Czerepinski, PJ., Davies, C., Canagarajah, CN., & Bull, DR. (2000). Matching pursuits video coding: dictionaries and fast implementation. *IEEE Transactions on Circuits and Systems for Video Technology*, *10*(7), 1103 - 1115. [7]. https://doi.org/10.1109/76.875515

Matching Pursuits Video Coding: Dictionaries and Fast Implementation

Przemysław Czerepiński, *Member, IEEE*, Colin Davies, *Member, IEEE*, Nishan Canagarajah, *Member, IEEE*, and David Bull, *Member, IEEE*

***Abstract*—Matching pursuits over a basis of separable Gabor functions has been demonstrated to outperform DCT methods for displaced frame difference coding for video compression. Unfortunately, apart from very low bit-rate applications, the algorithm involves an extremely high computational load. This paper contains original contribution to the issues of dictionary selection and fast implementation for matching pursuits video coding. First, it is shown that the PSNR performance of existing matching pursuits codecs can be improved and the implementation cost reduced by a better selection of dictionary functions. Secondly, dictionary factorization is put forward to further reduce implementation costs. A reduction of the computational load by a factor of 20 is achieved compared to implementations reported to date. For a majority of test conditions, this reduction is supplemented by an improvement in reconstruction quality. Finally, a pruned full-search algorithm is introduced, which offers significant quality gains compared to the better-known heuristic fast-search algorithm, while keeping the computational cost low.**

***Index Terms*—Displaced frame difference, low-complexity algorithm, matching pursuit, pruned full search, video coding.**

I. Introduction

DECOMPOSING a signal into a linear combination of basis vectors is a common problem in information processing. A matching pursuit [1] is a greedy algorithm that addresses the problem of signal decomposition over an overcomplete basis set. Matching pursuits can have advantages for compression, as the basis can be arbitrarily large and contain functions selected to closely match the structures comprising the coded signal. This contrasts with transform or subband coding, where the basis is complete and constrained by orthogonality or perfect reconstruction conditions. Then, signal structures which are not present in the basis must be represented by a linear combination of a larger number of basis functions, which spreads signal energy and detriments compression. However, the design of an optimal overcomplete basis in the rate-distortion sense is an np-.

Several matching pursuits algorithms for image [2]–[4] and video coding [5]–[8] were reported in recent years. This work pertains to the video codec of Neff and Zakhor [7], which employs matching pursuits to decompose the *displaced frame difference* (DFD) signal over a basis of Gabor functions. This codec was reported consistently to outperform the MPEG4 Verification Model [9]–[11], which decomposes the DFD signal over the *discrete cosine transform* (DCT) basis.

The drawback of matching pursuits DFD coding is the fact that, apart from very low bit rates, it necessitates a very high computational cost compared to other methods. The DFD signal is decomposed by correlating it with the basis functions. As a consequence of employing a redundant basis, the number of correlations is increased. In addition, a matching pursuits decomposition requires multiple passes over data as, in general, it is not possible to find a compact linear combination of basis functions approximating the coded signal in a single pass.

This paper describes original contribution to the issues of dictionary selection and fast implementation of matching pursuits DFD coding, and draws on work partially presented in the companion papers [12], [13]. Firstly, it is postulated that a basis for coding DFD signals should include functions well localized in space. Example bases which contain such functions are verified to offer a better trade-off between performance and implementation cost, compared to the basis reported by Neff and Zakhor [7], for a broad selection of video sequences. Further, the new bases are constructed in such a way that a basis vector can be factorized into another basis vector and an auxiliary low-complexity impulse response. Thus, the basis is constructed through a succession of short-kernel convolutions, which enables reusing previous filtering results within the matching pursuits framework and leads to a considerable reduction of the computational cost. Finally, it is shown that the new bases have advantages for a full-search matching pursuit and overcome the traditional problems of the known full-search algorithms: large storage requirements and an extremely high computational cost.

This paper is structured as follows. Section II reviews the developments in matching pursuits video coding. Section III presents alternative bases, which offer an improved PSNR performance at a reduced computational cost, compared to the basis reported by other authors. Section IV introduces bases obtained by cascaded short-kernel convolutions for low-complexity matching pursuits. Section V describes the original pruned full-search algorithm. Section VI compares the performances of the proposed bases. Conclusions are drawn in Section VII.

II. Matching Pursuits Video Coding

A. Background Theory

This section provides an informative introduction to the principles of matching pursuits decomposition. Please refer to [1] for a comprehensive treatment of the subject.

Manuscript received July 1999; revised April 2000. This work was supported by the EPSRC Grant GR-L43596. This paper was recommended by Guest Editor Y.-Q. Zhang.

P. Czerepiński, N. Canagarajah, and D. Bull are with the Centre for Communications Research, University of Bristol, Merchant Venturers Building, Bristol BS8 1UB, U.K. (e-mail: p.j.czerepinski@ieee.org; nishan.canagarajah@bristol.ac.uk; dave.bull@bristol.ac.uk).

C. Davies is with NDS UK, Chilworth Research Centre, Gamma House, Chilworth S016 7NS, U.K. (e-mail: cdavies@ndsuk.com).

Publisher Item Identifier S 1051-8215(00)08201-X.

Let H be the Hilbert space $\mathrm{H} = R^N$. The inner product of vectors $f, g \in \mathrm{H}$ is defined as $\langle f, g\rangle = \sum_n f[n]g[n]$ and the norm of a vector f is $\|f\| = \langle f, f\rangle^{1/2}$. Let $D = \{g_k\}$, where $\|g_k\| = 1$, $\mathrm{span}(D) = \mathrm{H}$ and $\mathrm{card}(D) \gg N$ be an overcomplete basis for H. A matching pursuit strives to compute a linear expansion of a vector $f \in \mathrm{H}$ over D by orthogonally projecting it onto basis vectors. The basis vector g_{k_0}, which maximizes $|\langle f, g_k\rangle|$ is selected as the first approximation of f

$$f = \langle f, g_{k_0}\rangle g_{k_0} + R_1 f. \tag{1}$$

Now, the residual $R_1 f$ is projected onto basis vectors. This process is repeated until the step m, which satisfies some convergence criterion. The vector f can now be expanded as

$$f = \sum_{n=0}^{m-1} \langle R_n f, g_{k_n}\rangle g_{k_n} + R_m f \tag{2}$$

where $R_0 f = f$. The basis vectors g_{k_n} in (2) are referred to as *atoms*. In contrast to the decomposition, signal expansion is a low-complexity operation and involves a linear combination of selected atoms; the ordering of atoms is not significant. The knowledge of the sequence $\{\langle R_n f, g_{k_n}\rangle, k_n\}$, $n = 0 \cdots m-1$, referred to as the *structure book*, is required to recover an approximation of the vector f with the error $R_m f$.

B. Matching Pursuits for DFD Coding

1) Description of the Algorithm: The codec proposed by Neff and Zakhor [7] employs a matching pursuit to expand the DFD signal over a basis of separable Gabor functions, defined as

$$g_{\vec{\alpha}\vec{\beta}}(i, j) = g_{\vec{\alpha}}(i) g_{\vec{\beta}}(j) \tag{3}$$

where $g_{\vec{\alpha}}(i)$ and $g_{\vec{\beta}}(j)$ are one dimensional functions, specified by the set of parameters $\vec{\alpha} = \{s, \xi, \phi, N\}$: the scale s, the frequency ξ, the phase ϕ and the domain size N

$$g_{\vec{\alpha}}(i) = K_{\vec{\alpha}} g\left(\frac{i - \frac{N-1}{2}}{s}\right) \cos\left(\frac{2\pi\xi\left(i - \frac{N-1}{2}\right)}{16} + \phi\right) \tag{4}$$

where $i = 0, 1, \ldots, N-1$, $K_{\vec{\alpha}}$ is a normalization factor, and $g(.)$ is a Gaussian window defined as $g(t) = e^{-\pi t^2}$.

The 2-D form of the dictionary of Neff and Zakhor is shown in Fig. 1(a). The $\{s, \xi, \phi, N\}$ parameters can be found in the reference [7]. This dictionary will be referred to as D0 in the following discussion, and was obtained experimentally by decomposing a training set of DFD images. In this paper, the set D0, as shown in Fig. 1(a), will be referred as a *dictionary*, whereas the *basis* D0 will consist of all integer translations of the functions from the dictionary D0 within the DFD signal.

