Politechnika Poznańska
Wydział Elektryczny
Instytut Elektroniki i Telekomunikacji
Zakład Telekomunikacji Multimedialnej i Radioelektroniki

Kompresja Cyfrowych Sekwencji Wizyjnych z Wykorzystaniem Poszukiwania Dopasowującego dla Reprezentacji Separowalnych

Leszek Górecki

Rozprawa doktorska
wykonana pod kierunkiem
prof. dr. hab. inż. Marka Domańskiego

Poznań, czerwiec 2006

Poznań University of Technology
Electrical Engineering Faculty
Institute of Electronics and Telecommunications
Division of Multimedia Telecommunications and Radioelectronics

Video Compression Using Matching Pursuit for Separable Representation

Leszek Górecki

Doctoral Dissertation

Advisor: Prof. Marek Domański

Poznań, June 2006

Abstrakt

Optymalna aproksymacja funkcji przy pomocy linowej kombinacji nieortogonalnych funkcji należących do nadmiarowego słownika jest problemem NP-trudnym. Opublikowany przez Mallata i Zhanga algorytm poszukiwania dopasowującego (ang. *matching pursuit*), jest sub-optymalnym, zachłannym algorytmem dla wyżej podanego problemu. Algorytm ten został użyty do kodowania błędu predykcji w systemie kompresji sekwencji wizyjnych przez Neffa i Zakhor. Niemniej jednak duży nakład obliczeń potrzebny do znalezienia pojedynczego atomu wymusił nałożenie wielu założeń. Jednym z takich założeń jest separowalnośc funkcji tworzących słownik.

Pomimo wielu założeń i ograniczeń algorytm poszukiwania dopasowującego użyty do kodowania błędu predykcji nadal cechuje się dużym zapotrzebowaniem na moc obliczeniową oraz uniemożliwia adaptację słownika do kodowanego sygnału. W rozprawie, została postawiona teza dotycząca zmiany procesu poszukiwania atomu. Kluczowym elementem nowego algorytmu jest dekompozycja separująca, pozwalająca na znalezienie najlepszej - w sensie błędu średniokwadratowego - funkcji separowalnej dla dowolnego sygnału wejściowego.

Użycie dekompozycji separującej w algorytmie poszukiwania dopasowującego pozwoliło nie tylko na ponad siedmiokrotne zmniejszenie nakładu obliczeń, ale także umożliwiło zaproponowanie schematu adaptacji słownika. Zaproponowany schemat uczenia się posłużył do stworzenia nowego słownika, który dawał wyraźnie lepszą aproksymację sygnału w stosunku do słownika zaproponowanego przez Neff i Zakhor. Dodatkowo, schemat uczenia się pomógł w przeprowadzeniu eksperymentów pozwalających na sformułowanie ogólnych spostrzeżeń. Eksperymenty dowiodły, że istnieje górna granica dla liczby funkcji w słowniku, która powoduje, że dalsze rozszerzanie słownika staje się niekorzystne z punktu widzenia kompresji danych. Oznacza to, że dalszą poprawę jakości aproksymacji sygnału można osiągnąć poprzez adaptację słownika do pojedynczego obrazu sekwencji. W rozprawie został podany przykład adaptacji słownika do kodowanego sygnału. Uzyskane wyniki, potwierdziły oczekiwany wzrost w jakości aproksymacji.

Abstract

The problem of optimal approximation of function with a linear expansion using overcomplete dictionary of non-orthogonal waveforms is NP-hard. Matching pursuit, introduced by Mallat and Zhang, is a greedy sub-optimal algorithm for finding an approximate solution to the above problem. This technique has been adopted by Neff and Zakhor as an alternative to the conventional DCT-based method for coding a prediction error frame. Despite the greedy strategy, the most significant problem of the matching pursuit is its intensive computation in the encoding step. Therefore, many assumptions and limitations -as separability of functions from dictionary- are applied in real-world applications.

Nevertheless, despite limitations and assumptions, the computational load of matching pursuit is still huge. In addition, another fundamental problem of matching pursuit, i.e. the lack of feedback between an input signal and a dictionary, still exists. Therefore, in order to break through the drawbacks of the matching pursuit, in the thesis of the dissertation, new strategy for searching atoms is proposed. The key element of this technique is separable decomposition that allows for computing a separable function that minimises the Euclidean norm of approximation error.

The results showed that the proposed algorithm is over seven times faster than the classic matching pursuit algorithm. Additionally, the novel algorithm allows for designing a dictionary. Dictionaries obtained using a proposed learning scheme, outperforms the dictionary proposed by Neff and Zakhor in term of PSNR. The experiments confirmed that the separable decomposition efficiently exploits separability of an input signal and gives a way to improve the representational performance of a dictionary. Moreover, the learning scheme described in the dissertation, gives great support for experiments. An important observation is that the highly redundant dictionary does not improve the quality of approximation. The experimentally obtained results indicate the bound of the number of separable functions in the dictionary that makes compression process unattractive. This fact means that further improvement in signal approximation lies in adequate prediction or adaptation to a current context i.e. to the frame or to the region of frame. The proposal for image-adapted dictionary is also presented in the dissertation. Results obtained by using dynamic dictionary adapted to the context of frame prove high compression efficiency of such system.

Acknowledgements

First and foremost, I would like to express my sincere thanks to my advisor Prof. Marek Domański for his invaluable help, for his support, and for his words of criticism. Moreover, I would like to ensure him that I would never want to explore the borders of his probably infinite patience. Therefore, once again, I thank him for his patience and generous guidance.

I owe many thanks to everyone that attempted to my seminary for their constructive questions, valuable remarks and helpful discussions. These words are particularly directed to Dr. Maciej Bartkowiak and Prof. Ryszard Stasiński.

In addition, I owe thank to Prof. Jerzy Nawrocki for introducing me to Marek Domański.

I am particularly aware of the invaluable personal support of my wife, Małgorzata. I also owe a great debt of thanks to my parents.

List of symbols and abbreviations

$\lvert\ \rvert$	- absolute value,
$\lVert\ \rVert$	- Euclidean norm,
$\langle\ ,\ \rangle$	- inner product,
$\lceil\ \rceil$	- round up (ceiling),
$\lfloor\ \rfloor$	- round down (floor),
1-D	- one-dimensional,
2-D	- two-dimensional,
α	- expansion coefficient,
AVC	- Advanced Video Coding (or Advanced Video Compression),
B	- number of functions in dictionary,
B-frame	- bi-directionally interframe encoded frame,
bpp	- bits per pixel,
CABAC	- Context-based Adaptive Binary Arithmetic Coder,
CAVLC	- Context-based Adaptive Variable-Length Coder,
CIF	- progressive 4:2:0, 352×288 luma-pixels video sequence,
DCT	- Discrete Cosine Transform,
DFD	- Displaced Frame Difference,
DS	- Diamond Search,
F	- average size of one-dimensional functions in dictionary,
F_{levels}	- number of reconstruction levels of linear function quantizer,
FFS	- Four-Step-Search,
GLA	- Generalised Lloyd Algorithm,
$\mathcal{H}$	- Hilbert space,
IDCT	- Inverse Discrete Cosine Transform,
I	- frame-intraframe encoded frame,
ITU	- International Telecommunication Union,

JPEG	- Joint Picture Expert Group,
K	- stepsize of linear quantizer,
KLT	- Karhunen-Loève Transform,
MAD	- Mean Absolute Difference,
MCP	- Motion-Compensated Prediction,
MMX	- Multimedia Extensions (or Matrix Math Extensions),
MP	- Matching Pursuit,
MPEG	- Motion Picture Expert Group,
MPwithSD	- Matching Pursuit with Separable Decomposition,
MSE	- mean squared error,
$\mathbb{N}$	- set of natural numbers,
N_{atoms}	- number of atoms used to encode prediction error,
N_{decomp}	- number of iterations of separable decomposition,
NINT	- the nearest integer value,
P-frame	- interframe encoded frame,
PSNR	- Peak Signal to Noise Ratio,
$Q_{stepsize}$	- stepsize of linear quantizer,
QCIF	- progressive 4:2:0, 176×144 lumina-pixels video sequence,
$\mathbb{R}$	- set of real numbers,
RLC	- Run Length Coding,
S	- size of local search,
TML4	- Test Model Long Term Number 4,
TSS	- Three-Step-Search,
VLC	- Variable Length Coding,
VLSI	- Very Large Scale Integration,
VQ	- Vector Quantization,
Y-PSNR	- PSNR for luminance,
$\mathbb{Z}$	- set of integer numbers,

Chapter 1
Introduction

1.1. Outline of problems

Generally speaking, video sequences contain significant amount of statistical and subjective redundancy within and between frames. The ultimate goal of video source coding is bit-rate reduction for storage and transmission by exploiting both statistical and subjective redundancies and to encode information using entropy coding techniques.

Dependent on the applications requirements we may consider *lossless* and *lossy* coding of the video data. The aim of lossless coding is to reduce video data for storage and to retain the original data representation. In contrast, the aim of lossy coding techniques is to achieve as good video quality as possible for available resources. For video sequences, some loss of information can usually be tolerated. There are at least three reasons for this. Firstly, one can assume that the digital signal is, in fact, an imperfect representation of a real-world scene. Secondly, significant loss of information can be tolerated by the human visual system without interfering with perception of the still image or video sequence. Thirdly, lossless compression is usually insufficient to store information using required number of bits. One way or another, lossy video compression is achieved by degrading the video quality. The smaller the file-size or the target bit-rate of the channel the higher the necessary compression of the video data and usually the more coding artefacts become visible. The main purpose of lossy coding techniques is to optimise the quality for a given target bit rate subject to objective or subjective optimisation criteria. It should be noted that the degree of distortion in video sequences (both the objective degradation as well as the amount of visible artefacts) depends on the complexity of the video scene as much as on the sophistication of the compression technique.

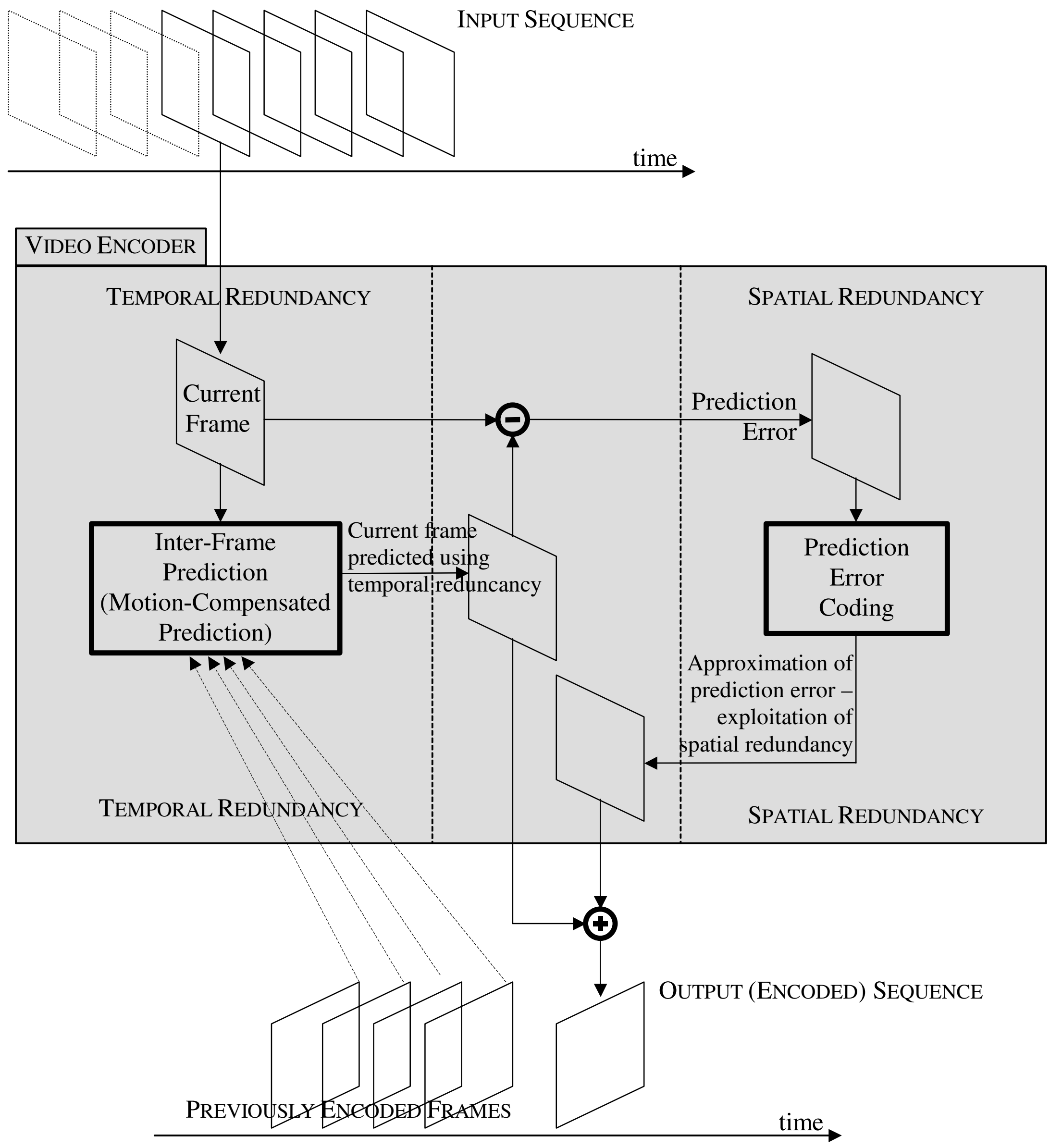


Fig.1. 1. The exploitation of temporal and spatial redundancy in hybrid video encoder. General scheme.

Most existing video compression systems have hybrid construction. The concept of such systems is based on the fusion of two techniques exploiting two different types of statistical redundancies [Tekalp95], [Doma98], [Ghan99], [Ohm04]. The first process i.e.

motion-compensated prediction, predicts each frame from its neighbouring frames, compresses the prediction parameters, and produces the prediction error frame. The second process codes the *prediction error* and is within the scope of this dissertation. Widespread video compression standards [H.261], [H.263], [MPEG-1], [MPEG-2], [MPEG-4a] use a DCT-based technique to code the residual error. As it was mentioned above, the prediction stage exploits temporal redundancy while the transform coding tries to efficiently get an advantage from spatial correlation (see Fig.1. 1).

Recently, the new generation of standardized video compression systems has been announced [MPEG-4c], [AVC], [H.264a], [VC-1]. In AVC-alike (*Advanced Video Coding*) system, better compression efficiency as compared to former standards has been attained due to:

- improved temporal (inter-frame) prediction,
- improved entropy coding,
- improved intra-frame coding with intra-frame prediction,
- more efficient motion vector representation,
- long-term memory,

(for more details see Section 2.2.3). Nevertheless, in context of the dissertation, it is worth mentioning, that while the prior video coding standards specified the DCT for transform coding of the prediction error, the AVC employs integer transform. In contrast to the DCT, this transform allows for a bit-exact specification.

Although DCT-based video coding is efficient, it introduces undesirable blocking artefacts especially at low bit rates [Derv96], [Fukuda00], [Sama04]. Due to bit rate restrictions, some blocks are represented by a small number of coarsely quantized transform coefficients, resulting in artefacts commonly known as blocking, blurring etc. Other approaches such as wavelet coding [Vett92], [Ant92], [Shap93] introduce ringing or rippling artefacts, which may become bothersome in the vicinity of image edges. Nevertheless, orthogonal transform coding is a fundamental technique of contemporary lossy compression systems and is particularly ubiquitous in image and video coding.

On the other hand, non-orthogonal overcomplete transforms present several interesting properties [Mall93], [Davis94], [Goyal95a] which position them as an alternative to orthogonal transforms [Bati03], [Durka96], [Ziyad98] especially in very low bit-rate data compression systems [Berg94], [Neff95], [Heus02]. It is well known that using simple and elementary words one can explain any complex theory or idea. However, such a description can be neither short nor elegant. Moreover, to express subtle thought (or delicate deviation in a signal) one needs to use a much larger set of words (i.e. dictionary) with sophisticated and expert vocabulary. This example has direct impact on functional analysis. An orthogonal basis is a special case of a dictionary since it is the smallest complete set that allows for exact representation of any signal. Nevertheless, the approximation of a signal with the assistance of the orthogonal dictionary is like the explanation with the assistance of simple words. It cannot be compact. An overcomplete dictionary is more flexible and more adequate to approximate an input signal using small number of elements, that is in finding the solution where most of the signal energy is captured by only few functions. Nevertheless, the overcomplete dictionary introduces difficulties to the choice of functions which should be selected to the approximation [Nafi96]. Additionally, this leads to the ambiguity. The reason of this comes into existence form the fact that a dictionary contains many „synonyms". This means that some functions from a dictionary have a common field of apply. At first sight, such ambiguity seems to be an advantage; nevertheless, it produces an unwanted increase of computational complexity. The number of feasible decompositions is infinite, and finding the best solution under a given criteria is a NP-hard problem [Davis94]. Therefore, there exists a great demand for rough methods in this field. Matching pursuit is one of the sub-optimal approaches that greedily approximate the solution to this NP-hard problem.

Matching pursuit (MP), introduced by Mallat and Zhang [Mall93], is an algorithm for decomposing a signal into a linear combination of functions chosen from possibly a redundant dictionary of functions. It is an iterative scheme that attempts to approximate input signal as closely as possible in a greedy manner at each step. The approximation of an input signal obtained in a single step is called an atom. The above technique has some very useful signal representation properties. For example, the dictionary element chosen at each stage is the element that provides the greatest reduction in mean square error between the true signal and the coded signal. In this sense, the signal structures are coded in order of importance, which is desirable in situations where the bit budget is very limited. For image and video coding applications, this means that the most visible features tend to be coded first. Weaker image features are coded later, if at all. It is even possible to control which types of image features are coded well by choosing dictionary functions to match the shape, scale or frequency of the desired features.

Matching pursuit algorithm provides an interesting way to iteratively decompose the signal in its most important features with limited complexity. This technique has been adopted by R.Neff and A. Zakhor in [Neff95], [Neff96a], [Neff96b], [Neff97a] as an alternative algorithm for coding a prediction error frame. Many results [Neff97], [AlSh99] have indicated that better visual quality at low bit rates can be obtained if DCT-based residual coders are replaced with matching pursuit coders. The video compression systems where coding of prediction error frame is realized using matching pursuit, are within the scope of this dissertation.

Since this dissertation takes focus on coding displaced frame differences in very low bit rate video systems, then the most important aspect is a character of this signal [Bhask95], [Tekalp95], [Doma98], [Ohm04]. The prediction error comes into existence as the difference between the original frame and the prediction of the current frame. It is a high frequency signal that is characterised by a local concentration of energy (see Fig.1. 2). The approximation of such a signal with respect to human visual system consists in coding the most visible and locally concentrated features first. Weaker signal features should be coded later, if at all. This simple and intuitive algorithm is very similar to matching pursuit strategy on condition that matching pursuit technique will choose exactly the same local concentration of energy to code, and the dictionary contains functions, which approximate the chosen region well. In theory, the satisfaction of above conditions is close to realization, but from a practical point of view, one must devise -once again- a rough solution.

Fig.1. 2. Prediction error from video sequences: Akiyo, Claire, Foreman.

Despite the greedy strategy, the most significant problem of the matching pursuit is its intensive computation in the encoding step. In real applications, this huge disadvantage is compensated with the aid of the following assumptions:

- searching of the best fitted function i.e. atom, is limited to the region which size is much smaller than the size of the prediction error,
- the dictionary contains separable functions,

- the dictionary is universal i.e. the matching pursuit uses functions from dictionary without any knowledge concerned the input signal and without any knowledge about how well the functions are suited to the prediction error.

The first assumption is well justified, since the displaced frame difference is characterised by local concentration of energy. In addition, Neff and Zakhor [Neff97] experimentally verified this assumption. They proposed an additional sub-step that reduces the space of searching an atom to the certain local region and allows it to take profit from the fact of concentration of the energy.

The second point is necessary for the existence of the fast inner product method. In addition, it is worth noticing that well known orthogonal transforms use separable functions to represent any signal, and separability is, in fact, an advantage. Finally, the works [Red98] [Neff00], [Neff02a] showed that the system with separable dictionary is slightly worse then the same system but supplemented by non-separable functions.

The last assumption results from the algorithm structure of the matching pursuit. The algorithm takes all functions from the dictionary and tries to find the best fitting for them. The direction of the search takes place from the dictionary towards the prediction error and makes it impossible to design the dictionary. Due to this, the matching pursuit uses the dictionary *a priori* and demands universality of the chosen set of functions to effectively represent an input signal.

1.2. Goals and thesis of the work

It has been experimentally proven [AlSh99], [Neff95], [Vlee98] that the matching pursuit method is an interesting alternative to the conventional DCT-based method in terms of both visual quality and PSNR. Especially, it does not suffer from the blocking artefacts in contrast to the block-based counterpart. However, the most significant problem of the matching pursuit is its computation load and the lack of feedback between an input signal and the dictionary.

The **main goal** of this dissertation is to propose new strategy for searching atoms in matching pursuit algorithm in order to break through the drawbacks of matching pursuit. The **particular goal** of this dissertation is to adopt a new technique for very low bit rate video coding system to encode a displaced frame difference and to prove high efficiency of such a system. An additional goal is to propose a learning scheme for designing dictionaries.

Taking into consideration the presented assumptions (Section 1.1), the **main thesis** of the dissertation is as follows:

- Using the property of separability of functions in the dictionary, it is possible to propose a strategy for searching atoms that allows for significant reduction of the computational complexity with a small decrease of efficiency. Moreover, the novel strategy may be used for designing the dictionary by exploiting information concerning the input signal.

The verification of the thesis will be performed based on the experimental results. Results for comparison of algorithms will be calculated based on the PSNR (*Peak Signal to Noise Ratio*) measure, which is defined as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right), \tag{1.1}$$

where, the MSE is mean squared error between an original and decoded samples of video frames. Although the PSNR measure does not correspond to subjective quality, nevertheless, the usage of PSNR is well motivated. The most important feature of the PSNR is that for an individual sequence and for a selected method, the higher PSNR values correspond to better quality of approximation. Moreover, verification of the thesis using the above objective measure is trustworthy since compared algorithms use Gabor functions to approximate a signal therefore the character of distortions is similar.

In order to verify the thesis, the experimental platform of video encoder will be implemented. Therefore, the existing AVC codec platform will be extended by the matching pursuit implementation. The environment obtained in this way will constitute the new reference platform and gives the reference results. Furthermore, the novel algorithm presented in this dissertation will be implemented and put to the tests. The comparison of the results will give the decisive answer concerning the efficiency of the novel method. Additional experiments will verify the thesis.

1.3. Overview of the dissertation

The dissertation is organised in seven chapters. Chapter 2 is substantially of a review nature, collecting background materials, which is important to the reminder of the document. Chapters 3 to 6, on the other hand, present author's contributions to the field.

Chapter 2 briefly describes the basic elements of video data compression. In particular, the main mechanisms that are involved in the inter-frame video coding like the motion-estimation and the transform coding are presented. The further part of this chapter is devoted to foundations of the matching pursuit. Our objective is not only to give a conceptual understanding to matching pursuit theory, but also to reflect on its implications for the video compression scheme. The adaptation of matching pursuit for video coding is described in details. In addition, important elements of the matching pursuit, like the dictionary and the searching process, are presented in separated sections. Finally, the implementations of the MP for the AVC video codec are presented. Chapter 2 contains the reference results for further comparisons to the proposed solutions.

Chapters 3 through 6 present the author's own contributions to the very low bit-rate video coding using matching pursuit. Chapter 3 deals with foundations for the new scheme of matching pursuit, which has been called matching pursuit with separable decomposition (MPwithSD). The separable decomposition, i.e. the key element of the novel solution is also described in this chapter. In addition, similarly as in the previous chapter, some kernel properties of the matching pursuit with separable decomposition are discussed in separate topics.

Chapter 4 proposes the video encoder based on the matching pursuit with separable decomposition. The verification of the main thesis of the dissertation takes place in this chapter. Examples of numerous simulation experiments using test sequences are given.

Further experiments and novel learning scheme for designing dictionary are presented in Chapter 5. This chapter contains also the conclusions concerning the universal and separable dictionaries. In addition, the dictionary adaptations are pointed and partially discussed. The example of dynamic adaptation performed at stage of region of approximation is separately presented in Chapter 6.

Chapter 2

Hybrid Video Compression and Matching Pursuit

2.1. Introduction

This chapter briefly describes both the general idea and the selected methods of contemporary hybrid video compression systems. It is worth noting that the name "hybrid" refers to many techniques that are combined to obtain a common purpose. In fact, two different types of redundancy are exploited into two separate stages. The temporal redundancy is exploited using inter-frame motion-compensated prediction. The spatial redundancy of the prediction error is exploited by the inter-frame process. The intra-frame coding stage of the motion residual is within the scope of the dissertation. In addition, this chapter shortly presents the key features of the H.264/AVC codec.

The last part of the chapter deals with video compression using matching pursuit. A dictionary and procedure of atom searching is described. In addition, many auxiliary techniques and solutions are presented in detail.

Some sections contain author's own improvements to the matching pursuit algorithm. Moreover, the reference environments based on two different H.264/AVC implementations are described in detail.

2.2. Hybrid Block-Based Video Coding

The most of the current video coding standards are based on two principles:

- block-based coding,
- hybrid structure.

The first principle means that an input sequence is divided into blocks which size is much smaller than the input image. As a result, each frame is considered as a sequence of independent blocks. Each block is separately coded using the selected methods.

The latter principle means that temporal and spatial redundancies are exploited in two very different methods. The first compression method exploits temporal redundancy between frames. It is assumed that it is possible to predict current frame based on the previous frame and motion information. Motion information must be estimated first based on two consecutive frames and such process is called motion estimation. The predicted frame is generated through the process called motion compensation. Both processes are described in the next section. On the other hand, spatial compression exploits spatial correlation within frame. Pixels in blocks are decorrelated through the Discrete Cosine Transform (or other similar transform), which packs most energy into as few coefficients as possible in the low frequency region. After spatial decorrelation, the transform coefficients are quantized. As a result, most of the coefficients have values that are equal to zero (or very close to zero). Next, the obtained approximation of the transform is effectively encoded. The transform coding with DCT is described in Section 2.2.2.

The assumed principles enforce the general scheme of existing video compression systems (and standards like MPEG-1/2/4 (see: [MPEG-1], [MPEG-2], [MPEG-4a], [MPEG-4b]), H.261 (see: [H.261]), H.263 (see: [H.263]), AVC (see: [H.264])). The video encoder employs two basic techniques: block-based motion compensation and block-based transform coding (see Fig. 2. 1). Motion compensation technique can be applied in both the forward and backward direction. The remaining signal i.e. prediction error, is transformed with the DCT and then coded using a run length coding (RLC) and variable length coding

(VLC) or arithmetic coding successively. The motion predictors, called motion vectors, are transmitted together with the spatial information. The video decoder decodes entropy-coded parameters received from an input bit-stream. Then, the motion compensation is performed to obtain the prediction of current frame. The frame prediction is supplemented by the approximation of the prediction error, which is represented by quantized DCT coefficients.

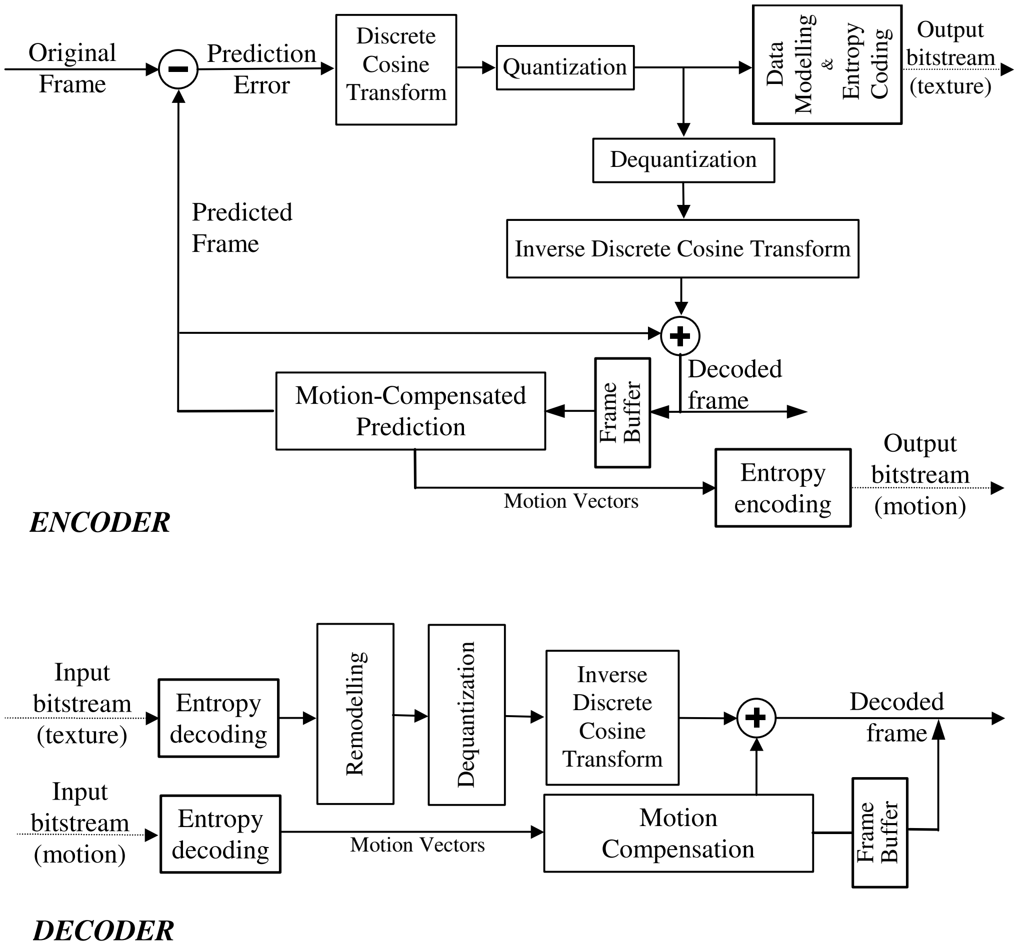


Fig. 2. 1. Scheme of DCT-base motion-compensated codec.

It is worth mentioning that the video standards do not specify an encoding process. Instead, they only specify binary formats for representing an input data to a decoder and a decoding process. Therefore, every decoder which is compliant with the standard, should be able to understand the syntax of an incoming bitstream and decode it. This decoder-only specification provides enough flexibility for manufacturers to design encoders of different complexities for different applications. Moreover, even after the standards are established, manufacturers can still continually improve and optimise decoding implementation algorithms or specific elements in a decoder if these improvements comply with the semantics defined the standard. Foremost, the standards that are defined in this way give a great degree of liberty in designing algorithms for encoding process (e.g. for motion estimation).

2.2.1. Motion Compensation

Motion-compensated prediction assumes translational model of motion for small image blocks from one video frame to another [Kell29], [Taki74], [Jain81]. The motion of every block is described using a motion vector, which is applied to all pixels inside the block. The prediction cannot be based on a source frame because the prediction has to be repeatable in the decoder, where the source images are not available (the decoded frames are not identical to the source frames because the bit rate reduction process introduces distortions into the decoded picture). Consequently, the encoder contains a local decoder, which reconstructs pictures exactly as they would be in the decoder, from which predictions can be formed.

As mentioned above, the motion vectors are calculated by the motion estimation step. In addition, the motion estimation is applied to luminance signal only. The process of motion detection is complex and computationally intensive. Generally, one can devise two classes of motion detection algorithms. The first broad class is based on optical flow, more properly called gradient-based methods [Horn81]. Optical flow algorithms determine translations between images from the estimates of spatial and temporal derivatives of brightness [Nagel95].

The second class is based on the block matching [Jain81]. In this method, the current frame is divided into blocks of size $N_1 \times N_2$, and the motion vector $< v_{v,} v_h >$ is obtained by minimizing a cost function D measuring the mismatch between the reference and the current block.

$$D(n_v, n_h, t, v_v, v_h) = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} C(f_{original}(n_v + i, n_h + j, t), f_{decoded}(n_v + v_v + i, n_h + v_h + j, t-1)), \quad (2.1)$$

where :

$f_{original}(y, x, t)$ -value of pixel in t-th original frame at point (y,x),

$f_{decoded}(y, x, t)$ -value of pixel in t-th decoded frame at point (y,x),

(n_h, n_v) -coordinate of the upper-left corner of the current block,

(v_h, v_v) -motion vector,

N_1, N_2 -size of block. Depending on the standard, the size of block may obtain different values:

- $N_1 = N_2 = 16$ in MPEG-1, H.261,
- $N_1 = N_2 = 16$ or $N_1 = N_2 = 8$ in MPEG-2, H.263,
- $N_1 \in \{8,16\}, N_2 \in \{8,16\}$ or $N_1 \in \{4,8\}, N_2 \in \{4,8\}$ in H.264/AVC.

$C(a,b)$ -cost function. The cost function can take different forms [Hask97], [Tekalp04]:

- $C(a,b) = |a - b|$ called MAE (Mean Absolute Error)/ MAD (Mean Absolute Difference) [Lin88],[Jaur90],
- $C(a,b) = (a - b)^2$ called MSE (Mean Square Error)/MSD (Mean Square Difference) [Ahma90],
- $C(a,b) = \begin{cases} 0 & if\ |a-b| \le T_h \\ 1 & if\ |a-b| > T_h \end{cases}$

  where T_h means a certain threshold. The cost function is called MPC (Mean Pel Count) / PDC (Pel Difference Classification) [Chan94].

Although any cost function can be used, the most widely used choice is the mean absolute difference (MAD / MAE) [TM5], [VM8], [TML4], [TML8], [JM8], [VC-1].

In order to find the best matching block producing the minimal mismatch error, one needs to calculate the cost function D at all locations in the search range i.e. for:

$$-R_h \le v_h \le R_h,$$
$$-R_v \le v_v \le R_v,$$

where:

R_h, R_v - horizontal and vertical range for the block matching method.

The above strategy, known as the *full search* or *exhaustive search*, is conceptually the simplest one, but simultaneously the most compute-intensive one. There are many techniques that significantly reduce the computational complexity with a small decrease in efficiency. Nevertheless, despite existing so-called fast methods, this process takes over 50% of the computational power of encoding process. Therefore, the alternative methods to the exhaustive search are important and worth to mention.

Lower computational complexity has *logarithmic search* [Jain81], [Hask97], which assumes $R_v = R_h = M = 2^k$. This algorithm first evaluates D at the centre and eight surrounding locations with distant *M*. The location that produces the smallest value of cost function becomes the centre of the next stage. Then, the search range is half reduced and the sequence is repeated *k* times to reach presumed exactness. (Often, such algorithms are called Three-Step-Search (TSS), [Koga81], [Li94] or Four-Step-Search (FSS/4SS) [Po96] depending on the number of steps. Similar concept is used in such fast motion estimation algorithms as: diamond search (DS) [Zhu00] or hexagon-based search [Zhu02]).

Another method that computes motion vector is *hierarchical search* [Barb02]. This technique uses an exhaustive search for *k*-times decimated frame. This trick reduces both the size of the matched block and the range of search. In the next step, the full search is performed for (*k-1*)-times decimated frame in local area of previously found location.

It should be noted that the subsequent improvement of the motion-compensated prediction techniques was the major reason for coding efficiency improvements achieved by modern standards when comparing them from generation to generation. The price for the use of the motion-compensated prediction in more sophisticated ways was always the same, i.e. increase in complexity requirements. Nevertheless, rapid development of VLSI technology led to significant increase of computational power. As a result, new aspects were involved with the motion-compensated prediction process. These options are:

- *Fractional-sample accuracy* [Broff77]. This term refers to the use of spatial displacement of motion vector that has more than integer precision. A theoretical motivation for this can be found in [Girod87], [Girod93]. Intuitive reasons include having a more accurate motion representation. Half-sample accuracy was considered during the design of H.261 [H.261] but was not included due to the complexity limits of the time. Later, as processing power increased and algorithm designs improved, video codec standards increased the precision of motion vector support from half-sample (in MPEG-1, MPEG-2, and H.263) to quarter-sample (for luminance in MPEG-4's advanced simple profile and H.264/AVC) and beyond (with eighth-sample accuracy used for chroma in H.264/AVC).
- *Variable block size* [Sull91a]. This term refers to the ability to select the size of the block (ordinarily a rectangular block-shaped region) for motion estimation. Intuitively, this provides the ability to effective trade off between the accuracy of the motion field and the number of bits needed for representing motion vector.

- *Bi-predictive motion-compensated prediction* [Hidaka89]. This term refers to the averaging of two MCP signals. Bi-predictive MCP was first put in a standard in MPEG-1, and it has been present in all other succeeding standards. Intuitively, such bi-predictive MCP particularly helps when the scene contains uncovered regions or smooth and consistent motion.
- *Multi-picture motion-compensated prediction* [Wieg99], [Wieg01]. This term refers to the using more than just one or two previous decoded pictures. This allows the exploitation of long-term statistical dependencies in video sequences, such as backgrounds, scene cuts, and textures with aliasing shown earlier in a sequence.
- *Motion vectors over picture boundaries* [Sull91b]. The approach solves the problem for motion representation for samples at the boundary of a picture by extrapolating the reference picture. The most common method is just to replicate the boundary samples for extrapolation. This method was standardized in [H.263].

2.2.2. Prediction Error Coding

In the compression of video sequences, the most important thing is the decorrelation of a signal, which is -except edges- spatially correlated. It is well known that the optimum Karhunen-Loève Transform (KLT) [Hotell33], [Joll86] can efficiently decorrelate pixels spatially and pack most energy in the fewest coefficients. However, KLT is not a fixed transform and can only be determined on the basis of the statistical group of frame regions. Therefore, the core of KLT must be calculated at the encoder and sent to the decoder along with the transform coefficients. The computation complexity of KLT makes this solution unattractive, especially if we take into consideration that the discrete cosine transform (DCT) gives similar decorrelation for signals such as natural images.

In many hybrid codecs [Tekalp95], [MPEG-1], [MPEG-2], [MPEG-4], [H.261], [H.263] the two-dimensional discrete cosine transform is used for decorrelation of both the current frame and the prediction error. The 2-D DCT is defined as follows:

$$F(m,n) = \frac{2}{N} C(m) C(n) \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} f(k,l) \cos\left(\frac{\pi m(2k+1)}{2N}\right) \cos\left(\frac{\pi n(2l+1)}{2N}\right), \qquad (2.1)$$

where:

$$C(v) = \begin{cases} \frac{1}{\sqrt{2}} & for \ v = 0, \\ 1 & else, \end{cases}$$

$F(m,n)$ - transform coefficient,

$f(k,l)$ - 2-D input signal sample,

k,l - spatial samples,

m,n - samples of spatial frequencies.

The transform is performed on small blocks and most compression standards use blocks of size 8×8 pixels. The DCT does not directly reduce the number of bits required to represent the block. In fact, for an 8×8 block of 8-bit samples, the DCT produces an 8×8 block of 11-bit coefficients (the range of coefficient values is larger than the range of pixel values). The reduction in the number of bits follows from the observation that, for typical blocks from natural images, the distribution of coefficients values is non-uniform. The transform tends to concentrate the energy into the low-frequency coefficients and many of the other coefficients are close to zero. The bit-rate reduction is achieved by quantizing and coding the remaining coefficients.

In many standards, a uniform quantizer with a different step size for each DCT coefficient is used (e.g. MPEG-2, H.263). Since the subjective perception of the quantization error varies with the frequency, higher frequency coefficients are quantized more coarsely. In addition, different quantization matrices are used for intra-coded and inter-coded blocks, since the signal from intra-coding has a different statistic. Intra-coded blocks contain energy in all frequencies and are likely to produce blocking artefacts if too coarsely quantized. On the other hand, blocks in the displaced frame difference contain predominantly high frequencies and can be subject to much coarser quantization.

Further processes, i.e. data modelling and entropy coding, lead to efficient compression of quantized coefficients (Fig. 2. 4). The quantized DCT coefficients are rearranged into a one-dimensional array by scanning them in a zig-zag order (Fig. 2. 2) or other alternative scanning order (e.g. [MPEG-4] see Fig. 2. 3).

