the required inner products are computed.
- The singular value decomposition is derived.
- The matrix E is computed.

An obvious question is whether or not we can do without the computation of the matrix E. One possibility is finding a fixed matrix E_{fixed} which is a good average over all statistically relevant matrices E (as the DCT is a good approximation of the optimal decorrelating matrix). Another possibility is the use of a small set of fixed matrices $\mathcal{E}$.

Thirdly and lastly, in our experiments we have selected a particular set of DCT basis functions as an addition to the motion dictionary. For which bit-rate which set of DCT functions (or any other set of orthogonal basis functions) is optimal, is still an open question.

7. REFERENCES

[1] A.Gersho and R.M.Gray, Vector Quantization and Signal Compression, Kluwer Academic Pub., Boston, 1992.

[2] S.Mallat and Z.Zhang, "Matching pursuits with time-frequency dictionaries," IEEE Trans. on SP, Vol. 41, No. 12, pp.3397-3415, Dec. 1993.

[3] K.M.Uz, M.Vetterli and D.LeGall, "Interpolative multiresolution coding of advanced television with compatible subchannels," IEEE Trans. on CAS for Video Technology, Vol.1, No.1, pp.86-99, March 1991.