The matching pursuits codec terminates the decomposition when a pre-selected rate or distortion criterion is satisfied. The complex entropy coding of the structure book hampers a rate-distortion optimization of the decomposition. Nevertheless, the codec of Neff and Zakhor was demonstrated consistently to outperform the DCT-based algorithms such as H.263 or the MPEG4 Verification Model, both subjectively and in the PSNR sense [9]–[11]. The power of the technique depends on the fact that the Gabor dictionary matches the structures present in most DFD signals better than the dictionary consisting of 64 DCT basis functions. The matching pursuit eliminates the blockwise segmentation of the motion residual and avoids the blocking

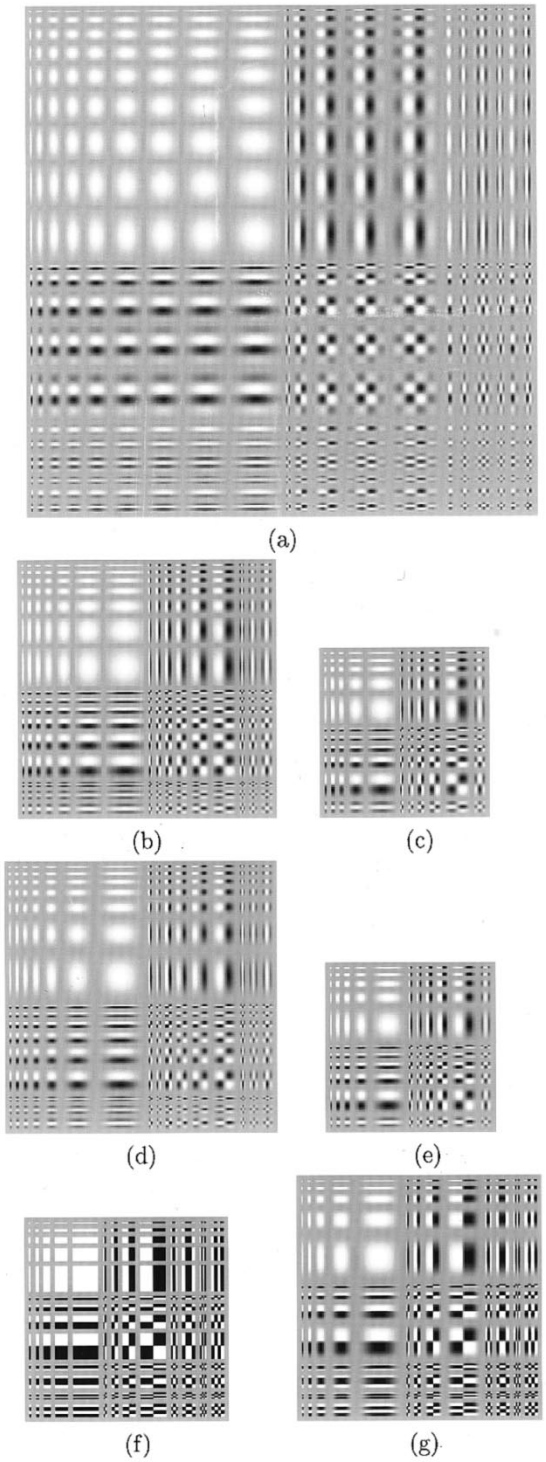


Fig. 1. The 2-D dictionaries. (a) D0 (20 × 20), Neff and Zakhor [7]. (b) D1 (16 × 16). (c) D2 (11 × 11). (d) C1 (16 × 16). (e) C2 (11 × 11). (f), (g) C3 (13 × 13), De Vleeschouwer and Macq [18], [19]. Haar function dictionary used for atom search and its smoothed version used for expansion. Functions are indexed from top left to bottom right, starting from 0.

and ringing artifacts. Moreover, the Gabor dictionary was reported to outperform the DCT dictionary, even if the restriction of the fixed blockwise structure of the DCT decomposition was relaxed [14].

2) Computational Cost and Memory Requirements: Due to the iterative nature of a matching pursuit, the implementation cost is related to the number of atoms stored in the structure book. Hence, with the exception of very low bit rates, the computational cost of the video encoder is extremely high. Truncation of basis functions' domain sizes as well as separability help keep the cost of every iteration at a reasonable level. An atom search can be split into correlating the coded signal with the basis and picking the atom which maximizes the inner product. The cost associated with the correlation stage can be expressed as[1]

$$T_c = (B+1)LS \text{ multiplications} + (B+1)(L-B)S \text{ additions} \tag{5}$$

where

B	number of 1-D functions in the dictionary (20 in the case of D0);
$L = \sum_{i=0}^{B-1} L_i$	total of lengths of the dictionary functions;
S	size of the search area (the whole frame or its subset).

T_c corresponds to the number of operations performed during a single atom search. Neff and Zakhor set $S = 16 \times 16$, which gives rise to the "fast" search algorithm, in contrast to a "full" search where the entire frame is searched. In the case of the fast algorithm, the search area is considerably smaller than the frame size, which reduces the computational cost at the expense of a slightly deteriorated performance. However, before the search can commence, an attempt is made to identify the region of size S to be searched for atoms. This involves finding the energy of overlapping subblocks, with the center of the maximum energy subblock adopted as the origin for atom search. The implementation cost associated with this "energy search" procedure is small and can be ignored.

The cost associated with maximizing the inner product can be estimated as

$$T_m = 2B^2S. \tag{6}$$

T_m stands for the required number of comparisons and ignores the temporary assignments and control flow. The factor of two accounts for the fact that the absolute value maximization is performed over a set of signed numbers. Using relations (5) and (6), the cost of a single atom search with the basis D0 is equal to $6132S$ multiplications, $5712S$ additions, and $800S$ comparisons.

The traditional serial implementation of a matching pursuit individually correlates every basis function with the residual. If the dictionary is separable, then three image buffers are required (at most) to store the correlation results, as illustrated in Fig. 2. Buffer 0 stores the coded signal. The results of filtering the signal in the vertical direction are stored in buffer 1. These are subjected to filtering in the horizontal direction, and the results are stored in buffer 2.

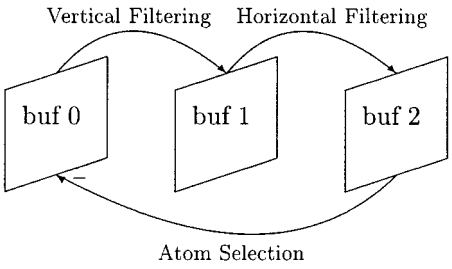


Fig. 2. Memory requirements of a serial implementation of the matching pursuits algorithm [7].

III. Improved Dictionaries for matching Pursuits DFD Coding

This section explains the derivation of new, improved dictionaries for expanding DFD signals. A good dictionary is one which provides a compact representation of the coded signal. Since coding the structure book may involve quantization as well as differential and adaptive entropy techniques, it is difficult to estimate the coding cost, associated with a given structure book, before the decomposition terminates. Therefore, a simplification is adopted in the following discussion and the term "compact representation" refers to a small number n of entries in the structure book. For a fixed n, a more compact representation yields a smaller value of $\|R_n f\|$. In the terminology of Mallat and Zhang [1], this is equivalent to improving the decay rate of the residual $R_n f$, which can be achieved in two ways:

1) decomposing the signal over a larger basis;
2) including in the basis functions that are better correlated to the coded signal.

Inherent in increasing the size of the basis set is an increase of the implementation cost, and therefore the first of the above methods is not attractive. On the other hand, improving the performance by a better selection of dictionary functions (if possible) is an appealing way forward, as it need not imply an increase of the implementation cost.

Since the design of an optimal overcomplete basis is computationally intractable, the derivation of new dictionaries presented in this paper was empirical. First, the new dictionaries were obtained through a progression of tests using a training set of video sequences. Second, the improvements associated with the proposed dictionaries were verified using a broader selection of video sources. The training set consisted of 100 CIF resolution luminance frames from the sequences "Silent Voice," "Foreman," "Table Tennis," and "Mobile and Calendar," which were encoded using a standard motion-compensated architecture. DFDs were decomposed using the fast atom search, with a search area $S = 21^2$. For any sequence, the number of atoms coded per frame was kept constant, and was equal to: 300 for "Silent Voice," 600 for "Foreman," 1000 for "Table Tennis," and 2000 for "Mobile and Calendar." Average atom distribu-

[1]Equation (5) is a simplification of the formula derived in [7] and does not account for the fact that localized filtering in one direction (e.g., vertical) is carried out over a slightly larger area than S, so that the second stage of filtering (horizontal) operates on correct data. For the coding parameters selected in [7], the approximation error is less than 5% and will be further reduced as the search area increases, to disappear completely in the case of a full-frame search.

tions, quoted in the following, refer to an average of normalized distributions obtained separately for every training sequence.

A. Critique of Dictionary D0

It is widely accepted [15]–[17] that the random process underlying DFD signals is nonstationary. Indeed, single DFDs are reminiscent of line drawings, where signal energy is concentrated into narrow, elongated regions along moving edges, due to the inaccuracies of the motion model. Therefore, it is postulated that a good basis for coding DFD signals should include functions, which are well localized in space, to account for the motion-model failure areas. However, as can be observed from Fig. 1(a), dictionary D0 appears instead to put a strong emphasis on coding smooth structures of a relatively large spatial support. For example, consider the subset of zero frequency functions. It begins with function 0, which is the unit impulse with the domain of size $N = 1$, followed by function 1, whose domain size is equal to $N = 5$. The increment between these two domain sizes seems to be too abrupt, which will cause the energy of nonzero mean structures with a spatial support of, say, 2 or 4 to be spread over multiple atoms. Such structures are expected to be present in the coded DFDs. Similarly, the dictionary lacks the high-frequency impulse response [1,−1], and the steepest edge that can be accounted for is [1, 0,−1], by means of function 9.