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Fig. 2. 2. Zig-zag order.

0	1	2	3	10	11	12	13
4	5	8	9	17	16	15	14
6	7	19	18	26	27	28	29
20	21	24	25	30	31	32	33
22	23	34	35	42	43	44	45
36	37	40	41	46	47	48	49
38	39	50	51	56	57	58	59
52	53	54	55	60	61	62	63

0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

Fig. 2. 3. Alternative scan orders considering horizontal and vertical properties of transform.

Such orders exploit the fact that most of the non-zero coefficients are in the low frequencies, thus, they basically cluster the non-zero coefficients at the beginning and zero coefficients at the end. The rearranged array is coded into a sequence of the run-level pairs. Finally, the run-level pairs are encoded using entropy coding technique such as variable-length coding or arithmetic coding.

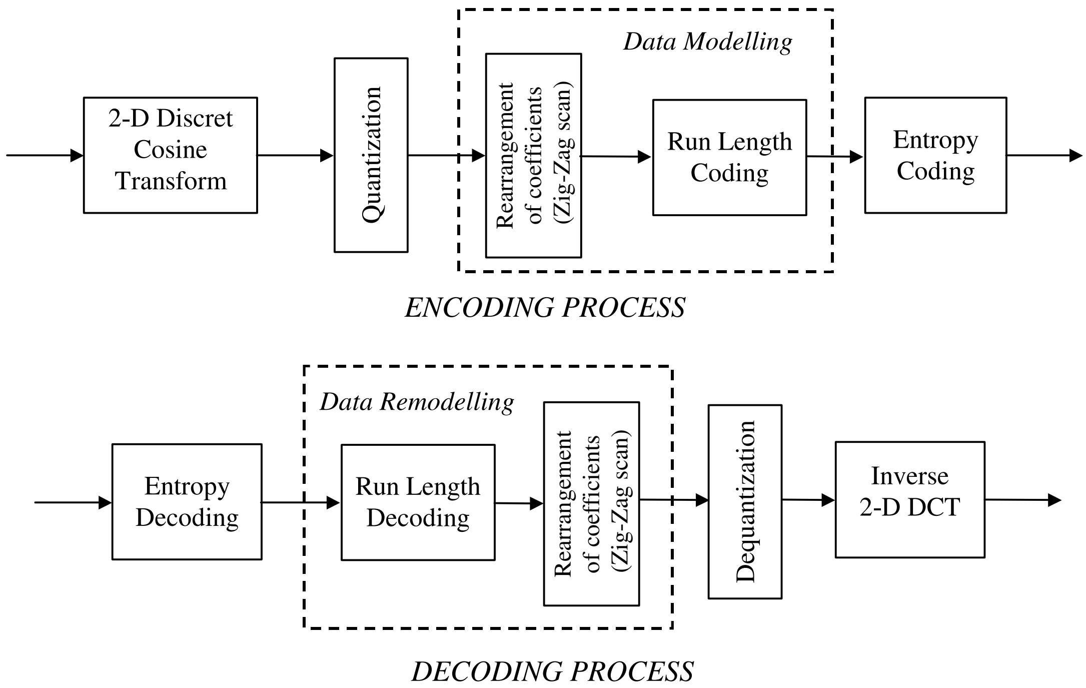


Fig. 2. 4. Scheme of a DCT-based codec.

The reconstruction of a signal consists of the inversion of the encoding process. The one-dimensional array is obtained from an input bit-stream using entropy decoding and run-length decoding. The two-dimensional matrix of quantized transform coefficients is gained as the result of zig-zag process. The approximation of the source discrete cosine transform comes into existence after dequantization. Finally, IDCT reconstructs an input signal. The 2-D Inverse Discrete Cosine Transform is defined as follows:

$$f(k,l) = \frac{2}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} C(m)C(n)F(m,n)\cos\left(\frac{\pi k(2m+1)}{2N}\right)\cos\left(\frac{\pi l(2n+1)}{2N}\right) \qquad (2.2)$$

(symbols are described at DCT) .

The DCT is commonly used in transform coding of images and video, because it is a close approximation to the statistically-optimal Karhunen-Loève transform, for a wide class of signals. Nevertheless, one disadvantage of the DCT is that its core contains real numbers. As the result, the transform coefficients have approximate representation in set of computer floating-point numbers. In a digital processing, when the direct and inverse transform is computed in a cascade, it is impossible to get the source data back. Moreover, in a motion-compensated video encoder, past decoded frames are used as reference information for prediction of the current frame. Therefore, if the encoder and the decoder use different floating-point formats, then the computed signals are also different. This leads to drift between the decoded data at the decoder and encoder. One solution to the data drift problem is to approximate the core transform by a matrix containing only integer numbers.

Integer approximations to the DCT should preserve the symmetries in the rows [Cham89]. In addition, the norms for the rows should be equal. Such approximation can be obtained from general formula [Rao90]:

$$Q(i,j) = NINT(\lambda \cdot H), \tag{2.3}$$

where:

$\lambda \in \mathbb{R}$ -scaling parameter,

$$\mathbf{H} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ c & s & -s & -c \\ 1 & -1 & -1 & 1 \\ s & -c & c & -s \end{bmatrix},$$

$c = \sqrt{2}\cos(\pi/8)$,

$s = \sqrt{2}\sin(\pi/8)$.

Integer approximation to the DCT is used by the newest video standard i.e. the H.264/AVC. It worth to mention that the transform matrix in the H.26L drafts (up to version 8 [TML8]) was obtained for $\lambda = 26$ (this value gets orthogonal matrix with equal-norm rows). Finally, due to Malvar's researches [Malvar01], the H.264/AVC standard uses the integer matrix generated for $\lambda = 2.5$. The main advantage of the Malvar's transform is very fast implementation and negligibly small loss in performance [Xin04].

2.2.3. The H.264/AVC Advanced Video Coding

The concepts of H.264/AVC are very similar to established standards like H.261, H.263, and MPEG-1/2/4. The standard is based on hybrid coding scheme, i.e. motion between frames of the sequence is predicted using motion vectors, and the prediction error is then transformed, quantized, and transmitted. There is no single coding element that provides the majority of the significant improvement in compression efficiency in relation to prior video coding standards. It is rather a plurality of smaller improvements that add up to the significant gain. Some of the improvements of the H.264/AVC are described below and depictured on Fig. 2. 5:

- *Variable block-size motion compensation with small block sizes* [Girod87], [Sull91a]. This standard supports more flexibility in the selection of motion compensation block sizes and shapes than any previous standard, with a minimum motion compensation block size as small as 4×4.
- *Quarter-sample-accurate motion compensation* [Chen02], [Girod93]. Most former standards provide half-sample motion vector accuracy at most. Although, quarter-sample motion vector accuracy was used in an advanced profile of the MPEG-4 Visual (part 2) standard, but in the H.264/AVC the complexity of the interpolation processing was reduced.
- *Multiple reference picture motion compensation* [Wieg99], [Wieg01]. Predictively coded pictures (called P-pictures) in MPEG-2 [MPEG-2] and its predecessors used only one previous picture to predict the values in an incoming picture. The new design extends upon the enhanced reference picture selection technique found in H.263++ [H.263] to enable efficient coding by allowing an encoder to select, for motion compensation purposes, among a larger number of pictures that have been decoded and stored in the decoder. The same extension of referencing capability is also applied to motion-compensated bi-prediction, which is restricted in MPEG-2 to using two specific pictures.
- *Decoupling of referencing order from display order* [Wieg01]. In prior standards, there was a strict dependency between the ordering of pictures for motion compensation referencing purposes and the ordering of pictures for display purposes. In H.264/AVC, these restrictions are largely removed, allowing the encoder to choose the ordering of pictures for referencing and display purposes with a high degree of flexibility constrained only by a total memory capacity bound imposed to ensure decoding ability. Removal of the restriction also enables removing the extra delay previously associated with bi-predictive coding.
- *Weighted prediction* [Wieg03], [Boyce04]. A new innovation in H.264/AVC allows the motion-compensated prediction signal to be weighted and offset by amounts specified by the encoder. This can dramatically improve coding efficiency for scenes containing fades, and can be used flexibly for other purposes as well.
- *In-the-loop deblocking filtering* [Hong01], [List03]. Block-based video coding produces artefacts known as blocking artefacts. Application of an adaptive deblocking filter is a well-known method of improving the resulting video quality, and when

designed well, this can improve both objective and subjective video quality. Building further on a concept from an optional feature of H.263+, the deblocking filter in the H.264/AVC is brought within the motion-compensated prediction loop, so that this improvement in quality can be used in inter-picture prediction to improve the ability to predict other pictures as well.

- *Directional spatial prediction for intra coding* [Meng03], [Pan03]. Extrapolating the edges of the previously-decoded parts of the current picture is applied in regions of pictures that are coded as intra. This improves the quality of the prediction signal, and also allows prediction from neighbouring areas that were not coded using intra coding.
- *Small block-size transform* [Wien03]. All major prior video coding standards used a transform block size of 8x8, while the new H.264/AVC design is based primarily on a 4x4 transform. This allows the encoder to represent signals in a more locally adaptive fashion.
- *Exact-match inverse transform* [Malvar01]. In previous video coding standards, the core of transform contains floating (pseudo-real) numbers. Due to this, it was very difficult to obtain an exact match to the ideal specified inverse transform. As a result, each decoder design would produce slightly different decoded video, causing a drift between encoder and decoder representation of the video and reducing effective video quality. The H.264/AVC is the first standard to achieve exact equality of decoded video content from all decoders.
- *Short word-length transform* [Malvar01]. All former standards used complex processing for transform computation and generally required at least 32-bit processing. The H.264/AVC design requires only 16-bit arithmetic. Moreover, the calculation of transform does not require a multiplication operation.
- *Arithmetic entropy coding* [Moffat95], [Marpe01b], [Wieg03], [Marpe03]. An advanced entropy coding method known as arithmetic coding is included in H.264/AVC. While arithmetic coding was previously found as an optional feature of H.263, a more effective use of this technique is found in H.264/AVC to create a very powerful entropy coding method known as CABAC (context-adaptive binary arithmetic coding).
- *Context-adaptive entropy coding* [Bj02], [Marpe01a], [Wieg03]. There are two entropy coding methods applied in H.264/AVC: CAVLC (context-adaptive variable-

length coding) and CABAC (context-adaptive binary arithmetic coding). Both methods use context-based adaptation to improve performance relative to prior standards.

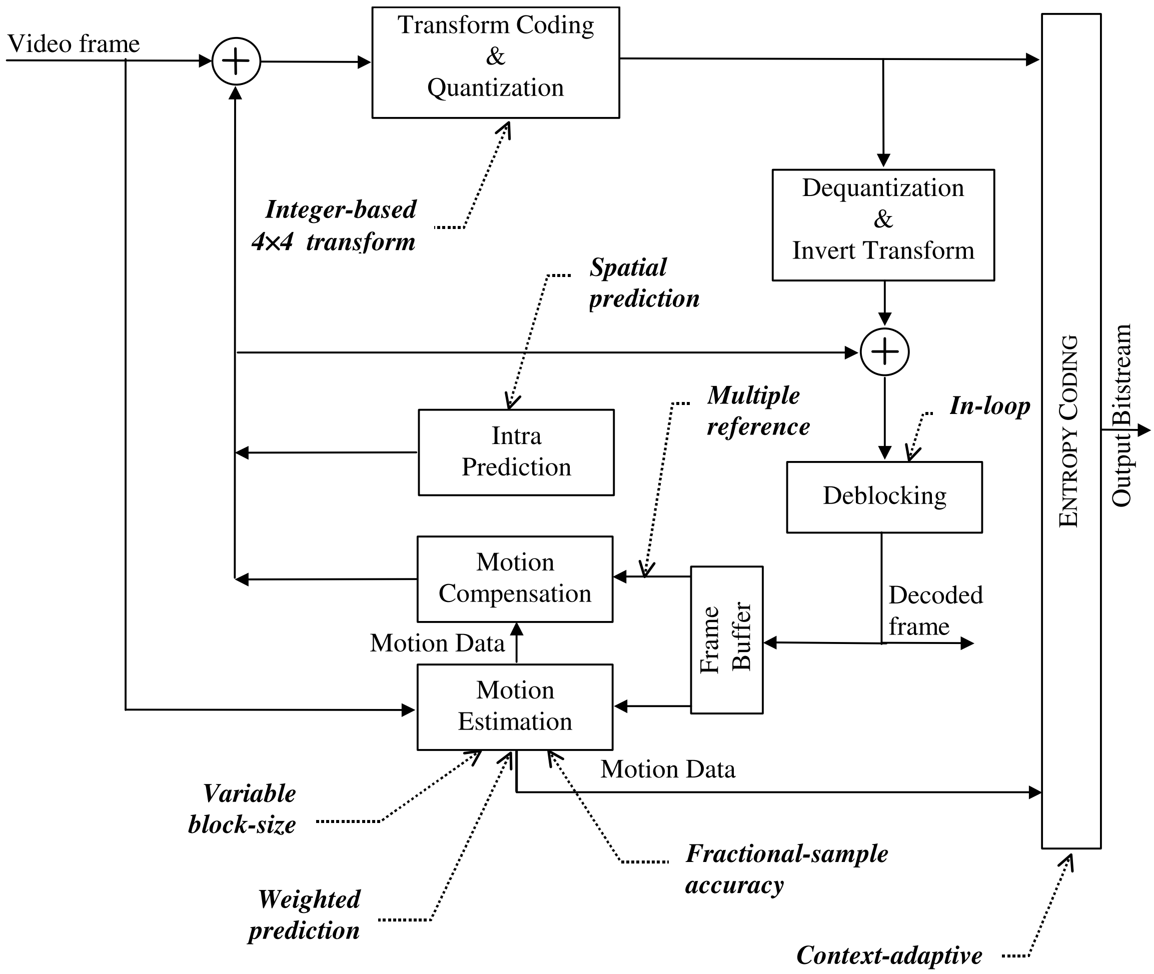


Fig. 2. 5. The H.264/AVC encoder.

2.3. The Basics of Matching Pursuit

This section presents the foundations of matching pursuit. The *M-optimal approximation* problem briefly presented in the section is described not only to show the genesis of the matching pursuit but also to justify the computational load of matching pursuit.

2.3.1. Optimal Approximation

Let us consider N-dimensional Hilbert space $\mathcal{H}$. Let $\mathcal{D} = \{\varphi_k\}_{k=1}^{L}$, $L \geq N$ be a dictionary i.e. at least complete set of unit vectors in $\mathcal{H}$. It is known [Daub91], [Davis94], [Goyal95] that any vector $f \in \mathcal{H}$ can be represented as a linear combination of elements of $\mathcal{D}$:

$$f = \sum_{k=1}^{L} \alpha_k \varphi_k \ , \tag{2.4}$$

where $\alpha_k \in \mathbb{R}$.

The smallest possible dictionary is a basis of $\mathcal{H}$ and such dictionary guarantees the unique representation. Nevertheless, in general, the dictionary $\mathcal{D}$ is overcomplete and defines set of non-orthogonal vectors. Thus, vectors in $\mathcal{H}$ do not have unique representation as linear combination of vectors from $\mathcal{D}$ [Davis94], [Nafi96].

Let $\varepsilon \in \mathbb{R}^+$ and $0 < M \leq N$. For a given $f \in \mathcal{H}$ an (ε, M)*-approximation* is an expansion,

$$\tilde{f} = \sum_{i=1}^{M} \alpha_k \varphi_{k_i} \ , \tag{2.5}$$

for which,

$$\left\| \tilde{f} - f \right\| < \varepsilon . \tag{2.6}$$

An expansion (2.5) that for a given $f \in \mathcal{H}$ and M minimizes $\left\| \tilde{f} - f \right\|$ is called *M-optimal approximation*.

It has been proved [Davis94], that for any $\varepsilon > 0$, $1 < M \leq \mathbb{N}$, determining whether (ε, M)-approximation exists is NP-complete. In addition, finding the *M*-optimal approximation is NP-hard [Davis94]. The intractability of *M*-optimal approximation results from the number of possible choices of dictionary functions. The complexity can be reduced if the dictionary elements are chosen one at a time instead of *M* at once. This reduction of a basic problem to simpler problems is the defining characteristic of a greedy algorithm. Matching pursuit, introduced by Mallat and Zhang [Mall93], is a greedy algorithm for finding approximate solutions to the *M*-optimal approximation problem.

Finally, it worth noticing that Hilbert space is used by many branches of science (e.g. functional analysis, quantum mechanics). In general, the elements of Hilbert space are called "vectors". Nevertheless, in applications, they are typically sequences of numbers [Neff95], [Durka96], [Heus02] or functions [Mall93], [Davis94], [Goyal95a] or wave-functions [Neuma55]. As the result, in this dissertation the phrases "vector" and "function" are used exchangeably. In addition, the elements of dictionary are sometimes called waveforms (see Chapter 2.4.1).

2.3.2. The Theory of Matching Pursuit

Matching pursuit algorithm [Mall93], [Neff95], [Goyal95b], [Durka96], [Vlee98], [Fross01] iteratively decomposes any function f of the N-dimensional Hilbert space $\mathcal{H}$, into linear combination of non-orthogonal functions chosen from overcomplete dictionary $\mathcal{D} = \{\varphi_k\}_{k=1}^{L}$, $\varphi_k \in \mathcal{H}$ and $L \geq N$. Additionally, it is imposed the normality of functions i.e. $\|\varphi_k\| = 1$.

The algorithm attempts to approximate an input signal as closely as possible in a greedy manner at each step. Firstly, the function f is decomposed as follows:

$$f = \left\langle \varphi_{k_1}, f \right\rangle \varphi_{k_1} + R_1 f = \alpha_1 \varphi_{k_1} + R_1 f \ , \qquad (2.7)$$

where:

k_1 - index of function from dictionary,

$R_1 f$ - residual signal,

$\alpha_1 = \left\langle \varphi_{k_1}, f \right\rangle$ - expansion coefficient.

The index k_1 is chosen in such way that the absolute value of the inner product $\left|\left\langle \varphi_{k_1}, f \right\rangle\right|$ is maximal. Due to this, the Euclidean norm of the residual function is minimal.

In the next step, the residual $R_1 f$ is expressed in the same way as the original signal f. The algorithm continues until either a predefined number of expansion coefficients is exceeded or the norm of the residual exceeds a predefined threshold. Each step yields a dictionary function $\varphi_{k_i} \in \mathcal{D}$, an expansion coefficient α_i and a residual $R_i f$, which is an input to the next stage of the procedure. After the *M* steps, the given signal f is approximated by the linear combination of the dictionary elements as follows:

$$f = \sum_{i=1}^{M} \alpha_i \varphi_{k_i} + R_M f \ . \qquad (2.8)$$

It is worth noticing, that $R_i f$ is orthogonal to φ_{k_i} and this leads to the principle of energy conservation,

$$\left\|R_{i-1}f\right\|^2 = \left|\left\langle \varphi_{i_1}, f \right\rangle\right|^2 + \left\|R_i f\right\|^2, \tag{2.9}$$

or equivalently to:

$$\left\|f\right\|^2 = \sum_{i=1}^{M} \alpha_i^{\,2} + \left\|R_M f\right\|^2. \tag{2.10}$$

The above technique gives very useful properties in signal representation. An interesting feature of the matching pursuit is that it puts few restrictions on the dictionary. The original Mallat and Zhang paper [Mall93] considers Gabor function dictionaries for time-frequency analyses, but the algorithm itself does not require such limitation to Gabor structures. In fact, any collection of arbitrarily sized and shaped functions can be used, as long as completeness is satisfied.

One can mark out some properties that derive directly from the matching pursuit algorithm, this is:

- invertibility (if the dictionary is at least complete [Mall93]),
- energy conservation (see equation 2.9 [Mall93]),
- exponentially bounded error decay (implies fast initial error decay [Davis97], [Mall99], [Fross04]),
- robustness to quantization (results from the fact that the coding space is of higher dimension than the signal space [Goyal95a], [Davis97]).

These properties are very important because they show that the general behaviour of the algorithm is independent from a dictionary. Nevertheless, one should remember that the chosen set of functions plays a crucial role in matching pursuit coding, especially in very low bit rate coding systems.

2.4. Matching Pursuit Video Coding

An adaptation of the matching pursuit for video coding has been presented in [Neff95], [Neff97], [AlSh99] and extended [Neff02a], [Neff02b] - mainly by Neff and Zakhor. Simplified block diagrams of the encoder and decoder in which motion residual is coded using matching pursuit are shown in Fig. 2. 6. As can be seen, firstly, the current frame is predicted by the motion-compensated process using the previous reconstructed frame. Then, the prediction error is treated as an input signal for the matching pursuit block. An approximated displaced frame difference is added to a predicted frame forming -in that way- the final shape of reconstructed frame. Extracted atoms parameters are coded and sent to the decoder. The dictionary and the process of finding and coding atoms are described in detail in the next sections.

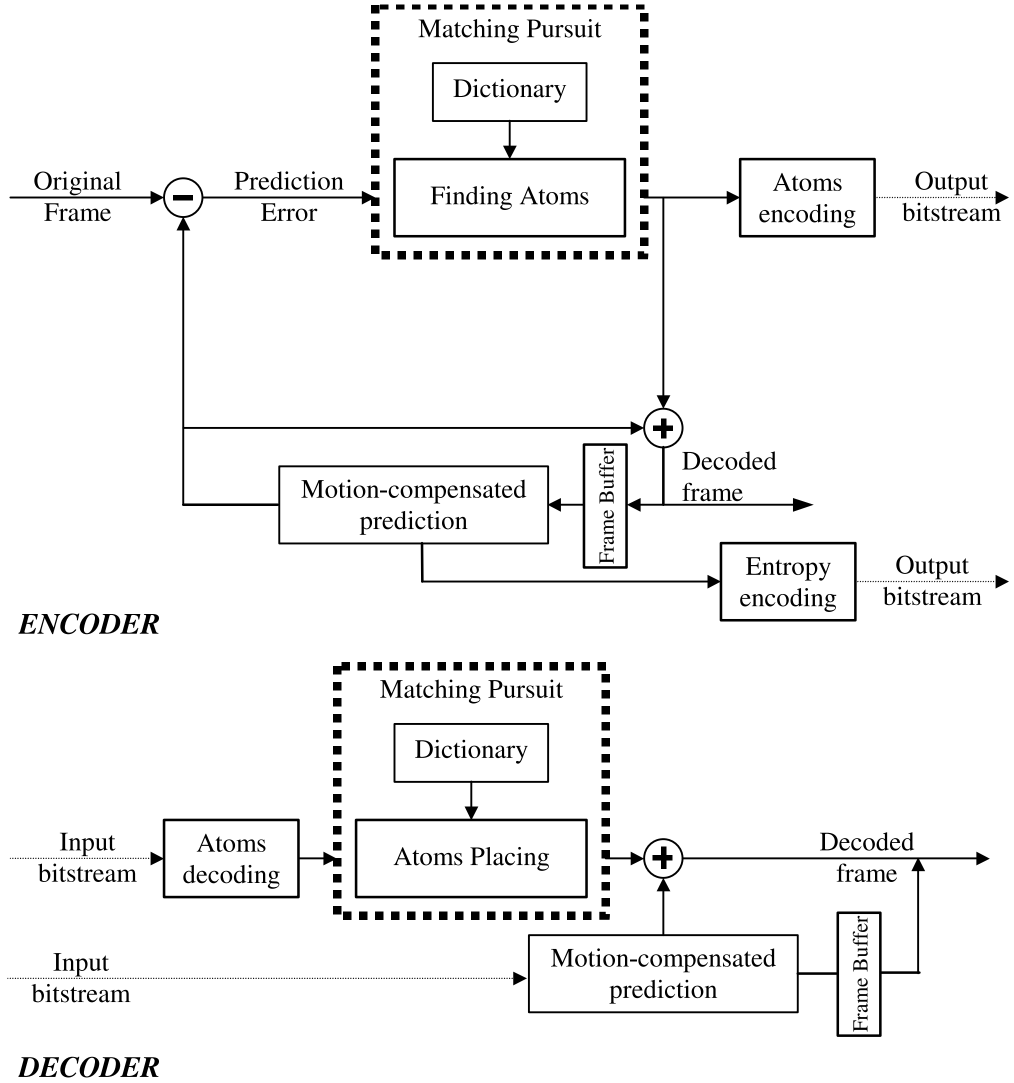


Fig. 2. 6. Scheme of video codec with matching pursuit.

2.4.1. The Dictionary

In theory of matching pursuit, it is assumed that an approximated input vector and vectors in a dictionary belong to the same Hilbert space. Such assumption simplifies the mathematical description of the matching pursuit algorithm and the approximation problem in theory of frames. Nevertheless, from practical point of view, this assumption is not necessary.

First of all, we can presume that vectors are supplemented by zeros and thus, transformed to the common Hilbert space. Alternatively, we can presume that space of one of vectors is limited in order to obtain the common Hilbert space. The later transformation is more practical since it allows considering vectors from lower dimensionality.

In real-world situations, the size of the input vector is much larger than the size of vectors in the dictionary. This means that the individual vector from a dictionary approximates small fragment of the input vector. Therefore, it is better to assume that the dictionary contains waveforms [Mall93], [Durka95], [Figu00] which are placed at any possible position within an input vector. Due to this, a waveform with the displacement is equivalent to the dictionary function in the theory of frames [Daub91], [Davis94], [Goyal95]. In the remainder of this dissertation, both definitions will be used exchangeably. Finally, the waveforms, along with the corresponding value of the inner product computed in a certain location, form a set of five parameters, as shown in Table 2.1. We say that these five parameters define an atom.

Table 2. 1. Parameters defining an atom.

	Practical point of view	**Theory of frames**
v_1, v_2	Indices of waveforms from dictionary.	$<v_1,v_2,x,y>$ defines a function in a Hilbert space for the theory of frames
x, y	Location of waveform within the prediction error	
α	Expansion/scale coefficient	Expansion/scale coefficient

Considering the fact that a prediction error is a two-dimensional discrete signal, it forces 2-D discrete functions in the dictionary. It must be repeated, that a fundamental problem of pure matching pursuit is the lack of feedback between an input signal and a dictionary, since the algorithm uses a dictionary *a priori*. This fact implies the great need for the designing of a universal dictionary.

The universal dictionary for video coding was proposed in [Neff96a], [Neff96b], [Neff97]. The authors took into consideration the computational load of the matching pursuit algorithm. As a result, they decided to put separable functions into the dictionary.

The universal dictionary consists of an overcomplete collection of 2-D non-orthogonal separable Gabor functions. Each function from the dictionary is directly based on two 1-D discrete Gabor functions, which are defined as follows:

$$\varphi_v(i) = K_v g\left(\frac{i - N/2 + 1}{s}\right)\cos\left(\frac{2\pi\xi\left(i - N/2 + 1\right)}{N} + \phi\right), \qquad (2.11)$$

where:

$i \in \{0,1,...N-1\}$

$g(x) = e^{-\pi x^2}$ - Gaussian-alike function,

$v = (s, \xi, \phi)$ -triple consisting respectively:

s -scale,

ξ -frequency,

ϕ -phase shift,

K_v - normalisation coefficient,

One can easily extend the above 1-D basis functions into 2-D ones in the following way:

$$\varphi_{v_1,v_2}(i,j) = \varphi_{v_1}(i)\varphi_{v_2}(j). \qquad (2.12)$$

The most commonly known dictionary of matching pursuit [Neff96a], [Neff96b], [AlSh99] is presented in details in Table 2. 2.

Table 2. 2. The definition of matching pursuit dictionary.

i	0	1	2	3	4	5	6	7	8	9
s_i	1	3	5	7	9	12	14	17	20	1.4
ξ_i	0	0	0	0	0	0	0	0	0	1
ϕ_i	0	0	0	0	0	0	0	0	0	π/2

i	10	11	12	13	14	15	16	17	18	19
s_i	5	12	16	20	4	4	8	4	4	4
ξ_i	1	1	1	1	2	3	3	4	2	4
ϕ_i	π/2	π/2	π/2	π/2	0	0	0	0	π/4	π/4

The above set of waveforms was constructed in the following manner [Neff95]. At first, a large set of parameterized Gabor functions was used to define a 2-D dictionary. Then, a set of residual images from training video sequences was decomposed by matching pursuit using a large dictionary. The subset of functions, which were most often selected by the matching pursuit algorithm were retained in the universal dictionary. We can say that they used quantitative criteria to build the universal dictionary.

Frossard and Vandergheynst in [Figu00] proposed a quite different learning rule. They observed that atoms that appear more often are usually the ones that come after a high number of iterations. Due to this, the energy they bring to the final result is very small. On the other hand, it is known that convergence speed and thus coding efficiency are strongly related to the choice of dictionary set [Mall93], [Mall99]. In a very low bit-rate video coding system where a prediction error is approximated using a small number of atoms, the most important aspect is the speed of convergence. Therefore, the dictionary containing waveforms that are more correlated with an input signal (i.e. that give larger absolute inner product values) implies faster convergence. Frossard and Vandergheynst used the above qualitative criteria to construct their dictionary. It is worth mentioning, that the universal dictionary created by Frossard was applied for approximation of still images. Nevertheless, the qualitative strategy [Figu00] has been presented as an alternative to the quantitative one [Neff95].

2.4.2. Locality of Approximation

Matching pursuit algorithm used to encode a prediction error should take profits from local concentration of energy of such signal. This fact has been considered during the construction of the universal dictionary (see Section 2.4.1). The waveforms chosen to the dictionary have small sizes and small region of support - just to approximate a single concentration of energy within a DFD.

Nevertheless, the matching pursuit requires examination of each 2-D dictionary structure at all possible locations in the prediction error and computes all of the resulting inner products. Considering up-to-date hardware productivity, it seems impossible to use this technique without simplifications. In order to understand the real size of the problem, let us consider an example. It is known that the complete basis set for a 176×144 QCIF image must contain 25344 basis functions. It is also easy to check, that video encoder based on 8×8 block-DCT satisfies this criteria, since a prediction error is divided on 396 distinguish blocks and each block is represented using 64 cosine functions. The calculation of all DCT coefficients requires performing 25344 inner products. The universal dictionary presented in the previous section contains 400 waveforms, thus, in order to find the best fit, we would have to compute over 10 million inner products $(400 \cdot 176 \cdot 144 = 10137600)$. The number of inner products results from the fact that each waveform from dictionary is positioned at each possible location within a residual image.

As it has been mentioned above, the displaced frame difference signal is characterised by a local concentrations of energy and this means that single step of matching pursuit algorithm can be limited to its certain region. Due to this, the prediction error can be pre-scanned for high-energy packet in order to select sub-region of a DFD and to reduce the computational intensity. This strategy is based on the assumption that the atom selected in the region which concentrates most of the energy is the best possible atom within residual image. The above strategy has been verified by Neff [Neff97].

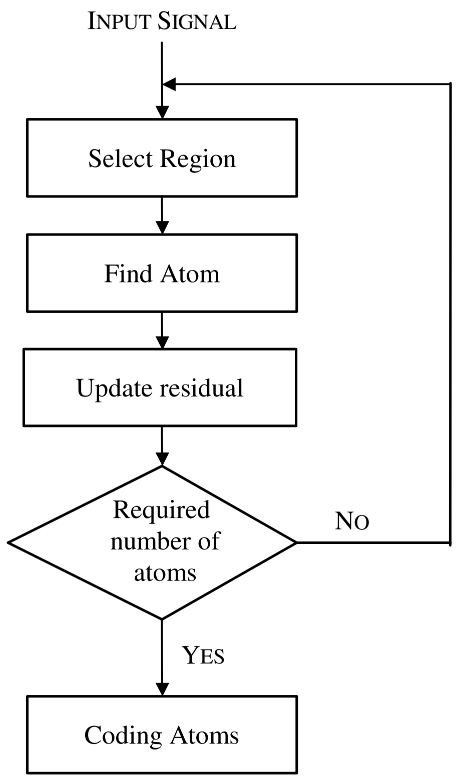


Fig. 2. 7. Matching pursuit encoding process.

The matching pursuit encoding process with step that limits the prediction error is depicted in Fig. 2. 7. In the first step, a location to encode is found. For this purpose, a 12×12 overlapped window is used to localize a region with the largest energy. The centre of the block with the largest energy value is adopted as an initial estimate of the inner product search. Then, each dictionary waveform is exhaustively matched to 24×24 window around the found centre (Fig. 2. 8). The largest absolute inner product value, along with its location and the function defines the atom. The residual is updated using the found atom. If the required number of atoms is not achieved the process is repeated.

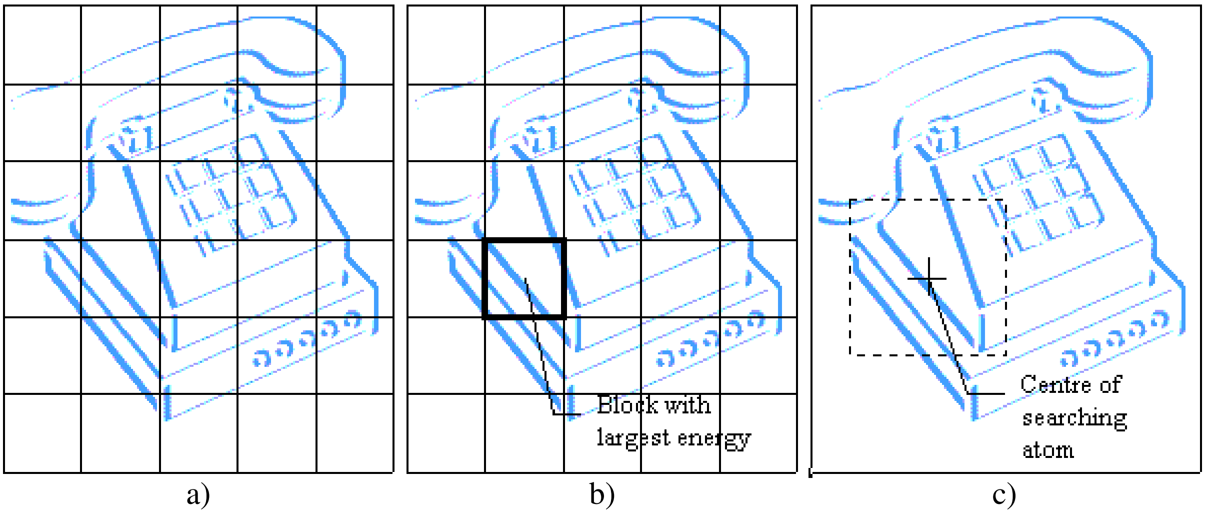


Fig. 2. 8. Algorithm providing locality of approximation: energy measurement (a), block searching (b) and selection of centre of atom search.

In [Banh97], Banham and Brailean have modified the above search strategy in such a way that the blocks closer to the centre of the prediction error are more likely to be chosen for exhaustive search. The strategy has been based on an assumption that more interesting information from a human point of view is located in the centre of an image.

A much more sophisticated method has been described in [AlSh99]. In this search strategy, each block is associated with the number of visits. The incremental counter of visits influences on the weight, which decreases the importance of energy contained in the block. In this way, the permanent selection of a poorly fitted region is avoided. It is worth noticing that the number of visits does not refer strictly to the atoms located in this block. In fact, it is possible that certain block is never selected as a centre of search, but it contains a few atoms - and vice versa.

2.4.3. Locality of Approximation. Experiments and Improvements

The presented strategies [Neff95], [AlSh99] that utilise a local approximation have been verified by the author of this dissertation in implementation based on the H.264/AVC encoder (Section 2.6.1). The experiments have shown that different methods give different subjective and objective results. This fact caused that the author took into investigation this problem and finally proposed new strategy. The author noticed that some regions of a prediction error were approximated using a few atoms that overlap each other. This was in the opposition to the assumption that a concentration of energy should be represented using a single atom. Furthermore, the author assumed that if the region demands a representation using a few atoms, this approximation should be performed in successive frames. Therefore, the main idea of the novel strategy is based on avoiding the regions that have been already chosen.

The novel strategy divides the prediction error on overlapping 12×12 pixels blocks as previous methods do. Each block is characterised by its energy and a penalty coefficient. The value of the penalty coefficient refers to the number of iterations on which the block will be discarded from the approximation process. At the beginning, all blocks have the penalty coefficient of zero, and this means that all blocks are considered. A selection of a block with the highest energy value changes its penalty coefficient value to the $T_{penalty}$. It means that the selected block will be disabled for the $T_{penalty}$ successive iterations. As experiments have shown, the objective quality criteria increases, if the global value $T_{penalty}$ is increased (Fig. 2. 9).

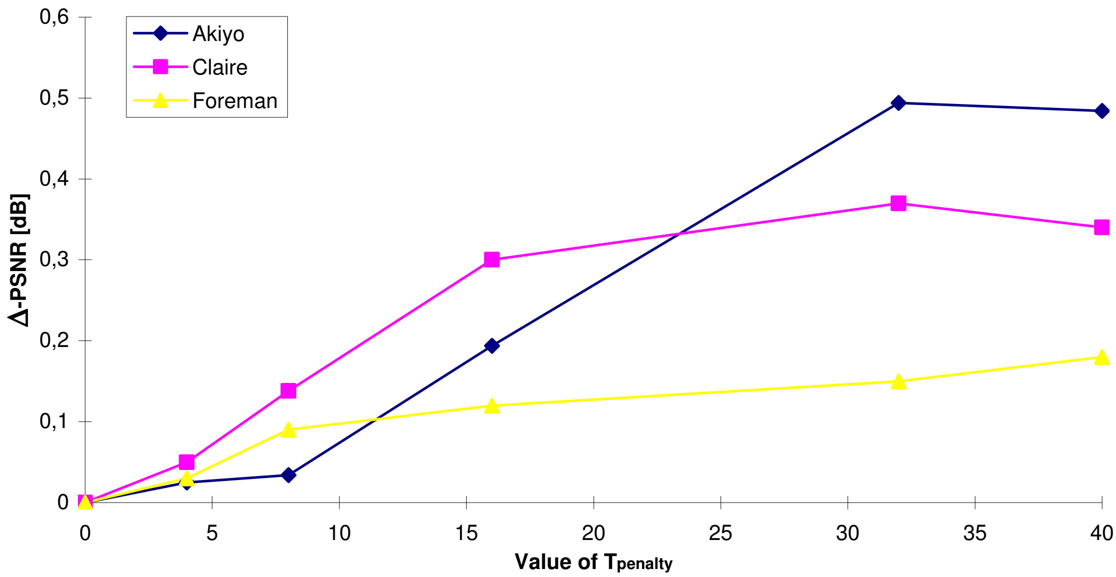


Fig. 2. 9. Dependency of average Y-PSNR on $T_{penalty}$ in the proposed process of selection of approximation region.

Experiments were performed on standard QCIF sequences: Akiyo, Claire and Foreman. For the simplicity, all inter-coded frames in each sequence were encoded using 40 atoms. The sequences were encoded for $T_{penalty} = \{0,4,8,16,32,40\}$.

The trustworthiest trend reflects sequence Foreman, since almost a whole scene is changed from frame to frame. Akiyo and Claire contain static backgrounds therefore small improvement in the background at the beginning of the sequence results in average Y-PSNR. The proposed by the author method outperforms introduced by Neff and Zakhor technique which is actually placed at the beginning of the above chart since it is equivalent to $T_{penalty}$=0. The latter method i.e. [AlSh99] would be placed somewhere in the middle of the chart, since it allows for multiple selection of blocks but under some restrictions.

2.4.4. Searching atoms

In order to find an atom, one should examine each waveform from dictionary at all possible locations in the prediction error and compute all of the resulting inner products. As it has been mentioned in the previous section, the search for an atom can be limited to certain region. Nevertheless, the computation load may be quite high even under the above conditions. Finding the maximum absolute value of the inner product requires [Neff95]:

$$O_{inner\_nonsep} = F^2 \cdot B^2 \cdot S^2 \tag{2.13}$$

operations, where:

O_{inner_nonsep}	-the number of multiply-accumulate operations required to find a single atom in the case where the separability of functions is ignored,
S	- the size of local search,
B	-the number of 1-D waveforms in the separable dictionary,
F	-the average size of 1-D waveforms.

