DICTIONARIES FOR MATCHING PURSUITS VIDEO CODING

David Bull, Nishan Canagarajah and Przemysław Czerepiński

Centre for Communications Research, University of Bristol
Merchant Venturers Bldg., Woodland Rd., Bristol BS8 1UB, United Kingdom
{dave.bull, nishan.canagarajah, p.j.czerepinski} @bristol.ac.uk

ABSTRACT

Matching pursuits is a dictionary-based coding method, which has recently been demonstrated to outperform standard techniques, such as H.263, at low bit rate video compression. Unfortunately, the method involves an extremely high computational cost, compared to the DCT-based H.263. The contribution of this paper relies on reporting three new dictionaries for matching pursuits video coding, whose advantages include both a reduced computational cost and improved PSNR performance, compared to dictionaries reported to date.

1. INTRODUCTION

1.1. Matching Pursuits

The theory of matching pursuits was developed by Zhang and Mallat [1]. The method was initially utilized as a signal analysis rather than compression tool. More recently, it was applied to coding the *displaced frame difference* (DFD) signal, that arises during the process of digital video compression [2], and reported to outperform the H.263 standard for low bit rate video coding.

A matching pursuit strives to characterize the coded signal in terms of a number of *dictionary functions*. Contrary to other compression algorithms, such as orthogonal or subband transforms, the dictionary is not restricted by orthogonality or perfect reconstruction conditions. This fact has two important consequences:

1. Good compression performance can be achieved by selecting dictionary functions that are well-matched to the signal structure.
2. The function that best matches the signal is determined by an exhaustive search. The contribution of this function is subtracted from the signal, and the search process is repeated on the residual, until a quality criterion is achieved. The iterative search renders method's computational cost high. Heuristic fast search algorithms, which reduce the computational cost at the expense of a slightly deteriorated performance have been developed.

An item of coded data, that consist of the dictionary function index, the value of the scalar product between the function and local DFD data, and the position of the function within the frame is referred to as an *atom*.

1.2. Objectives and Contribution of this Work

The main goal of this work is to critically analyse the dictionaries proposed by other authors [2], and to investigate the possibility of improving the performance of matching-pursuits based compression by a better selection of dictionary functions. As a result, three new separable dictionaries are reported, whose advantages include reduced computational cost requirements and improved compression performance, compared to dictionaries reported to date.

2. SEPARABLE GABOR FUNCTION DICTIONARIES FOR VIDEO CODING

2.1. Introduction

A 1-dimensional discrete Gabor function $g_{\vec{\alpha}}(i)$ is specified by a set of four parameters, $\vec{\alpha} = \{s, \xi, \phi, N\}$, the scale s, the frequency ξ, the phase ϕ and the domain size N:

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

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

Typically, a dictionary comprises separable two-dimensional Gabor functions, defined as:

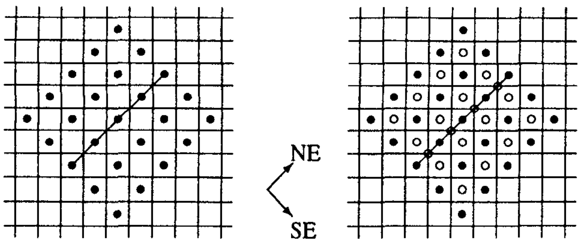
$$g_{\vec{\alpha}\vec{\beta}}(i,j) = g_{\vec{\alpha}}(i) g_{\vec{\beta}}(j) \quad (2)$$

The use of separability enables a considerable reduction of the computational cost, associated with searching the DFD signal for atoms [2].

In addition to simple horizontal-vertical separability, diagonally separable Gabor functions are also employed in this paper. Consider figure 1: without loosing any generality, it can be assumed that diagonal separability is achieved by firstly filtering in the south-east (SE) direction, and secondly in the north-east (NE) direction. Suppose an image is filtered using a single filter of length 5 in the SE direction. Then, filtering in the NE direction, as shown in figure 1a, will leave out every other SE diagonal. An alternative approach, adopted in this work, is shown in figure 1b. Filtering in the SE direction is performed using two filters: an odd-length filter, whose domain is indicated by black disks, and an even-length filter, whose domain is indicated by circles. The result of correlating the signal with the odd-length filter lies on a full pixel location, whereas the result of correlating the signal with the even-length filter lies on a half pixel location, between two diagonally contiguous pixels. With such an approach, a sample is available per every SE diagonal for filtering in the NE direction.