The dictionary D0 was used to code the training set, and a histogram of dictionary function counts, shown in Fig. 3(a), was collected in the process. As can be observed from the histogram, the contribution of certain functions, such as the pairs 5 and 6 or 12 and 13 to the reconstructed signal is negligible. These are pairs of similar impulse responses, which exemplify the relatively high level of aliasing present in the dictionary. On one hand, aliasing is inherent in an overcomplete basis and enables a compact representation of the coded signal. However, from the point of view of implementation and coding efficiency, the selection of functions in the dictionary D0 does not seem to offer a good balance, and for the training set of sequences, the functions that correspond to the least-frequently selected atoms could be removed without affecting the PSNR performance.

B. The Proposed Dictionary D1

As an alternative to D0, a new dictionary D1 is proposed. This dictionary was derived by gradually modifying the dictionary D0 to overcome the deficiencies described above. The benefits (if any) of every modification were evaluated experimentally using the training set. Only the modifications that improved the overall system performance by either leading to a better reconstruction quality or a reduced computational cost were kept.

The derived dictionary D1 consists of 256 (16×16) functions, and is shown in Figs. 4 and 1(b). All dictionary D1 functions can be described by Gabor parameters, shown in Table I. The following list summarizes the introduced modifications.

1) Removing from D0 functions which corresponded to the least frequently occurring atoms.
2) Introducing the short-kernel even length functions 1, 7, 9 (see Table I).
3) Redesigning the set of zero frequency functions. In short, the progression of function lengths: 1, 2, 3, 5, 9, 17, 25 (functions 0–6 in Table I) was found to be superior to the progression 1, 5, 9, 11, 15, 21, 23, 29, 35 of zero frequency function lengths in the dictionary D0.
4) Redesigning the set of high-frequency functions. Most importantly, the functions 7 and 13, with the frequency $\xi = 8$, were introduced.

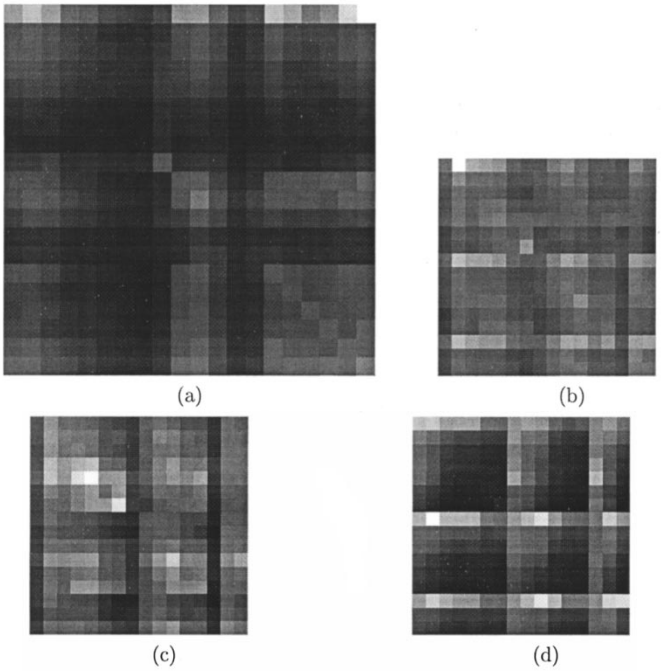


Fig. 3. Histograms of functions counts; white corresponds to frequently occurring functions and black to infrequently occurring functions. (a) Dictionary D0, evaluated over four training set sequences. (b) Dictionary D1, evaluated over four training set sequences. (c) Dictionary D1, evaluated for "Silent Voice." (d) Dictionary D1, evaluated for "Mobile and Calendar."

The PSNR improvement achieved by D1 over D0 ranged between 0.1 dB in the case of "Silent Voice" to 0.7 dB in the case of "Mobile and Calendar." Most importantly, owing to a reduction of the dictionary size and basis function lengths, this gain was achieved at a computational cost of $T_c = 2023S$ multiplications $+1751S$ additions, $T_m = 512S$ comparisons per atom, a reduction by factors of 3 and 1.6, respectively. An additional benefit issues from the fact that an index into dictionary D1 can be coded with 8 bits, compared to 8.64 bits required for an index into dictionary D0. These costs can be lowered by entropy coding. Experiments performed show that the histograms of atom counts can vary considerably for different video sources [see Fig. 3(c) and (d)], which suggests that the source associated with the sequence $\{k_n\}$ in the structure book should be modeled adaptively.

C. The Proposed Dictionary D2

The dictionary D1 outperformed the dictionary D0 both in terms of rate distortion and implementation cost. Therefore, a question arose whether D1 could be further simplified while still maintaining superiority over D0.

The dictionary D2 was derived by progressively removing functions from the dictionary D1. At every substep of the derivation, one function was removed from the dictionary used during the previous substep, starting from dictionary D1.

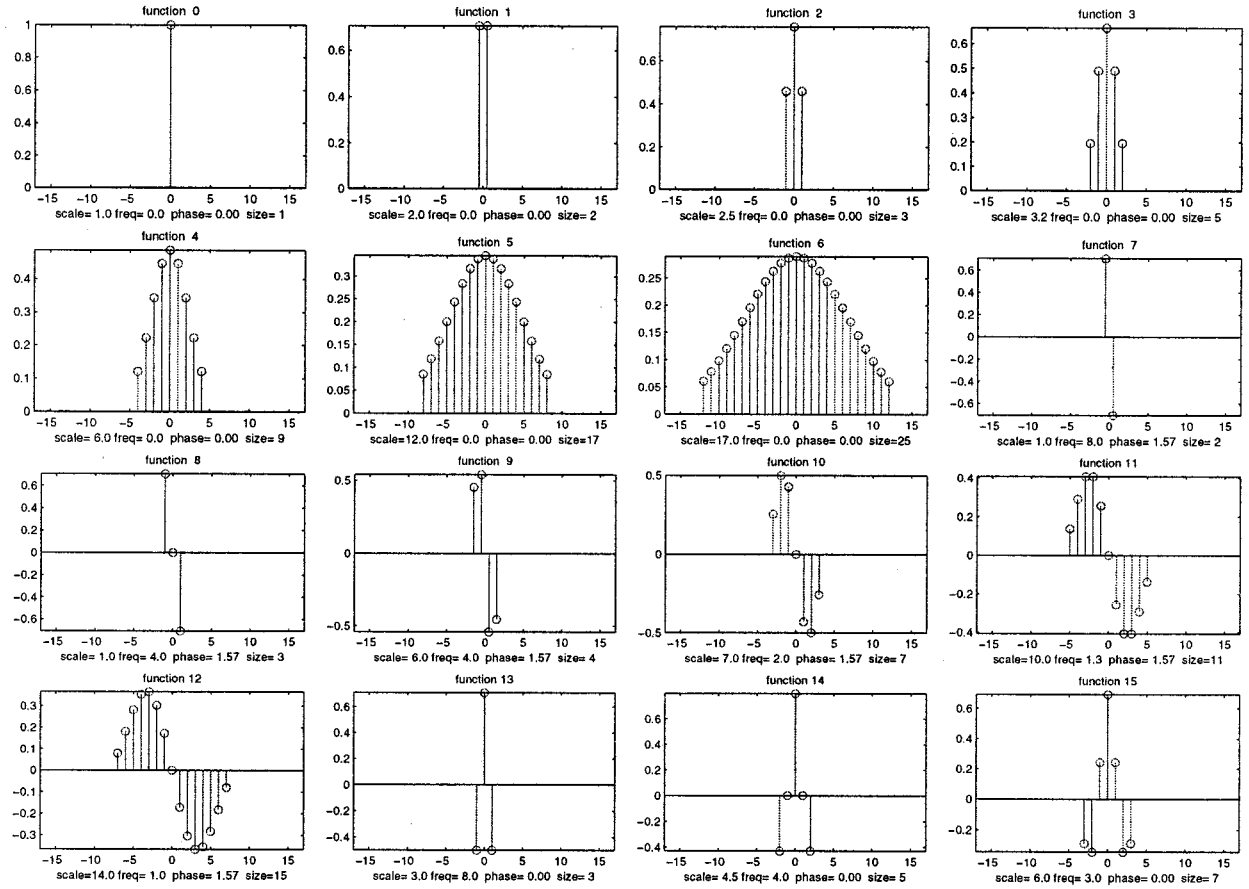


Fig. 4. The 16 impulse responses comprising dictionary D1.