If we assume the commonly used values [Neff97], [AlSh99], [Jeon00] for the above parameters (i.e. S=20, B=20, F=16) it gives over 40 million multiply-accumulate operations. To reduce the computational complexity of the inner product, one should utilise the separability property. This process has been presented in detail in [Neff95] and [Neff97]. The calculation process is divided into two stages. In the first stage, one calculates the inner product of a 2-D prediction error with a certain vertically oriented 1-D dictionary waveform and the intermediate result is kept for the second stage (Fig. 2. 10).

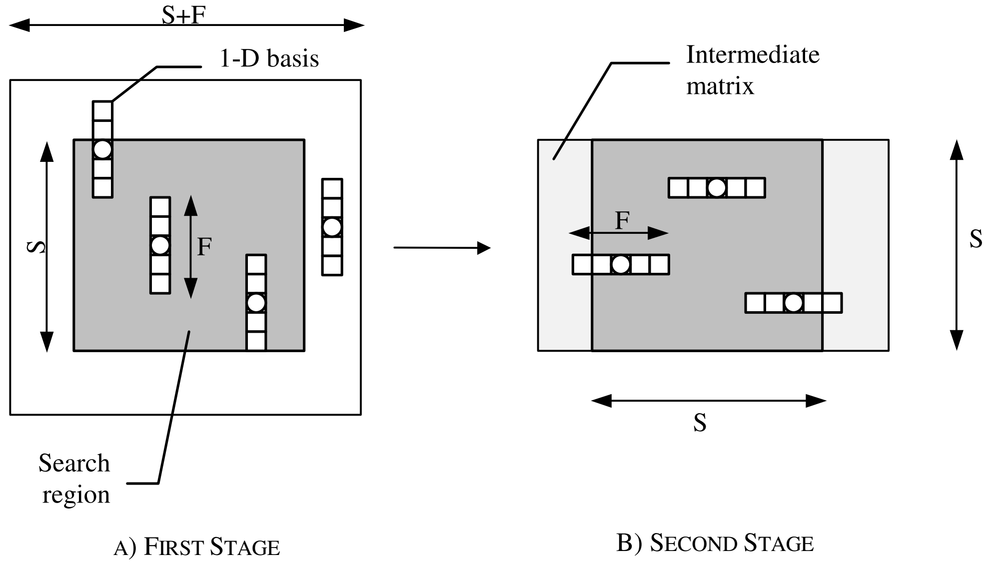


Fig. 2. 10. Separable fast method.

At the second stage, the process is repeated for all horizontally oriented 1-D waveforms instead. The number of operations is reduced to [Neff95]:

$$O_{inner\_sep} = B \cdot (S + F) \cdot S \cdot F + B \cdot S^2 \cdot F = B \cdot S \cdot F \cdot (2S + F), \qquad (2.14)$$

where:

O_{inner_sep}	-the number of multiply-accumulate operations required to find single atom utilising the separability of waveforms in the dictionary,
S	- the size of local search,
B	-the number of 1-D dictionary waveforms,
F	-the average size of 1-D waveforms.

In [Oh00], [Oh01] the calculation of the inner product value for 1-D basis has been speeded up by the utilisation of the symmetry property. It has been noticed that almost all waveforms in a dictionary from Table 2. 2 have odd or even symmetry. As a result, it is possible to simplify the mathematical formula of the inner product and reduce the number of multiplication to about one half (Fig. 2. 11).

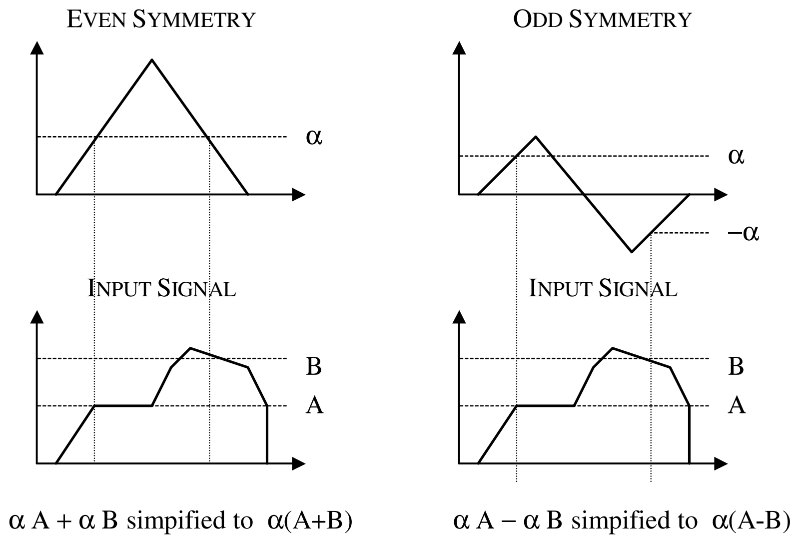


Fig. 2. 11. Simplification obtained using property of symmetry.

Many works have been recently devoted to the efficient computation of inner product of 2-D signals [Oh00], [Jeon00], [Oh01]. A very interesting idea has been presented in [Jeon00]. The proposed method allows precluding of a substantial number of dictionary functions from the inner product search using simple distance comparison without any degradation of the image quality. The method is based on Schwartz's inequality, this is:

$$\left|\langle f, g\rangle\right| \leq \|f\|\|g\|, \tag{2.15}$$

which leads us to :

$$\left|\langle f, g\rangle\right| \leq \|f\| \tag{2.16}$$

(taking into consideration the fact, that functions in dictionary are normalised, thus $\|g\| = 1$).

As mentioned above, the first stage of the fast inner product procedure gives the intermediate matrix, which is essential for the second stage to finish the whole process. Calculations for all horizontally oriented dictionary waveforms can be avoided in advance, if the value of Euclidean norm for sub-region of intermediate matrix is less than the maximum absolute value $\alpha_{current}$ of the inner product found out so far, i.e.:

$$if\ \|f_{sub-region}\| \leq |\alpha_{current}| \Rightarrow \forall g \in D\ \ \left|\langle f_{sub-region}, g\rangle\right| \leq |\alpha_{current}|. \tag{2.17}$$

The proposed method requires additional operations in order to calculate the norm for sub-region i.e. $\|f_{sub-region}\|$. However, this computation is negligibly small in comparison with the second stage. The advantage of the technique is that it can reduce up to 70% of the inner product calculations, which helps one to apply the matching pursuit method for real-time video coding.

It is worth noticing that the DCT-based technique requires similar number of operations to encode and to decode a signal. In contrast to this, the matching pursuit algorithm behaves very similar to the motion-compensated prediction, since its encoding process is complex and computationally intensive. The decoding process is simple and its activity is based on information provided by the encoder. Another difference between the complexity of the DCT and MP comes from the fact that transform coding requires a constant number of operations independently of the quality of approximation. The matching pursuit method demands to compute more atoms to better approximate an input signal. Consequently, the MP encoder and decoder transform more atoms, thus their computational load is also increased.

As can be seen on Fig. 2. 12, the computational load of matching pursuit is comparable with the inter-frame prediction process (i.e. the motion-compensated prediction). If dictionary contains non-separable waveforms, then searching atoms is much more computationally intensive than motion estimation. Therefore, there is a great demand for time-efficient algorithm that finds an atom in shorter time. The computational complexity comparison of an encoder based on a matching pursuit to the DCT-based encoder has been presented in [Neff95] and [Neff97]. Results obtained from [SIM3] and [TMN5] codecs have shown that MP-based encoder is four and eight time slower than DCT one for 10 kbits/s and 24 kbits/s respectively. In practice, in real-time applications used for videoconferences, these results are reduced by the fact that a video decoder based on matching pursuit demands fewer operations to reconstruct coded frames.

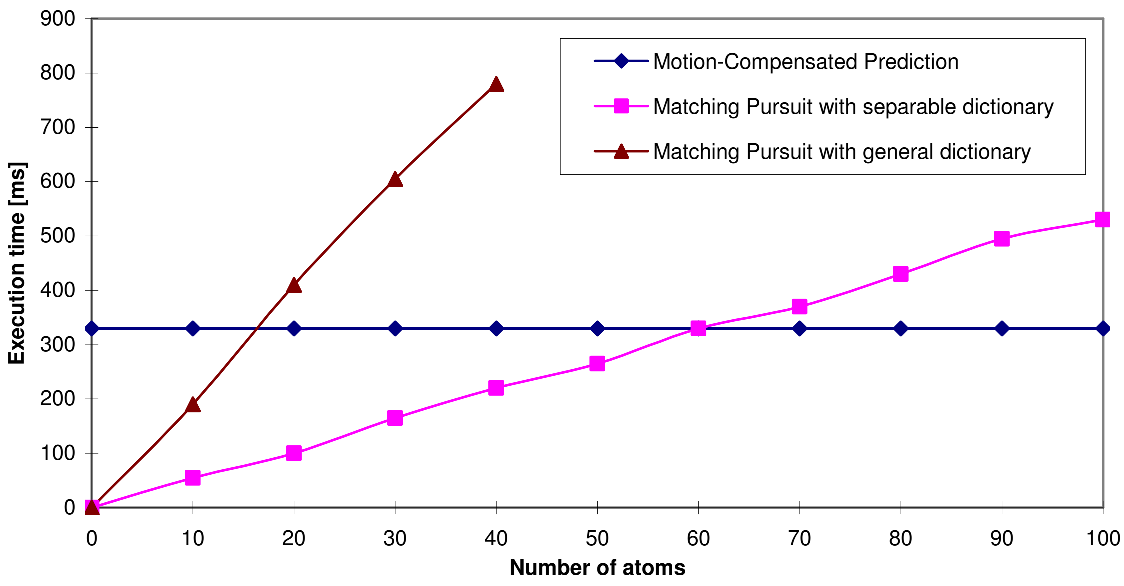


Fig. 2. 12. The comparison of matching pursuit implementations to the AVC motion-compensated prediction for QCIF sequence performed at ±16 pixels search range, qurter-pel accurancy, and all possible blocks sizes.

The comparison of decoder complexity and the performance for matching pursuit and DCT-based video systems have been presented in [Neff98]. Authors suggest that, in order to reduce blocking effects, the DCT-based decoder uses de-blocking and de-ringing filters increasing in that way a computational complexity. The matching pursuit does not suffer from the blocking artefacts, due to this MP-based decoder is seven times faster for QCIF and four times faster for CIF than similar DCT-based decoder with post-filtering.

2.4.5. Atoms coding

When the residual image is decomposed, the encoder should efficiently code a list of the chosen atoms. Each atom is defined by five parameters which describe the position $(\mathbf{x,y})$, waveform $(\mathbf{v_1,v_2})$, and the expansion coefficient α (Table 2. 1). The method of encoding atoms parameters has been proposed in [Neff95], [Neff96a], [Neff96b], [Neff97], [Neff97a] and improved in [AlSh99]. The main idea is based on the assumption that the most efficient order of encoding atoms is the position order. Due to this, atoms are sorted in order: from left to right and top to bottom. Each position of an atom is coded as a vertical displacement with respect to the previous atom and its displacement within the line. The waveform is specified by horizontal and vertical components, which are represented by an index equivalent to i in Table 2. 2. The indexes are coded using Huffman codes based on statistics from training video sequences. The expansion coefficient is quantized by a linear quantizer with a fixed stepsize and coded using variable length codes. The results obtained by using this method are similar to the sum of entropy of individual atom parameters.

Nevertheless, in real residual images, atoms are placed at the locations where motion estimation is ineffective. For this reason atoms are not distributed uniformly and independently on prediction error. Miroslavsky and Zakhor in [AlSh99] showed that it is possible to take advantage of these facts and spend fewer bits per atom position than the theoretical lower bound for the uniform independent atom distribution. Their algorithm, so-called *NumberSplit,* is based on a divide and conquer idea. The image is divided into two halves along a larger dimension and the number of atoms in the left or top half is coded. Then, the total number of atoms and the number of atoms in the first half allows the decoder to calculate the number of atoms in the second half. This algorithm is applied recursively until there are no more atoms in the given half of the image or until the size of the image half, is less than a certain threshold. Atoms in the last half are reordered according to the spiral scan [Banh97] and coded using Huffman codes (see Fig. 2. 13).

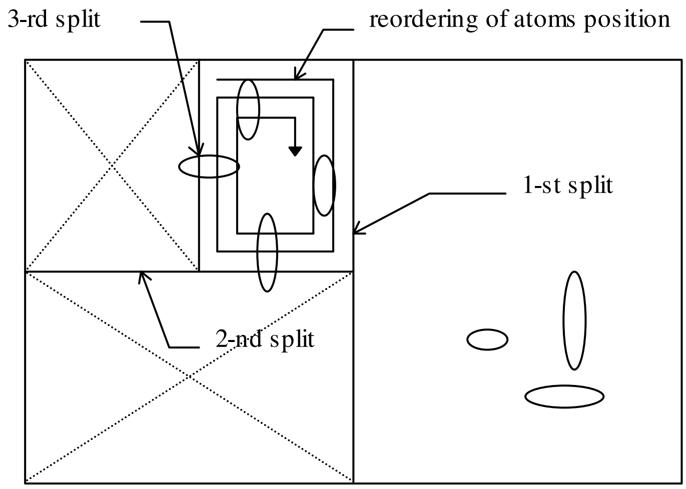


Fig. 2. 13. Scheme of "NumberSplit" algorithm.

The tremendous improvement in coding of atoms position has been presented in [Lin03]. Authors have used most of the temporal correlation and have coded atoms position utilising quadtree and quadtree prediction. The above technique saves 1-2 bits with respect to the *NumberSplit* algorith. Similar ideas and results have been also presented in [Garr05], [Garr06].

The researches have led to algorithms that get an advantage from spatial and temporal coherence of the atom positions and allow encoding the atom using fewer bits than the entropy. Moreover, the rate-distortion models are based on the entropy models [Ghara98], [Vander01]. As a result, this shows that the entropy model is credible and trustworthy.

In spite of the importance of efficient atoms coding, this problem is beyond the scope of this dissertation. It is motivated by the above works; foremost by the fact that the sum of entropy of five parameters defining an atom is very close (practically higher) to the number of bits required to code an atom. Taking into consideration the above fact, the number of bits per atom is estimated with the assistance of entropy.

2.5. Atoms Post-Selection

It has to be emphasised that the complexity of the M-optimal approximation problem was mainly reduced by the transformation of a basic problem to M simpler problems. In fact, the greedy strategy could be optimal, only if a local region would be properly selected. In a very low bit rate system, where a signal is approximated using small number of atoms, the task of choosing desired regions is particularly important.

The experiments performed by the author showed, that a sequence of absolute values of successive expansion coefficients does not preserve decreasing order (see Fig. 2. 14). Taking into consideration the fact that error decay is exponential [Mall93], [Davis97], this reveals problem concerning with the approximation using small number of atoms. This problem is particularly related to systems where the bit budget is very limited since, smaller number of atoms leads potentially to worse quality of solution for the M-optimal approximation problem. Let us notice, that non-monotonic order of selected atoms gives worse solution (i.e. more distant from optimum) at the beginning of the order then in its further parts.

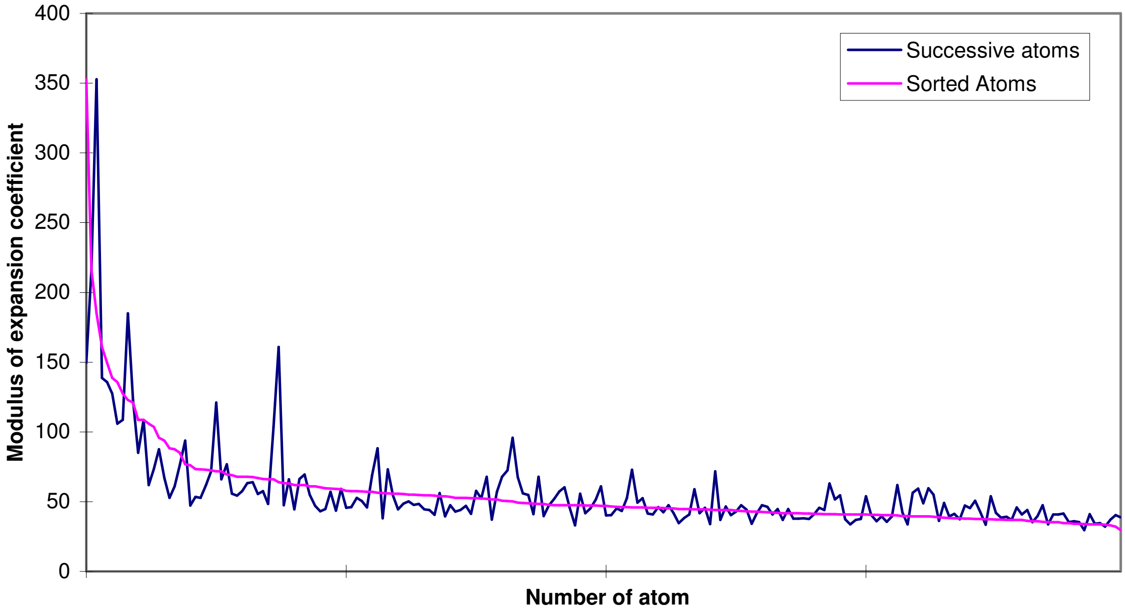


Fig. 2. 14. Order of successively computed atoms for Foreman sequence at 4th frame.

To compensate an unwanted effect related to the fact that atom is searched in a limited region of an input signal, the author proposed a simple modification to a matching pursuit. The scheme of a new algorithm is presented in Fig. 2. 15. In the first step, 2N atoms are selected by using matching pursuit. Then, atoms are sorted and N atoms with the greatest absolute value of expansion coefficient are taken to approximate an input signal.

The *„Atoms Post-Selection"* algorithm compensates improper selections of local regions and additionally avoids atoms that poorly represent an input signal. The main disadvantage of this solution is its computational complexity that is doubled by the fact that the algorithm needs to find the 2N atoms. The results in Fig. 2. 20 show slight increase in the PSNR (average 0.15 dB).

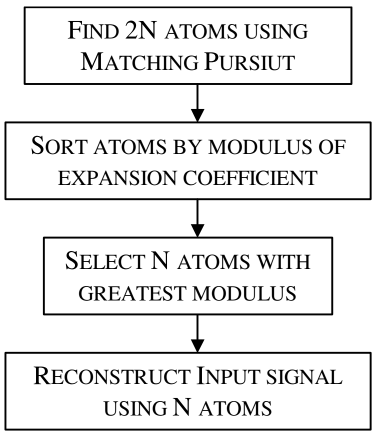


Fig. 2. 15. Atoms Post-Selection.

It is worth mentioning that the post-selection of atoms changes the statistic of atoms parameters and allows it to approximate a frame using larger number of atoms. This phenomenon results from the fact that the atoms selected to represent a signal are more ordered by the absolute value of expansion coefficient. Due to this, modulus density in so-called pure matching pursuit is very similar to Gausian function while the density of absolute values of inner products in post-selection scheme is Lapleace-alike (see Fig. 2. 16).

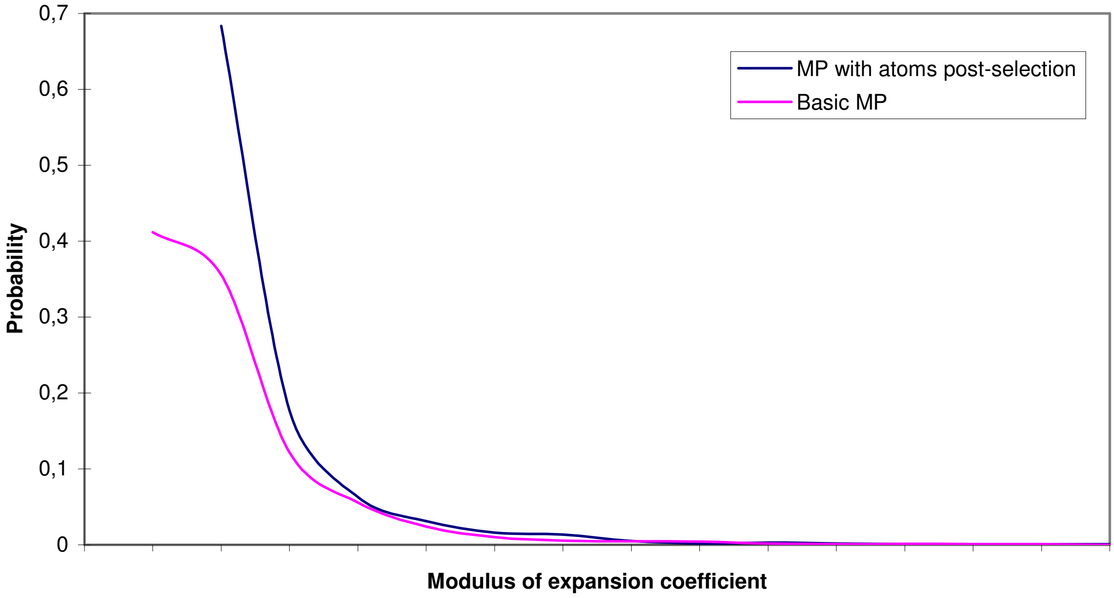


Fig. 2. 16. The modulus density comparison of two strategies:
Matching pursuit and Matching pursuit with post-selection of atoms.

2.6. Implementations of Video Coder with Matching Pursuit

First of all, the implementations performed by the author were built on top of the H.264/AVC codec. The reasons for choosing the H.264/AVC codec were determined by the fact that this specification is a field of contemporary researches and comparisons.

For the purposes of this dissertation, the author has created two different implementations: one for researches, and the later for comparisons.

2.6.1 Implementation based on the TML codec

The first experimental platform has been built by the author on top of the Test Model Long Term Number 4 [TML4] i.e. H.26L ver.4.3. codec of Telenor. There were three reasons why the version of TML4 has been chosen:

- The source for version 4.3 is well written i.e. the idea and plan of the encoder is clearly decomposed on separate parts of the project. In addition, only few people were involved in the project, therefore, the code is being written in a very coherent style.
- The dissertation demanded to perform many experiments. Taking into consideration the fact that a matching pursuit algorithm is computationally intensive, there was great need of existence of the fast and stable environment like the TML4. In addition, a lot of experiments did not refer to results obtained from the other video encoders since the main purpose of such experiments were to check the behaviour of certain solutions. The author assumes that in such instances the implementation based on TML4 is a good solution.
- The source of version 4.3 is the highest version of the encoder that does not contain the B-frame encoder. The author assumes -similarly to others [Neff96a], [Sch01]- that sufficiently good environment for comparison of different motion residual coding methods arises in the P-frame scheme. (It is assumed that B-fame scheme introduces the same unwanted artefacts to the encoded sequence as P-frame scheme). Taking into consideration the historical point of view [Neff95], [AlSh99], [Neff02a], the above version of the codec seems to be a proper selection, because all works devoted to comparison to the DCT and matching pursuit had switched the B-frame technique off.

The TML4 had not implemented a mechanism that allows for a sequence to be encoded with an assumed bit-rate. In order to obtain this, the source of the encoder has been supplemented with a rule that controls the value of the quantization step and indirectly influences the number of bits produced to encode a frame. The rule is presented in the pseudo-C style in Fig. 2. 17. As can be seen, the quantization step is changed, if calculated current bit-rate differs more than 1.5% or less than 1% of the assumed bit-rate.

```
If (CurrentBitRate > DefinedBitRate * 1.15)
   {
   If (QuantizationLevel < MAX_LEVEL)  QuantizationLevel++
   }
Else
   {
   If (CurrentBitRate < DefinedBitRate * 0.99)
      {
      If (QuantizationLevel > 0) QuantizationLevel--
      }
   }
```

Fig. 2. 17. The encoder rule that defines a quantiaztion level in TML4.

Changes in the decoder source have not been necessary because each header of an encoded frame contains information about quantization. The introduced changes cause that new video system slightly outperforms the base TML4 system with constant value of quantization level in term of an average PSNR. The TML4 video encoder significantly outperforms the MPEG-4 system.

The TML4 video encoder with a matching pursuit has most coding tools of the AVC encoder. In particular, it contains all the advantages of the AVC motion prediction process. The source has been modified in such a way that the matching pursuit algorithm has replaced the transform coding. The algorithm of matching pursuit has been implemented in the way described above. Selected atoms are stored outside the H.26L bitstream in a separate file. As it has been mentioned, atoms parameters are not encoded but, in order to estimate a number of bits per atom, the entropy is being calculated. The number of bytes required to store the complete information concerning atoms is estimated as follows:

$$MP\_stream = \left\lceil \frac{E_{atom} T_{atom} + H_{atom}\, N_{P-frame})}{8} \right\rceil, \qquad (2.18)$$

where:

E_{atom} -entropy of atoms set for encoded sequence,

T_{atom} -total number of atoms in whole sequence,

$N_{P-frame}$ -number of P-frames,

H_{atom} -number of bits per header (constant estimated as $H_{atom} = 24$)

2.6.2. Implementation based on the JM codec

The main purpose of implementation of a video encoder based on the Joint Model version 8.4. [JM8] was to compare the results of the very new AVC transform-based encoder to the matching pursuit solution.

Highly complex scheme of the encoder as well as many techniques co-operated in order to obtain common purpose enforced on the author to perform a trick. Thus, the transform coefficients of inter-frame coded macroblocks are set to zero. As a result, such macroblocks are approximated using motion vectors only, or alternatively, the macroblocks are not intra-coded. The obtained in this way image form the predicted frame, which is used to create the prediction error signal. The matching pursuit algorithm approximates the resulted prediction error. The general schemes of encoder and decoder are presented on Fig. 2. 18.

In the implementation, some simplifications have been made. Firstly, the atom parameters were not encoded, but instead the number of bits required to encode an atom was estimated using statistical model based on entropy calculations (see equation 2.18.). Secondly, each frame was encoded using the number of bits known from a respective JM8 bitstream. This means that the same bit allocation as in the standard AVC was used for the consecutive frames encoded by matching pursuit. We can say that work of encoders (i.e. JM8 and JM8 with matching pursuit) is synchronized by the output bitstream (see further for details). The synchronisation of bitstreams gives a very good comparison model and simultaneously simplifies the control block in the experimental encoder.

The synchronisation of output bitstreams has been performed as follows (see Fig. 2. 19). At first, the current frame is encoded using the AVC encoder. Then, all transform coefficients for inter-frame coded macroblocks are set to zero and decoded frame is produced. The number of bits spent to encode the current frame decreases the number of bits, which has been spent to encode the adequate frame in the AVC coder. As a result, the number of remaining bits is obtained. Then, the number of atoms for encoding the prediction error is estimated. Next, the chroma components are represented as closely as it is possible to the PSNR values known from the AVC encoder. The remaining number of atoms is used to encode luminance component of the prediction error.

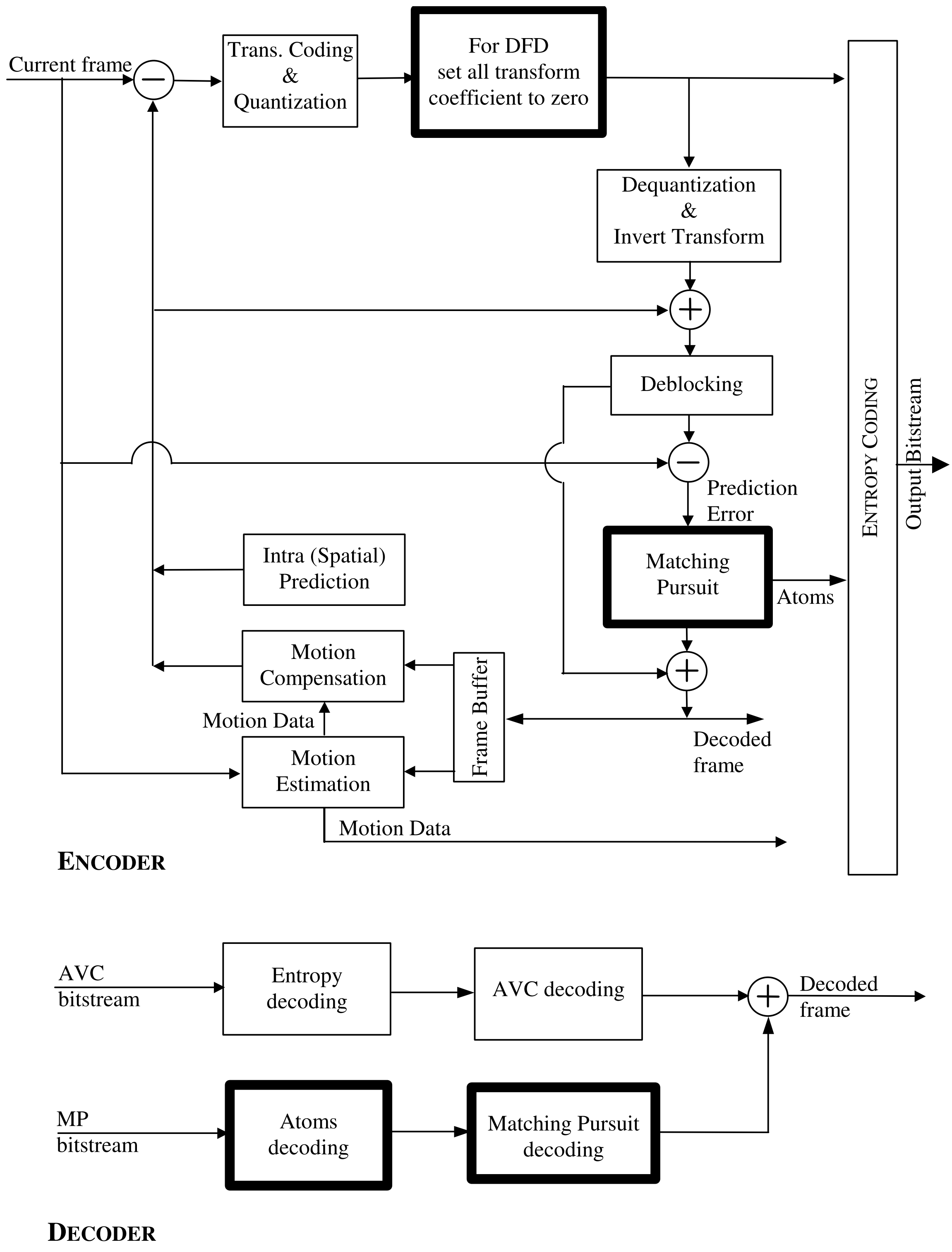


Fig. 2. 18. The AVC video codec with matching pursuit.

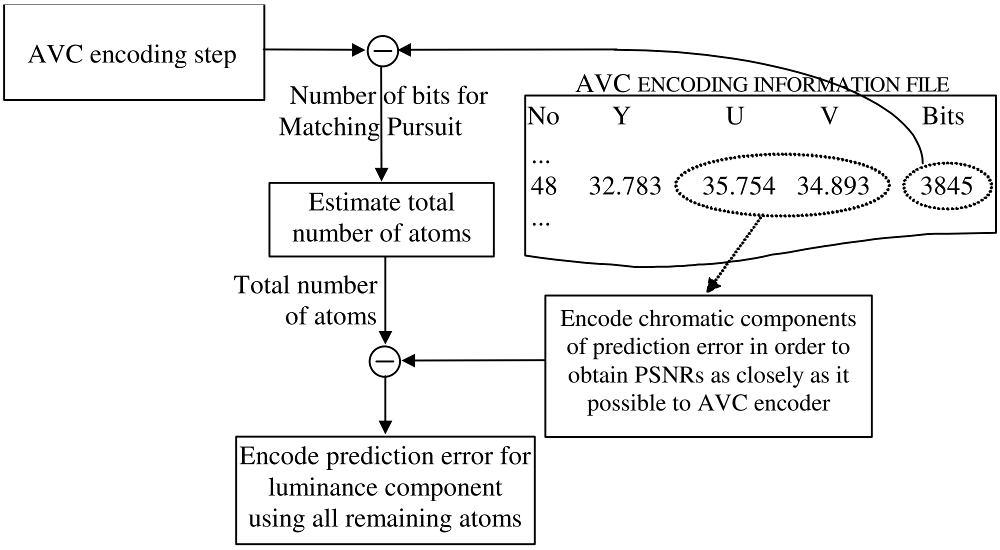


Fig. 2. 19. Scheme of synchronization the AVC with Matching Pursuit encoder to the AVC JM8 encoder.

As can be seen in Fig. 2. 20, the basic implementation of the video encoder with matching pursuit gives satisfactory results while the improved version of the basic implementation gives very similar results to the AVC transform-based encoder. The improved version has been extended of about the new process of selecting regions (Section 2.4.2) and the proposed „Atoms Post-Selection" algorithm (Section 2.5).

Fig. 2. 21, Fig. 2. 22, Fig. 2. 23 show bits allocation for the different encoders. The basic and the improved implementation of the video encoder with matching pursuit have very similar bits allocation. Differences between the transform-based AVC and the matching pursuit-based counterpart result from the fact that transform coefficients in the latter encoder are set to zero. As the result, more macroblocks are coded in so-called skip mode. Thus, fewer bits are required to encode the mode for macroblock. For example, some motion vectors disappear from output bitstream since the AVC rate-distortion optimization process prefers the skip mode.

Nevertheless, the competing of the matching pursuit method with the transform-based methods is out of the scope of this dissertation. The most important is to build stable and credible environment for further comparisons.

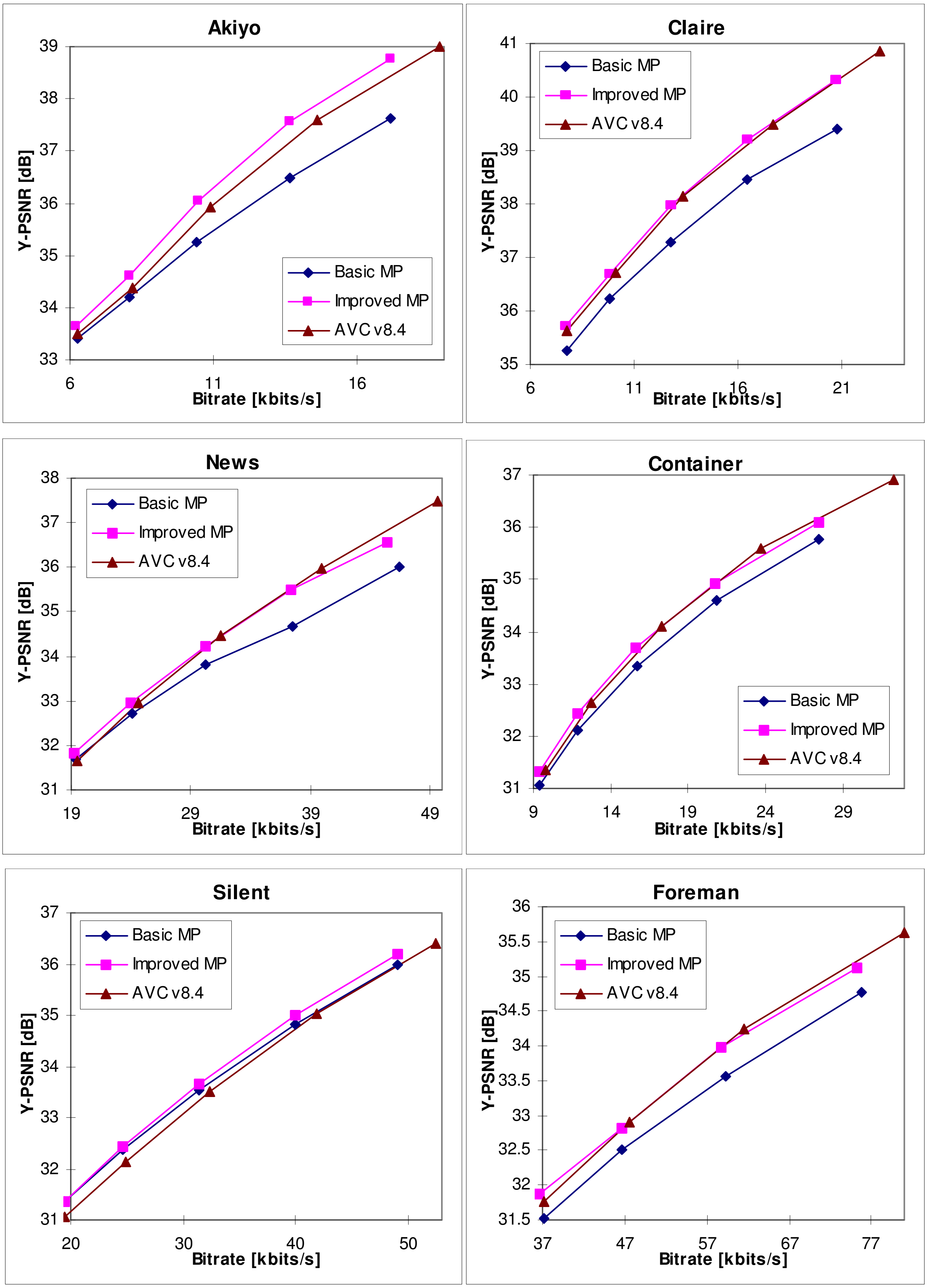
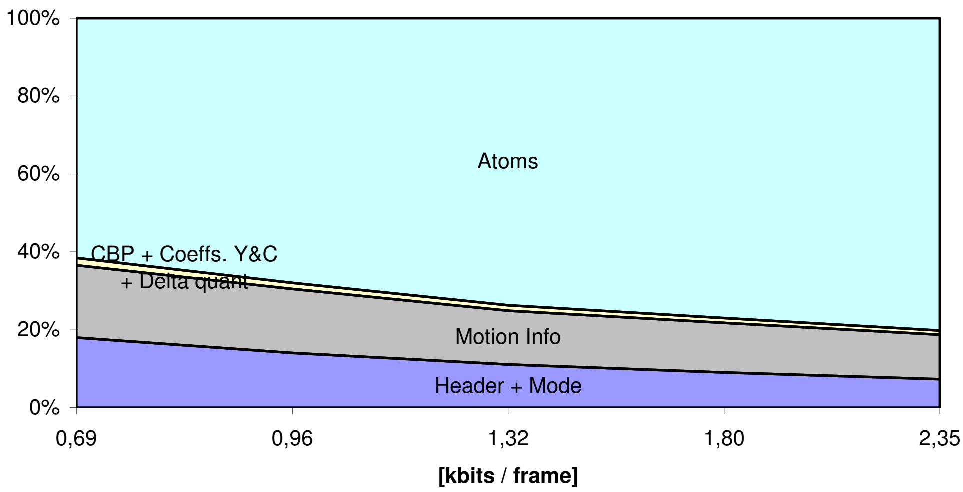
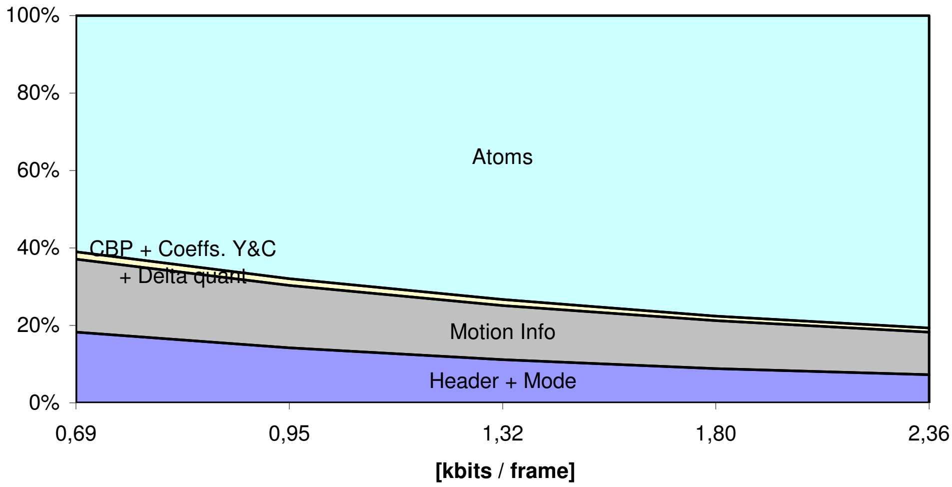