One consequence of the implementation of diagonally separable filters detailed above is that three impulse responses are now required: (1) an odd-length filter for convolution in the SE direction, (2) an even-length filter for convolution in the SE direction

0-7803-5471-0/99/$10.00©1999 IEEE



(a) (b)

Figure 1: Implementation of diagonally-separable Gabor functions. (a) drawbacks of employing a single filter; (b) correct approach, employing two filters.

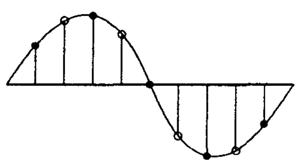


Figure 2: Sampling a Gabor function for diagonal separability. Samples marked with disks form the odd-length response for filtering in the SE direction, samples marked with circles form the even-length response for filtering in the SE direction, and samples marked either with disks or circles form the response for filtering in the NE direction.

and (3) a filter, composed of all the samples of even and odd responses, for filtering in the NE direction. The sampling process of a Gabor function for diagonal separability is shown in figure 2.

The following sections describe the proposed separable Gabor function dictionaries for matching pursuits video coding.

2.2. Dictionary D0

This is the 20×20-function dictionary reported by Neff and Zakhor. Its 2D form is shown in figure 3; the $\{s, \xi, \phi, N\}$ parameters can be found in the reference [2]. This dictionary is used as a basis for comparison to the proposed dictionaries, described below.

2.3. Dictionary D1

The dictionary D0 was adopted as the starting point for the derivation of the dictionary D1. Then, the contribution of every dictionary function to the reconstructed signal was scrutinized, cf. figure 6. It was concluded that selected functions can be removed from the dictionary D0 without compromising the compression performance. Following that, a further group of heuristically-motivated modifications was proposed. The benefits (if any) of every modification were evaluated experimentally, using four test sequences. Only the modifications that improved the overall system performance by either leading to a better reconstruction quality or a reduced coding cost were kept on board. The derived dictionary D1 consists of 17 functions, and is shown in a 2D form in figure 4a. See table 1 for the $\{s, \xi, \phi, N\}$ quadruplets that describe dictionary functions. The relative complexity of a matching pursuit with this dictionary is shown in table 2. The following list summarizes the introduced modifications:

1. Removing from D0 the functions which corresponded to the least frequently occurring atoms, cf. figure 6.
2. Introducing even-length functions 1, 14, 15 (table 1). (All functions in the dictionary D0 are odd length.)
3. Redesigning the set of lowpass functions. In short, the progression of function lengths: 1, 2, 3, 5, 9, 17, 25 (functions 0–6 in table 1) was found to be superior to the progression 1, 5, 9, 11, 15, 21, 23, 29, 35 of lowpass function lengths in dictionary D0.
4. Redesigning the set of highpass functions. Most importantly, the functions 15 and 16, with the frequency $\xi = 8$ were introduced. ($\xi = 4$ is the highest frequency present in the dictionary D0.) This is illustrated in figure 7.

2.4. Dictionary D2

A lower computational complexity and a better PSNR performance demonstrate the superiority of the designed dictionary D1 over the dictionary D0. The derivation of the dictionary D2 was motivated by the need to further reduce the size (and hence the complexity) of the dictionary D1, while still maintaining the performance equal or superior to that of the original 20-function dictionary, D0.

The derivation consisted of a number of substeps. At every substep, one function was removed from the dictionary used during the previous substep, starting from dictionary D1. The removed function was alway the one, whose contribution to the reconstruction quality, averaged over four test sequences, was the smallest during a given step. The procedure was terminated when the PSNR performance, associated with the current dictionary was inferior to the PSNR performance, associated with the dictionary D0. Thus, functions were removed in the following order (cf. table 1): 12, 6, 11, 8, 3, 15, 7, leaving the following ten functions: 0, 1, 2, 4, 5, 9, 10, 13, 14, 16 in the dictionary D2. This dictionary is shown in a 2D form in figure 5.