TABLE I
$\{s, \xi, \phi, N\}$ QUADRUPLETS THAT DESCRIBE THE DICTIONARY D1

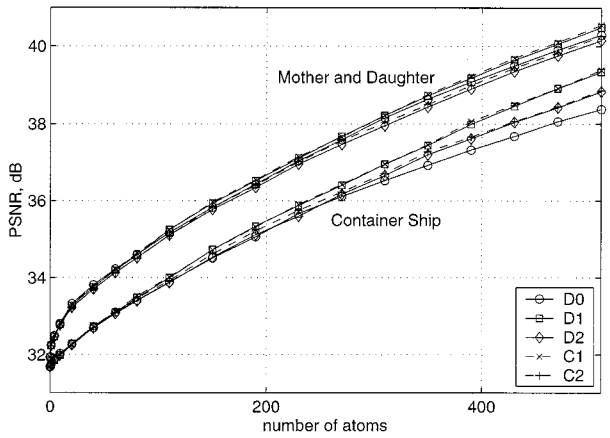
fn index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s	1.0	2.0	2.5	3.2	6.0	12.0	17.0	1.0	1.0	6.0	7.0	10.0	14.0	3.0	4.5	6.0
ξ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.0	4.0	4.0	2.0	1.3	1.0	8.0	4.0	3.0
ϕ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	0.0	0.0	0.0
N	1	2	3	5	9	17	25	2	3	4	7	11	15	3	5	7

The function to be removed was always the one which was the least-frequently matched to the coded signal (on average for the training set). The procedure was terminated when the PSNR performance of the current dictionary became inferior to the PSNR performance of D0. This occurred when the dictionary was narrowed down to between 12 and 8 functions, depending on the sequence. A decision was taken that the dictionary D2 should consist of 11 1-D functions for the simple reason that, in absence of entropy coding, 11×11 functions can be enumerated with 7 bits with very little redundancy. Thus, dictionary D2 was formed by removing functions from D1 in the following order (indices into 1-D functions): 14, 6, 11, 3, 13, leaving functions 0, 1, 2, 4, 5, 7, 8, 9, 10, 12, and 15. Dictionary D2 offers a considerably lower computational cost, compared to D0: $T_c = 840S$ multiplications and $708S$ additions, $T_m = 242S$ comparisons per atom, a reduction by factors of 7.3 and 3.3, respectively.

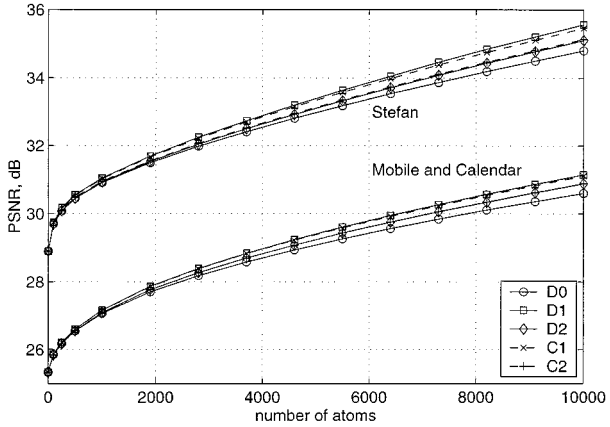
D. Comparison of Dictionary Performance

In order to verify the improved performances of dictionaries D1 and D2, they were employed to code data from outside the training set. Fig. 5 shows the PSNR versus the number of atoms plots, obtained by coding single DFD frames from example SQCIF and 4CIF resolution sequences (similar results were also obtained for QCIF and CIF resolution sources). Fast atom search was employed and atom products did not undergo any quantization. For SQCIF sequences, overlapped motion compensation was used, as defined in H.263. Non-overlapped motion compensation was used in the case of the 4CIF sequences. The following observations can be made from Fig. 5.

1) For all tested sequences, the decay rate of the residual is fastest in the case of the dictionary D1.
2) The decay rate achieved with the dictionary D2 is slower than the decay rate achieved with the dictionary D0 only



(a)



(b)

Fig. 5. Performance plots of matching pursuits video coding with dictionaries D0, D1, D2, C1, and C2. (a) SQCIF resolution. (b) 4CIF resolution.

in the case of "Mother and Daughter." For remaining sequences, D2 either matches or outperforms D0.

3) The proposed dictionaries perform consistently well for different sequences, resolutions, and in the presence of different motion compensation techniques.

It can be concluded that D1 offers an all-round rate-distortion-complexity improvement over dictionary D0. The performance of dictionary D2 is, on average, equivalent to that of dictionary D0, at a significantly lower computational cost.

IV. Reduced Cost Correlation for Matching Pursuits

The computational cost figures of dictionaries D0, D1, and D2 quoted in Section III show that correlations form the most expensive part of a matching pursuit. In this section, it is shown how a factorization of dictionary functions can be employed to achieve a substantial reduction of that cost.

A. Factorizing Basis Functions

The implementation cost required by a matching pursuits decomposition can be considerably reduced by factorizing the basis functions. The idea is to design the dictionary in such a way that longer dictionary functions arise through a convolution of shorter dictionary functions with low complexity auxiliary

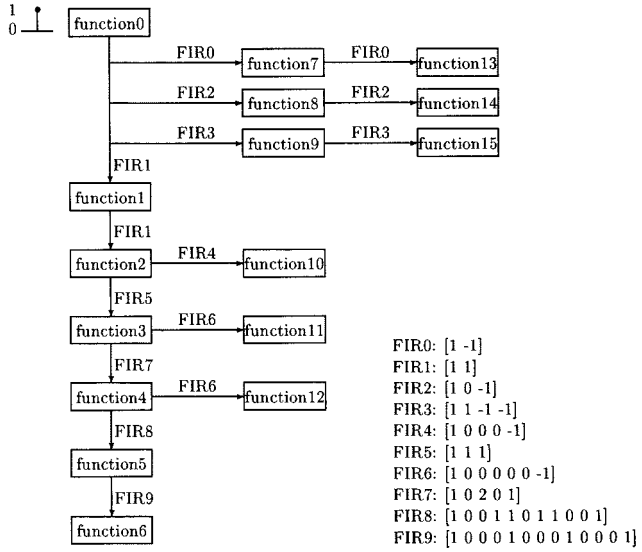


Fig. 6. Succession of convolutions forming the dictionary C1.

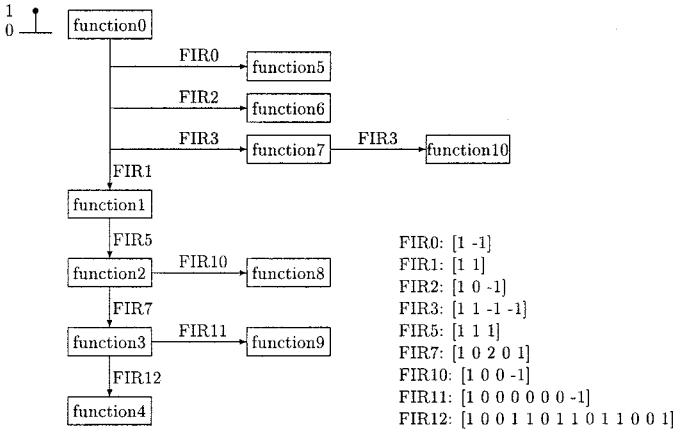


Fig. 7. Succession of convolutions forming the dictionary C2.

filters. This enables reusing previous filtering results within the matching pursuits framework.

Thus, dictionaries C1 and C2 were designed to approximate the dictionaries D1 and D2 respectively, and arise through a cascade of convolutions shown in Figs. 6 and 7. Fig. 1(d) and (e) show dictionaries C1 and C2 in a 2-D form. The coefficients of the auxiliary impulse responses FIR are equal to 1, -1, or (in one case) 2. Therefore, the correlation stage can be implemented with add and shift operations alone. If, for simplicity, a multiplication by 2 is treated as two additions, then the computational cost associated with dictionary C1 is equal to $476S$ additions, 57 multiplications, and $512S$ comparisons per atom. The multiplications are required to normalize the inner product values. Prior to normalization, dictionary functions of the same energy can be grouped and maximized separately. Then, only one normalization per group is required before the final maximization. There are 57 such groups in C1. Similarly, the computational cost associated with dictionary C2 is equal to $276S$ additions, 43 multiplications, and $242S$ comparisons per atom.