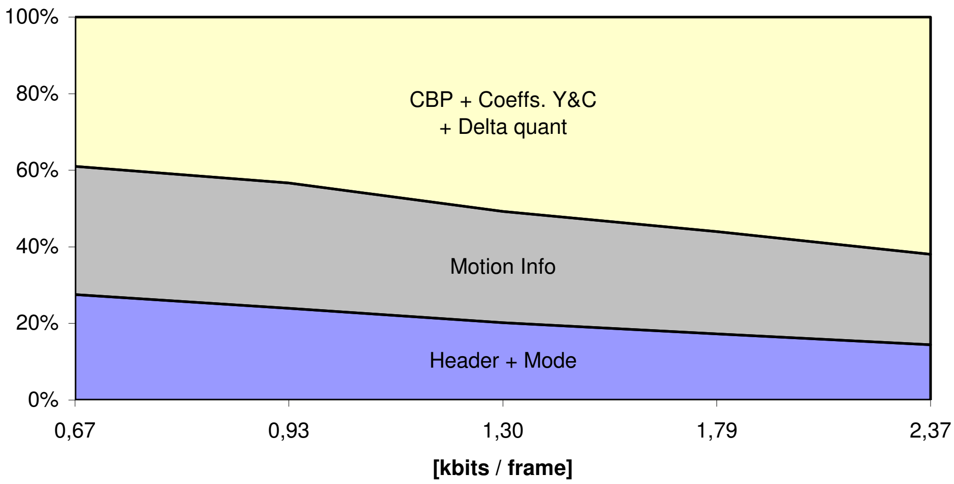
Fig. 2. 20. PSNR values for luminance (Y-PSNR) for QCIF video test sequences: Akiyo, Claire, News, Container, Silent and Foreman.



a)

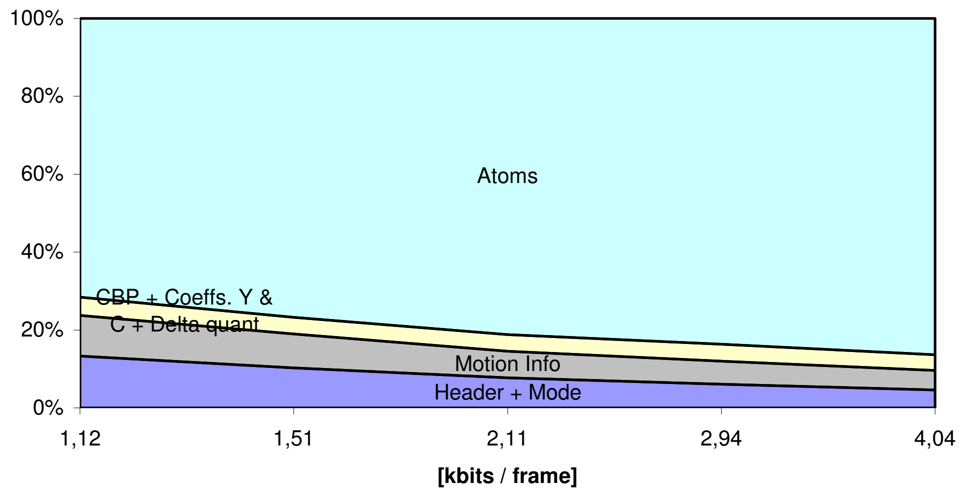


b)

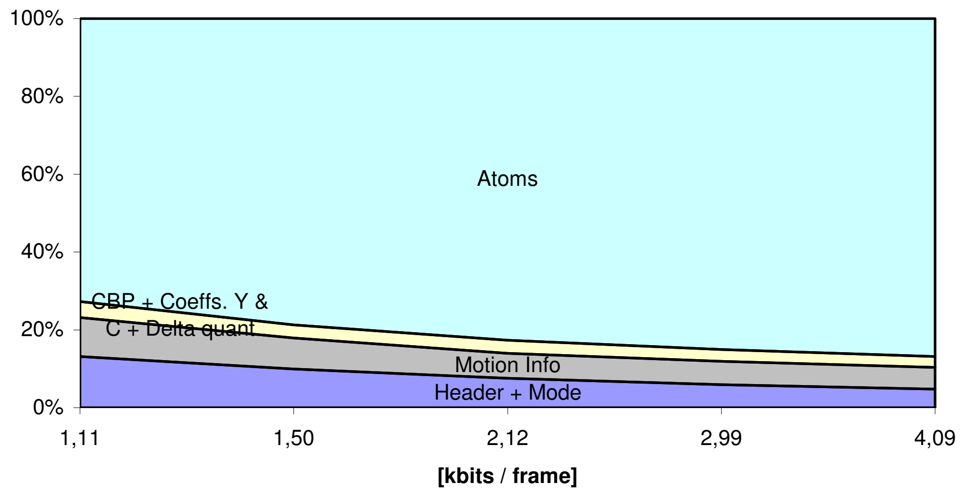


c)

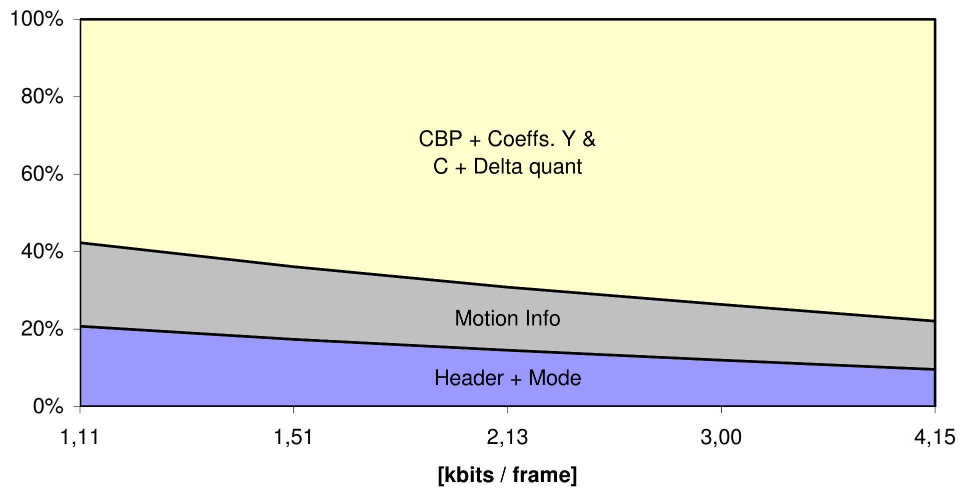
Fig. 2. 21. The comparison of streams for Akiyo sequence for Basic MP (a) and Improved MP (b) and the AVC v8.4 (c).



a)



b)



c)

Fig. 2. 22. The comparison of streams for Container sequence for Basic MP (a) and Improved MP (b) and the AVC v8.4 (c).

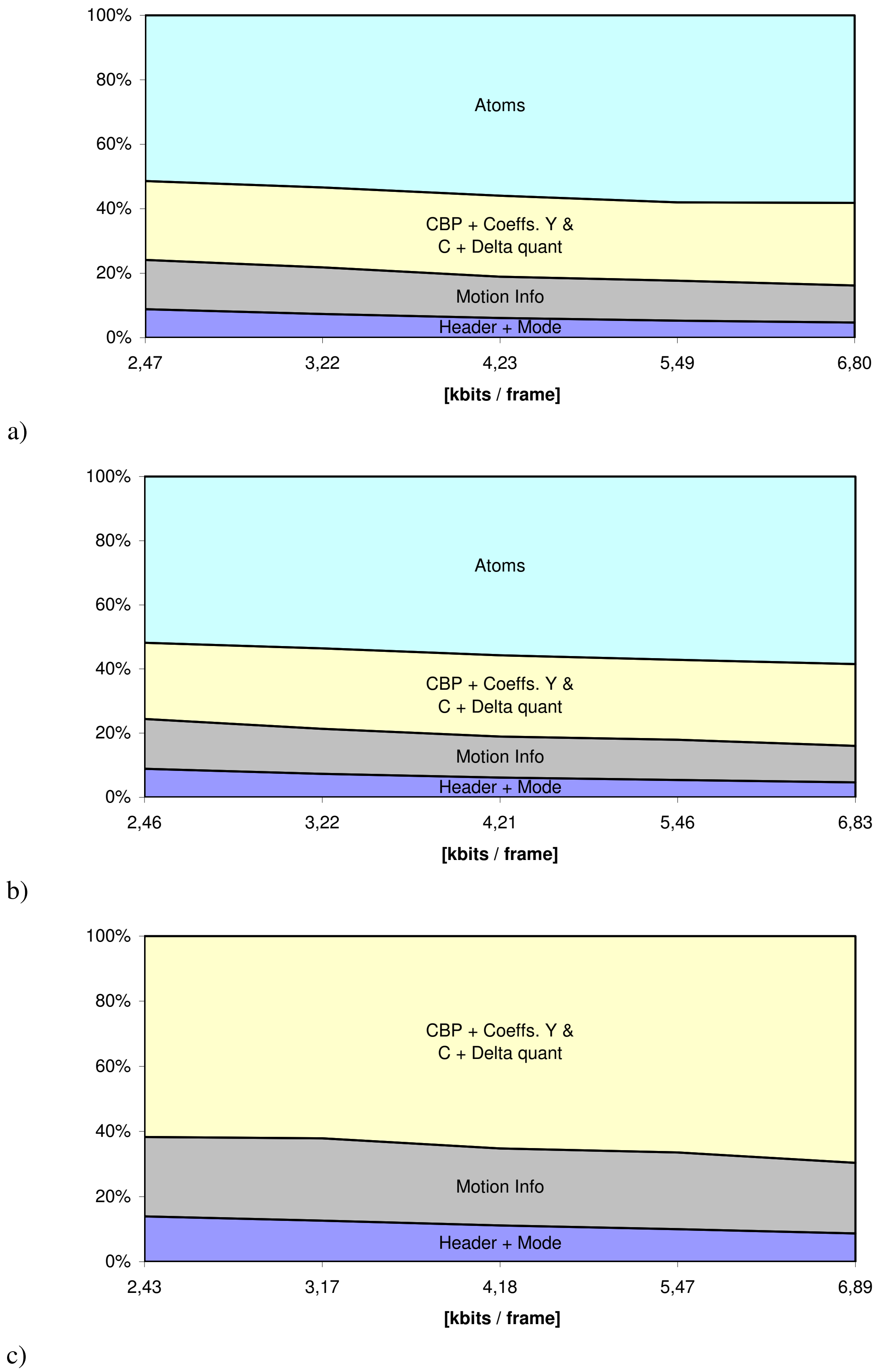


Fig. 2. 23. The comparison of streams for Silent sequence for Basic MP (a) and Improved MP (b) and the AVC v8.4 (c).

2.7. Conclusions

In this chapter, the foundations and implementations of the matching pursuit have been presented. In addition, the author proposed two algorithms that allow it to improve the objective quality criteria. The first algorithm concerns the method of choosing the local approximation region (see Section 2.4.2). The later algorithm, i.e. post-selection of atoms, tries to correct improper representation of the selected region (Section 2.5). The implementation of the matching pursuit for video coding confirmed that this method is competitive to the transform-based solutions. In addition, the implementation made certain that the matching pursuit technique in spite of the separable environment is a very computational intensive process. Due to this, the existence of faster solution for the two-dimensional separable MP is desired.

Chapter 3

Matching Pursuit with Separable Decomposition

3.1. Introduction

Matching pursuit is a technique that is able to represent a signal using a small number of atoms. However, the computational load related to finding a single atom is significant. Moreover, a fundamental problem of matching pursuit in video coding is the lack of feedback between the motion compensated residual image and a dictionary. It is worth mentioning that the computational load depends to hardware power and it is quite possible that in a few years time the hardware performance will be enough to handle this problem in real-time applications. In addition, the finding of atoms can be easily performed as a parallel process [Norc02]. One should notice that the above computational load refers to a simplified model of matching pursuit that uses both the separability of functions from dictionary and procedure of atom search limited to a selected region. The first point implies the existence of a fast inner product algorithm and simultaneously guaranties a good approximation of motion residual. The latter point is well motivated and allows for searching into significantly reduced region. Even if we suppose that the problem of computational complexity is just a matter of time, it is clear that the lack of feedback between an input signal and a dictionary is still an unsolved question.

In the above context, finding of an atom may be viewed as a process of searching for the best separable function of the chosen region, which simultaneously belongs to a dictionary. The direction of search performs from *a priori* known dictionary towards an input signal. Due to this, the best real separable function remains unknown and, furthermore, it is impossible to supplement the dictionary by the most expected function. In order to change this, the author proposes a novel method that allows for decomposing N-dimensional signal into N one-dimensional signals, i.e. into a best separable function.

This chapter presents the author's own contributions to the matching pursuit. In particular, the matching pursuit with separable decomposition is described. The chapter also contains the experimental results that confirm utility of the proposed method.

3.2. Theory of Matching Pursuit with Separable Decomposition

3.2.1. Preliminary

It seems that the procedure for finding atoms is the weakest stage of matching pursuit. The reason for this lies in the way of searching for the best solution, i.e. atom. The proposed method [Neff95], [Neff97] is very general and does not take into consideration the properties of separable functions in a dictionary. Let us remember that the algorithm takes all functions from the dictionary and tries to find the best fitting for them in all possible locations. This means that the space of solutions for N-dimensional signal increases rapidly since this space comes into existence as a product of all dimensions. The dissertation takes focus on the stage of searching atoms and presents a new strategy that allows for the reduction of the computational complexity of the matching pursuit utilising the separability of functions from the dictionary (see Fig. 3. 1).

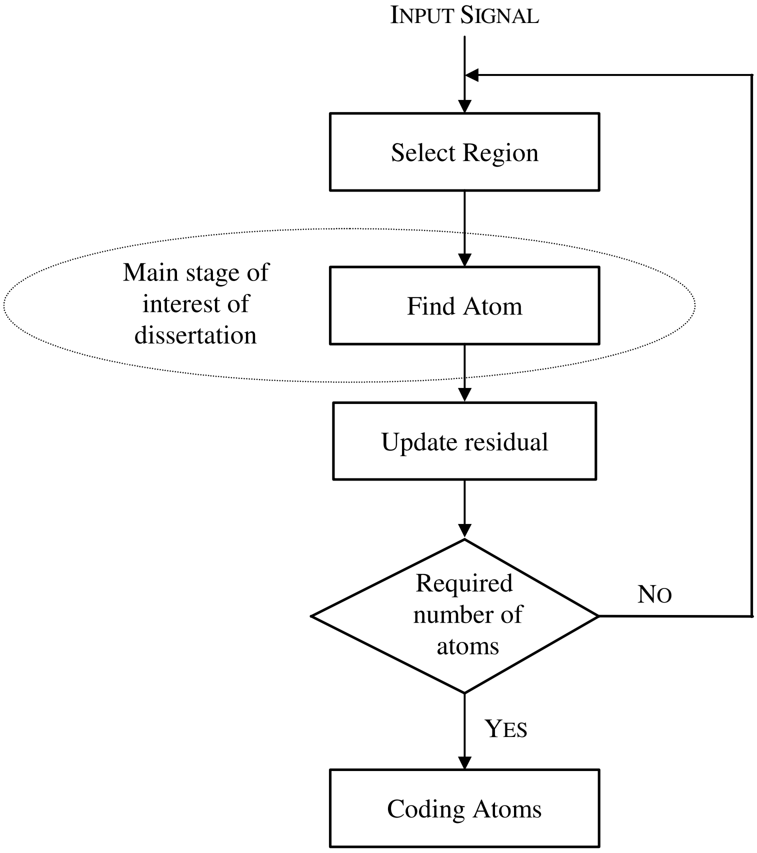


Fig. 3. 1. Matching pursuit encoding process.

It is worth mentioning that the process of signal reconstruction may be unchanged; furthermore, this means that the decoder is able to use both matching pursuit and the matching pursuit with separable decomposition bit-stream. The above situation is very similar to a motion-compensated prediction process where coding standard does not specify how motion vectors are to be computed.

In general, the matching pursuit with separable decomposition is a type of matching pursuit algorithm. The key element of this technique is separable decomposition [Doma03], [Doma05b].

3.2.2. Separable Decomposition

For the sake of simplicity, let us consider 2-D space $\mathcal{H}$ of real-valued functions:

$$\mathcal{H} = \{ f : X \times Y \to \mathbb{R} \}, \tag{3.1}$$

where:

$X = [0,1 \ldots I-1]$,

$Y = [0,1 \ldots J-1]$.

In this space $\mathcal{H}$, a measure $s(a,b)$ of similarity of $a \in \mathcal{H}$ and $b \in \mathcal{H}$ is defined as follows:

$$s(a,b) = \frac{|\langle a,b \rangle|}{\|a\|\|b\|}, \tag{3.2}$$

where $\langle \cdot \rangle$ and $\|\cdot\|$ denote the inner product and Euclidean norm, respectively.

Let $\mathcal{H}_s \subset \mathcal{H}$ be a subset containing all separable functions of the space $\mathcal{H}$. The main goal of separable decomposition is to find $t \in \mathcal{H}_s$ that for a given $f \in \mathcal{H}$ satisfies:

$$\forall q \in \mathcal{H}_s \quad s(f,t) \geq s(f,q). \tag{3.3}$$

In order to obtain the best separable representation $t \in \mathcal{H}_s$ of $f \in \mathcal{H}$, one needs to iterate a few steps of a transformation defined as follows:

$$\alpha_k(i) = \sum_{j=0}^{J-1} f(i,j)\beta_{k-1}(j), \tag{3.4}$$

$$\beta_k(j) = \sum_{i=0}^{I-1} f(i,j)\alpha_k(i)$$

or alternatively,

$$\beta_k(j) = \sum_{i=0}^{I-1} f(i,j)\alpha_{k-1}(i), \tag{3.5}$$

$$\alpha_k(i) = \sum_{j=0}^{J-1} f(i,j)\beta_k(j),$$

where:

$\alpha_k : X \to \mathbb{R}$ - 1-D function,

$\beta_k : Y \to \mathbb{R}$ - 1-D function,

and additionally,

$t_k = \alpha_k \beta_k$ - 2-D separable function from $\mathcal{H}_s$.

The whole process starts with arbitrarily chosen constant functions α_0 and β_0. After each iteration, new functions are computed thus, a new 2-D separable function t_k is obtained. The Theorem 4 (see Appendix) shows that the sequence of separable functions calculated in the above way satisfies:

$$s(f,t_0) \leq s(f,t_1) \leq \ldots \leq s(f,t). \tag{3.6}$$

In this sequence, the last function is the best separable decomposition of f.

3.2.3. Matching Pursuit with Separable Decomposition

Let us remember that the complexity of the *M*-optimal approximation problem (see Chapter 2.3.1) has been reduced by the greedy sub-optimal algorithm in such a way that the *M* dictionary elements are chosen individually instead of *M* at once. In a space of separable functions, it is possible to apply a technique that reduces the complexity of the problem in a very similar way. Note that separable decomposition finds a separable function that approximates an input signal in the best manner, i.e. it minimises Euclidean norm. This fact allows it to consider *N* one-dimensional functions instead of one *N*-dimensional signal, since a separable function can be treated as a product of one-dimensional functions. The above observation laid the foundations of the novel concept for matching pursuit. The proposed algorithm has been called the matching pursuit with separable decomposition. In general, the algorithms are the same; the only difference concerns the part of searching for an atom.

The detailed scheme of searching atom in matching pursuit algorithm with separable decomposition is presented in Fig. 3. 2. In the first stage, the best separable representation of an input signal is found using separable decomposition. To understand further, it is better to see this process as a transformation to *N* one-dimensional signals.

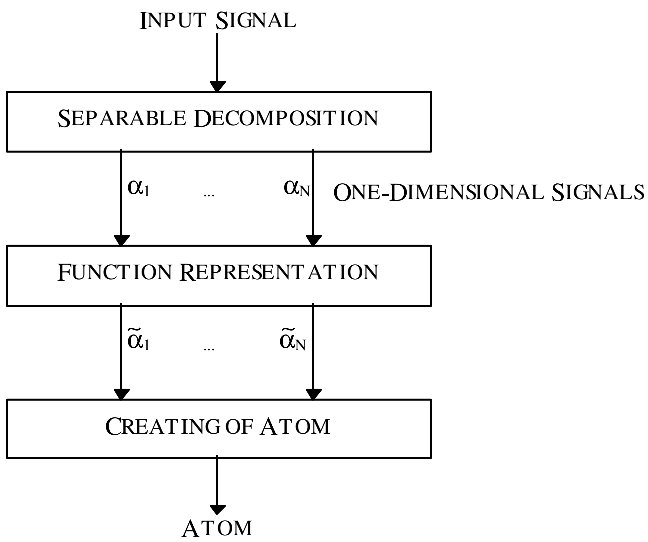


Fig. 3. 2. Finding of atom in matching pursuit with separable decomposition.

The second stage tries to represent one-dimensional functions. The step, hereafter called *FunctionRepresentation,* may adopt a vary form. In the easiest case, an output function $\tilde{\alpha}_i$ may be exactly the same as an input function α_i. Another solution may consist of applying one-dimensional matching pursuit to a function α_i. The computational complexity comparison of the last scheme to the fast matching pursuit algorithm used in [Neff95], [Neff97] shows superiority the first method over the latter. Let us assume that a dictionary contains B×B separable functions. In addition, let S denote the size of a atom search in a single direction, and F be an average size of one-dimensional functions (similarly to Chapter 2.4.3.). In accordance with (2.13) [Neff95], the number of operations required to find an atom in the matching pursuit scheme is expressed as follows:

$$O_{inner\_sep} = B \cdot S \cdot F \cdot (2S + F)\,, \tag{3.7}$$

where:

O_{inner_sep}	- the number of multiply-accumulate operations required to find single atom utilising the separability of functions in the dictionary,
S	- the size of local search,
B	- the number of 1-D dictionary waveforms,
F	- the average size of 1-D dictionary element.

In the case of matching pursuit with separable decomposition, where the process must be performed twice -this is for both the horizontally and vertically oriented functions- the computational complexity is expressed in the following way:

$$O_{inner\_rep\_sep} = B \cdot S \cdot F + B \cdot S \cdot F = 2 \cdot B \cdot S \cdot F\,, \tag{3.8}$$

where:

$O_{inner\_rep\_sep}$	- the number of multiply-accumulate operations required to find single atom in MPwithSD.

The speed up is notable even if we take into consideration the number of calculations needed to perform the separable decomposition process:

$$O_{speed\_up} = \frac{O_{inner\_sep}}{O_{inner\_rep\_sep}} = S + \frac{F}{2}\,, \tag{3.8a}$$

where:

O_{speed_up}	- the speed up for O_{inner_sep} to $O_{inner\_rep\_sep}$.

For practical applications, one can assume $S = 20,\ F = 16$ (see Section 2.4.4).

Generally speaking, the *FunctionRepresentation* approximates the N-dimensional separable function in N individual stages. In next parts of this dissertation, it will be shown a very important role of this step especially in the context of data compression.

Very interesting similarities can be perceived by comparison of the matching pursuit with separable decomposition to DCT-based coding (Fig. 3. 3). Both the former and the latter technique restrict their activity to some local regions. For this purpose, DCT-based encoder divides an input signal into non-overlapping distinct blocks and attempts to express each block in term of basis functions taking a particular quantization into consideration.

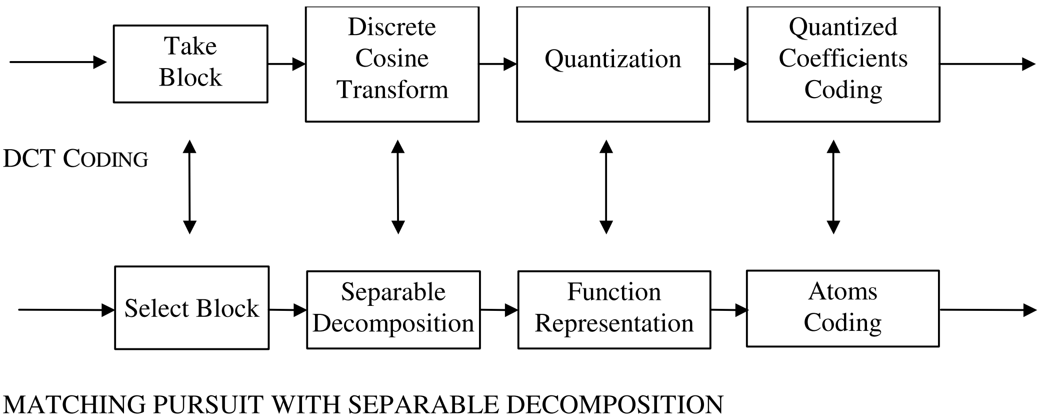


Fig. 3. 3. The comparision of DCT-based coding to matching pursuit with separable decomposition.

The first step of matching pursuit that limits the region of approximation, fulfils similar task. Let us remember that this procedure chooses a region with the largest energy and avoids repeated selections of regions. There are no problems with repeated selections in DCT-base system since all blocks are disjointed. On the other hand, as a result of quantization, some blocks in DCT-technique may be skipped if all quantized coefficients are set to zero. This situation is equivalent to matching pursuit in which a certain region is never chosen. Continuing this analogy, the second stages (i.e. Discrete Cosine Transform and Separable Decomposition) express the selected block into the space of separable functions. The main goal of the last but one stage is efficient quantization (Fig. 3. 3).

3.3. Matching Pursuit with Separable Decomposition Video Coding

Simplified block diagrams of the encoder and decoder in which motion residual is coded using matching pursuit with separable decomposition are shown in Fig. 3. 4. The schemes of the encoder and the decoder are very similar to those presented in Chapter 2.6. The only differences concern the ways of approximation and reconstruction of a prediction error. As can be seen, the current frame is first motion compensated using the previously reconstructed frame. Then, the prediction error is approximated using the matching pursuit with separable decomposition. The final shape of the reconstructed frame is formed by the addition of the temporally utilised motion-compensated frame and the approximation of the displaced frame difference. As it was mentioned above, the decoder is so similar to the matching pursuit decoder that it can be left without any changes (Fig. 3. 4).

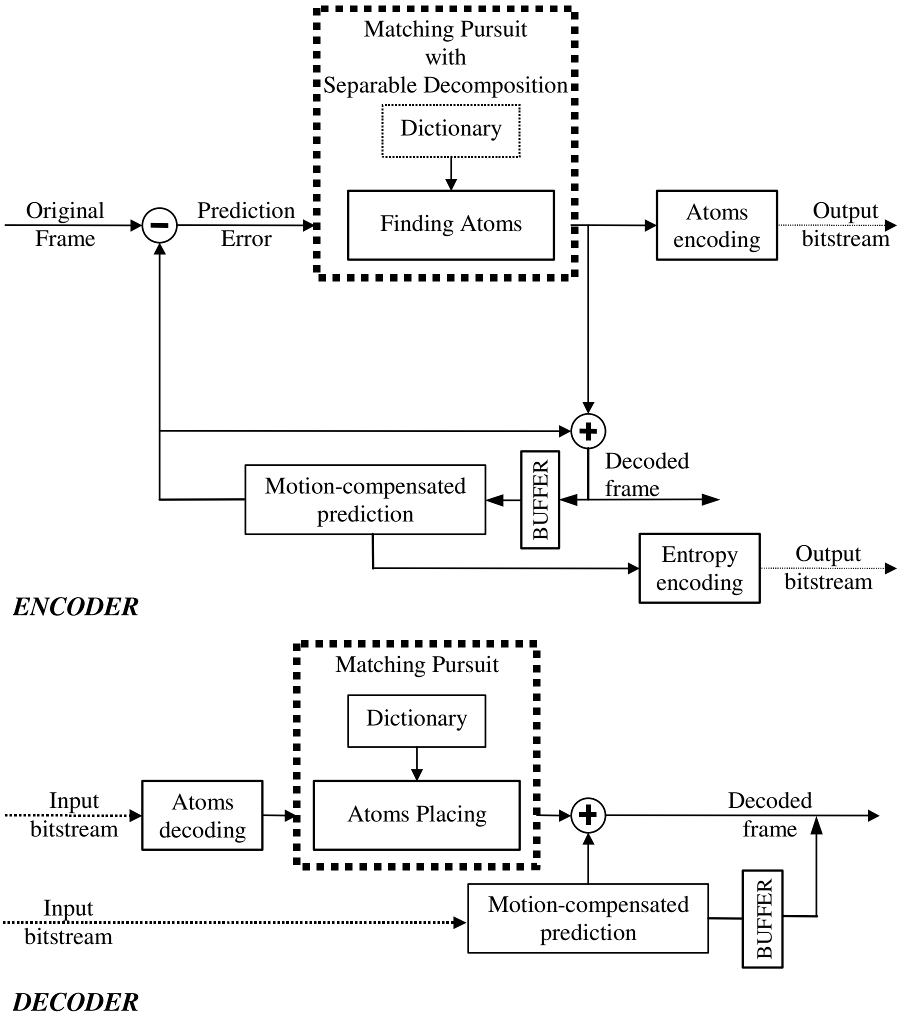


Fig. 3. 4. The scheme of video codec using matching pursuit with separable decomposition.

3.3.1. Implementation of Video Encoder using Pure Matching Pursuit with Separable Decomposition

The purpose of the implementation of a video encoder using the matching pursuit with separable decomposition was to demonstrate the efficiency and flexibility of this technique. The main goal was concerned with verification of how some quantization parameters influence the quality of approximation.

The encoder was built on top of the Test Model Long Term Number 4 (TML4) i.e. H.26L ver.4.3. codec of Telenor. Firstly, the DCT-based encoder was removed and in its place the matching pursuit algorithm was implemented (see Section 2.5). Then, the MP was modified to get the matching pursuit with separable decomposition (MPwithSD) algorithm. The final MPwithSD implementation contains:

- Region selection,
- Searching of atom - *FindAtom*,
- Atoms post-selection.

The *"Region selection"* and *„Atoms post-selection"* are discussed in the previous chapters (see Section 2.4.3 and 2.5). The *FindAtom* procedure is new. For experimental purposes, some elements of the *FindAtom* have been distinguished and parameterized in an external configuration file (see Fig. 3. 5). Due to this, it is possible to manipulate some of parameters and influence the algorithm behaviour. The most interesting parameters describe the number of iterations of separable decomposition (N_{decomp}) and the number of reconstruction levels of linear quantizer in the *FunctionRepresentation* (F_{levels}). The additional parameters contain information about the number of atoms used for individual P-frame (N_{atoms}) and the stepsize of linear quantizer, which is applied for an expansion coefficient ($Q_{stepsize}$).

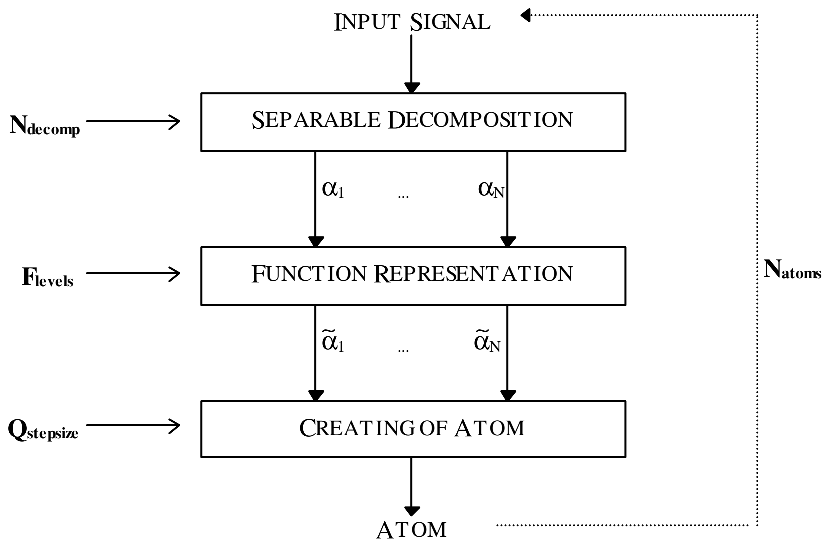


Fig. 3. 5. The corelations between the configuration file parameters and stages in matching pursuit with separable decomposition.

The complete process of motion residual approximation is performed as follows. At first, the centre of region that will be locally decomposed is selected. Then, the block with the largest concentration of energy is enlarged to the block of 24×24 pixels. The separable decomposition is performed for five different partitions of the selected region (Fig. 3. 6). The above strategy comes from the observation that in some cases the selected region contains two separate packets of energy that exclude an efficient separable representation. To avoid such situations and simultaneously improve the representation in the space of separable functions, one should eliminate one of many hypothetical functions by reducing the range of the decomposition.

Among the five separable functions obtained after the above stage, the separable function with the largest absolute value of expansion coefficient is selected. In the *FunctionRepresentation* stage, the selected normalised one-dimensional signals, which create the separable function, are quantized using a midtread uniform quantizer.

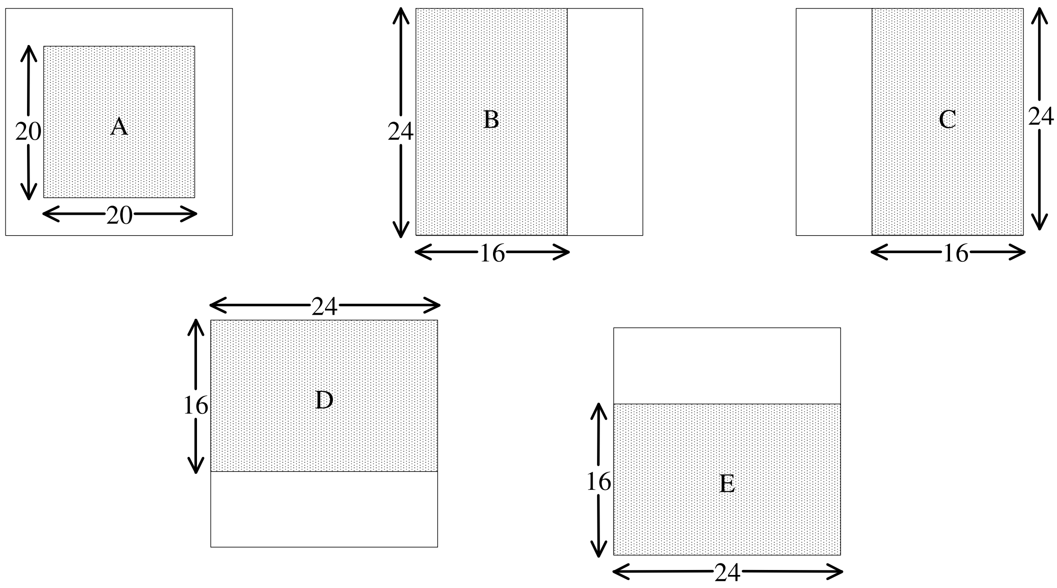


Fig. 3. 6. Regions of support for the separable decomposition.

In detail, this quantization process is performed as follows:

$$\tilde{a}(i) = \frac{NINT(a(i)F_{levels})}{K_{norm}} \tag{3.9}$$

where:

$a(i)$	-an input function,
$\tilde{a}(i)$	-quantized function,
F_{levels}	-quantization parameter,
K_{norm}	-constant which is defined to normalize an output function,
$NINT$	-the nearest integer value.

After the *FunctionRepresentation* process, a quantized separable function is obtained as a tensor product of two 1-D quantized signals. Then, the expansion coefficient is calculated and quantized with a stepsize $Q_{stepsize}$. At last, the atom is removed from the motion residual and an incoming new residual is submitted to the next approximation step. In accordance with rule of atoms post-selection, $2N_{atoms}$ atoms are generated but only N_{atoms} atoms are selected to represent an input signal.

3.3.2. Convergence of Separable Decomposition

In order to verify the convergence of an initial function to the optimum separable function in the separable decomposition process, some experiments were performed. In the experiments, five standard QCIF video test sequences were compressed using the above encoder implementation. In order to minimise influence of different parameters, only the N_{decomp} parameter was changed and all other parameters were fixed. The encoder was configured in such a way that the quantization processes had not influenced on the final results (i.e. F_{levels} was big and $Q_{stepsize}$ was very small). The first frame of each 10-seconds sequence was an I-frame and all the consecutive frames were P-frames. All P-frames in all sequences were coded using the same number of atoms (N_{atoms} =20). Let us remember that the separable decomposition started to approximate the selected region with constant functions.

The experiments proved that a separable decomposition approaches the calculated separable function to the optimal separable function in a few iterations. Increases of PSNR with respect to situations where the separable decomposition are performed in two steps are plotted in Fig. 3. 7. A very good approximation is achieved over eight iterations (Fig. 3. 7 and Fig. 3. 8). From a practical point of view, it seems profitable to perform 12 to 16 steps of a separable decomposition, since increasing the number of iterations does not compensate gains.

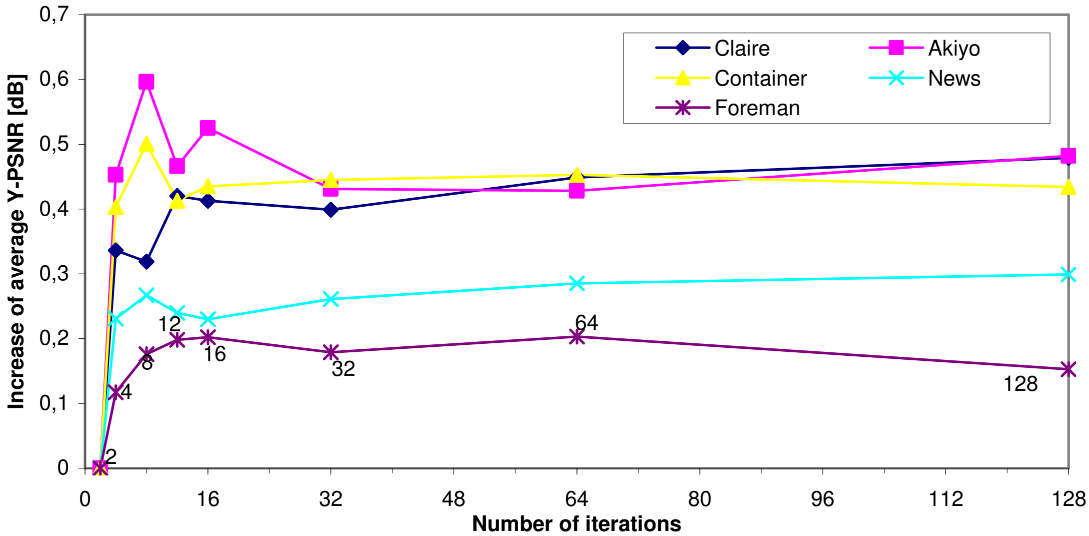


Fig. 3. 7. The increases of PSNR with respect to two-steps separable decomposition.

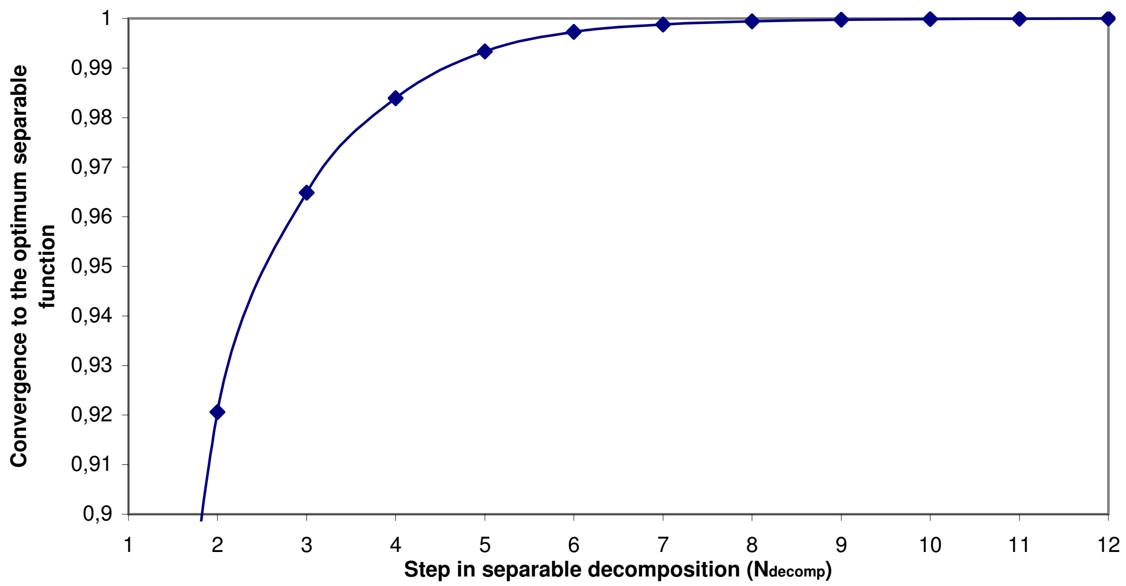


Fig. 3. 8. The average convergence in Akiyo sequence depending for a number of steps (N_{decomp}) in the separable decomposition.