2.5. Dictionary D3

Dictionaries that exploit the horizontal-vertical separability only have been reported to date. The motivation for deriving dictionary D3 was to investigate the performance improvements that can be achieved by including diagonally-separable Gabor functions in the dictionary. Thus, the dictionary D3 consists of two parts:

- the part that supports horizontal-vertical separability—dictionary D1;
- the part that supports diagonal separability—an appropriately resampled version of dictionary D1 (see figure 4b).

3. RESULTS

3.1. Coder Configuration

Matching pursuits have been investigated in the context of coding the DFD signal. Thus, in order to avoid the influence of other methods' coding artefacts, the first frame of any sequence was left uncoded. Remaining frames were coded in the *predictive mode*, i.e. the motion compensated prediction for the current frame was formed from the reconstructed previous frame.

The coder employed standard block-matching motion estimation, with a block size 16×16. The search radius was limited to 16 pixels; the search was performed on full pixel locations, and then refined on half pixel locations within a two pixel radius. Standard H.263-like entropy coding was employed on the motion field.

Because of its recursive nature, the computational cost involved in a full-search matching pursuit is extremely high. In order to speed up the coding process, a fast search algorithm used in [2] was employed.

The magnitude of atom's scalar product was always quantized uniformly, with a quantizer value $Q = 4$. Atom parameters, such as horizontal and vertical components of the 2-dimensional Gabor function, the scalar product, atom position within the frame and the 'last atom' flag were all coded using simple first order models. An adaptive arithmetic coder was used for that purpose.

The source material consisted of 4 sequences: 'Silent Voice', 'Foreman', 'Table Tennis' and 'Mobile and Calendar'. All results quoted below have been obtained for 100 luminance frames at CIF resolution.

The coder operated in a fixed rate mode, where the number of bits spent coding any frame was kept constant. Every sequence was coded at one bit rate, selected to match its relative complexity. The coding rates were 256 kbps for 'Silent Voice', 512 kbps for 'Foreman', 1000 kbps for 'Table Tennis' and 2000 kbps for 'Mobile and Calendar'. All coding parameters were kept fixed throughout the tests.

3.2. Coder Performance

Table 2 summarizes the performance of the proposed dictionaries. The following observations can be made from there:

- The 17-function dictionary, D1, outperformed the dictionary D0 by 0.1 to 0.9 dB at approximately half the computational cost.
- The 10-function dictionary, D2, matched the performance of the dictionary D0, at approximately 20% of the computational cost.
- The combined 17-function h-v and diagonally separable dictionary, D3, outperformed the dictionary D0 by 0.3 to 1.2 dB at a similar computational cost.

Figure 8 presents example subjective results of utilizing dictionaries D0, D2 and D3.

4. CONCLUSIONS

Three dictionaries (D1, D2 and D3) for matching pursuits based compression have been developed in the course of this work. Each possesses advantages over the previously reported dictionaries, and provides a different trade off between the computational cost and performance.

Acknowledgement

The authors wish to acknowledge the EPSRC grant GR-L43596 and NDS UK for their support of this work.

5. REFERENCES

[1] S.G. Mallat and Z. Zhang. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[2] R. Neff and A. Zakhor. Very Low Bit-Rate Video Coding Based on Matching Pursuits. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):158–171, February 1997.