Employing a factorized dictionary increases storage requirements. The required number of frame buffers depends on the

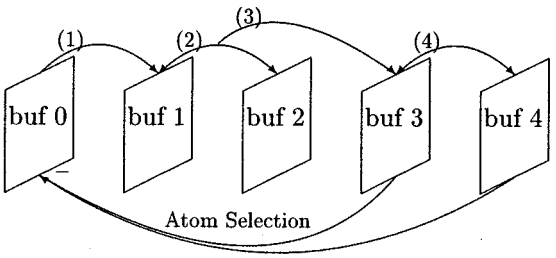


Fig. 8. Memory requirements of a fast matching pursuits algorithm. Arrows (1) and (2) illustrate possible data flow directions during vertical filtering, and arrows (3) and (4) illustrate possible data flow directions during horizontal filtering.

topology of the diagrams shown in Figs. 6 and 7. It can be shown that, in the case of dictionaries C1 and C2, a serial implementation requires five data buffers (Fig. 8): buffer 0 stores the coded signal. The results of filtering the signal vertically are stored in buffer 1. It will be necessary to further vertically filter the data stored in buffer 1, and to store both the original and the filtered signal versions. Frame buffer 2 is provided for that purpose. Vertically filtered data, stored either in buffer 1 or 2 is then subjected to horizontal filtering, and the result is stored in buffer 3. Again, it will be necessary to further horizontally filter the data stored in buffer 3, and to store both the original and the filtered signal versions. Frame buffer 4 is provided for that purpose.

It should be stressed that alternative factorizations exist for dictionaries C1 and C2 to those shown in Figs. 6 and 7, and small further reductions of computational cost could be achieved by rearranging the order of convolutions. However, this would increase storage requirements.

We acknowledge that another factorized dictionary was independently proposed by De Vleeschouwer and Macq [18]. This dictionary consists of Haar wavepackets and is shown in Fig. 1(f). Since the Haar basis arises through convolutions with the impulse responses [1,1] and [1,−1] at different scales, the computational cost associated with this dictionary is low: $168S$ additions, 9 multiplications, and $338S$ comparisons are required per atom. However, storage is increased to seven frame buffers. While this dictionary offered implementation simplicity, its PSNR performance was found to be inferior to that of dictionaries C1 and C2, due to the blocky nature of Haar functions. Since the submission of this manuscript, De Vleeschouwer and Macq reported a modified version of their algorithm [19]. The "blocky" Haar functions are now only used to search the DFD signal for atoms, while the decomposition is accomplished using a smoothed version of the dictionary, shown in Fig. 1(g). This dictionary will be referred to as C3 in the following, and its performance will be compared to other dictionaries in Section VI.

B. Performance

The performance of dictionaries C1 and C2 was evaluated using the procedure explained in Section III-D, and the results are shown in Fig. 5. It can be observed that for all tested sequences, the factorizations C1 and C2 maintain the performance of the prototype dictionaries.

C. Extension to Nonseparable Bases

In this section, successive factorizations were used to approximate prototype separable Gabor dictionaries. However, this approach can be applied in a more flexible manner, to construct arbitrarily shaped functions. For example, it seems logical to enrich the dictionary with diagonal functions of various orientations [8], [12]. Further research is required to establish whether such extensions justify the associated increase of computational and memory requirements.

V. Pruned Full Search

In order to keep the computational cost of a matching pursuits decomposition down to a reasonable level, a fast-search algorithm which limits the search area to a subset of the whole frame was proposed [7]. However, it is recognized that the fast search leads to suboptimum results, and one way of improving the decay rate is to increase the search area S. Since atom domains are limited, then for a sufficiently large value of S, the spatial support of any atom will occupy a small fraction of S. Mallat and Zhang [1] took advantage of this fact by proposing the following full-search algorithm. Suppose that at the $(n+1)$th stage of the decomposition an atom g_{k_n} is selected. Then, the residual $R_{n+1}f$ can be written as

$$R_{n+1}f = R_nf - \langle R_nf,\, g_{k_n}\rangle g_{k_n}. \tag{7}$$

Now, the inner product between any basis function $g_k \in D$ and the residual $R_{n+1}f$ can be written as

$$\langle R_{n+1}f,\, g_k\rangle = \langle R_nf,\, g_k\rangle - \langle R_nf,\, g_{k_n}\rangle \langle g_{k_n},\, g_k\rangle. \tag{8}$$

If the values $\langle R_nf,\, g_k\rangle$ and $\langle R_nf,\, g_{k_n}\rangle$ have been stored, then the calculation of the products $\langle R_{n+1}f,\, g_k\rangle$ only requires the computation of the inner products $\langle g_{k_n},\, g_k\rangle$. It is feasible to tabulate the values $\langle g_{k_n},\, g_k\rangle$ to reduce the complexity even further. Then, the computation of an inner product $\langle R_{n+1}f,\, g_k\rangle$ requires one multiplication and one addition only.

The products $\langle g_{k_n},\, g_k\rangle$ only take nonzero values for those functions g_k which overlap g_{k_n}. Therefore, the average cost of recomputing the products $\langle R_{n+1}f,\, g_k\rangle$ is independent from the size of the coded signal. Instead, it is governed by the average search area S_f, estimated using the following formula, which assumes that all atoms are selected with equal probabilities

$$S_f = (L/B + L_{\max} - 1)^2 \tag{9}$$

where $L_{\max}$ is the size of the longest impulse response in the dictionary. For dictionaries D0, C1, C2, and C3, S_f evaluates to 49^2, 38^2, 27^2, and 36^2, respectively.

Unfortunately, this method requires an extremely high amount of memory, as storage for $B \times B$ filtered versions of the residual signal is needed. This practically precludes it from video applications. However, instead of employing the update procedure of (8), the region affected by the previously picked atom can be simply recorrelated with the dictionary. If a factorized basis is employed, the correlation is accomplished with a small number of additions per function. For example, dictionaries C1 and C2 require an average of 1.86 and 2.28

TABLE II
SUMMARY OF THE COMPLEXITIES OF DIFFERENT DICTIONARIES

Dictionary	D0	D1	D2	C1	C2	C3
comments	Gabor fns Neff [7]	Gabor fns section III-B	Gabor fns section III-C	factorized section IV-A	factorized section IV-A	smoothed Haar De Vleeschouwer [19]
no. of functions	20×20	16×16	11×11	16×16	11×11	13×13
multiplications/atom	$6132S$	$2023S$	$840S$	57	43	9
additions/atom	$5712S$	$1751S$	$708S$	$476S$	$276S$	$168S$
comparisons/atom	$800S$	$512S$	$242S$	$512S$	$242S$	$338S$
no. frame buffers	3	3	3	5	5	7
S_f	49^2	32^2	23^2	38^2	27^2	36^2

S_f—average area filtered during the pruned full-search algorithm.

additions per function per every location in S_f, a complexity equivalent to the multiply-add operation of (8). This is the main idea behind the pruned full-search algorithm, which operates as follows.

1) The entire frame is searched during the first stage of the decomposition. For every location, an atom is stored that maximizes the absolute value of the product $\langle R_0 f, g_k \rangle$ at that location. Then, the best atom is picked by maximizing inner products of atoms stored at all individual locations.
2) During remaining stages, the search is only repeated on that subset of the signal, which has been affected by the previously picked atom. The size of this subset depends on the domain of dictionary functions and not on the size of the DFD signal.

The above algorithm was implemented with one modification: instead of storing an atom for every full-pixel location, the residual is segmented into fixed subblocks of size $M \times M$, and a single atom is stored for every subblock. This has a number of advantages, such as: reducing storage requirements for locally best atoms, reducing the number of normalization operations (if a factorized dictionary is used), and reducing the number of comparisons. A disadvantage of this modification is a small increase of the search area, as the region affected by the previously selected atom must be rounded up to fully cover an integer number of subblocks. The choice of the subblock size M is implementation dependent; we found that a choice of $M = 8$ actually led to a reduction of the search time compared to $M = 1$, while offering considerable storage savings.

The computational cost, associated with various dictionaries, is summarized in Table II. If a simplifying assumption is made that the costs of multiplication, addition and comparison operations are equivalent, dictionaries C1, C2 and C3 reduce the cost of atom search by factors of 13, 24 and 25 compared to dictionary D0 for the same search area. Apart from the first stage, a full pruned search with the dictionaries C1, C2 and C3 is smaller than the cost of a fast search with the dictionary D0 by factors of 4, 15, and 8 respectively. Another observation that can be made from Table II is that, for factorized dictionaries, the cost of maximizing the inner product is comparable to the cost of correlation. This is confirmed by the example coding times, shown in Table III. It should be noted that these times are implementation dependent and serve only as an illustration. Similarly, Table II is intended as a general guide. For example, issues such as operand fetch and result store cycles were not taken into account. In addition, the duration of product maximization may depend on data ordering.

TABLE III
NUMBER OF SECONDS SPENT DURING AN EXAMPLE 100 ATOM DECOMPOSITION. EVALUATED FOR "FOREMAN" AT QCIF RESOLUTION, USING A SILICON GRAPHICS O2 WORKSTATION WITH A MIPS R10000 PROCESSOR CLOCKED AT 195 MHz

Dictionary	D0	D1	D2	C1	C2	C3
fast search, 100 atoms, $S = 21 \times 21$						
t_c	16.98	6.07	2.59	0.84	0.50	0.53
t_m	0.84	0.59	0.28	0.59	0.28	0.39
pruned full search, first atom, S = whole frame						
t_c	-	-	-	0.85	0.43	0.42
t_m	-	-	-	0.54	0.25	0.23
pruned full search, remaining 99 atoms, $S = S_f$, $M = 8$						
t_c	-	-	-	4.74	1.53	2.25
t_m	-	-	-	2.76	0.82	1.15
motion estimation						
t	0.98	0.98	0.98	0.98	0.98	0.98

t_c— number of seconds spent correlating basis with residual. t_m—number of seconds spent maximizing the inner product value. Half-pixel motion estimation was performed within a radius of 16 pixels, using SAD as the matching criterion and the "spiral" evalutation order.