It must be mentioned that fluctuations which can be seen on Fig. 3. 7 (particularly visible at Akiyo sequence) result from the motion-compensation process.

Let’s note, that used test sequences are characterised by different intensities of motion. The sequences can be grouped in three categories: the sequences with large static background (Akiyo, Claire, Container), the sequence with moderate motion intensity (News) and the sequence with very intensive scenes (Foreman). The above classification clearly corresponds with plot in Fig. 3. 7 and with individual increases of PSNR. The correlation can be expressed in the following way: the more motion intensive sequence, the less increase of PSNR value in successive iteration of separable decomposition. The author can not explain the reason for this. The most probable hypothesis assumes that this phenomenon depends on an initial function. In a sequence where the energy of a prediction error is dispersed, on the one hand, it is hard to find an efficient separable representation of region, on the other hand, the best separable representation is similar to a constant function. Since an error in a motion-predicted frame coming from a high motion sequence is dispersed, it is very probable that an initial constant function will approximate a chosen local region well.

3.3.3. Number of atoms. Scalability

A very important feature of each compression method is its scalability, i.e. how a given method changes their properties if an instance of the method grows up. In other words, it is the question about the area of application. In order to answer this question, some experiments with five standard QCIF video test sequences were performed. The encoder was configured in such a way that quantization processes had a meaningless importance (i.e. F_{levels} was big and $Q_{stepsize}$ was very small). The number of iterations N_{decomp} in the separable decomposition was set to 12 (according to the results from the previous section). The first frame of each 10-seconds sequence was an I-frame and all the consecutive frames were P-frames. All P-frames during a single test were coded using the same number of atoms. The results are shown in Fig. 3. 9.

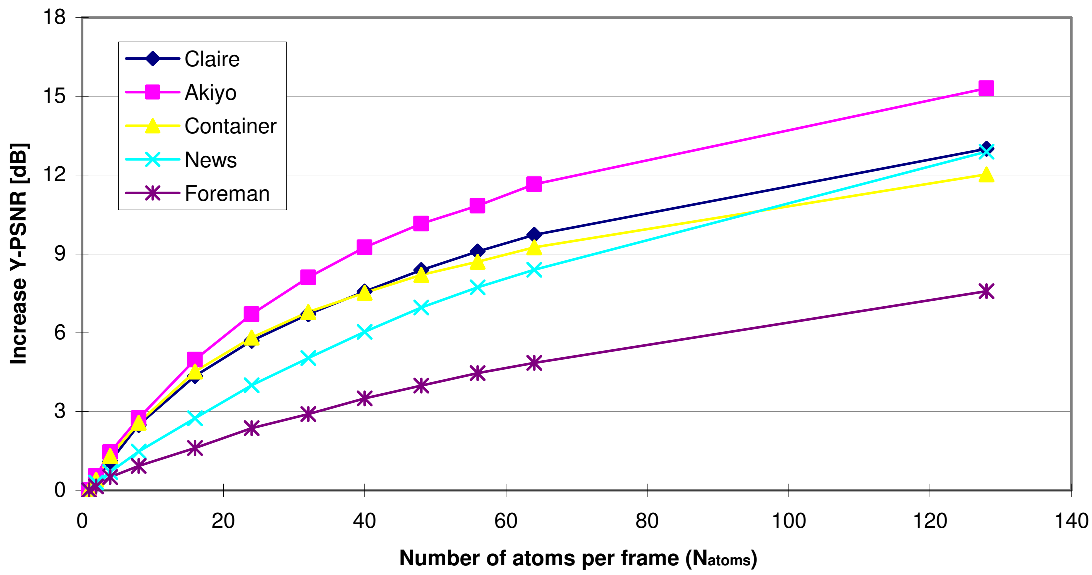


Fig. 3. 9. Scalability of matching pursuit with separable decomposition.

As can be seen, the video encoder based on matching pursuit with separable decomposition is characterised by good scalability. It must be mentioned that the MPwithSD technique similarly as the MP guarantees an asymptotic convergence to an approximated function ([Davis97] and Section 2.3.2), but simultaneously it does not mean that after any finite number of iterations the error will be close to zero [Mall93], [Davis97]. Nevertheless, utilisation of MPwithSD not only to a very low bit-rate system seems the advisable idea. At last, it is worth mentioning that the MPwithSD method is predisposed to construction of SNR scalable video systems, since its features result from the properties of atoms.

The algorithmic scalability of MPwithSD has been confirmed by experiments performed on still images [JPEG]. In the experiments, each JPEG test image was transformed using the separable decomposition to 2-D separable function of image size. Then, the obtained separable function was quantized and the expansion coefficient was calculated. The computed elements were used to represent whole 512×512 pixels image by a single atom. The increase of PSNR is presented on Fig. 3. 10. As can be seen on **Błąd! Nie można odnaleźć źródła odsyłacza.**, the quality of images is poor, nevertheless the content of images is legible. First of all, decoded pictures do not suffer from blocking artefacts.

Taking into consideration the uncompressed data needed to store images using 16 and 32 atoms correspond to 0.375 bpp and 0.75 bpp respectively, and it seems very promising for the method. In addition, the above experiments confirmed that 12 iterations for the separable decomposition ensure the representation very close to the optimal separable solution.

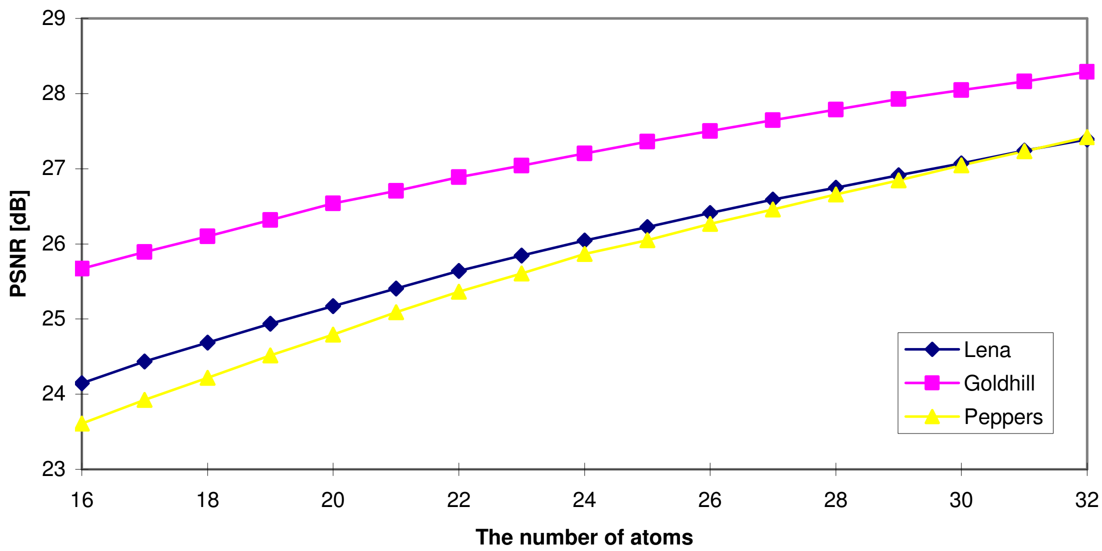


Fig. 3. 10. The PSNR increase for 512×512 pixels still images with respect to the number of atoms.

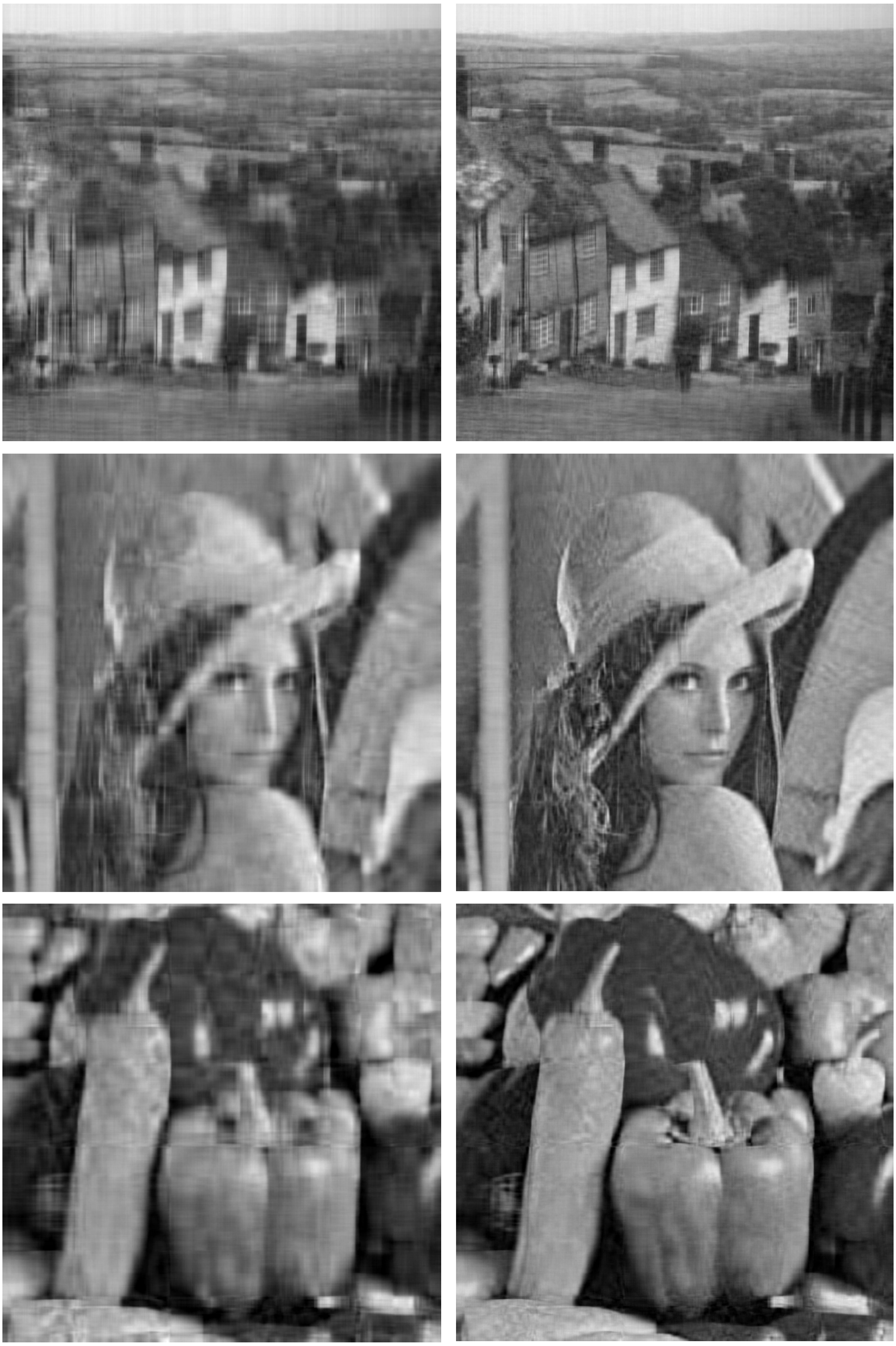


Fig. 3.11 The test images: Goldhill, Lena, Peppers encoded using separable functions. The left images are approximated using 16 2-D functions obtained by separable decomposition. The right images are approximated using 32 2-D separable functions. The number of iterations for separable decomposition was established to 14.

3.3.4. The Expansion Coefficient Quantization

Generally speaking, the expansion coefficient is a value that scales a normalised waveform from a dictionary in order to approximate a residual. In spite of the essential meaning of this component in the structure of an atom, the quantization of expansion coefficient is not often deeply discussed, although many works describe this problem in a trite way. In-depth analysis can be found in monograph [Neff99]. In order to confirm observations concerning the way of quantization, some experiments were performed. During these experiments, expansion coefficients were split on modulus values and signs. As a result of this, the table of VLC codes was reduced to half. At the beginning of the experiments, expansion coefficients was quantized using uniform midtread quantizer with the stepsize $Q_{stepsize}$ as follows:

$$\tilde{\alpha} = NINT\left(\frac{|\alpha|}{Q_{stepsize}}\right). \qquad (3.10)$$

The process of inverse quantization of expansion coefficient was defined by:

$$\alpha = sign(\alpha)\, Q_{stize}\tilde{\alpha}\,, \qquad (3.11)$$

where:

$Q_{stepsize}$	-stepsize of quantizer,
α	-expansion coefficient,
$\tilde{\alpha}$	-quantized expansion coefficient,
$NINT$	-the nearest integer value.

The probability of quantized expansion coefficients are presented on Fig. 3. 11 and Fig. 3. 12. As can be seen, the probability functions are similar to the Gaussian function. The inclination of curve depends on stepsize value $Q_{stepsize}$; the bigger value of quantization the higher density of probability round about a median. Nevertheless, it is known that matching pursuit and matching pursuit with separable decomposition guarantee an asymptotic convergence to an input signal, therefore, one expects that the probability of quantized expansion coefficients will be similar to the Laplace function.

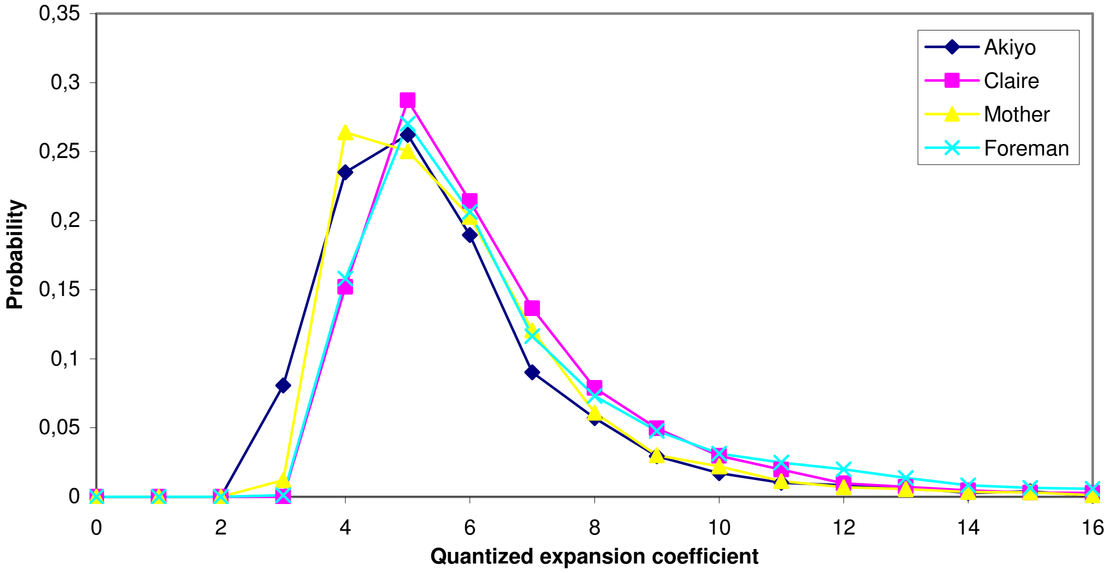


Fig. 3. 11. Probability of quantized expansion coefficients for video encoder based on the matching pursuit with separable decomposition with $Q_{stepsize}$=8.

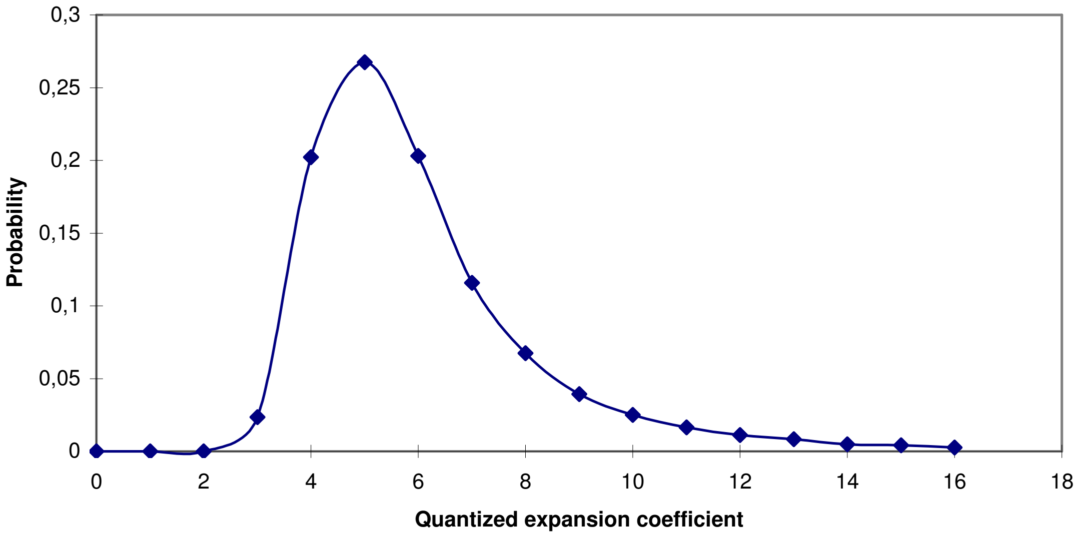


Fig. 3. 12. An average probability of quantized expansion coefficient for the matching pursuit with separable decomposition with $Q_{stepsize}$=8.

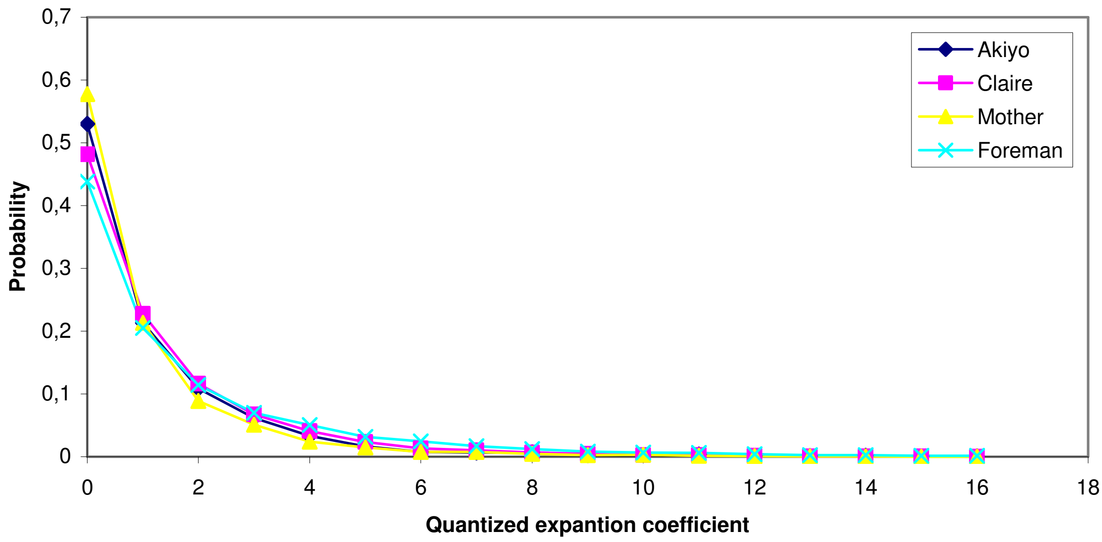


Fig. 3. 13. Probability of quantized expansion coefficients for video encoder based on the matching pursuit with separable decomposition with $Q_{stepsize}$=8 and adaptive dead-zone.

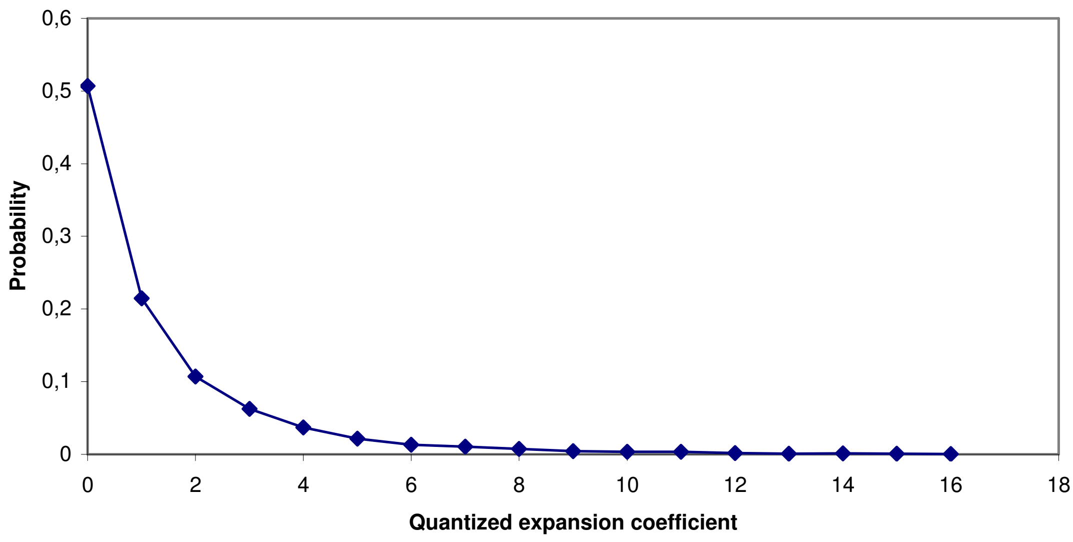


Fig. 3. 14. An average probability of quantized expansion coefficient for the matching pursuit with separable decomposition with $Q_{stepsize}$=8 and adaptive dead-zone.

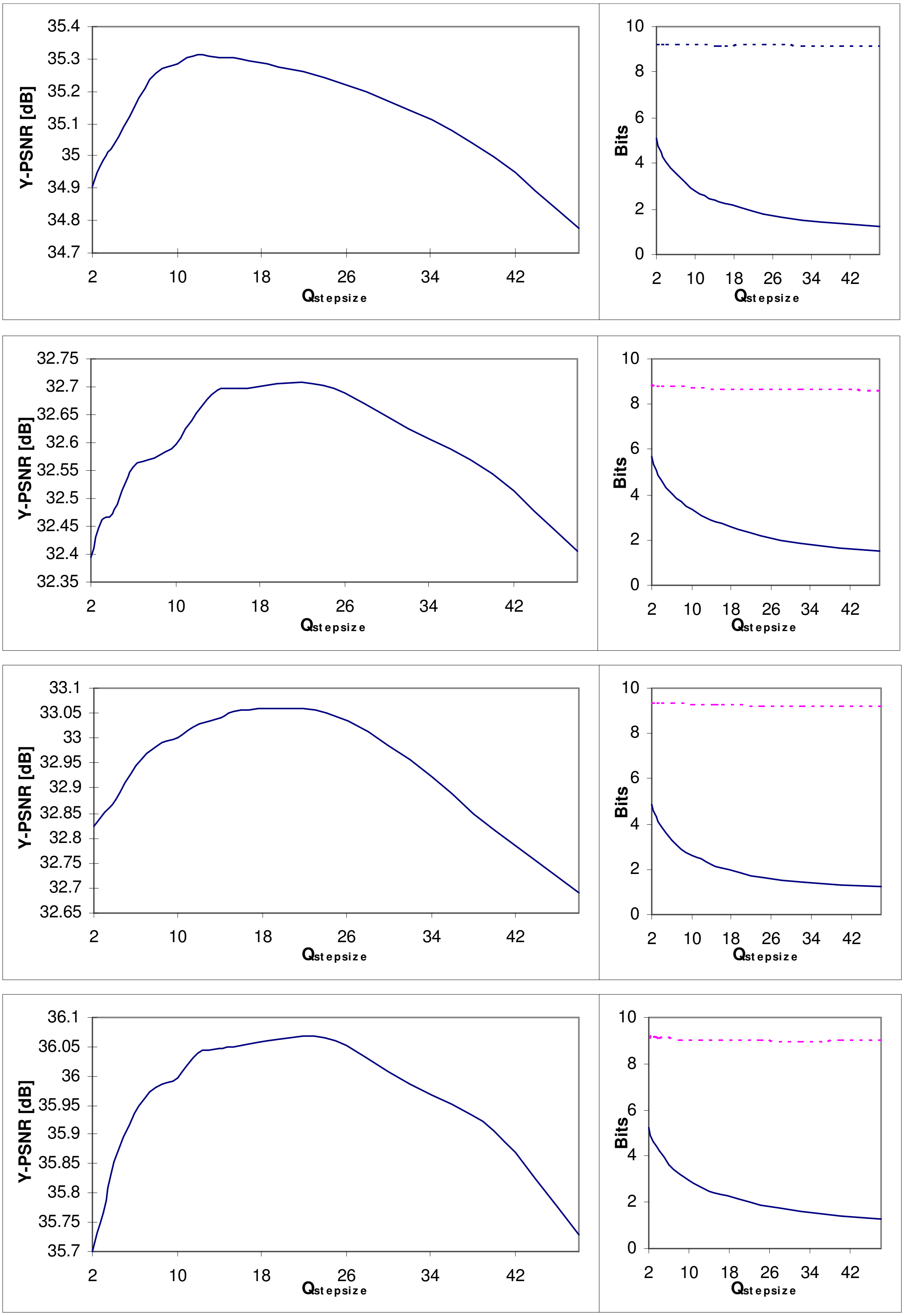


Fig. 3. 15. **The changes in Y-PSNR (left plots), the number of bits required to encode a position of an atom (dotted line on right plots) , and the number of bits required to encode a modulus (solid right plots) depending on the stepsize $Q_{stepsize}$ for test sequences: Akiyo, Foreman, Mother and Claire.**

In order to improve the ratio of compression, one should supplement a process of quantization with an adaptively calculated dead-zone. This technique was implemented in the next version of the encoder in which the value of dead-zone for each frame of sequence was obtained after the post-selection process.

The smallest absolute value of expansion coefficients is taken as the value of the dead-zone. Then, the quantization process is performed as follows:

$$\tilde{\alpha} = \left\lfloor \frac{|\alpha| - DeadZone}{Q_{stepsize}} \right\rfloor. \tag{3.12}$$

The process of inverse quantization of expansion coefficient is defined by:

$$\alpha = sign(\alpha)\left(Q_{stize}(\tilde{\alpha} + 0.5) + DeadZone\right), \tag{3.13}$$

where:

$Q_{stepsize}$	-stepsize of quantizer,
α	-expansion coefficient,
$\tilde{\alpha}$	-quantized expansion coefficient,
$\lfloor \cdot \rfloor$	-the biggest integer value not greater than an input value,
$DeadZone$	-quantizer dead-zone.

As can be seen on Fig. 3. 13 and Fig. 3. 14, the curves of probability are similar to the curve of Laplace function. In accordance with an exponential character of approximation in MP and MPwithSD, the above form of the probability is very profitable - especially from a compression point of view. Increasing of $Q_{stepsize}$ influences on the probability and improve the rate of compression. On the other hand, the increase of quantizer step-size affects the quality of the sequence, i.e. causes an increase of distortions. To solve this classical two-dimensional problem some experiments were performed. Fig. 3. 15 shows dependencies between Y-PSNR and $Q_{stepsize}$. It must be mentioned that the change in quantizer step-size value influences the number of bits required to encode the modulus of an expansion coefficient (see right plots in Fig. 3. 15) and it indirectly allows for the representation of an input signal using different number of atoms. The experiments reveal that the range of profitable values for quantizer step-size is very wide (since for $8 \le Q_{stepsize}$ and $Q_{stepsize} \le 24$, the changes in Y-PSNR are very small). In spite of the quite different concept for the quantization process, this conclusion coincides with the observations from monograph [Neff99].

Further analysis of the quantization error narrows down the range for step-size values towards lower numbers. Let α_k be an expansion coefficient in *k*-th step in MPwithSD (or MP). Let d_k be a quantization error in *k*-th step:

$$d_k \in \left[-\frac{Q_{stepsize}}{2}, \frac{Q_{stepsize}}{2} \right], \tag{3.14}$$

where:

$Q_{stepsize}$ -stepsize of quantizer.

In accordance with (2.8) and (2.9), energy of a residual after N steps may be expressed in the following way:

$$\|R_N f\|^2 = \|f\|^2 - \sum_{k=1}^{N} \alpha_k^{\ 2} + \sum_{k=1}^{N} d_k^{\ 2} . \tag{3.15}$$

This important formula means that the energy of the residual changes proportionally to the square of the expansion coefficients. Simultaneously, the energy is increased by a square of quantization errors [Mall93], [Davis97], [Fross01]. It is easy to lead out the formula for the total error $E_{Total}(\cdot)$:

$$E_{Total}(Q_{stepsize}) = \sum_{k=1}^{N} d_k^{\ 2} \leq \sum_{k=1}^{N} \left(\frac{Q_{stepsize}}{2} \right)^2 = \frac{N\, Q_{stepsize}^{\ 2}}{4} . \tag{3.16}$$

It is worth mentioning that the upper bound of $E_{Total}(\cdot)$ depends only on the value of $Q_{stepsize}$. Nevertheless, one should stress that meaning of this expression is relative to values of expansion coefficients and indirectly to the quality of a compressed sequence. Note that values of expansion coefficient are proportional to a distortion and for that reason the total error has different strength (see Fig. 3. 16). The above aspect demonstrates that a value of quantizer step-size $Q_{stepsize}$ should be dependant of a dead-zone, since the *DeadZone* parameter is a pseudo-measure of a prediction error. On the other hand, in the context of the previous experiments, the wide range of admissible values for $Q_{stepsize}$ allows it to avoid additional computation and fix the value of quantizer step-size to 12. In this way, a fixed sub-optimal solution is obtained.

As can be seen on Fig. 3. 17, the changes in Y-PSNR for test sequences in a real implementation correspond to the theoretical results from Fig. 3. 16.

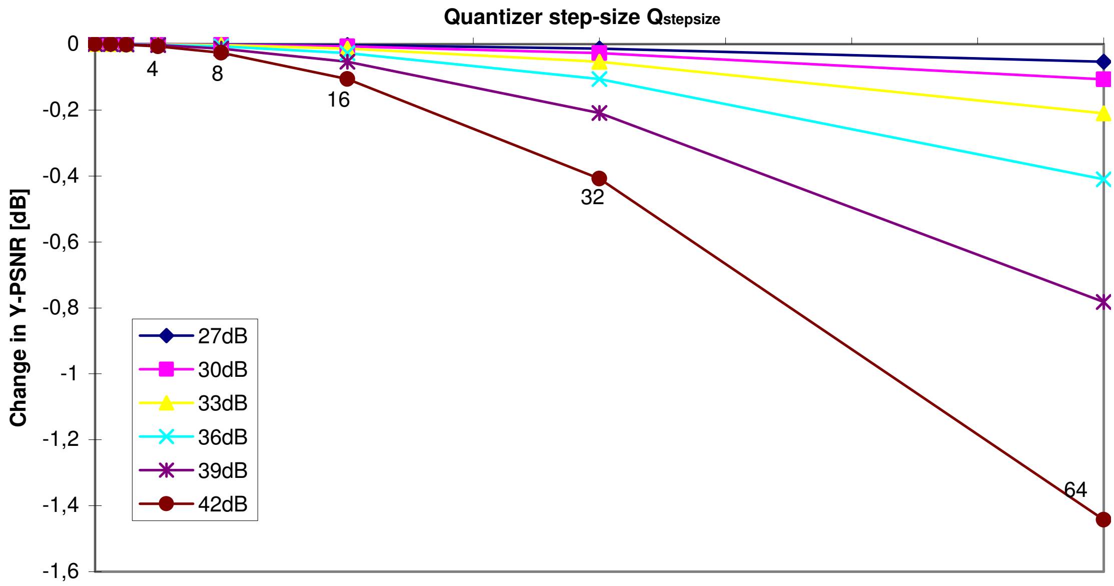


Fig. 3. 16. Theoretical upper bounds for changes in Y-PSNR depending on the value of quantizer step-size $Q_{stepsize}$. An additional assumption is that each P-frame is represented using 40 atoms.

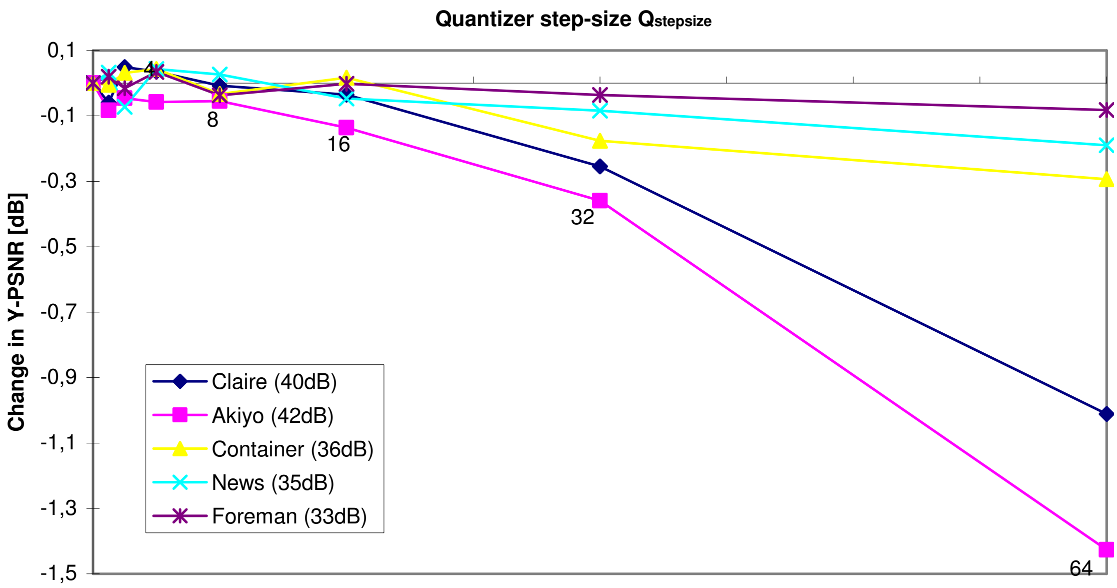


Fig. 3. 17. Changes in Y-PSNR depending on the value of quantizer step-size $Q_{stepsize}$. The average Y-PSNR for each sequence is placed in brackets near by the name of the sequence. Each P-frame of each sequence was represented using 40 atoms.

3.3.5. Function Quantization

Probably the most interesting but surely the most important parameter is the number of reconstruction levels of linear quantizer in the *FunctionRepresentation* (F_{levels}). In accordance with (3.9), the process of quantization for 1-D functions in the described test implementation is performed as follows:

$$\tilde{a}(i) = \frac{NINT(a(i)F_{levels})}{K_{norm}}, \tag{3.17}$$

where:

$a(i)$	-an input function,
$\tilde{a}(i)$	-quantized function,
F_{levels}	-quantization parameter,
K_{norm}	-constant which is defined to normalize an output function,
$NINT$	-the nearest integer value.

As was mentioned in Section 3.3.1, the best found separable function $f_{sep}(i,j)$ is quantized in the *FunctionRepresentation* process into two separate steps. Let $a(i)$ and $b(j)$ denote 1-D normalised functions such that:

$$f_{sep}(i,j) = a(i)b(j). \tag{3.18}$$

Furthermore, let:

$$\tilde{a}(i) = a(i) + d_a(i) \ , \tag{3.19}$$

and

$$\tilde{b}(j) = b(j) + d_b(j), \tag{3.20}$$

where:

$$d_a(i) \in \left(-\frac{1}{2F_{leves}}, \frac{1}{2F_{levels}} \right], \tag{3.21}$$

$$d_b(j) \in \left(-\frac{1}{2F_{leves}}, \frac{1}{2F_{levels}} \right]. \tag{3.22}$$

In order to study the behaviour of functions during the quantization process, the measure of functions similarity has been defined as follows:

$$s(f_1, f_2) = \frac{\left|\left\langle f_1, f_2 \right\rangle\right|}{\|f_1\|\|f_2\|}. \tag{3.23}$$

In our case, the above measure can be reduced to equivalent forms:

$$s(a, \tilde{a}) = 1 - \frac{1}{2}\sum d_a{}^2, \tag{3.24}$$

$$s(b, \tilde{b}) = 1 - \frac{1}{2}\sum d_b{}^2. \tag{3.25}$$

It is easy noticing, that:

$$s(f_1, \tilde{f}_1) \geq 1 - \frac{1}{2}\sum\left(\frac{1}{2F_{levels}}\right)^2 = 1 - \frac{F}{8F_{levels}{}^2} \tag{3.26}$$

(where F - is the number of non-zero coefficients in function).

The short analysis of (3.26) allows it to admit that for $F_{levels} \geq 16$, a similarity of function to its quantized version is sufficient to represent an input signal without perceptible differences (since for commonly used F=20, a value of similarity measure $s(\cdot,\cdot)$ is greater than 99%). In order to verify an influence of a function quantization process on a quality of representation, some experiments were performed. In the experiments, five standard QCIF video test sequences were compressed with the assistance of the above encoder implementation. The first frame of each 10-second sequence was an I-frame and all the consecutive frames were P-frames. All P-frames in all sequences were coded using the same number of atoms (N_{atoms}=40). The number of iterations N_{decomp} in the separable decomposition was set to 12. The quantization of expansion coefficients was switched off (i.e. $Q_{stepsize}$ was very small).

As can be seen on Fig. 3. 18, the results confirm the above supposition. The degradation of quality for $F_{levels} \geq 16$ is insignificant, and simultaneously for $F_{levels} < 16$ becomes more and more visible.

Further analysis performed for function quantization process is related to an expansion coefficient, which may be a pseudo-measure of a representation quality. Let us assume that:

$$a = \left\langle f, f_{sep} \right\rangle, \tag{3.27}$$

where f is a certain local region of an input signal. For the same region f and for the quantized separable function $\tilde{f}_{sep}$, the new value of expansion coefficient may be expressed in the following way:

$$\left\langle f, \tilde{f}_{sep} \right\rangle \approx \alpha - \frac{\alpha}{2} \sum d_a^{\,2} - \frac{\alpha}{2} \sum d_b^{\,2} \,. \tag{3.28}$$

As can be seen, the degradation strongly depends on an absolute value of the expansion coefficient a and on the parameter of quantization F_{levels}. This observation shows that the process of quantization is very complex and has different meanings depending on energy brought by the function. Due to this, the next chapter of this dissertation is devoted to the *FunctionRepresentation* process.

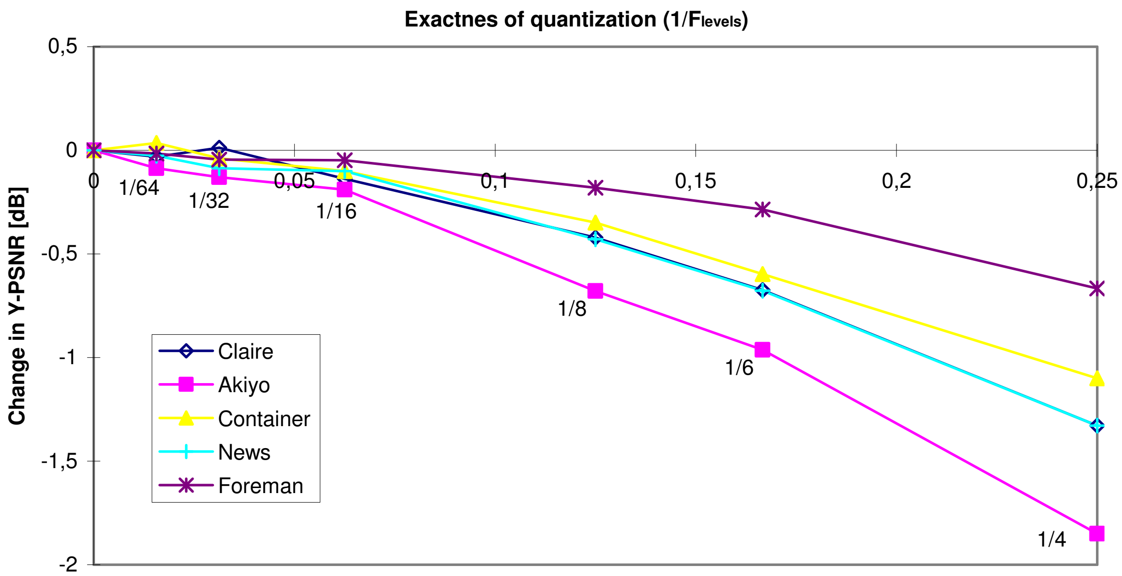


Fig. 3. 18. Influence of function quantization on PSNR.