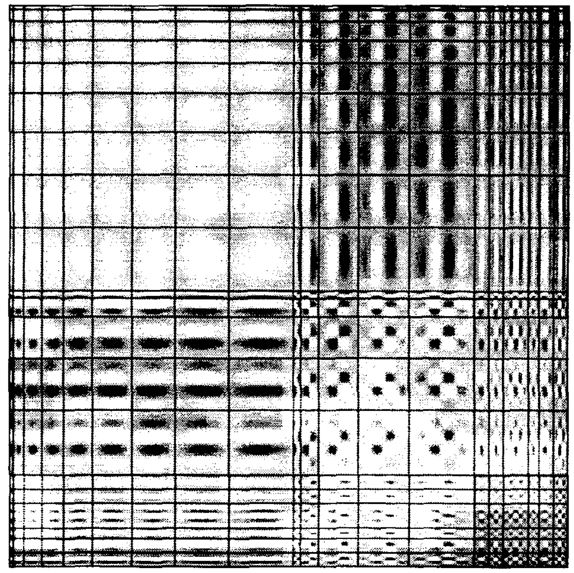
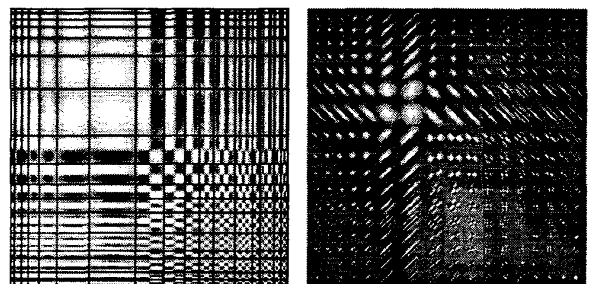


Figure 3: The two dimensional dictionary D0 (20×20 functions).



(a) (b)

Figure 4: (a) The two dimensional dictionary D1 (17×17 functions); (a)(b) The two dimensional dictionary D3 (17×17 + 17×17 functions).

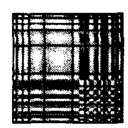
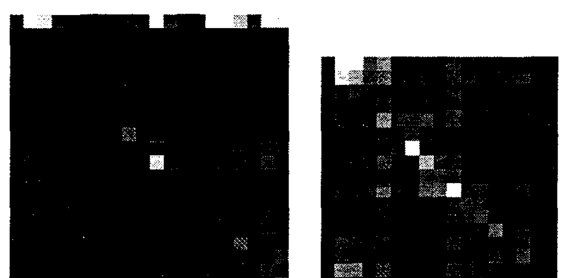
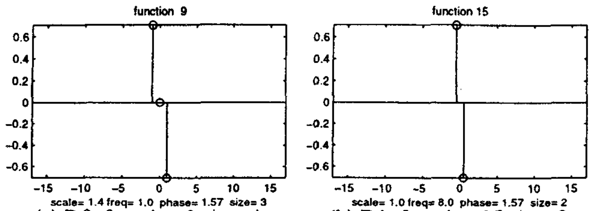


Figure 5: The two dimensional dictionary D2 (10×10 functions).



(a) dictionary D0 (b) dictionary D1

Figure 6: Relative frequencies of separable dictionary functions. White corresponds to frequently occurring functions, and black to infrequently occurring functions.



(a) D0, function 9, $\xi = 4$ (b) D1, function 15, $\xi = 8$

Figure 7: Highest frequency functions in the dictionaries D0 and D1.

(a) H.263, 34.94 dB (b) D0, 35.70 dB (c) D2, 35.72 dB (d) D3, 36.13 dB

Figure 8: Fragments of the reconstructed frame 100, 'Akiyo', CIF resolution, coded at 48 kbps.

fn index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
s	1.0	2.0	2.5	3.2	5.0	12.0	17.0	14.0	10.0	7.0	6.0	8.0	8.0	1.0	4.0	1.0	3.0
ξ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.3	2.0	3.0	3.0	4.0	4.0	4.0	8.0	8.0
ϕ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	$\pi/2$	$\pi/2$	$\pi/2$	0.0	$\pi/2$	0.0	$\pi/2$	$\pi/2$	$\pi/2$	0.0
N	1	2	3	5	9	17	25	15	11	7	7	11	9	3	6	2	3

Table 1: $\{s, \xi, \phi, N\}$ quadruplets that describe the dictionary D1.

dictionary	no. of functions	relative computational complexity	'Silent Voice'	'Foreman'	'Table Tennis'	'Mobile & Calendar'
D0	20	1.0	37.67	35.81	33.62	25.41
D1	17	0.5	37.81	35.97	34.05	26.33
D2	10	0.2	37.66	35.81	33.86	26.08
D3	17+17	1.0	37.98	36.36	34.17	26.67

Table 2: Summary of obtained PSNR results, dB.