The benefits of employing a full-search matching pursuit are presented in Section VI.

VI. CODING RESULTS

This section compares the performances of dictionaries presented in this paper for QCIF and CIF resolution sequences.

A. Coder Configuration

Matching pursuits was investigated for coding the DFD signal. The first frame of every test sequence was coded in the H.263 intraframe mode. Remaining frames were coded as P-frames, using the standard motion-compensated architecture, with half pixel motion estimation and a block size of 16 × 16. The test sequences were 10 s long; CIF resolution sequences were coded at 30, and QCIF sequences were coded at 15 frames/s. Overlapped motion compensation was used in the

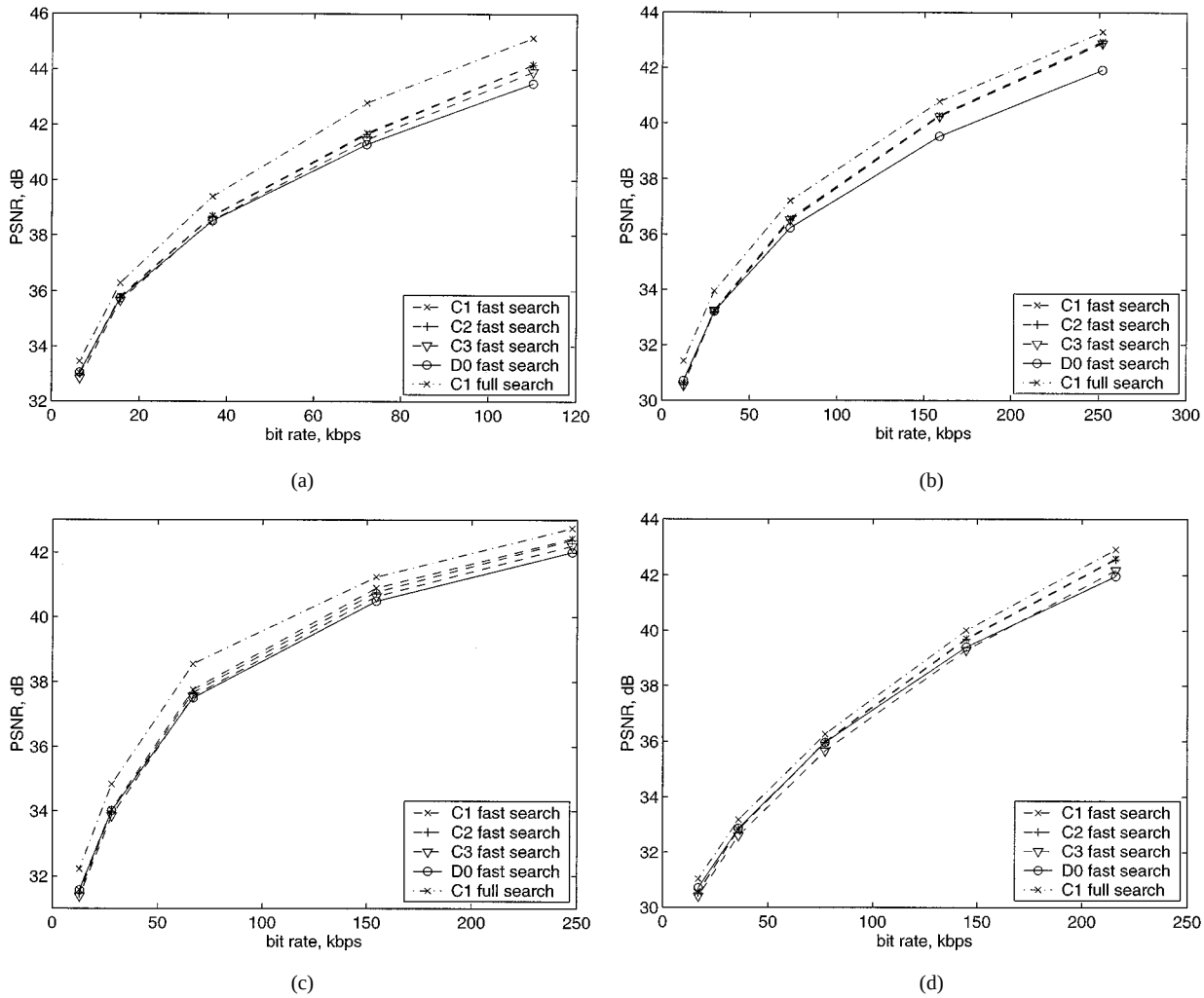


Fig. 9. PSNR performance of various dictionaries, QCIF resolution, 15 fps. (a) Akiyo. (b) Container Ship. (c) Hall Monitor. (c) Silent Voice.

case of the QCIF sequences, and the H.263 median-prediction algorithm was implemented to entropy code the motion field.

In the case of the fast search, the search area was set to $S = 21^2$. At every stage of the decomposition, a decision must be made whether to code an atom belonging to the luminance plane or an atom belonging to one of the chrominance planes. It was found that if this decision is taken solely based on inner-product values, then very few chrominance atoms are encoded in the case of dictionaries C1, C2, and C3. Therefore, inner products of the chrominance atoms were biased by a factor of 1.15 to force the encoding of color information.

The structure book was divided into three subsets, corresponding to the luminance and color component atoms, which were then coded separately. Atom positions were coded using the algorithm of Zeng and Ahmed [20]. Atom products were quantized inside the decomposition loop to the reconstruction levels 5, 9, 15, 25, 45, 80, 140, and 240. Thus, product magnitudes were coded with 3 bits and followed by a sign bit. Functions were coded using a single index into the appropriate 2-D dictionary. A variable length code designed assuming a uniform distribution of dictionary functions was used to code function indices. For dictionaries D0, C1, C2, and C3, the average cost of coding an index was equal to 8.72, 8.00, 6.94, and 7.49 bits, respectively.

B. Rate Control

Prior to the experiments, the test sequences were H.263 coded under constant quantization conditions. The matching pursuits codecs were then configured to match the bit expenditure of H.263 for every frame.

The cost of coding a C1, C2, or C3 dictionary index is lower than the cost of coding a D0 dictionary index. Consequently, in the case of factorized dictionaries, a given bit budget will be met by a higher number of atoms than in the case of D0, as illustrated in Table IV. This counteracts the reduction in computations. However, the computational penalty is very small, compared to the speed-up figures quoted in Section V. Secondly, it is usually compensated by an improvement in the PSNR performance.

C. Coder Performance

Figs. 9 and 10 show the coding results obtained for dictionaries D0, C1, C2, and C3. The following observations can be made from the plots corresponding to the fast-search strategy.

1) For very low bit rates, where a small number of atoms is coded, the performance of all dictionaries is very close. In some cases ("Silent Voice" QCIF, "Foreman" CIF,