3.4. Conclusions

First of all, the experiments show that the matching pursuit with separable decomposition is a very promising technique. In addition, the results proved that the assumption, which concerns the representation with the assistance of separable functions, was a proper choice.

One way or another, matching pursuit with separable decomposition solves some problems and simultaneously opens others. A large scalability of method (see Section 3.3.3) allows us to assume that the application of the MPwithSD should not be limited to very low bit-rate systems. Satisfactory convergence of separable decomposition (see Section 3.3.2) guarantees low computational complexity. It is worth mentioning that the algorithm of separable decomposition can be simply performed in assistance of SIMD (single instruction / multiple data) processing model (e.g. MMX family). For further analyses and experiments, it is assumed that twelve iterations for the separable decomposition are sufficient to obtain optimal representation.

Although matching pursuit with separable decomposition transforms the initial problem into the N one-dimensional problems, it still does not solve the question concerning the efficient representation of separable functions. Partial answers to this question are contained in the next chapter.

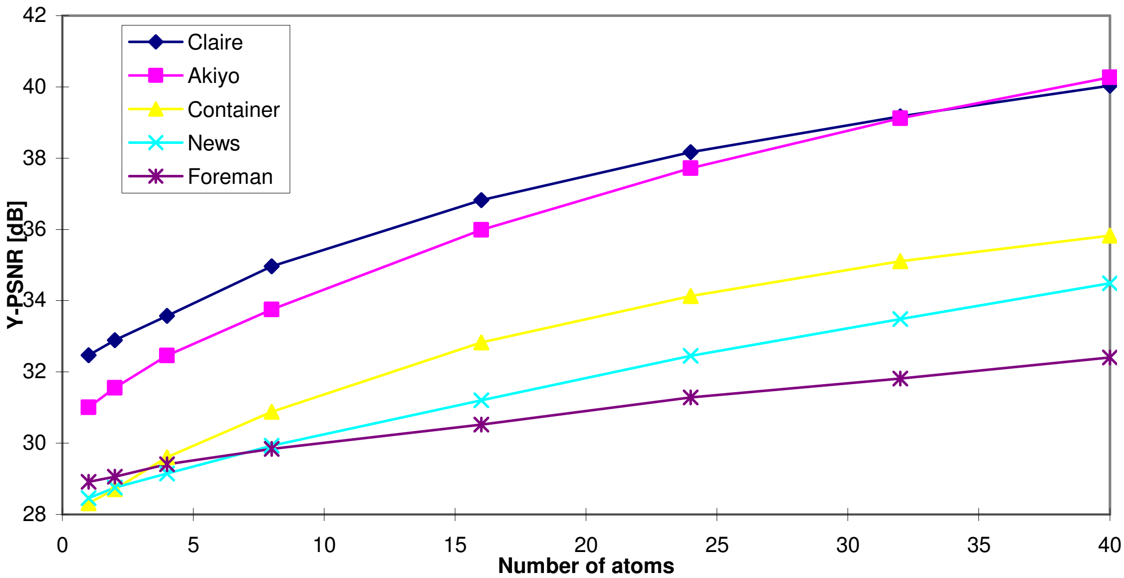


Fig. 3. 19. The final results for video encoder using matching pursuit with separable decomposition.

At the end, the results obtained from test implementation are plotted in Fig. 3. 19. Comparison of the results presented in the previous chapter demonstrates that similar PSNRs may be achieved with the assistance of a few atoms.

Chapter 4

Matching Pursuit with Quantized Separable Decomposition

4.1. Introduction

Efficient representation of signals is the main purpose of data compression. Although this goal is easy to define, nevertheless each fundamental solution for the above issue constitutes many additional problems, which appear around the main technique. Generally speaking, each main technique used in a data compression system represents or approximates an input signal in a way that is profitable from a compression point of view. After the above transformation, the resulting new form of input signal is quantized either in order to remove some information, which is not important in the reconstruction, or in order to retain the signals with indispensable features. The obtained quantized signal must be efficiently coded using several lossless techniques. As can be seen, the general description shows that in data compression systems at least two additional problems accompany the main technique, they are:

- how to perform the quantization process,
- how to efficiently encode the quantized signal.

In fact, the above additional problems are very important because in these steps a fundamental compression is performed. The first process removes subjective redundancy and influences the quality of representation. The latter process exploits statistical redundancy in order to store the entire quantized information in a very compact way. One way or another, considering transform coding, one should distinguish three steps:

- transform calculation,
- quantization of transform coefficients,
- modelling and lossless encoding of quantized data.

It must be mentioned, that the main task of the transform calculation in compression systems is the decorrelation of the input data i.e. minimisation of the number of essential coefficients. The decorrelated signal has properties that are desirable from a compression point of view; nevertheless, the transform calculation does not perform any direct compression steps.

The separable decomposition, which is used in the matching pursuit algorithm, satisfies double task since it not only transforms an input signal into a profitable form but also performs some compression steps. Note that transformation of a two-dimensional signal of size $N \times N$ into two-dimensional separable space allows different representation since, each 2-D separable function of size $N \times N$ can be represented by two 1-D functions of size N. This means that an exact representation of 2-D separable function suffices to store the 2N coefficients instead of the N^2 ones.

The separable decomposition used in the matching pursuit scheme gives in fact a new technique called the matching pursuit with separable decomposition. In a particular case, the separable decomposition can be used in the matching pursuit algorithm strictly for finding an atom to speed-up this process. This chapter is devoted to the above issue. Nevertheless, it must be mentioned that the matching pursuit is a very general algorithm therefore, the application of the separable decomposition in this scheme results from the assumption that all functions in the dictionary are separable.

4.2. The Matching Pursuit with Separable Representation with a Priori Known Dictionary

The previous chapter describes the modification of the matching pursuit scheme that introduces the application of the separable decomposition. It also contains information about the main elements and properties that influence the behaviour of the proposed technique, such as:

- the convergence of separable decomposition to the optimum separable representation (Chapter 3. 3. 2),
- the scalability of method i.e. how the number of atoms influences the quality of approximation (Chapter 3. 3. 3),
- the manner of quantization of expansion coefficients (Chapter 3. 3. 4),
- the influence of quantization on the quality of approximation (Chapter 3. 3. 5).

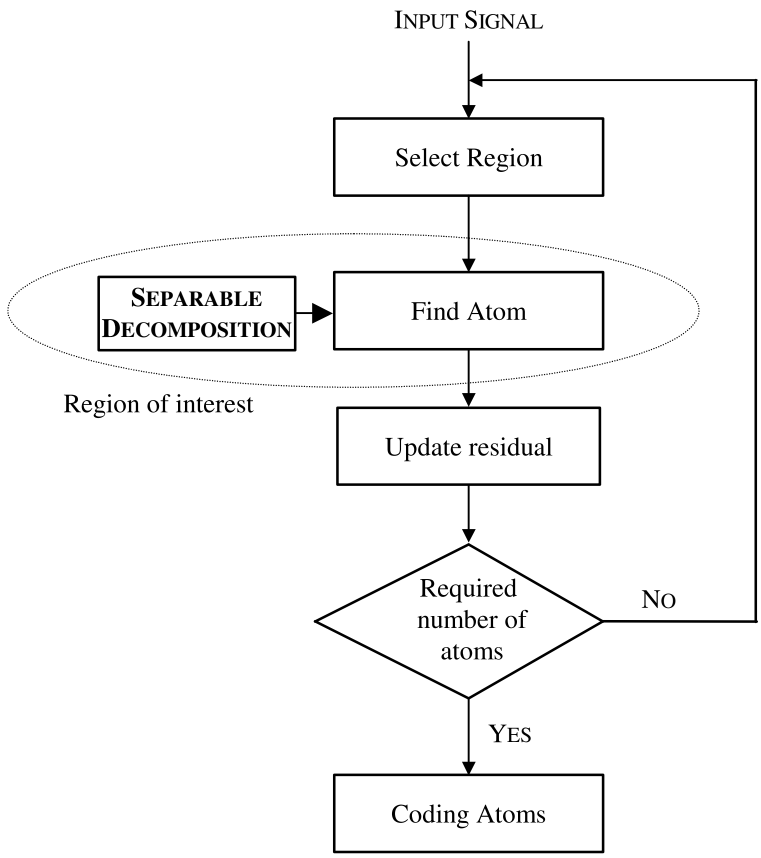


Fig. 4. 1. Matching pursuit with separable decomposition.

The previous chapter gives the answers to many of the above topics. In particular, it allows us to fix the number of iterations in the separable decomposition. In addition, the efficient quantization method of expansion coefficients has been proposed and described. Nevertheless, the most flexible part of the MPwithSD algorithm is the part that is responsible for a representation of one-dimensional signals. As has been shown, this process is very sensitive to the level of quantization. In addition, an improper or rough quantization can rapidly decrease the value of the objective quality criteria (PSNR). For that reason, the process of function representation plays a crucial role in the matching pursuit with separable decomposition strategy.

In a case when an input signal must be represented using *a priori* known separable functions, the matching pursuit with separable decomposition seems to be a good alternative to the classic matching pursuit. In such a solution, obtained 1-D functions should be efficiently represented with the assistance of a known set of functions. This observation coincides with the thesis. The main goal of this dissertation is to prove that when utilising the separable property of functions in the dictionary, it is possible to design a strategy for matching pursuit that allows for reducing the computation complexity of the algorithm with a small decrease on efficiency. Such a strategy can be obtained, if the 1-D waveforms from the dictionary will be used to represent one-dimensional signals at the stage of function representation in the matching pursuit with separable decomposition algorithm (see Fig. 4. 2).

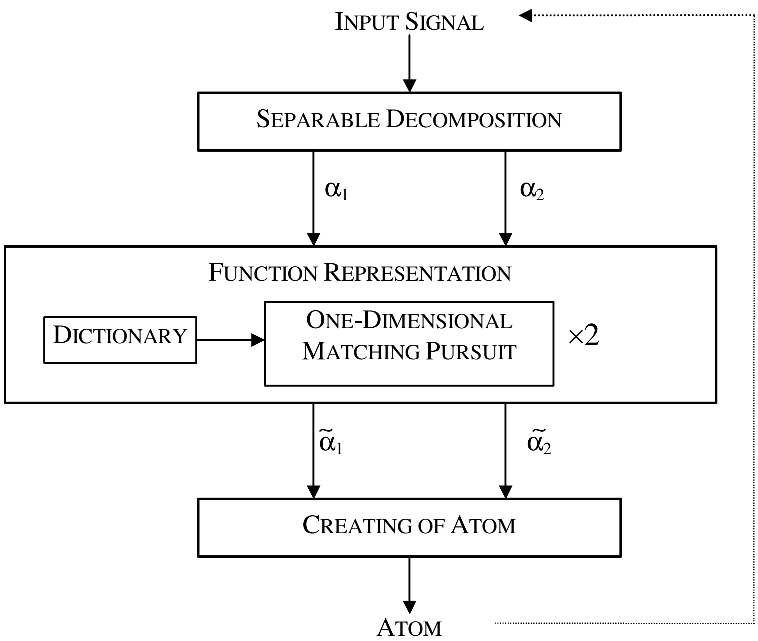


Fig. 4. 2. The proposed implementation of finding atom in the matching pursuit with separable decomposition.

The best representation of a one-dimensional signal using a single waveform may be obtained by computing inner products of the 1-D input signal with respect to all dictionary functions that can be located at any point within the input signal. In fact, this is a single step in the matching pursuit algorithm. After the above process, which must be performed twice, the indices of waveforms are obtained.

It must be mentioned that in spite of the optimality of the 2-D separable function obtained after the separable decomposition, the optimal separable representation of an input signal with respect to the 2-D separable dictionary is not guaranteed. This phenomenon can appear not only for a true input signal but also for the separable representation of an input signal. Note that approximation of a vertically oriented function may introduce some artefacts which change the optimal form of a corresponding horizontally oriented component, and *vice versa*. Let us remember that the main goal concerns the approximation of a true input signal, and also that the process of approximation of an input signal is only temporarily replaced by another approximation process which refers to the optimal separable function of an input signal. Due to this, after the approximation of a horizontally or vertically oriented function, the corresponding complementary function should be recalculated and then approximated. This gives us at least two models of approximation - i.e. models that comply with correlation of 1-D functions and models that attempt approximation independently for each 1-D function.

In order to compare how a model of approximation influences the value of the objective quality criteria, some experiments were performed. For this purpose, two models of function representation were implemented:

- the simple model - in which one-dimensional functions are approximated independently of each other,
- the conjunctive model - in which better approximation among the approximations of 1-D signals is chosen at first. Then, the latter function is recalculated and represented by a function from the dictionary.

The experiment (Fig. 4. 3) reveals two aspects:

- the conjunctive model is about 30% more computationally intensive,
- the conjunctive model is about 0.15 dB better than the simple model.

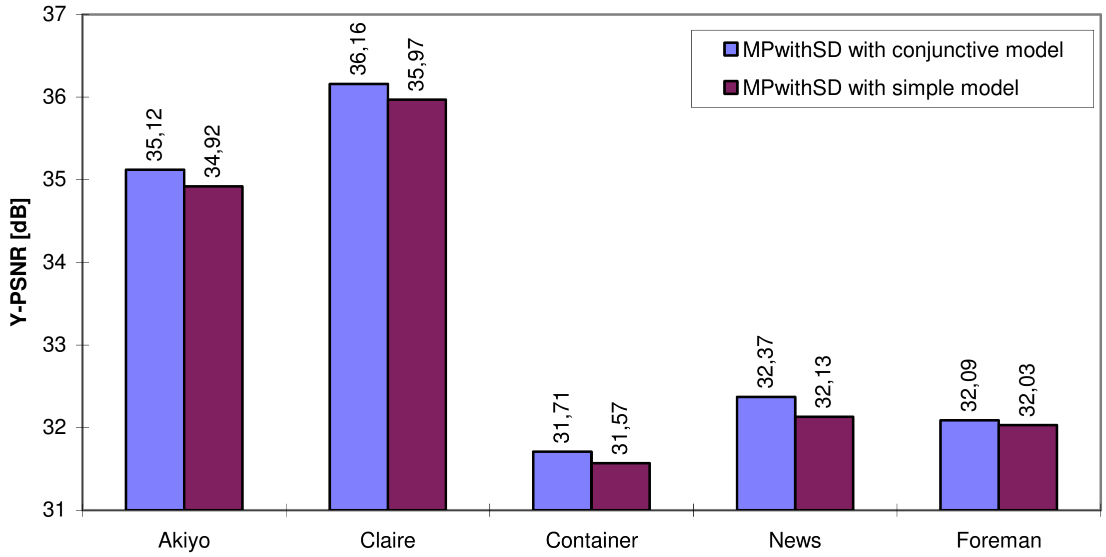


Fig. 4. 3. The comparison of function representation models.

For further experiments, the conjunctive model has been selected as the model that is more similar in behaviour to the pure matching pursuit. In addition, the above experiments showed that approximation in separable space is close to optimal only if the representation of one-dimensional functions does not introduce too many artefacts. Let us assume that the similarity of functions is measured in the following way:

$$s(f_1, f_2) = \frac{\left|\langle f_1, f_2 \rangle\right|}{\|f_1\|\|f_2\|}, \qquad (4.1)$$

where:

f_1 - 1-D function obtained from separable decomposition,

f_2 -1-D properly displaced waveform from the dictionary.

In the above experiment, the average value of measure of similarity between 1-D function f_1 and its best approximation f_2 amounted to about 75%. Under these conditions, it seems that degradation of the PSNR is not substantial and, in some cases, the simple model of representation can be used in order to speed-up the process of approximation.

4.3. Results of Experiments

The main goal of the experiments was to confirm the thesis of this dissertation. For this purpose, four different encoders were built:

- [MP] - In this implementation, the displaced frame difference is encoded using a pure matching pursuit algorithm with a fast inner product method (see Chapter 2).
- [MP+S] - It is an improved version of the [MP] implementation, in which the Schwartz inequality is used in order to reduce the computational load (see Section 2.5).
- [MP+S+Post] - It is an improved version of the [MP+S] implementation (see Section 2.5.1).
- [MPwithSD] - In this implementation, a prediction error is encoded using the matching pursuit with separable decomposition algorithm (see Section 4.2). In addition, the post-selection of atoms is used.

In this experiment, five standard QCIF video test sequences were compressed with the assistance of the above encoder implementations. The first frame of each 10-second sequence was an I-frame and all the consecutive frames were P-frames. All P-frames in each sequence were coded using the same number of atoms.

The figure Fig. 4. 4 presents the computational load for four different methods of error prediction coding i.e. it contains the amount of time that was devoted to approximate a displaced frame difference. The experiment showed that the representation of a prediction error with the assistance of the matching pursuit with separable representation method (i.e. [MPwithSD]) is 2.8 times faster than the fastest matching pursuit implementation (i.e. [MP+S]). If we take into consideration the fact that the conjunctive model was taken in the experiments, it will be clear that the MPwithSD algorithm with a simple model will be four times faster than the [MP+S] algorithm.

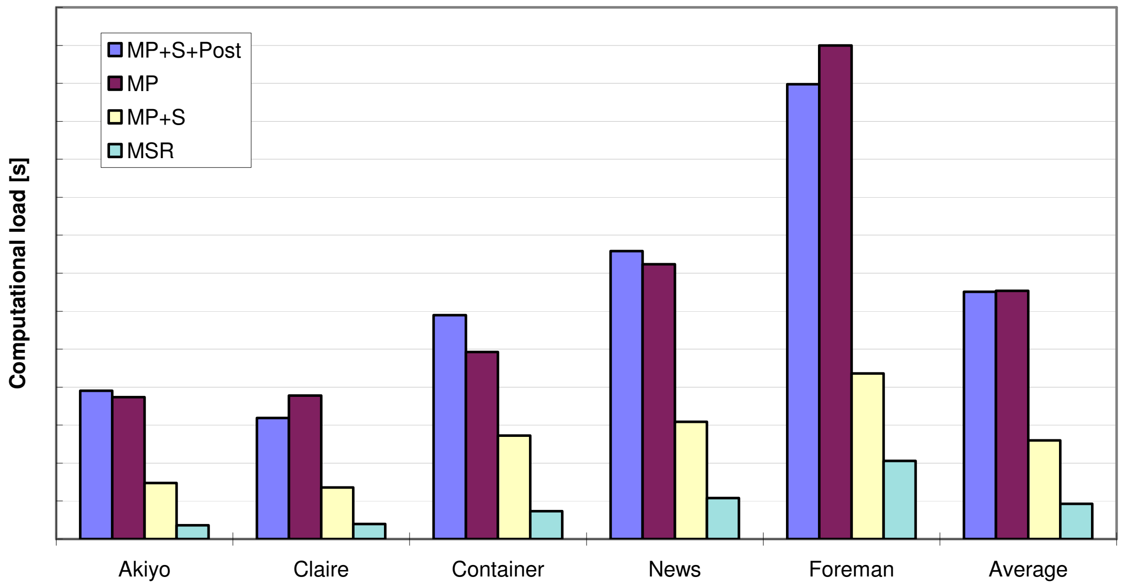


Fig. 4. 4. The computational load comparision of different error prediction coding implementations. The Akiyo and Claire were coded using 40 atoms. The sequences Container, News and Foreman were coded using 50, 75 and 110 atoms respectivly.

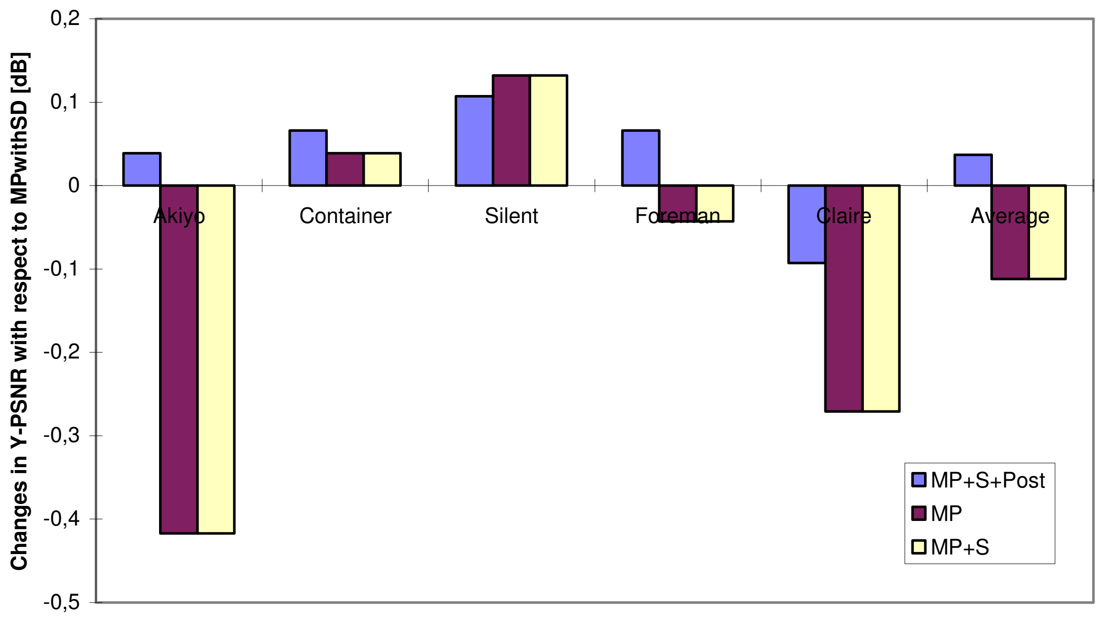


Fig. 4. 5. The Y-PSNR comparision with respect to MPwithSD implementation.

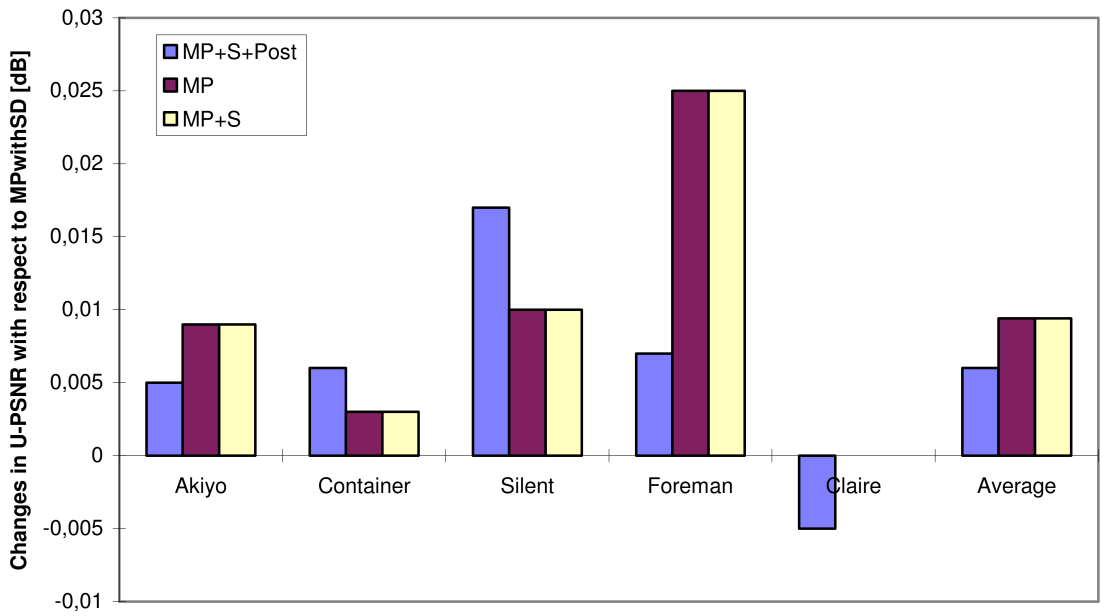


Fig. 4. 6. The U-PSNR comparision with respect to MPwithSD implementation.

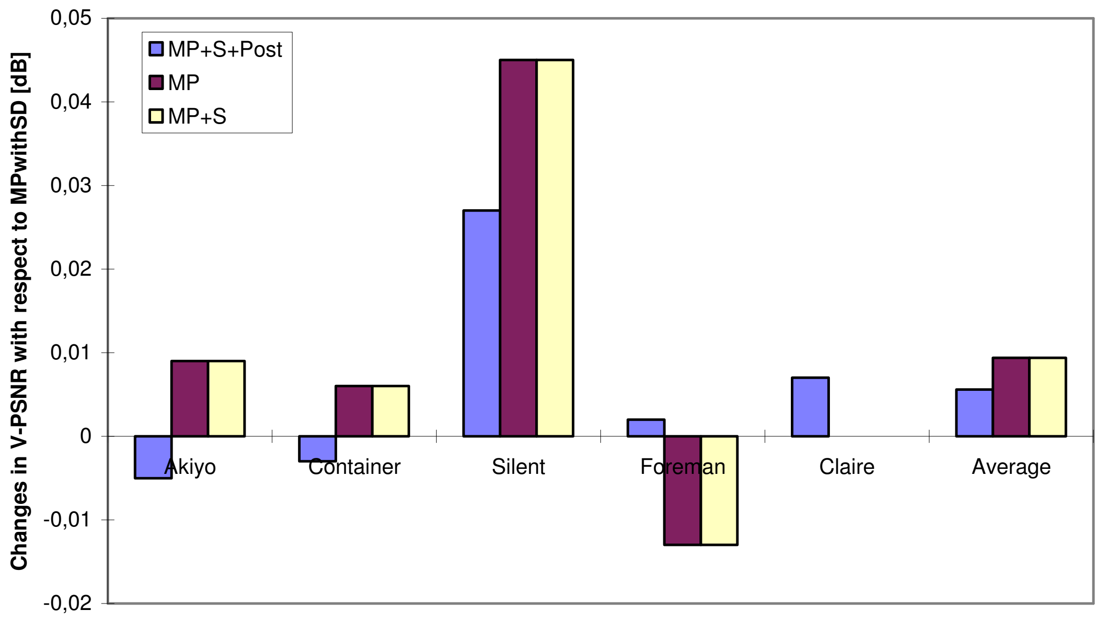


Fig. 4. 7. The V-PSNR comparision with respect to MPwithSD implementation.

On the other hand, the [MP+S+Post] implementation is about 0.1 dB better than the matching pursuit representation algorithm. However, the computational load of this method is 7 times greater than the [MPwithSD] algorithm (or 10 times greater if we consider the simple model).

The experiment showed that the use of Schwartz inequality in the matching pursuit algorithm causes that a computational load is strongly reduced but the reduction is unpredictable.

Further results showed that the probability for function indices (Fig. 4. 8 and Fig. 4. 9) and average values of expansion coefficients (Fig. 4. 10 and Fig. 4. 11) are very similar to each other. It means that the matching pursuit with separable decomposition does not introduce characteristic artefacts and the algorithm behaves like the matching pursuit method.

The experiment reveals that the dictionary proposed by R. Neff and A. Zakhor in [Neff96], [Neff96a], [Neff96b], [Neff97] is not uniformly useful, i.e. in particular, some functions are extremely rarely used. The explanation of this is that the used dictionary was designed for a MPEG-2 video hybrid encoder, which has slightly different properties and furthermore, a prediction error has different features. Due to this, there is a need for a redesign of the dictionary for the H.26L/AVC video system.

It is worth mentioning that functions that have narrow region of support are more frequently to be used. This observation coincides with the small size of the block in the motion estimation process and the transform coding in the AVC encoder.

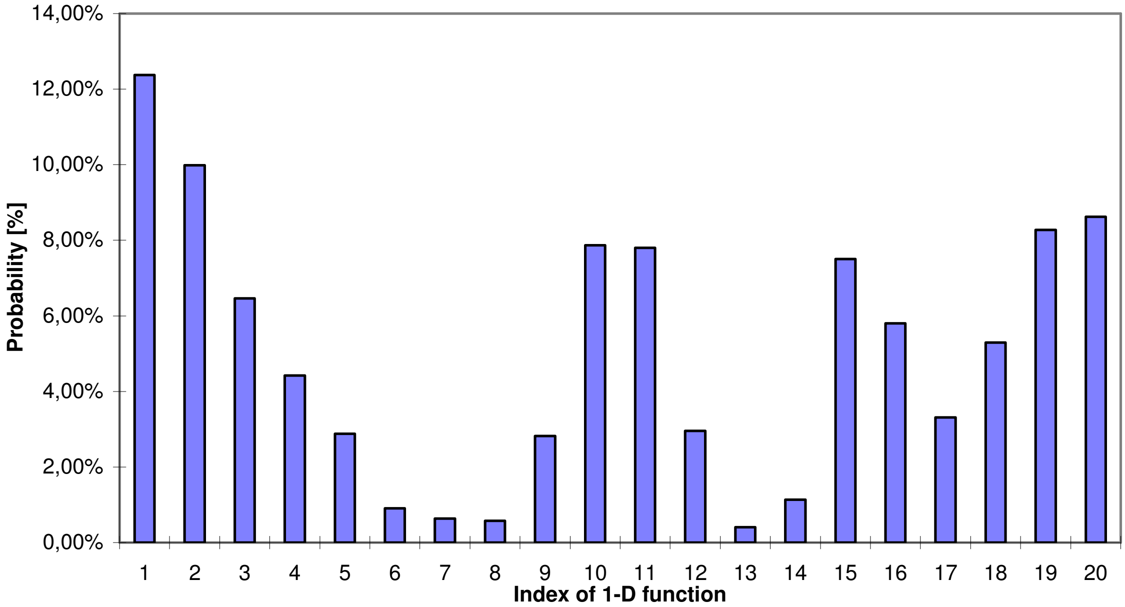


Fig. 4. 8. The histogram of function indexes in the matching pursuit [MP+S+Post] for Neff&Zakhor dictionary.

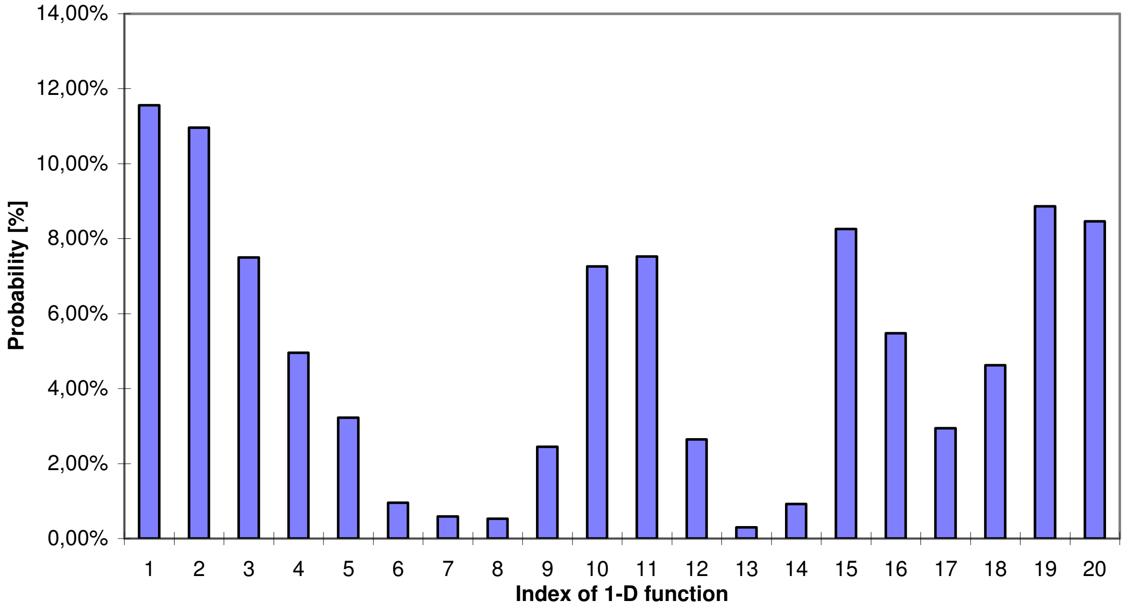


Fig. 4. 9. The histogram of function indexes in the matching of separable representation [MPwithSD] for Neff&Zakhor dictionary

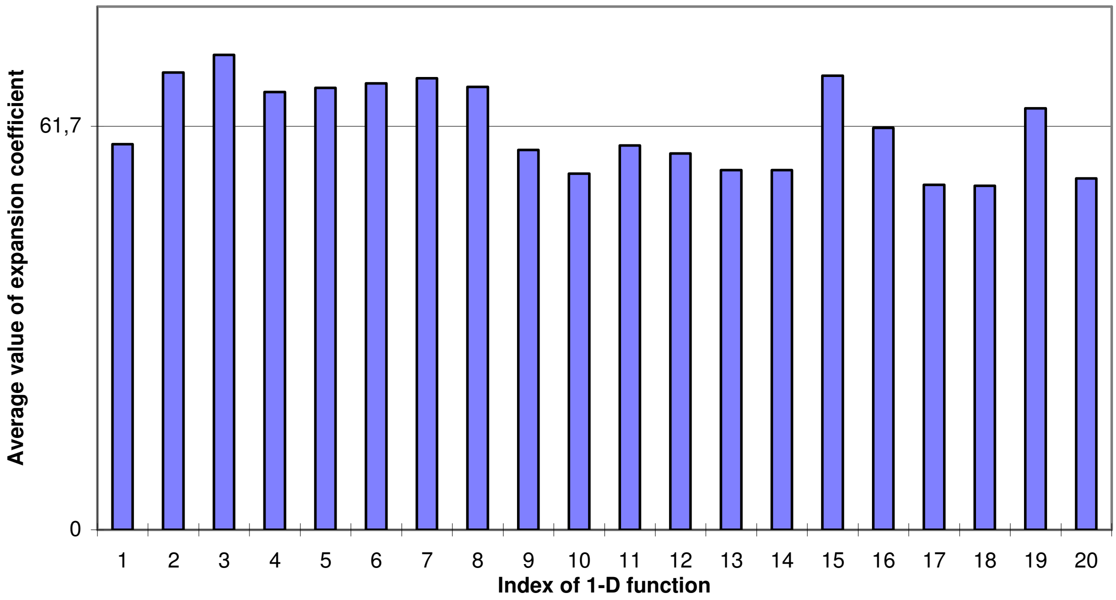


Fig. 4. 10. An average value of expansion coefficients for functions from Neff&Zakhor dictionary in the matching pursuit algorithm [MP+S+Post]

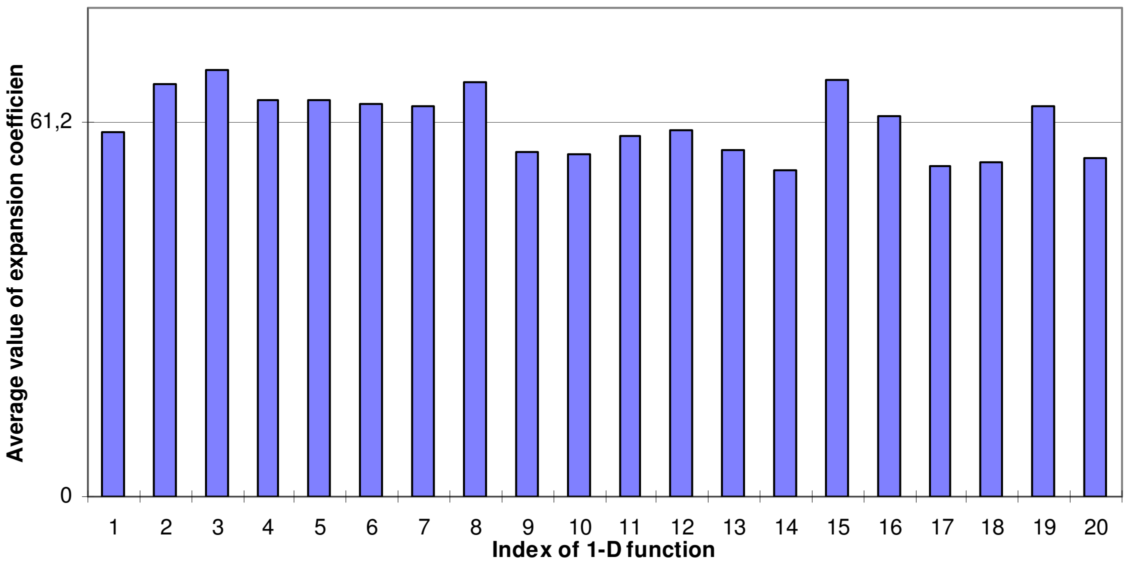


Fig. 4. 11. An average value of expansion coefficient for function from Neff&Zakhor dictionary in the matching pursuit with separable decomposition [MPwithSD].

4.4. Conclusions

The main goal of this dissertation has been attained. It has been proven that utilisation of the property of separability of functions from a dictionary provides to the new strategy. This strategy allows for reducing the computational complexity of the matching pursuit algorithm with a small decrease in compression efficiency. The novel strategy has been obtained by using the separable decomposition in the matching pursuit algorithm [Doma03], [Doma05b]. The reduction of complexity has been attained in a similar way as for the M-optimal approximation problem. Let us remember that the complexity of the M-approximation problem has been reduced by the matching pursuit algorithm in such a way that the M dictionary elements are chosen individually instead of M at once. Similarly, the application of separable decomposition allows for considering N one-dimensional functions instead of one N-dimensional signal.

The results showed that the proposed algorithm is 7-10 times faster than the classic matching pursuit algorithm with Schwartz inequality and the post-selection of atoms. The novel algorithm gives slightly worse PSNR results (up to 0.1 dB).

The proposed implementation approximates each one-dimensional function using a single waveform from the dictionary. This scheme can be generalised and the representation using two or more waveforms seams very promising. In this way, the SNR scalability system can be obtained.

Chapter 5

Learning Dictionaries for Matching Pursuit with Separable Decomposition

5.1. Introduction

The idea of matching pursuit is to use a large overcomplete basis set called a dictionary to ensure perfect reconstruction of the original residual image. The choice and construction of the dictionary strongly affects coding performance. Nevertheless, a fundamental problem of the hitherto matching pursuit coding techniques is the lack of feedback between an input signal and a dictionary, since this technique uses a dictionary a priori. This fact implies a great need for the designing of a universal dictionary.

Moreover, it is assumed that the overcomplete dictionary contains functions that are able to approximate local concentrations of energy in a very accurate way. However, in practical applications, a much smaller set of basis functions is usually adopted to speed up the matching pursuit algorithm. As a result, the representation using the universal and static dictionary is rough and not suitable enough to express subtle parts of a signal. In most matching-pursuit-based video codecs reported in the literature [Neff95], [AlSh99], [Oh00] a set of separable Gabor functions is used as a dictionary. This leads to a fast implementation of the matching pursuit algorithm.

Nevertheless, the most important question is if it is possible to improve representation using a constant number of functions in a dictionary. Partial answers on this question are included in [Sch01] and [Chien00]. Both solutions exploit a vector quantization technique. The learning scheme proposed in this chapter is based on vector quantization too.

5.2. Vector Quantization

Vector quantization (VQ) is the generalisation of scalar quantization to the case of a vector [Linde80], [Gray84], [Abut90], [Gar92], [Akrout94], [Cherk98]. The basis structure of VQ is essentially the same as a scalar quantization and consists of an encoder and decoder.

Let $p_X(x)$ be the probability density function for the N-dimensional random variable X we wish to quantize. Let us assume that there is a training sequence consisting of M source vectors:

$$T = \{t_1, t_2, \ldots, t_M\}\,. \tag{5.1}$$

Additionally, M is assumed to be sufficiently large so that all statistical properties of the source are captured by the training sequence.

Let $D(x, y)$ be an appropriate distortion measure defined as follows:

$$D(x, y) = \|x - y\|\,. \tag{5.2}$$

The encoder is defined by a partition of the training set T into sets V_k called *cells of Voronoi*. All elements of T that lie in V_k will be encoded to index k and decoded to $\hat{t}_k$. Each vector $\hat{t}_k$ is called *a code vector* or *codeword*. In addition, the set of all codewords is called *a codebook*. The decoder is defined by specifying the reproduction value $\hat{t}_k$ for each partition V_k.

An optimal quantizer that minimises the average distortion $D(x, y)$ must satisfy the following conditions:

- *Nearest neighbour condition:*

$$V_k = \left\{ t \in T : D(t, \hat{t}_k) \leq D(t, \hat{t}_j), \mathit{for\,all}\ j \neq k \right\}. \tag{5.3}$$

The condition says that the encoding partition V_k should consist of all vectors that are closer to $\hat{t}_k$ than any of the other codewords.

- *Centroid condition:*

$$\hat{t}_k = \frac{\sum_{t_i \in V_k} t_i}{\overline{\overline{V_k}}}. \tag{5.4}$$

This condition says that the codeword $\hat{t}_k$ for the cell of Voronoi V_k should be the average of all those training vectors that belong to cell V_k.

5.3. Generalised Lloyd Algorithm

Vector quantizers can be designed using an iterative procedure called the Generalised Lloyd algorithm (GLA) [Lloyd82]. This algorithm starts with an initial codebook concerning B initial codewords. The algorithm proceeds as follows:

1. *Optimise the encoder given the current decoder*. Using the current codebook, divide a training set into partitions V_k according to the nearest neighbour condition. This gives an optimal partitioning of the training data for the given set of codewords.
2. *Optimise the decoder given the current encoder*. Using the current partitioning, recalculate the centroid values.
3. If the new codebook is changed, go to step 1.

The Generalised Lloyd algorithm is a descent. Each step either reduces the average distortion or leaves it unchanged. For a finite training set, the distortion can be shown to converge to a fixed value in a finite number of iterations. The GLA does not guarantee a globally optimal quantizer, as there may be other solutions for the necessary conditions that yield a smaller distortion.

5.4. Learning scheme

The separable decomposition can be used in matching pursuit algorithm to speed up the process of finding atoms. Let us remember that the complexity of the M-approximation problem has been reduced by the greedy matching pursuit algorithm in such a way that the M dictionary elements are chosen individually instead of all M elements at once. In an environment of separable functions, it is possible to apply a technique that reduces the complexity of the problem in a very similar way. Note that separable decomposition finds a separable function that approximates an input signal in the best manner. This fact allows for considering the N one-dimensional functions instead of one N-dimensional signal.

On the other hand, the separable decomposition not only reduces the computational complexity of matching pursuit, but also gives a feedback to the dictionary. Note that separable decomposition computes 1-D functions and expects these functions in a dictionary. Weak representation of 1-D functions causes weak representation of the 2-D input function. Nevertheless, the fact that optimal 1-D functions are known lays the foundation of the proposed learning scheme.

The novel learning scheme uses separable decomposition and vector quantization to result in an improved dictionary. The whole process is performed as follows (Fig. 5. 1).

At first, an initial dictionary is used to encode the chosen sequence using matching pursuit with a separable decomposition. As a result, the set of „expected” functions is obtained. This set and the dictionary are treated as input parameters for GLA, i.e. as the set of training vectors and codebook respectively. The vector quantization algorithm computes the next version of dictionary. The whole process can be repeated using the new dictionary.

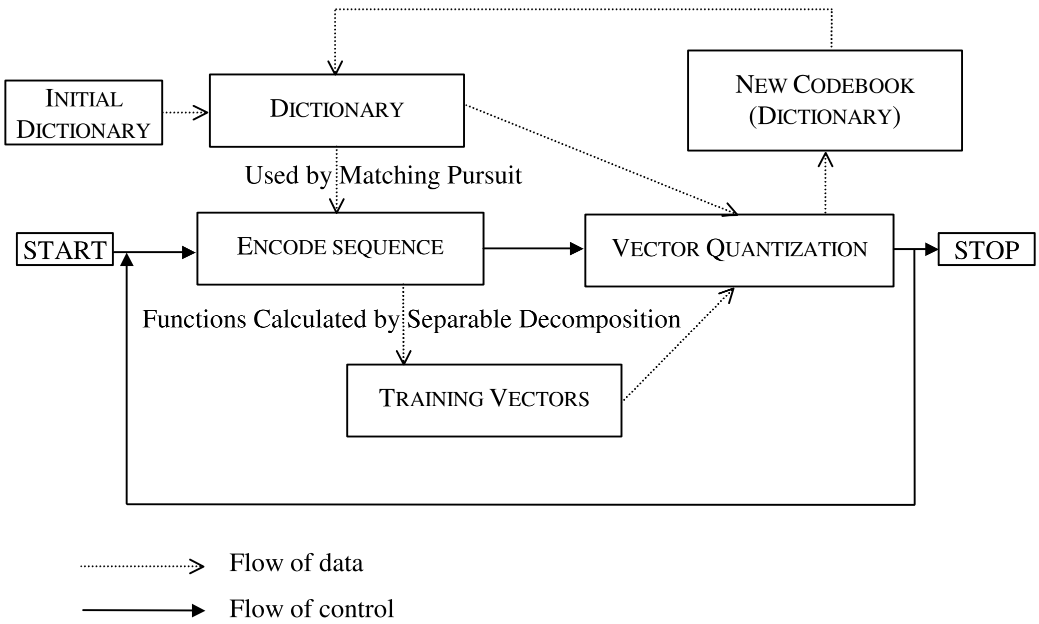


Fig. 5. 1. Proposed learning scheme.

The initial codebook for vector quantization (VQ) is the same as the dictionary used to encode a sequence and to get the training vectors $\{t_1, t_2, \ldots, t_M\}$. Note that at the first iteration of VQ, each training vector t_k already belongs to a cell of Voronoi V_k that is represented by a dictionary function. It implies that all training vectors that were approximated by the *k*-th function in the matching pursuit algorithm are classified to the same cell of Voronoi. As a result, all training vectors from a cell of Voronoi V_k define a new centroid being a new version of the *k*-th function in a dictionary.

The process of calculation of a centroid should be slightly modified to get proper results for the distortion measure defined as:

$$D(x, y) = 1 - \left| \frac{\langle x, y \rangle}{\|x\| \|y\|} \right| = 1 - \left| \langle x, y \rangle \right|, \tag{5.5}$$

for $\|x\| = 1$ and $\|y\| = 1$.

Since $D(x, y)$ depends on the absolute value of the inner product, the proper sign of the product should be used in the calculation of a centroid. This problem can be easily solved. There exists a coefficient $c_i = \pm 1$ that gives positive values for the inner products:

$$\forall t_i \in V_k \ \ \left\langle \hat{t}_k , c_i t_i \right\rangle > 0 \,. \tag{5.6}$$

In this way, a temporary centroid is calculated as follows:

$$\hat{t}_k^{'} = \sum_{t_i \in V_k} c_i t_i \,. \tag{5.7}$$

The final centroid is obtained from $\hat{t}_k^{'}$ by normalization i.e.:

$$\hat{t}_k = \frac{\hat{t}_k^{'}}{\left\| \hat{t}_k^{'} \right\|} \,. \tag{5.8}$$

Using expansion coefficients can modify the above scheme. In this way, the importance of training vectors may be taken into account. The experiments show that application of the weighted sum in expression (5.7) is not necessary as it leads to similar coding performance.

5.5. Experimental results

The main purpose of the experiments is to verify the proposed learning scheme. Since the most important task for each learning model is its efficiency, then the PSNR values were taken as the measure for the proposed method. An additional task was to compare the efficiency of the universal dictionary proposed by Neff and Zakhor and the trained dictionaries as proposed in this chapter.

The implementation of the video encoder was presented in Chapter 4. In the described implementation, some simplifications have been made. Firstly, the atom parameters were not encoded, but instead the number of bits required to encode an atom were estimated using a statistical model based on entropy calculations. Secondly, each frame was encoded by using the number of bits known from a respective AVC bitstream. This means that the same bit allocation as in standard AVC coding was used for the consecutive frames encoded by matching pursuit. The above simplifications are well motivated. The entropy model gives similar results as the model implemented in [Neff95]. The synchronisation of bit-streams gives a very good comparison model and simplifies the control block in the experimental encoder.

In the experiments, seven standard QCIF video test sequences were used: Akiyo, Container, Silent, Foreman, Claire, News and Mother. The experiments were performed at very low bitrates of about 8-48 kbps. For all test sequences, 10 seconds of video were compressed. The first frame was an I-frame and all the consecutive frames were P-frames. No B-frame was used in the experiments. For comparison purposes, some reference results were taken from the JM8 encoder and from the video encoder based on matching pursuit with the dictionary proposed by Neff & Zakhor (Table 5. 1).

Table 5. 1. The results for the AVC JM8 encoder and for the MP-based encoder using Neff&Zakhor dictionary.

Test sequence	Framerate [Hz]	Bitrate [kbps]	AVC Y-PSNR [dB]	MP with N&Z Y-PSNR [dB]
Akiyo	7.5	8.19	34.39	34.71
Container	7.5	12.71	32.64	32.49
Silent	7.5	24.24	32.14	32.50
Foreman	10	47.81	32.94	32.78
Claire	7.5	10.10	36.7	36.86
News	7.5	24.60	32.96	33.09
Mother	7.5	16.59	33.82	33.46

In the first experiment, the proposed learning scheme was used to obtain optimal static and separable dictionaries individually for each sequence. For this purpose, the Neff & Zakhor dictionary was used as the initial dictionary. Then, several cycles of a novel learning scheme have been run. We found no noticeable difference in performance between successive cycles. The obtained dictionaries give about a 0.28dB increase in objective quality criteria (PSNR for luminance) (see Table 5. 2) to the original dictionary.

The proposed learning scheme is a very stable method. Standard deviations taken from the last 16 of 32 cycles are not greater than 0.03dB. In fact, the motion compensation used in the video encoding process is responsible for more fluctuations on the PSNR than the learning scheme itself.

Table 5. 2. The average results for individually trained Neff & Zakhor dictionary

Sequence	Framerate [Hz]	Bitrate [kbps]	Average luminance PSNR [dB]	PSNR Standard Deviation [dB]	Gain to Neff&Zakhor [dB]
Akiyo	7.5	8.19	34.90	0.022	0.20
Container	7.5	12.71	32.97	0.025	0.48
Silent	7.5	24.24	32.74	0.019	0.24
Foreman	10	47.81	33.01	0.014	0.24
Claire	7.5	10.10	37.13	0.027	0.27
News	7.5	24.60	33.38	0.023	0.30
Mother	7.5	16.59	33.69	0.014	0.23

The next experiment used a randomly generated dictionary as an initial dictionary to the learning scheme. All generated dictionaries contained 20 one-dimensional waveforms (similar to the Neff & Zakhor dictionary). The dictionaries were generated in the following way. At first, the region of support for each generated waveform was randomly selected from the range 1 to 22. Then, the appropriate number of non-zero coefficients was generated. Finally, the generated waveform was normalized.

For each sequence, 12 randomly generated dictionaries were created. Then, each dictionary was used in the learning scheme to obtain the optimal dictionary for individual sequence. The average results (with standard deviation not greater than 0.03dB) are marked using bold style in Table 5. 3. Next, the trained dictionaries were used to encode the remaining sequences and the average results are shown also in Table 5. 3. (normal style of font).

The results show that the proposed learning scheme is very stable and gives similar results for any randomly generated dictionary. It is worth mentioning that the results obtained from the Neff & Zakhor dictionary are also similar (see Table 5. 1.) since this dictionary can be treated as the instance of the randomly generated dictionary.

Table 5. 3. The average Y-PSNR for twelve randomly generated and trained using the learning scheme dictionaries

	Dictionaries						
Sequence	Trained on Akiyo	Trained on Container	Trained on Silent	Trained on Foreman	Trained on Claire	Trained on News	Trained on Mother
Akiyo	**34.90**	34.83	34.83	34.88	34.84	34.91	34.86
Container	32.70	**32.95**	32.61	32.64	32.72	32.82	32.74
Silent	32.64	32.45	**32.73**	32.71	32.62	32.67	32.70
Foreman	32.94	32.80	32.99	**33.00**	32.96	32.97	33.01
Claire	37.04	36.91	37.07	37.09	**37.14**	37.11	37.11
News	33.31	33.30	33.31	33.32	33.30	**33.40**	33.33
Mother	33.67	33.53	33.67	33.66	33.65	33.68	**33.69**

As can be seen, dictionaries trained on all sequences excluding the Container gave similar results. This means that the above sequences contained similar characteristics of a prediction error and its optimal dictionaries contained similar waveforms simultaneously. So, Akiyo, Silent, Foreman, Claire, News and Mother belong to the same group of sequences. This means that the dictionary that gave good results for all sequences within one type of sequence can be obtained from a learning scheme performed on any representative using any initial dictionary. Therefore, it is possible to use dictionaries calculated to certain classes of video sequence. Content-class-adapted dictionaries may provide slightly higher compression efficiency than that obtained with a dictionary that is not adapted to the class of video sequence content. Note that almost 50% of the Container sequence presents waving water, which has, in fact, „no shape". This is the reason why this sequence is different from the others. It is worth mentioning that PSNR results taken from dictionaries trained on the other sequences are worse than the PSNR obtained using a dictionary trained on Container sequence, nevertheless the PSNR results are still better than the Neff&Zakhor solution.

One way or another, the proposed learning scheme is a very good method of generating a universal dictionary. The convergence of the novel learning scheme is presented on Fig. 5. 2. As can be seen, the efficiency of the dictionary grows up very quickly and after the eighth iteration it is very close to the optimal value.

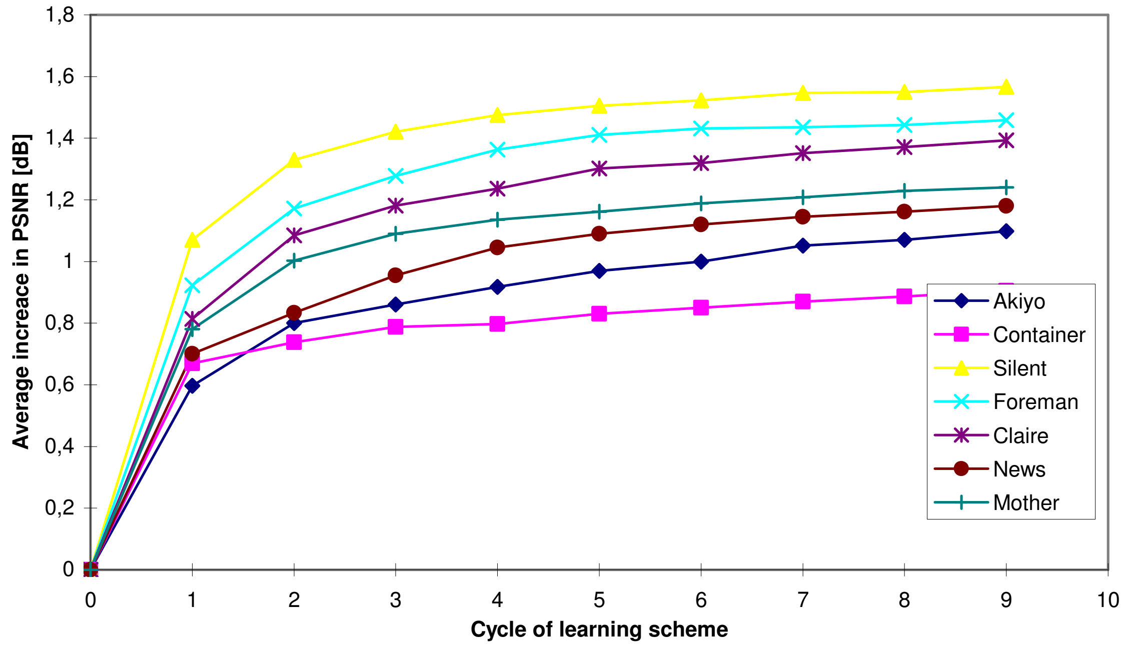


Fig. 5. 2. The convergence of the learning scheme for different sequences.

The most important aspect of the proposed learning scheme is its way of obtaining a new dictionary. The key element is a separable decomposition since it gives optimal one-dimensional functions that are expected to be in a dictionary. If we compare this scheme to the way proposed by Neff & Zakhor, then we will see the advantage. Let us remember that Neff & Zakhor generate a large set of Gabor functions. Then they used this large dictionary to encode MPEG video test sequences and, successively, they reduced the initial dictionary by removing rarely used functions. The final step consisted of choosing the functions which are more frequently used. In the solution proposed in this dissertation, it is not necessary to decide what functions should be in a dictionary. The knowledge concerning the character of the encoded signal is also not required since all information about an input signal results from the separable decomposition.

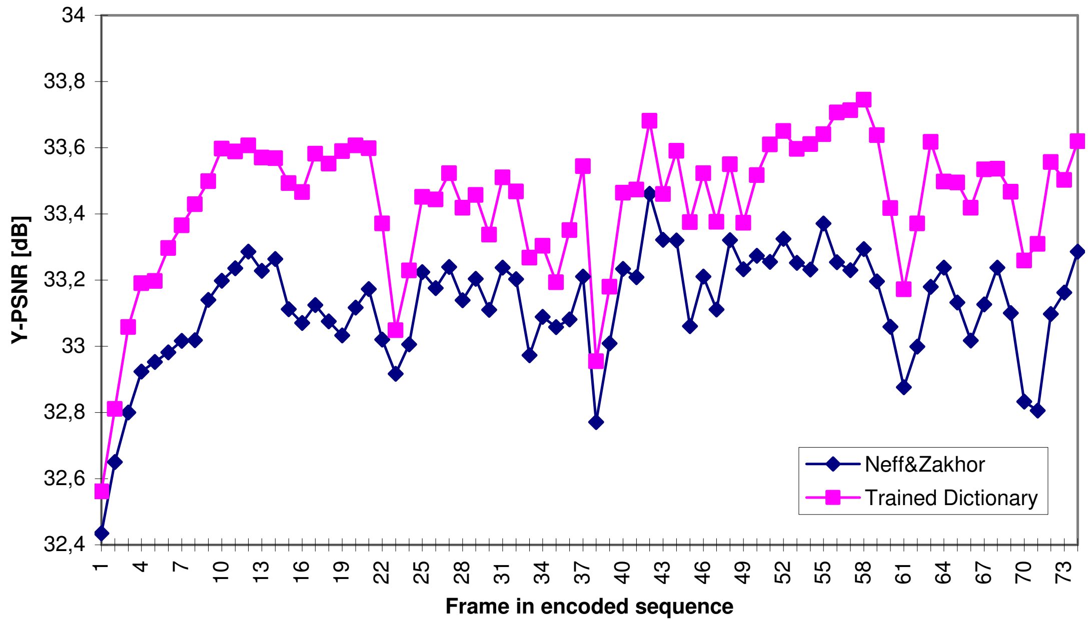


Fig. 5. 3. The comparision of results for the News sequence. The trained dictionary was computed on the News sequence in the proposed learning scheme based on matching pursuit with separable decomposition. The initial dictionary consisted of Neff&Zakhor waveforms.

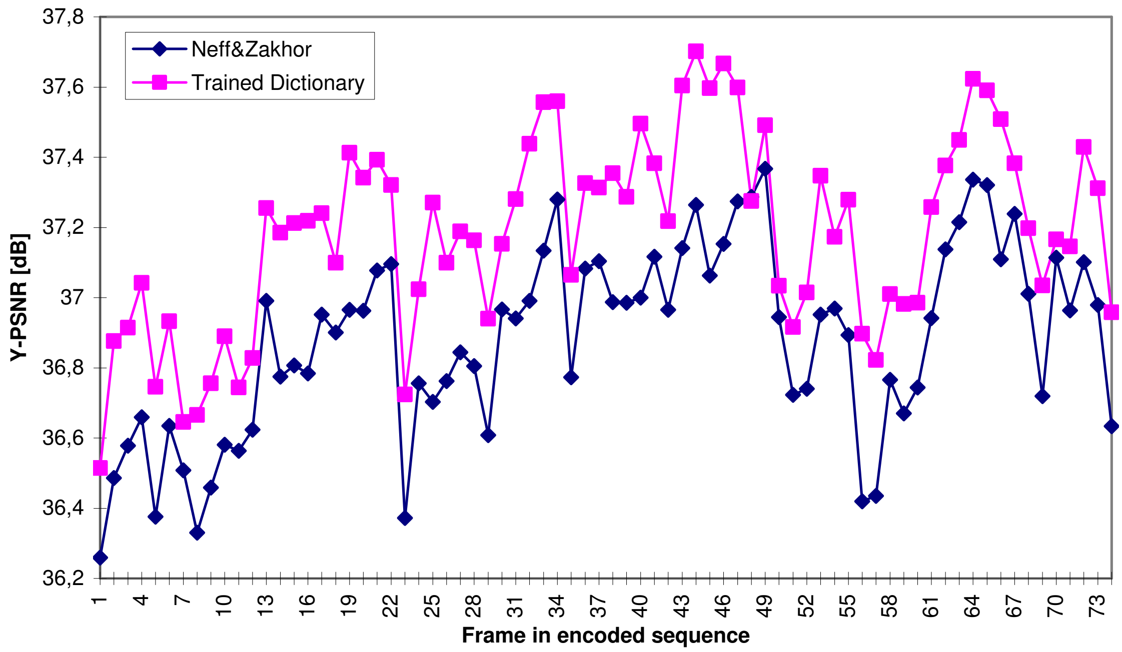


Fig. 5. 4. The comparision of results for the Claire sequence. The trained dictionary was computed on the Claire sequence in the proposed learning scheme based on matching pursuit with separable decomposition. The initial dictionary consisted of Neff&Zakhor waveforms .

The main goal of the next experiment was to check how the PSNR changes if the number of functions in the dictionary is increased. Four test QCIF sequences: Akiyo, Container, Silent and Foreman were used in the next experiment. At the beginning, four different dictionaries concerning eight functions were generated and trained on the above sequences using the described learning scheme. The number of cycles for the learning method was set to 16. Then, the last eight average Y-PSNR results were taken to estimate an efficiency of a dictionary containing eight functions. The whole process was repeated for sets containing 16, 24, 32, 64 and 128 functions. The results presented on Fig. 5. 5 show that 32 functions in a dictionary point out an upper bound of efficiency for QCIF video sequences. The small increase in efficiency for the Foreman sequence is related to the small increase in the bit-stream. In addition, as was mentioned previously, the Container should be discarded from our investigation because the sequence is not representative. The results proved that the Container consists of many shapes (waving water) that cannot be approximated in an accurate way with the assistance of a strongly limited set of functions.

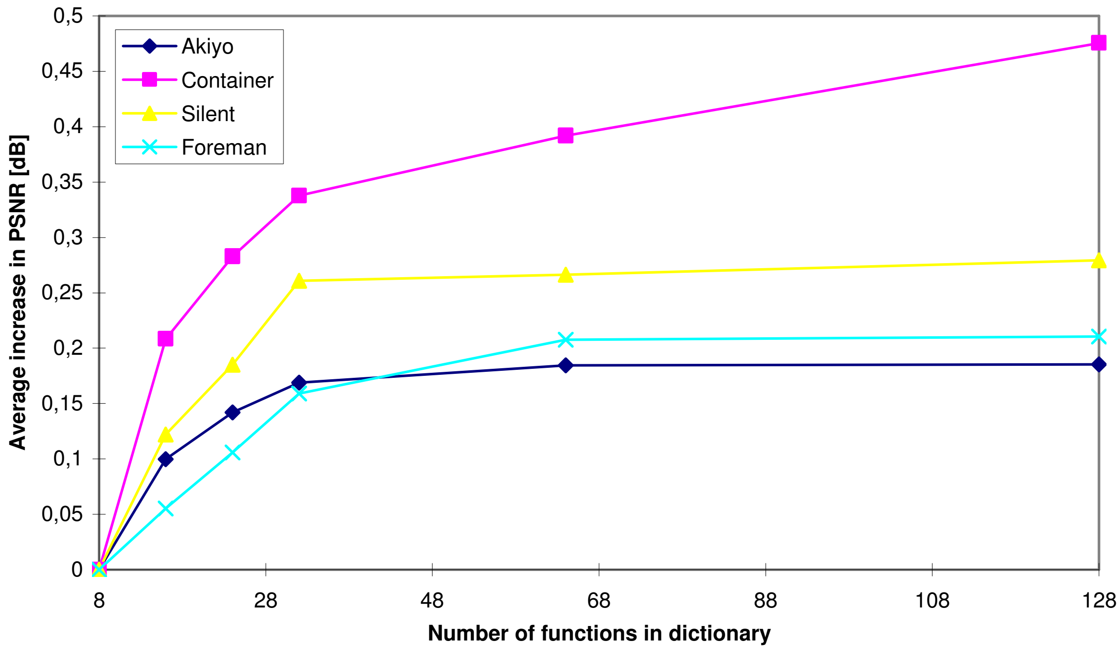


Fig. 5. 5. The increase of PSNR depending on the size of dictionary.

On the other hand, the experiment shows that it is not true that in order to improve PSNR value the number of functions in dictionary should be increased. The larger dictionary means that we need more bits to encode an index of a function. Consequently, this leads to a smaller number of atoms per frame and less approximated regions. However, the larger dictionary has a more computationally intensive process of finding an atom. Considering this, a good trade-off between efficiency and computational load is about 24 functions in dictionary. Moreover, the number of functions in universal dictionary does not influence significantly the objective quality criteria since the difference between extreme values i.e. for 8 and 128 functions is about 0.2dB on average. This leads to the observation that in order to improve the quality of the signal approximation, one needs to adapt the dictionary to the single frame or, even better, to the region of local search of atom.

In the next experiments, the QCIF sequences were encoded with the assistance of dictionaries which were adapted to a single frame. In order to realise the above purpose, the video encoder was modified in such a way that the vector quantization technique had been implemented in the coding loop of the prediction error. In fact, the novel learning scheme was placed into the process of approximation of the displaced frame difference. The whole dictionary adaptation was performed in the following way (see Fig. 5. 6). At first, the prediction error was approximated using the matching pursuit with separable decomposition. Then, one-dimensional functions obtained from the separable decomposition were used as the training vectors in the vector quantization algorithm. The Generalised Lloyd algorithm was applied to improve the current version of dictionary, i.e. to obtain its new version. Finally, the original shape of prediction error had been restored before the trained dictionary was used in the successive cycle of the learning scheme. The previously described steps were performed the predefined number of times.

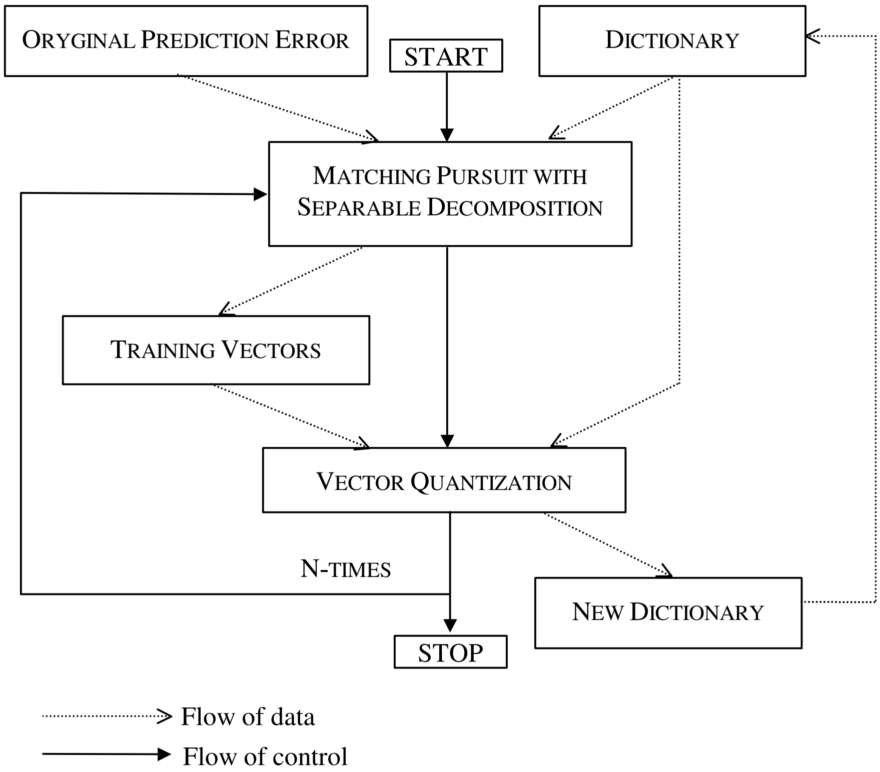


Fig. 5. 6. The adaptation of dictionary for single frame using the proposed learning scheme with separable decomposition.

The experiment was executed for dictionaries containing 8, 16 and 32 one-dimensional functions. Local adaptation of dictionary was performed using 8, 16, 32 and 64 cycles of the learning scheme. The final results for the experiment are presented on Fig. 5. 7.

The experiment revealed that the eight-functions-dictionary adapted to the single frame gives similar results as the video encoder using the universal 20-waveforms-dictionary trained on the specific sequence (compare Table 5. 3 and Fig. 5. 7). (Note that sequences Akiyo and Container slightly exceed its universal solutions. The reason of this is because the above sequences are encoded using small number of atoms. As a result, the vector quantization used relatively large number of codewords.) It is important to appreciate the fact that the bit-stream was estimated without information concerning the adopted dictionary. This means that the definitions of 1-D functions are omitted from the bit-stream. Taking into consideration the above fact, the increase of the objective quality criteria is possible only and only if it is possible to encode the definition of trained functions in a very compact way.

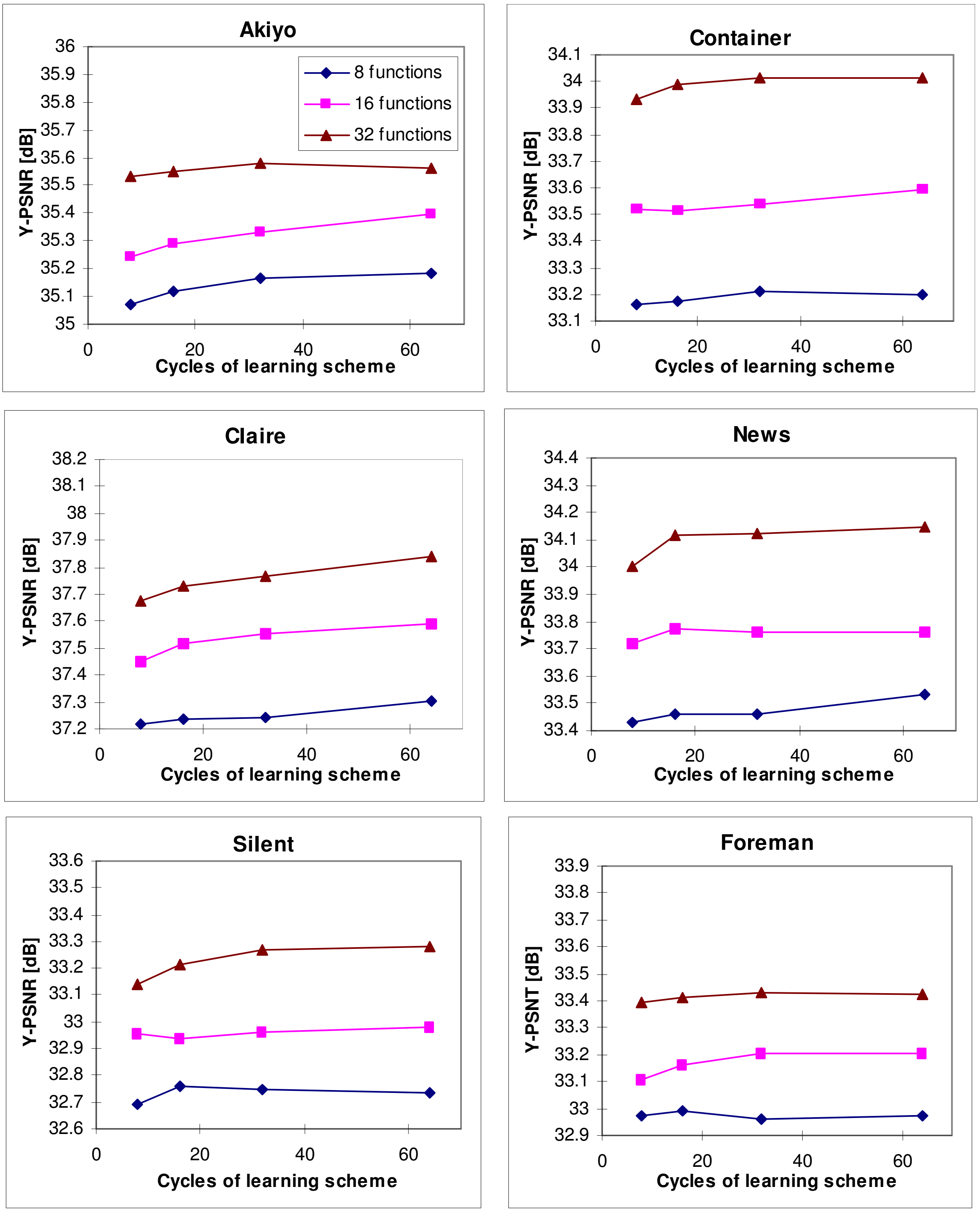


Fig. 5. 7. The experimental results taken from the video encoder using matching pursuit with the separable decomposition and in-loop learning scheme.

5.6. Conclusions

In this chapter of the dissertation, an original concept of new learning scheme for video coding based on matching pursuit with separable decomposition has been presented [Doma05b]. The novel scheme has been obtained using information from the separable decomposition, which gives a feedback to the dictionary. Small improvement (e.g., about 0.25dB) may be achieved by designing dictionaries for different classes of video content like landscapes, head and shoulders, etc. The results have been verified by a series of experiments with standard test video sequences and original software that implements matching pursuit coders on the platform of the advanced motion-compensated prediction of the AVC/H.264. As can be seen, further improvement of objective quality can be obtained by adaptation of dictionary to every single frame.

Chapter 6

Matching Pursuit with Dictionary Dynamically Adapted to Signal

6.1. Introduction

In matching pursuit theory, a large overcomplete basis set called a dictionary is used to ensure perfect reconstruction of the original image. The dictionary plays a crucial role for the matching pursuit algorithm since it strongly affects its convergence and visual performances [Mall93], [Goyal97], [Sch04]. On the other hand, the previous chapter showed that a large static dictionary does not improve the objective quality criteria if the bits budget is constant. Actually, experimentally obtained results showed that the upper bound for the number of functions in universal dictionary is not so big and amount to 32 one-dimensional functions for the QCIF video sequences. Furthermore, the experiments confirmed that in order to improve the quality of the signal approximation, one needs to adapt the dictionary to an individual frame (Chapter 5.5) or even better, i.e. to a local region. Moreover, the adaptation should refer to the known signal since this is the only way to avoid function definition in bit-stream. In other words, it is assumed that the definition of function should be reproduced using information contained in an input signal.

The signal approximated using matching pursuit in video compression system comes into existence as the difference of the current frame and the prediction of the current frame. It is high-frequency signal that is characterised by local concentrations of energy that mostly appear at edges of objects within the frame. Nevertheless, a fundamental problem of matching pursuit is the lack of feedback between an input signal and a dictionary, since this technique uses a dictionary a priori. To compensate this inconvenience, we propose supplementing the dictionary by some functions that are extracted from the motion-predicted image. It is well known that prediction error and a high-frequency signal obtained from a predicted image using upper-bound filter are very similar to each other. Of course, „similar" does not mean „exact", nevertheless in term of lossy compression it signifies a good approximation.

6.2. Dynamic Dictionary

A fundamental problem of matching pursuit is the lack of feedback between an input signal and a dictionary. This fact implies the great need for designing of a universal dictionary. In practice, this means that the overcomplete dictionary should be sufficiently redundant to express an input signal in a very good way. An extension of the set of functions implies two problems, which are in the opposition to the larger dictionary. The first problem results from the fact that the more functions in a dictionary the higher computational load. The latter problem signifies an increase of number of bits needed to code indexes of atom functions.

In order to minimise the number of functions in a dictionary and maximise the objective quality criteria, i.e. PSNR, one should apply a learning scheme proposed in the previous chapter or in [Peng00], [Sch01], [Sch04]. It is worth mentioning that such solutions demand at least two cycles of coding since each cycle provides new information to the learning process. Due to this, it is practically impossible to use the above learning schemes in real-time applications. In addition, the size of stream will be decreased because the obtained dictionary must be sent to the decoder.

In order to combine features of a universal dictionary and learning schemes, the author proposes novel model of dictionary. In the new solution, the static set of functions is locally supplemented by functions extracted from a motion-predicted image. The main idea is based on some general observations concerning the prediction error, i.e. (see Fig. 6. 1 and Fig. 6. 2):

- a prediction error is a high-frequency signal,
- a prediction error often occurs in place of image edges,
- a prediction error and a high-frequency signal obtained from an predicted image using upper-bound filter are very similar to each other.

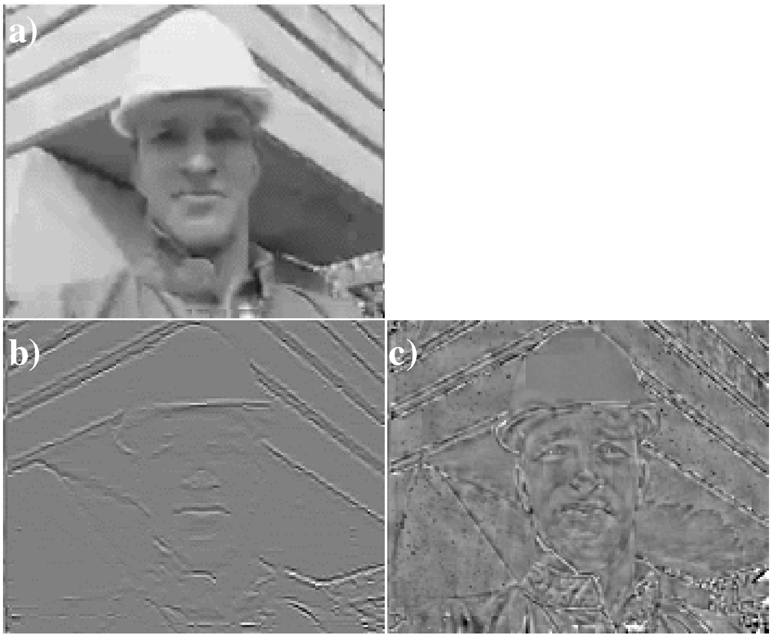


Fig. 6. 1 Predicted image (a) from sequence Foreman. Image (b) obtained from predicted frame (a) using upper-bound filter. Prediction error (c) from sequence Foreman.

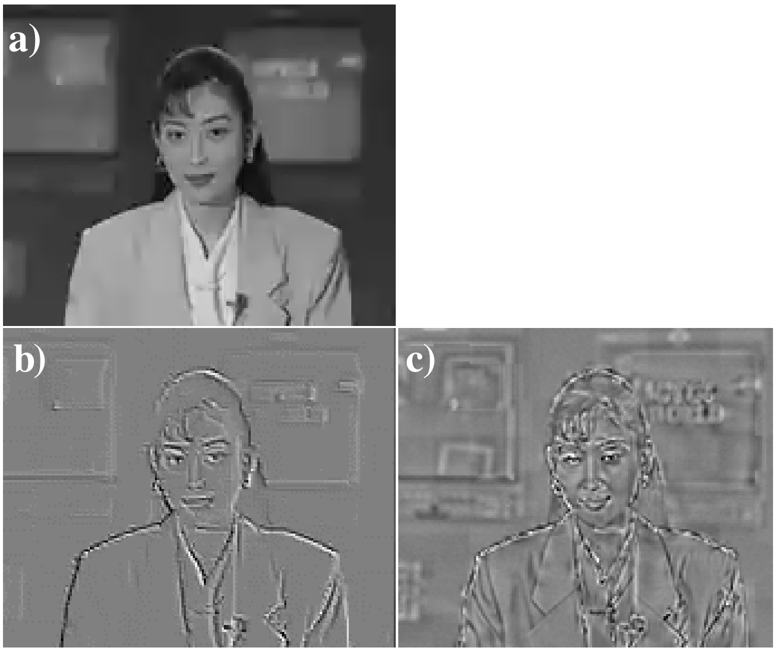


Fig. 6. 2. Predicted image (a) from sequence Akiyo. Image (b) obtained from predicted frame (a) using upper-bound filter. Prediction error (c) from sequence Akiyo.

One expects that some functions obtained from a predicted image using upper-bound filters could outperform predefined functions from the dictionary.

The general scheme of the above idea is presented on Fig. 6. 3. The whole process is very similar to the scheme form Chapter 2.4.2. In the proposed scheme, each approximation stage in matching pursuit algorithm is preceded by an additional step that dynamically and locally supplements a predefined dictionary.