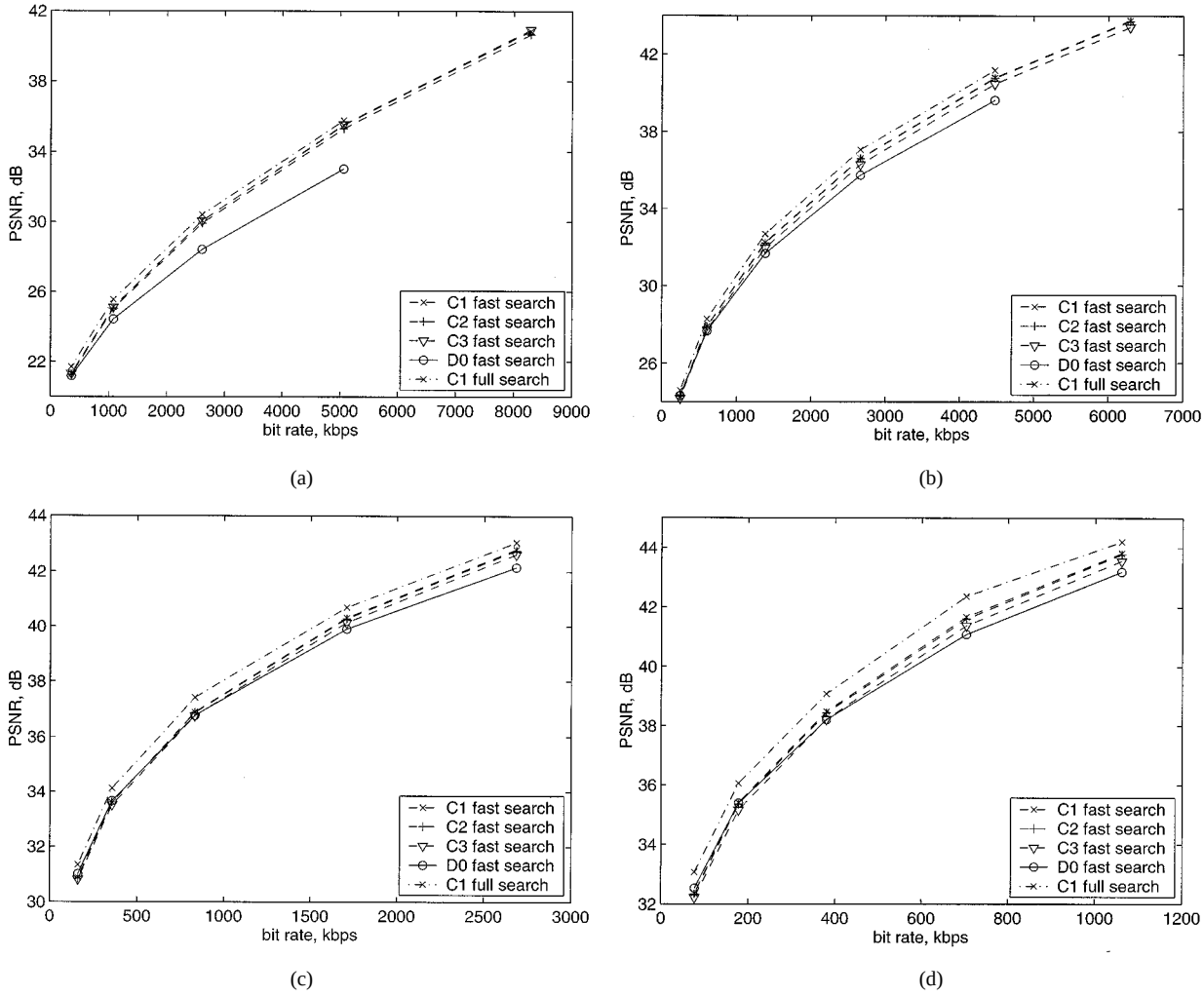


Fig. 10. PSNR performance of various dictionaries, CIF resolution, 30 fps. (a) Mobile and Calendar. (b) Stefan Edberg. (c) Foreman. (d) News.

TABLE IV
EXAMPLE CODING STATISTICS AND PSNR PERFORMANCE OF MATCHING PURSUITS DICTIONARIES

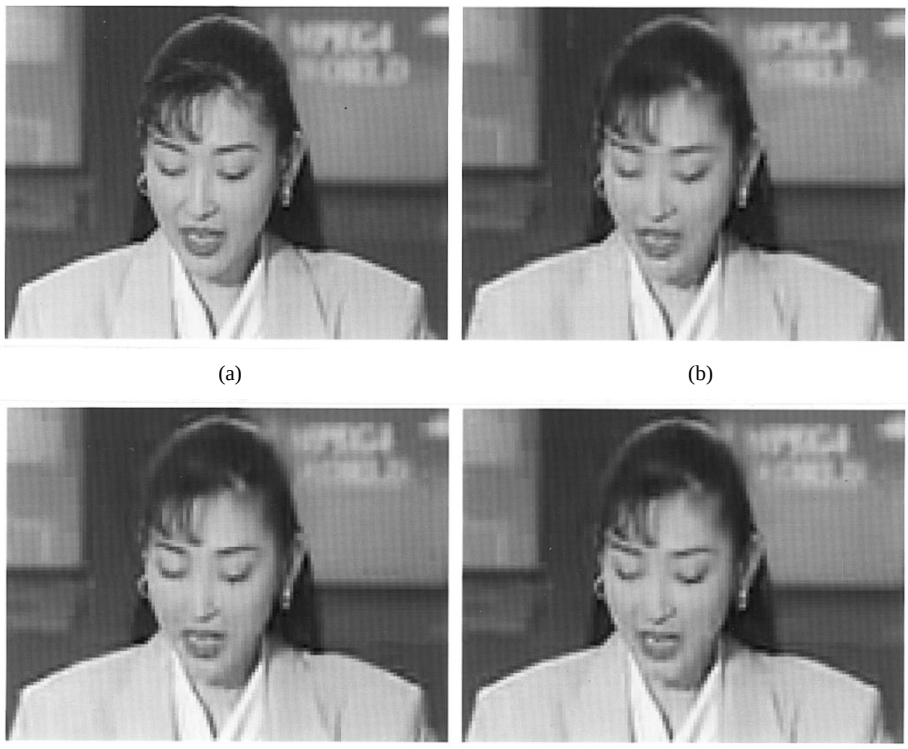
Sequence			D0 fast	C1 fast	C2 fast	C3 fast	C1 full
Hall	bit rate, kbps		27.96	27.96	27.95	27.96	27.96
Monitor	av atom/frame		73.6	76.0	80.0	78.4	76.5
QCIF		Y	34.01	34.04	33.98	33.83	34.83
15 fps	PSNR,	Cb	38.03	38.17	38.37	38.12	38.61
	dB	Cr	40.37	40.19	40.31	40.08	40.56
Stefan	bit rate, kbps		2663	2663	2663	2663	2663
Edberg	av atom/frame		4589	4710	5033	4875	4710
CIF		Y	35.75	36.63	36.60	36.30	37.07
30 fps	PSNR,	Cb	39.20	40.02	40.01	39.50	40.84
	dB	Cr	39.19	40.02	40.05	39.45	40.87

"News" CIF), dictionary D0 fractionally outperforms other dictionaries.

2) For all test sequences, the overall performance of the proposed dictionaries C1 and C2 is superior to that of dictionary D0. The gain increases with the bit rate and reaches up to 3 dB in the case of "Mobile and Calendar."
3) There is very little difference between the performances of dictionaries C1 and C2. Apart from the sequences "Mobile and Calendar" and "Hall Monitor," the corresponding PSNR curves cannot be distinguished.
4) Dictionary C3 (De Vleeschouwer and Macq [19]) matches the performance of the proposed dictionaries in the case of "Mobile and Calendar" and "Container Ship," and proves slightly inferior for the remaining sequences (by up to 0.4 dB).

For clarity, only a single full-search result is shown in Figs. 9 and 10. It was observed that the relationships between PSNR plots obtained with the full-search strategy are identical to those obtained with the fast-search strategy. The full-search matching pursuit offers a significant PSNR improvement over the fast search, ranging between 0.3–1.5 dB. Owing to the pruned full-search strategy, the cost of a full-search matching pursuit with dictionaries C1 and C2 is smaller than the cost of a fast matching pursuit with dictionary D0.

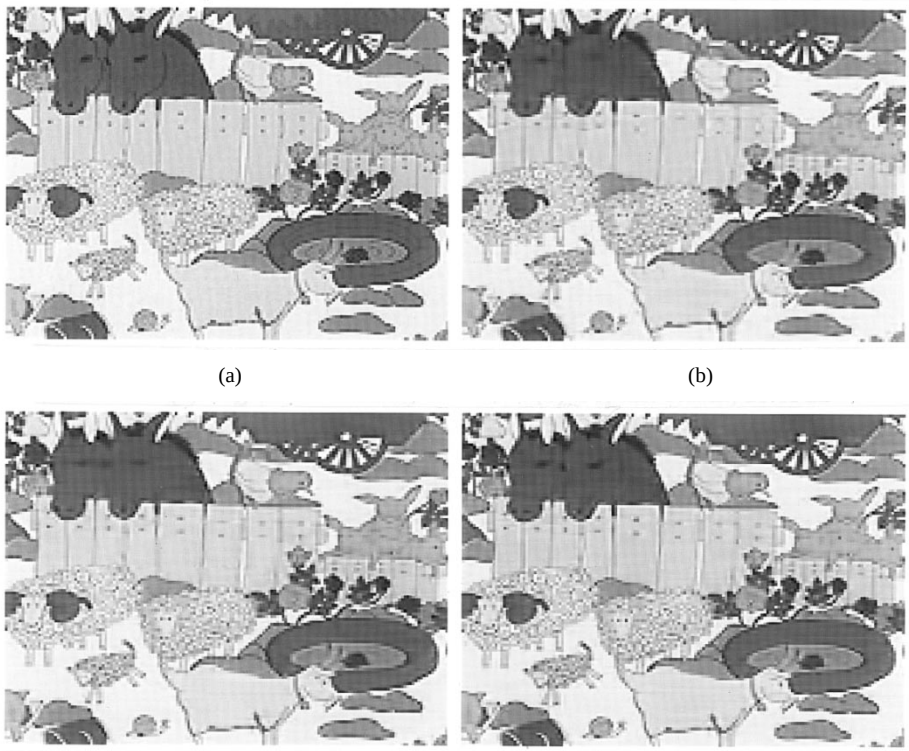
The reconstructed sequences also underwent an informal subjective quality assessment; example reconstructed frames are shown in Figs. 11 and 12. In the case of a fast-search algorithm, the subjective quality of reconstructions obtained with different dictionaries was often too close to identify any dictionary as superior. "Mobile and Calendar" is an exception that clearly favors the proposed factorized dictionaries over D0.



(a) (b)

(c) (d)

Fig. 11. Subjective performance of various dictionaries; "Akiyo" frame 298 (fragment), QCIF, 16 kbits/s. (a) Original. (b) D0, fast search, 35.65 dB. (c) C2, fast search, 35.74 dB. (d) C1, full search, 36.32 dB.



(a) (b)

(c) (d)

Fig. 12. Subjective performance of various dictionaries; "Mobile and Calendar" frame 299 (fragment), CIF, 2600 kbits/s. (a) Original. (b) D0, fast search, 24.42 dB. (c) C2, fast search, 29.91 dB. (d) C1, full search, 30.40 dB.

A noticeable temporal artifact in matching pursuits video coding is a sudden change from soft low contrast to sharp high contrast for some scene features. In the worst case, this may cause selected objects to go in and out of focus on

a frame-to-frame basis. The full-search algorithm usually succeeded in updating the scene more uniformly than the fast-search algorithm, thus producing a more pleasing, stable reconstruction with fewer soft-to-sharp transitions. It can be concluded that the full-search matching pursuit can offer significant advantages over the heuristic fast algorithm.

VII. Conclusion

Matching pursuits video coding has been cursed by an extremely high implementation cost. This has, so far, prevented a widespread acceptance of this method as state-of-the-art, despite excellent subjective and objective performance. This paper presented new dictionaries and implementation techniques for matching pursuits video coding, which significantly lessen the computational cost bottleneck. The total number of operations required to decompose the coded signal was reduced by over 20 times, compared to an implementation reported in the literature [7]. For a majority of test conditions, this reduction was supplemented by an improvement in objective and subjective reconstruction quality.

In Section III, new dictionaries D1 and D2 were derived as alternatives to the dictionary D0 reported by Neff and Zakhor [7]. The derivation was experimental and it was guided by the desire to improve the correlation between the dictionaries and the residual signal.

In Section IV, an original low-cost implementation for the correlation stage of a matching pursuit was introduced, which depended on factorizing the dictionary. Two low-cost factorized dictionaries—C1 and C2—were derived as approximations of dictionaries D1 and D2, and were shown to match the coding performance of their prototypes. Assuming that the costs of addition, multiplication, and comparison operations are equivalent, the dictionaries C1 and C2 reduce the implementation cost by factors of 13 and 24, compared to dictionary D0, while providing a superior reconstruction quality.

Section V introduced an efficient implementation of the full-search matching pursuit using the factorized bases. The full-search algorithm offered a clear subjective quality improvement compared to the fast-search algorithm. In terms of the PSNR, it provided an advantage ranging between 0.3–1.5 dB, depending on the sequence.

The proposed dictionaries C1 and C2 clearly offer a better balance between complexity and performance, compared to dictionary D0.

Acknowledgment

The authors acknowledge C. De Vleeschouwer for providing the coefficients of dictionary C3. P. Czerepiński is indebted to D. Redmill for numerous useful discussions and comments.

References

[1] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397–3415, Dec. 1993.

[2] H. R. Rabiee, R. L. Kashyap, and S. R. Safavian, "Adaptive image representation with segmented orthogonal matching pursuit," *IEEE Int. Conf. Image Processing*, vol. 2, pp. 233–236, 1998.

[3] H. R. Rabiee, S. R. Safavian, T. R. Gardos, and A. J. Mirani, "Low bit rate subband image coding with matching pursuits," *Proc. SPIE Visual Communications and Image Processing Conf.*, vol. 3309, pp. 875–880, 1998.

[4] M. Gharavi-Alkhansari, "A model for entropy coding in matching pursuits," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 778–782, 1998.

[5] M. Vetterli and T. Kalker, "Matching pursuit for compression and application to motion compensated video coding," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 725–729, 1994.

[6] M. Gharavi-Alkhansari and T. S. Huang, "Fractal video coding by matching pursuit," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 157–160, 1996.

[7] R. Neff and A. Zakhor, "Very low bit-rate video coding based on matching pursuits," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 158–171, Feb. 1997.

[8] D. W. Redmill, D. R. Bull, and P. Czerepiński, "Video coding using a fast nonseparable matching pursuits algorithm," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 769–773, 1998.

[9] S.-C. S. Cheung, R. Neff, and A. Zakhor, "hanges regarding matching pursuits in video VM V.11," Document ISO/IEC JTC1/SC29/WG11, MPEG 98/M3832, July 1998.

[10] R. Neff and S.-C. S. Cheung, "Cost and benefit analysis for matching pursuits as a version 2 tool," Document ISO/IEC JTC1/SC29/WG11, MPEG 98/M3834, July 1998.

[11] O. K. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor, "Video compression using matching pursuits," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 123–143, Feb. 1999.

[12] D. Bull, N. Canagarajah, and P. Czerepiński, "Dictionaries for matching pursuits video coding," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, pp. 540–543, 1999.

[13] P. Czerepiński, C. Davies, N. Canagarajah, and D. Bull, "Dictionaries and fast implementation for matching pursuits video coding," in *Proc. Picture Coding Symp.*, 1999, pp. 41–44.

[14] R. Neff and A. Zakhor, "Matching pursuits video coding at very low bit rates," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, 1995, pp. 411–420.

[15] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE J. Select. Areas Commun.*, vol. 5, pp. 1140–1154, Aug. 1987.

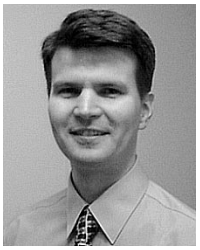
[16] P. Strobach, "Tree-structured scene adaptive coder," *IEEE Trans. Commun.*, vol. 38, pp. 477–486, Apr. 1990.

[17] W. Li and M. Kunt, "Morphological segmentation applied to displaced frame difference coding," *Signal Processing*, vol. 38, pp. 45–56, 1994.

[18] C. De Vleeschouwer and B. Macq, "New dictionaries for matching pursuits video coding," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 764–768, 1998.

[19] C. De Vleeschouwer and B. Macq, "Subband dictionaries for low-cost matching pursuits of video residues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 984–993, Oct. 1999.

[20] G. Zeng and N. Ahmed, "A block coding technique for encoding sparse binary patterns," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 778–780, May 1989.



Przemysław Czerepiński (S'95–M'98) received the M.Sc. degree in electronics and telecommunications from the Wrocław University of Technology, Wrocław, Poland, in 1994, and the Ph.D. degree from the University of Bristol, Bristol, U.K., in 1999, with a thesis on displaced frame difference coding.

Currently, he is a Research Associate at the Centre for Communications Research, University of Bristol. His research interests include all aspects of image- and video-coding technology.

Colin Davies (M'96) received the B.Eng. degree in electronics and the Ph.D. degree from the University of Southampton, Southampton, U.K., in 1990 and 1996, respectively. His doctoral work was in the area of sensing the 3-D shape of dynamic scenes using structured light applied to capture the shape of the human mouth during speech.

Since 1996, he has been working on new media coding and delivery technologies at NDS Ltd., U.K., a leading supplier of systems for the secure distribution of digital entertainment and information.

Nishan Canagarajah (M'00) received the B.A. (Hons.) degree and the Ph.D. degree in digital signal processing techniques for speech enhancement, both from the University of Cambridge, Cambridge, U.K.

He is currently a Senior Lecturer in Signal Processing at the University of Bristol, Bristol, U.K. He is also an Editor of a book on mobile multimedia technology. His research interests include image and video coding, nonlinear filtering techniques and the application of signal processing to audio and medical electronics.

Dr. Canagarajah is a Committee Member of the IEE Professional Group E5, a member of the Virtual Centre of Excellence in Digital Broadcasting and Multimedia Technology, and an Associate Editor of the *IEE Electronics and Communication Journal*.

David Bull (M'93) is currently a Professor of Signal Processing at the University of Bristol, Bristol, U.K., where he leads the Image Communications Group in the Centre for Communications Research, and has worked widely in the fields of 1- and 2-D signal processing. He has published over 150 papers and a book in these areas. Previously, he was an Electronic Systems Engineer at Rolls Royce, and a Lecturer at the University of Wales, Cardiff. His recent research has focused on the problems of image and video communications, in particular error-resilient source coding, linear and nonlinear filterbanks, scalable methods, content-based coding, and architectural optimization.

Dr. Bull is a member of the EPSRC Communications College and the Program Management Committee for the DTI/EPSRC LINK program in Broadcast Technology. Additionally, he is Director of the Virtual Centre of Excellence in Digital Broadcasting and Multimedia Technology, and a member of the U.K. Foresight ITEC panel.