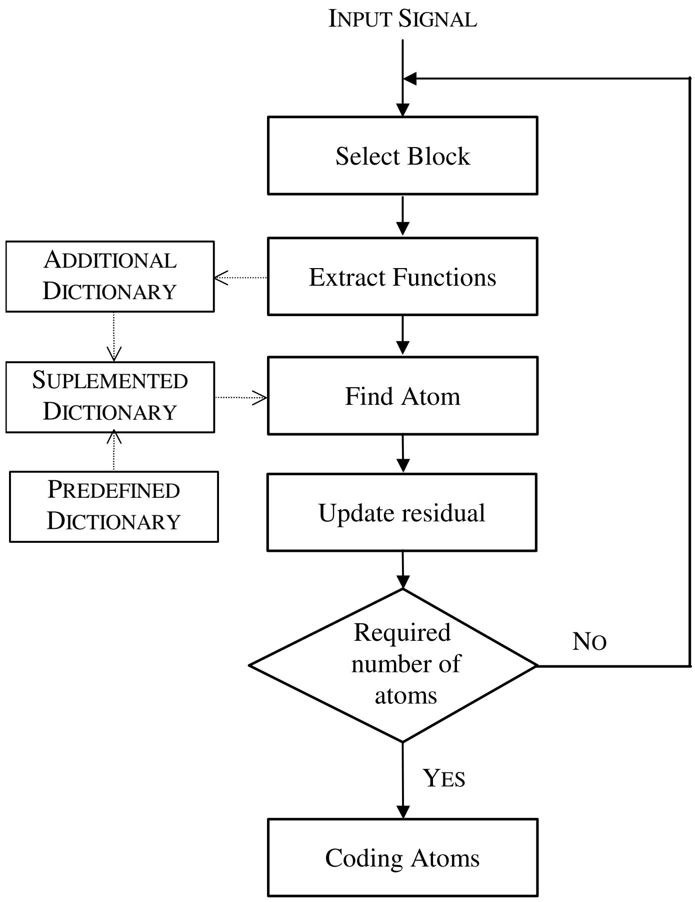


Fig. 6. 3. The matching pursuit scheme supplemented by additional step that extracts information for image-adapted functions.

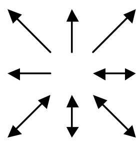
The proposed scheme is general, i.e. it can be used not only in the matching pursuit with separable decomposition but also in the classic matching pursuit algorithm. In addition, the scheme does not define how to supplement a dictionary. Nevertheless, the verification of the proposed scheme in this dissertation was performed with the assistance of observations concerning the motion-predicted image and the upper-bound filters.

6.3. Implementation

In the implementation of the proposed scheme, each the searching atoms stage is preceded by a step which dynamically supplements a universal dictionary. New functions are extracted from the motion-predicted image using 2-D non-recursive two-term digital filter, which are well known as sharpen-filters. The main goal of the above filters is to stand in relief edges contained in the motion-predicted frame. In our implementation, two types of filters in four different directions are applied to obtain eight additional shapes (Fig. 6. 4). After this step, the locally updated dictionary is used to find the best approximation of the prediction error.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 6. 4. The filters (in matrix form) used to obtain eight additional functions.

It must be mentioned that functions extracted from motion-predicted image are not separable. Thus, the process of finding atoms for these functions is general, i.e. it does not utilise the property of separability. As the result, the computational load of approximation of the prediction error is nearly doubled.

In order to avoid additional parameters related to atoms coding, a region of support for additional functions is limited to eight pixels around the centre of search in both horizontal and vertical direction. In this way, the decoder is able to reconstruct the search centre on the basis of atom position in the following way:

$$\begin{aligned} x_{centre} &= \lfloor (x_{atom} / 8 \rfloor + 4 \ , \\ y_{centre} &= \lfloor (y_{atom} / 8 \rfloor + 4 \ . \end{aligned} \tag{6.1}$$

where:

x_{atom}, y_{atom} -atom position,

x_{centre}, y_{centre} -centre of block, centre of atom search.

Due to this, the additional dynamic atoms are treated uniformly as the others. It is worth mentioning that new functions are removed from dictionary after each update of a residual. There are two reasons to do this. Firstly, we are able to reconstruct functions since the predicted image is known in the decoder. Secondly, the number of functions in the dictionary influences on the number of bits needed to code function index. Due to this, the additional functions are removed in order to minimise the amount of information.

6.3. Experimental Results

The purpose of the experiments was to compare an efficiency of static and dynamic dictionary in video coding scheme based on matching pursuit. In the experiments, four standard QCIF video test sequences: Akiyo, Foreman, Container and Carphone were compressed. The experiments were performed for bit rates of about 8-64 kbps. For all test sequences, 10 seconds of video were compressed.

The codecs were built on top of the AVC ver.8.4. The general schemes of encoders and decoders are very similar to the described in Chapter 2.

The results of experiments are presented in Table 6. 1 and Table 6.2. As can be seen, both implementations use -on average- similar number of atoms per frame for the individual sequence and for selected type of coding. Note, that in the proposed model of dictionary, about 20 percentages of atoms are defined on the basis of functions extracted form a predicted image (last but one column). Finally, the matching pursuit with dynamic dictionary is about 0.3dB better than the static solution.

Table 6. 1. The results for the matching pursuit with the static dictionary.

Sequence	Type	Frame rate [Hz]	Bitrate [kbits/s]	Bits per atom	Atoms per frame	Y-PSNR [dB]
Akiyo	IPPP	7.5	8	19.58	28.89	34.60
Akiyo	IBPBP	15	11	19.80	17.93	34.64
Foreman	IPPP	10	47.5	19.87	114.22	32.90
Foreman	IBPBP	15	62.5	19.46	93.13	32.80
Container	IPPP	7.5	12.5	18.55	54.86	32.46
Carphone	IBPBP	15	53	20.40	95.90	32.84

Table 6. 2. The results for the matching pursuit with the proposed dynamic model of dictionary.

Sequence	Type	Frame rate [Hz]	Bitrate [kbits/s]	Bits per atom	Atoms per frame	Y-PSNR [dB]	Dynamic atoms [%]	Gain [dB]
Akiyo	IPPP	7.5	8	19.94	29.18	34.74	23.14	0.14
Akiyo	IBPBP	15	11	20.23	18.24	34.70	22.77	0.06
Foreman	IPPP	10	47.5	20.20	114.77	33.33	19.93	0.43
Foreman	IBPBP	15	62.5	20.15	94.46	33.14	19.47	0.33
Container	IPPP	7.5	12.5	18.86	55.23	32.82	15.21	0.36
Carphone	IBPBP	15	53	20.74	94.99	33.11	18.70	0.28

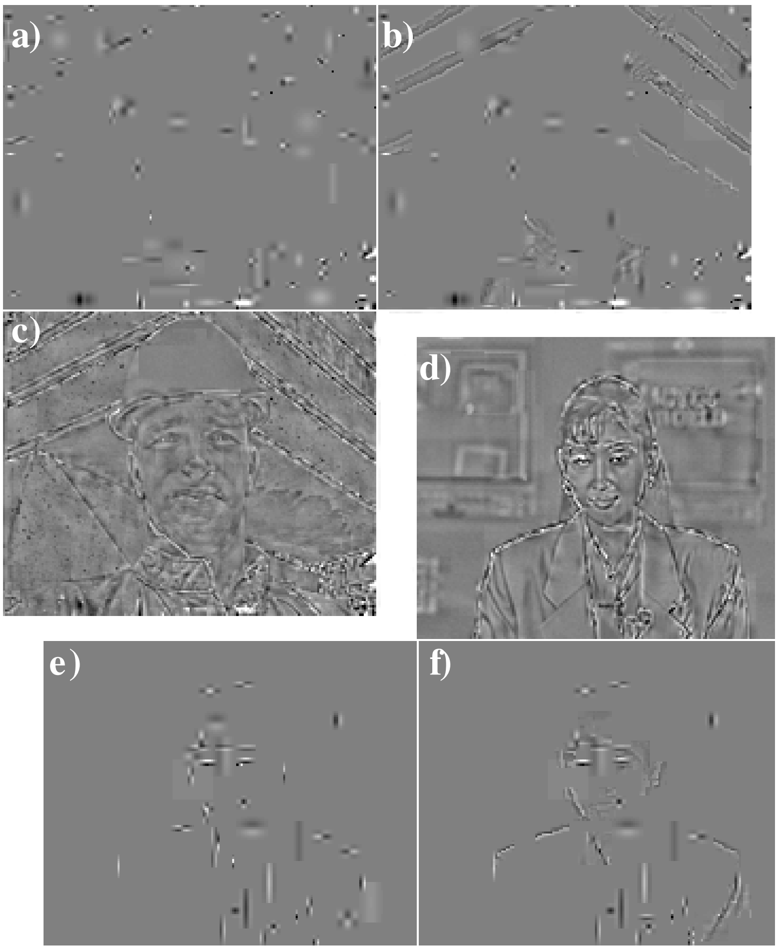


Fig. 6. 5. Foreman prediction error coded by 100 atoms using static (a) and dynamic (b) dictionary. Foreman (c) and Akiyo (d) original prediction error frames. Akiyo prediction error coded by 40 atoms using static (e) and dynamic (f) dictionary.

The Fig. 6. 5 shows differences in approximation between the novel dynamic model of dictionary and the old static solution. As can be seen, in the proposed method, many diagonal artefacts are approximated in a very good way. In addition, these diagonal artefacts are represented with the assistance of single atom. In the static case, some diagonal parts of the prediction error are represented using two, three and even more atoms.

6.4. Conclusions

This chapter of the dissertation verifies and confirms that dynamic image-adapted dictionary improves representation of an input signal [Doma04], [Doma05a]. Nevertheless, the computational load using non-separable atoms makes such a solution unattractive, especially for real-time applications. On the other hand, the presented method was verified using filters taken *ad hoc*. The author does not claim that selected filters ensure the best adaptation. Therefore, the obtained results might be far from optimal results. Nevertheless, this shows that dynamic adaptation of dictionary to the encoded signal opens new fields for further researches. It is assumed that efficient adaptation using properly selected filters may significantly increase the quality of approximation.

Chapter 7
Conclusions

7.1. Epilogue

The author's researches have focused on the video compression systems where coding of prediction error is realized using matching pursuit with an overcomplete set of separable functions. The greedy sub-optimal algorithm for solving the approximation problem was proposed by Mallat and Zhang [Mall93]. However, the most significant problem of the matching pursuit is its computational load and the lack of feedback between an input signal and the dictionary. In practical applications, the computational complexity of the algorithm is decreased by using the property of separability of functions from dictionary. Nevertheless, the process of approximation is still intensive and the computational load is comparable to motion estimation (Section 2. 4. 4). Thus, there is great demand for time-efficient algorithm that allows for significant reduction of computational complexity of matching pursuit. The author has suggested in the thesis of the dissertation, that using the property of separability of functions from the dictionary, it is possible to improve the process of searching atoms.

The key element of the proposed method is a separable decomposition (Section 3. 2. 2). The separable decomposition allows for computing the optimal separable representation for any selected region of an input signal. It is iterative process that converges to the optimal function practically after twelve iterations (Section 3. 2. 2, Fig 3. 8). Moreover, the speed of convergence is independent from the size of an input signal. As the result, the optimal separable function may be effectively computed for small regions of a prediction error and for whole large still images as well (Section 3. 3. 3).

The fact that the optimal separable representation is known allows for breaking through many drawbacks of matching pursuit and lays the foundation of the novel method. Foremost, the separable decomposition allows for building the efficient algorithm for searching atoms in matching pursuit scheme (Section 4). Moreover, the separable decomposition forms bridge between a dictionary and an input signal. As the result, the process allows for designing the dictionary giving information to the learning scheme.

7.2. Verification of Thesis

The thesis was experimentally proven in Chapter 4 and Chapter 5. The results confirm that the proposed algorithm is 7-10 times faster than the classic matching pursuit algorithm [Berg94], [Neff95]. The novel algorithm gives negligibly worse PSNR results. Additionally, Section 4 reveals that the dictionary proposed by Neff and Zakhor [Neff95], [Neff97] is not optimal for the AVC video encoder. Therefore, the feedback of the proposed matching pursuit with separable decomposition has been put to a test in the novel learning scheme. The experiments confirmed that the separable decomposition efficiently exploits separability of an input signal and gives a way to improve the representational performance of a dictionary. Dictionaries obtained by the proposed learning scheme have increased PSNR results of about 0.25dB. The final implementation of encoder using matching pursuit has included:

- an atom search algorithm improved by using a separable decomposition (Chapter 4),
- the algorithm of post-selection, where the best N atoms are chosen among 2N calculated atoms (Section 2.5),
- the modified method that selects the centre for searching an atom.

The gain between the final implementation to the reference implementation (see Section 2.6.2) is perspicuous (Fig. 7. 1). There is no single coding method that provides the majority of the significant improvement in compression efficiency in relation to classic matching pursuit. It is rather a plurality of smaller improvements that add up to the significant gain. Additionally, we must remember, that the matching pursuit algorithm in the final implementation of encoder is over seven times faster than the implementation based on Neff and Zakhor's concept [Neff95].

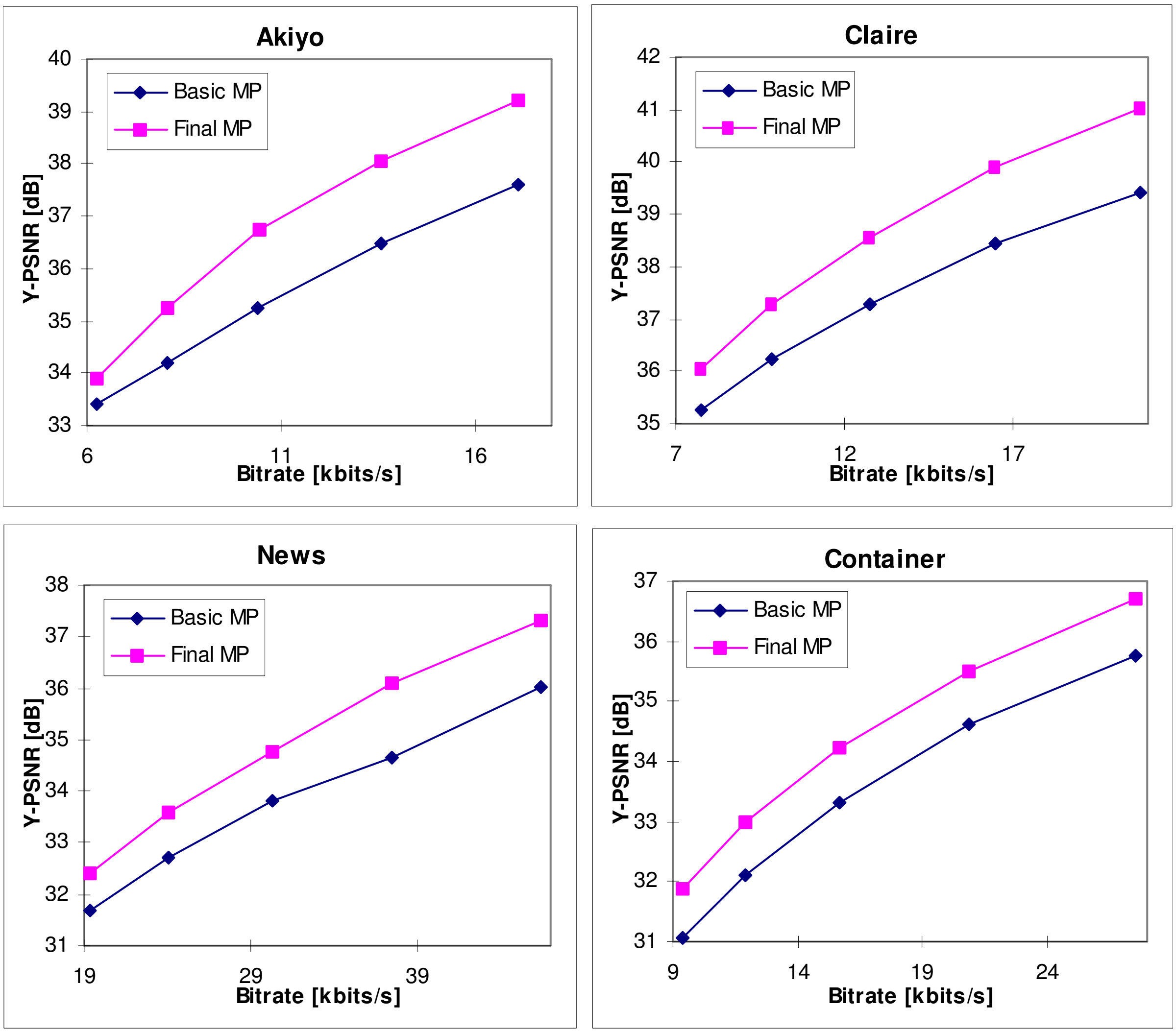


Fig.7. 1. The final results. The comparison of the basic matching pursuit-based video encoder to the encoder built on the basis of the author's proposals for QCIF test sequences.

Chapter 5 presents many experimental results concerning the dictionaries. An important observation is that the highly redundant dictionary does not improve the quality of approximation (Section 5. 5, Fig. 5. 5). The experimentally obtained results indicate the bound of the number of separable functions in the dictionary that makes compression process unattractive. Therefore, it is not true that in order to improve the objective quality of representation one needs to extend the universal dictionary. The only way to solve this problem lies in adequate prediction or adaptation of the functions from dictionary to a current context i.e. to the frame (Fig. 5. 7) or to the region of frame. In the author's opinion, this is the field of further researches. The author in Chapter 6 has proposed a solution for image-adapted dictionary. In the presented algorithm, the static dictionary is supplemented by functions extracted from predicted image using upper-bound filters. The dynamic dictionary adapted to the context of frame improves the objective quality of approximation of about 0.3 dB.

7.3. Achievements

Finally, the main original results of the dissertation are:

- The separable decomposition, this is the convergent transformation that allows it to consider the N one-dimensional functions instead of one N-dimensional signal. (Chapter 3).
- The matching pursuit with separable decomposition, this is the matching pursuit algorithm that uses the separable decomposition. (Chapter 4).

Other original achievements of the dissertation are:

- The proposal and implementation of a matching pursuit algorithm for prediction error coding in the H.264/AVC encoder (Chapter 2.6).
- The novel learning scheme for designing of a universal dictionary (Chapter 5).
- The novel model of dictionary that is locally adapted to the encoded signal (Chapter 6).
- The "Atoms Post-Selection" algorithm that corrects weak or improper approximation of an input signal (Section 2.5).
- The new method of electing block for the searching atoms process (Section 2.4.3).

Bibliography

[Abut90] Abut H., „Vector quantization", IEEE Reprint Collection, IEEE Press, Piscataway, NJ, USA, 1990.

[Ahma90] Ahmad Fadzil M.H., Dennis T.J., „A hierarchical motion estimator for interframe coding", IEEE Division Colloquium on Applications of Motion Compensation No. 1990/128, October 1990.

[Akrout94] Akrout N., Prost R., Goutte R., „Image compression by vector quantization: a review focused on codebook generation", Image and Vision Computing, Vol. 12, No. 10, pp. 627-637, December 1994.

[AlSh99] Al.-Shaykh O., Miroslavsky E., Nomura T., Neff R., Zakhor A., „Video compression using matching pursuits", IEEE Transactions on Circuits and Systems for video Technology, Vol. 9, pp. 123-143, February 1999.

[Ant92] Antonini A., Barlaud M., Mathieu P., „Image coding using wavelet transform", IEEE Trans. Image Proc., Vol. 1, pp. 205-220, April 1992.

[AVC] „Special issue on the H.264/AVC video coding standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, nr 7, July 2003.

[Banh97] Banham M., Brailean J., „A selective update approach to matching pursuits video coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol.7, No.1, pp.119-129, February 1997.

[Bati03] Batista V., Wu Yinghua, „Matching-pursuit for simulations of quantum processes", Journal of Chemical Physics, Vol. 18, No. 15, April 2003.

[Barb02] Barbarien J., Andreopoulos Y., Munteanu A., Schelkens P., Cornelis J., „Coding of motion vectors produced by wavelet-domain motion estimation", ISO/IEC JTC1/SC29/WG11 (MPEG), Awaji Island, Japan, MPEG Report M9249, December 7-12, 2002.

[Berg94] Bergeaud F., Mallat S., „Matching pursuit of images", IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, Vol. 2, pp. 330-333, October 1994.

[Bhas95] Bhaskaran V., Konstantinides K., „Image and video compression standards", Boston, Kluwer 1995.

[Bie88] Bierling M., „Displacement estimation by hierarchical block-matching”, Proc. Visual Comm. And Image Proc., SPIE Vol. 1001, pp.942-951, 1988.

[Bj02] Bjøntegaard G., Lillevold K., „Context-adaptive VLC coding of coefficients”, JVT Document JVT-C028, Fairfax, VA, May 2002.

[Boyce04] Boyce J.M., „Weighted prediction in the H.264/MPEG AVC video coding standard”, International Symposium on Circuits and Systems, Vol.3, pp. 789-92, 2004.

[Broff77] Brofferio S., Rocca F., „Inter-frame redundancy reduction of video signals generated by translating objects", IEEE Trans. on Commun., Vol. COM-25, pp. 48- 455, Apr 1977.

[Cham89] Cham W., „Development of integer cosine transforms by the principle of dyadic symmetry”, IEE Proc., Part 1, Vol. 136, pp. 276–282, August 1989.

[Chan94] Chan E., Rodriguez A.A., Ghandi R., Panchanathan S., „Experiments on block-matching techniques for video coding”, Multimedia Systems, Vol. 2, No. 5, pp 228-241, December 1994.

[Chen02] Chen P., Woods J. W., „Improved MC-EZBC with quarter-pixel motion vectors", ISO/IEC JTC1/SC29/WG11, MPEG2002/m8366, 2002.

[Cherk98] Cherkassky V., Mulier F., „Learning from data - concepts, theory, and methods”, John Wiley & Sons, USA, 1998.

[Chou99] Chou Yao-Tang, Hwang Wen-Liang, Huang Chung-Lin, „Very low-bit video coding based on gain-shape vector quantizer and matching pursuits”, IEEE International Conf. on Image Processing, Vol. 2, pp.76-80, October 1999.

[Daub91] Daubechies I., „Ten lectures on wavelets”, CBMS-NFS Series in Applied Mathematics, SIAM, 1991.

[Davis94] Davis G., „Adaptive non-linear approximations”, Ph.D. dissertation, Mathemetics Department, New York University, September 1994.

[Davis97] Davis G., Mallat S., Avellaneda M., „Adaptive greedy approximations,” Journal of Constructive Approximations, Vol. 13, pp. 57–98, 1997.

[Derv96] Derviaux C., Coudoux F., Gazalet M., Corlay P., „Blocking artifact reduction of DCT coded image sequences using a visually adaptive post-processing”, International Conf. on Image Processing, 1996.

[Doma98] Domański M., „Zaawansowane techniki kompresji obrazów i sekwencji wizyjnych”, Technical University of Poznan, 1998.

[Doma03] Domański M., Górecki L., „Very low bit rate video coding with matching pursuit and separable representations", European Conference on Circuit Theory and Design, Vol. II, pp. 85-88, Kraków, September 2003.

[Doma04] Domański M., Górecki L., „Matching pursuit for video coding with dictionary extracted from predicted image”, International Workshop on Systems, Signals and Image Processing, pp. 155-158, Poznañ, 2004.

[Doma05a] Domański M., Górecki L., „Video coding using matching pursuit with image-adapted dictionary”, International Workshop on Systems, Signals and Image Processing, Chalkidia, 2005.

[Doma05b] Domański M., Górecki L., „Adaptive dictionaries for matching pursuit with separable decomposition”, European Signal Processing Conference, Antalya, September, 2005.

[Durka96] Durka P., „Time-frequency analyses of EEG”, Ph.D. dissertation, Institute of Experimental Physics, Department of Physics, Warsaw University, August 1996.

[Figu00] Figueras R.M., Frossard P., Vandergheynst P., „Evolutionary multiresolution matching pursuit and its relations with the human visual system”, European Signal Processing Conference, 2002.

[Fross01] Frossard P., Vandergheynst P., „A posteriori quantized matching pursuit”, Proc. of IEEE Data Compression Conference, Snowbird, UT, March 2001.

[Fross04] Frossard P., Vandergheynst P., Figueras i Ventura R.M., Kunt M., „A posteriori quantization of progressive matching pursuit streams", IEEE Transactions on Signal Processing, Vol. 52, pp. 525-535, 2004.

[Fukuda00] Fukuda K., Kawanaka A., „Reduction of blocking artefacts by adaptive DCT coefficient estimation in block-based video coding”, International Conference of Image Processing, 2000.

[Garr05] Garrigues P., „Atom position coding in a matching pursuits based video coder”, MSc, University of California, Berkeley, 2005.

[Garr06] Garrigues P., Zakhor A., „Atom position coding in a matching pursuits based video coder”, Very Low Bit-rate Video, Lecture Notes in Computer Science 3893, proceedings 2006.

[Gar92] Gersho A., Gray R.M., „Vector quantization and signal compression”, Kluwer Academic Publishers, 1992.

[Ghan99] Ghanbari M., „Video coding: An introduction to standard codecs”, Institution of Electrical Engineering Press, London, 1999.

[Ghara98] Gharavi-Aikhansari M., „A model for entropy coding in matching pursuit”, IEEE International Conference on Image Processing, Vol. 1, pp. 778-782, 1998.

[Girod87] Girod B., „The efficiency of motion-compensating prediction for hybrid coding of video sequences”, IEEE Journal on Selected Areas in Commun., Vol.5, No.7, pp. 1140–1154, Aug. 1987.

[Girod93] Girod B., „Motion-compensating prediction with fractional-pel accuracy", IEEE Trans. on Commun., Vol. 41, No. 4, pp. 604–612, April 1993.

[Gore01] Gorecki L., „Kodowanie sekwencji wizyjnych z wykorzystaniem poszukiwania dopasowuj¹cego”, VI Poznañskie Warsztaty Telekomunikacyjne, III Sesja, pp. 3.2-1, December 2001.

[Goyal95a] Goyal V., „Quantized overcomplete expansions: analysis, synthesis and algorithms”, Mem. UCB/ERL Technical Report M95/57, June 1995.

[Goyal95b] Goyal V., Vetterli M., Thao N., „Quantization of overcomplete expansions”, Proc. of Data Compression Conference (DCC), pp. 13-22, 1995.

[Goyal97] Goyal V.K., Vetterli M., „Dependent coding in quantized matching pursuit”, Proceedings of the SPIE - Visual Communication and Image Processing, Vol. 3024, pp. 2-12, 1997.

[Goyal98] Goyal V., Vetterli M., Thao N., „Quantized overcomplete expansions in $\Re^n$: analysis, synthesis and algorithms”, IEEE Transaction on Information Theory, Vol. 44, pp. 16-31, January 1998.

[Gray84] Gray R.M., „Vector quantization”, IEEE Acoustics. Speech and Signal Processing Magazine, Vol.1, pp. 4-29, April 1984.

[H.261] „Video codec for audiovisual services at px64 kbit/s”, CCITT Recomendation H.261, CDM XV-R, 37-E Version 1: November 1990; ITU-T Rec. H.261, Version 2: March 1993.

[H.262] ITU-T and ISO/IEC JTC 1, „Generic coding of moving pictures and associated audio information – Part 2: Video”, ITU-T Rec. H.262 – ISO/IEC 13818-2 (MPEG-2 Video), November 1994.

[H.263] ITU-T, „Video coding for low bit rate communication”, ITU-T Rec. H.263, Version 1, November 1995; Version 2, January 1998; Version 3, November 2000.

[H.263+] Bormann C., Cline L., Deisher G., Gardos T., Maciocco C., Newell D., Ott J., Sullivan G., Wenger S., Zhu C., RFC 2429, Version of ITU-T Rec. H.263 Video (H.263+), available at: ftp://ftp.isi.edu/in-notes/rfc2429.txt, October 1998.

[H.264] ITU-T and ISO/IEC JTC 1, „Advanced video coding for generic audiovisual services”, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, 2003.

[H.264a] Joint Video Team of ITU-T and ISO/IEC JTC 1, „Draft ITU-T recommendation and final draft international standard of Joint Video Specification (ITU-T Rec.H.264 ISO/IEC 14496-10 AVC)”, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, March 2003.

[Hask97] Haskell B. G., Puri A., Netravali A.N., „Digital video: an introduction to MPEG-2”, Chapman & Hill, New York, 1997.

[Heus02] Heusdens R., Vafin R., Kleijn W., „Sinusoidal modelling using psychoacoustic-adaptive matching pursuit”, IEEE Signal Processing, Vol. 9, No. 8, August 2002.

[Hidaka89] Hidaka T., „Description of the proposing algorithm and its score for moving image (a part of the proposal package)", ISO/IEC JTC 1/SC 2/WG 8 MPEG 89/188, October 1989.

[Hong01] Hong Min-Cheol, Park Young Man, „Further experimental results of VCEG-N30 (loop filter for improving visual quality)”, ITU-T, 15th Meeting: Pattaya, Thailand, 4-6 December 2001.

[Horn81] Horn B.K.P., Schunck B.G., „Determining optical flow”, Art. Int. 17, pp. 185-203, 1981.

[Hotell33] Hotelling H., „Analysis of a complex of statistical variables into principal components”, Journal of Educational Psychology, 1933.

[ISO11172] ISO/IEC 11172: „Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 2: Coding of Moving Picture Information”, 1991.

[Jain81] Jain J.R., Jain A.K., „Displacement measurement and its application in inter-frame image coding”, IEEE Trans. Commun., Vol COM-29, pp. 1799-1808, December 1981.

[Jaur90] Jaureguízar J., Ronda J.I., García N., „Motion compensated prediction on digital HDTV”, Signal Processing V: Theories and Applications, Proceedings of EUSIPCO-90, Barcelona, Vol.2, pp. 753-756, September 1990.

[Jeon00] Jeon B., Oh Seokbyeung, Oh Seoung-Jun, „Fast matching pursuit method with distance comparison”, International Conference of Image Processing, Vol. 1, pp. 980-983, October 2000.

[JM8] JM Reference Software version 8.4, International Telecommunications Union, available at: http://iphome.hhi.de/suehring/tml/download/old_jm/jm84.zip, older JM versions available at: http://iphome.hhi.de/suehring/tml/download/old_jm/

[Joll86] Jolliffe I. T., „Principal component analysis”, Springer-Verlag, 1986.

[JPEG] Joint Photographic Experts Group, “Various Documents and Links,” available at: http://www.jpeg.org.

[Kell29] Kell R.D., „Improvements relating to electric picture transmission systems", British Patent No. 341,811, 1929.

[Koga81] Koga T., Iinuma K., Hirano A., Iijima Y., Ishiguro T., „Motion-compensated inter-frame coding for video conferencing", Proceedings of National Telecommunications Conference, Vol. 4, pp G5.3.1-G5.3.5, 1981.

[Li94] Li R., Zeng B., Liou M.L., „A new three-step search algorithm for block motion estimation", IEEE Trans. Circuits. Syst. Video Technology, Vol. 4, pp. 438-442, August 1994.

[Lin88] Lin C.M., Kwatra S.C., „Motion compensated interframe color image coding", International Conference on Communications, Part 1, Amsterdam, pp. 516-520, 1988.

[Linde80] Linde Y., Buzo A., Gray R. M., „An algorithm for vector quantizer design", IEEE Trans. Commun., Vol. COM-28, pp. 84-95, January 1980.

[Lin03] Lin Jian-Liang, Hwang Wen-Liang, Pei Soo-Chang, „FGS video coding based on bitplane coding of matching pursuit at low bit rates", 16th IPPR conference on Computer Vision, Graphics and Image Processing, 2003.

[List03] List P., Joch A., Lainema J., Bjøntegaard G., Karczewicz M., "Adaptive deblocking filter", IEEE Trans. on Circuits and Syst. for Video Tech., Vol. 13, no. 7, pp. 614-619, July 2003.

[Liu00] Liu Q., Wang Q., Wu L. „Dictionary with tree structure for matching pursuit video coding", IEE Journal, Vol. 36, pp. 1266-1268, July 2000.

[Lloyd82] Lloyd S.P., „Least squares quantization in PCM", IEEE Transactions on Information Theory, Vol. IT-28, No.2, pp. 129-137, March 1982.

[Mall93] Mallat S., Zhang Z., „Matching pursuits with time-frequency dictionaries", Technical Report 619, Computer Science Department, New York University, August 1993 (also IEEE Transactions in Signal Processing, Vol.41, No.12, pp. 3397-3415, December 1993).

[Mall99] Mallat S., „A wavelet tour of signal processing", 2nd ed. Academic Press, 1999.

[Malvar01] Malvar H. S., „Low-complexity length-4 transform and quantization with 16-bit arithmetic", Video Coding Experts Group, Q.6/SG16, VCEG-N44, Santa Barbara, September 2001.

[Marpe01a] Marpe D., Blättermann G., Wiegand T., „Adaptive codes for H.26L", JVT Document JVT-L13, Eibsee, January 2001.

[Marpe01b] Marpe D., Blättermann G., Heising G., Wiegand T., „Video compression using context-based adaptive arithmetic coding", Proc. IEEE ICIP., pp. 558-561, 2001.

[Marpe03] Marpe D., Schwarz H., Wiegand T., „Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard”, IEEE Transactions on Circuits and Systems for Video Technology, Vol.13, no.7, July 2003.

[Meng03] Meng B., Au O., Wong C.W., Lam H.-K., „Efficient intra-prediction mode selection for 4x4 blocks in H.264”, Proc. ICME, Baltimore, USA, July 2003.

[Moffat95] Moffat A., Neal R.M., Witten I.H., „Arithmetic Coding Revisited”, Proc. IEEE Data Comp. Conf., pp. 201-211, 1995.

[MPEG-1] ISO/IEC JTC 1, „Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s –Part 2: Video”, ISO/IEC 11172-2 (MPEG-1), Mar. 1993.

[MPEG-2] ITU-T and ISO/IEC JTC 1, „Generic coding of moving pictures and associated audio information – Part 2: Video”, ITU-T Rec. H.262 – ISO/IEC 13818-2 (MPEG-2 Video), November 1994.

[MPEG-4a] „MPEG-4 video verification model version 4.0”, ISO/IEC JTC1/SC29/WG11/ M1380, Chicago, October 1996.

[MPEG-4b] „MPEG-4 visual coding standard, Final draft of international standard”, ISO/IEC JTC1/SC29/WG11, December 1997.

[MPEG-4c] „Editors' text for ISO/IEC 14496-10:2005 (AVC 3rd Edition)”, ISO/IEC JTC1/SC29/WG11, Doc. MPEG05/N7081, Busan, April 2005.

[Nafi96] Nafie M., Ali M., Tewfik A., „Optimal subset selection for adaptive signal representation”, IEEE Int. Conf. Acoust. Speech Signal Proc., Vol 6, pp. 2511-2514, May 1996.

[Nagel95] Nagel H.H., „Optical flow estimation and the interaction between measurement errors at adjacent pixel positions”, International Journal of Computer Vision, Vol.15, pp.271-288, 1995.

[Neff95] Neff R., Zakhor A., „Matching pursuit video coding at very low bit rates”, IEEE Data Compression Conf., pp.411-420, March 1995.

[Neff96a] Neff R., Martinian E., Miroslavsky E., Zakhor A., „Experiment T3: Matching pursuit coding of prediction errors. Progress report”, ISO/IEC JTC1 SC29 WG11, M1365, October 1996.

[Neff96b] Neff R., Miroslavsky E., Al.-Shaykh O., Shang-Pin Chang, Martinian E., Zakhor A., „Experiment T3: Matching pursuit coding of prediction errors. Current status and results”, ISO/IEC JTC1 SC29 WG11, M1568, November 1996.

[Neff97] Neff R., Zakhor A., „Very low bit rate video coding based on matching pursuits”, IEEE Trans. Circuits and Systems for Video Technology, vol.7, No.1, February 1997.

[Neff97a] Neff R., Miroslavsky E., Al.-Shaykh O., Shang-Pin Chang, Martinian E., Zakhor A., „Experiment T3: Matching pursuit coding of prediction errors", ISO/IEC JTC1 SC29 WG11, M1817, February 1997.

[Neff98] Neff R., Nomura T., Zakhor A., „Decoder complexity and performance comparison of matching pursuit and MPEG-4 video codecs", IEEE International Conference on Image Processing, Vol. 3, pp. 783-787, October 1998.

[Neff99] Neff R., Zakhor A., „Modulus quantization for matching pursuit video coding", IEEE Trasactions no Circuits and Systems for Video Technology, Vol. 10. No. 6, pp. 895-912, September 1999.

[Neff00] Neff R., Zakhor A., „Dictionary approximation for matching pursuit video coding", IEEE Int. Conf. Image Processing, pp. 828-831, 2000.

[Neff02a] Neff R., Zakhor A., „Matching pursuit video coding. Part I: Dictionary approximation", IEEE Transactions Circuits and Systems for Video Technology, Vol. 12, pp. 13-26, January 2002.

[Neff02b] Neff R., Zakhor A., „Matching pursuit video coding. Part II: Operational models for rate distortion", IEEE Transactions Circuits and Systems for Video Technology, Vol. 12, pp. 27-39, January 2002.

[Neuma55] Neumann J. von, „Mathematical foundations of quantum mechanics", Princeton University Press, 1955.

[Norc02] Norcern R., Schneider P., Uhl A., „Approaching real-time processing for matching pursuit image coding", Media Processors 2002, Vol. 4674, pp. 84-90, January 2002.

[Oh00] Oh S., Jeon B., „Fast matching pursuit using the absolute symmetry of gabor-function", Image Proc. and Image Understanding, Vol. 12, pp. 365-369, January 2000.

[Oh01] Oh S., Jeon B., „Fast matching pursuit method using property of symmetry and classification for scalable video coding", IEICE Trans. Fundamentals, Vol.E84-A, No.6, June 2001.

[Ohm04] Ohm J.R., „Multimedia communication technology“, Springer, Heildeberg 2004.

[Pan03] Pan F., „Fast mode decision for intra prediction", JVT-G013, 7-th Meeting, Pattaya, Thailand, March 2003.

[Peng00] Peng Chien-Kai, Hwang Wen-Liang, Huang Chung-Lin, „Matching pursuits low bit rate video coding with codebooks adaptation", Int. Conf. on Acoustics, Speech and Signal Processing, Vol. 1, pp. 408-411, June 2000.

[Po96] Po L.M., Ma W.C., „A novel four step search algorithm for fast block motion estimation”, IEEE Trans. Circuits System Video Technology, Vol. 6, pp. 313-317, June 1996.

[Rao90] Rao K.R., Yip P., „Discrete Cosine Transform: Algorithms, Advantages, Applications”, Boston: Academic Press, Chapter 4, 1990.

[Red98] Redmill D.W., Bull D.R., Czerepinski P., „Video coding using a fast non-separable matching pursuits algorithm”, IEEE International Conference on Image Processing, Vol.5, pp. 769-773, October 1998.

[Rich03] Richardson I., „H.264 and MPEG-4 video compression”, Wiley, Chichester 2003.

[Sama04] Samadani R., Sundararajan A., Said A., „Deringing and deblocking DCT compression artifacts with efficient shifted transforms”, IEEE International Conference on Image Processing, 2004.

[SIM3] COST211ter Simulation Subgroup, „Simulation model for very low bit rate image coding (SIM3)”, April 1993.

[Sch01] Schmid-Saugeon P., Zakhor A., „Learning dictionaries for matching pursuits based video coders”, IEEE International Conference on Image Processing, Vol. 3, pp. 562-565, Thessaloniki, 2001.

[Sch04] Schmid-Saugeon P., Zakhor, A, „Dictionary design for matching pursuit and application to motion-compensated video coding”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 6, pp. 880-886, June 2004.

[Shap93] Shapiro J., „Embedded image coding using zero-trees of wavelet coefficients”, IEEE Transactions on Signal Processing, vol. 41, pp. 3445-3462, December 1993.

[Srin04] Srinivasan S., Hsu Pohsiang, Holcomb T., Mukerjee K., Regunathan S. L., Lin Bruce, Liang Jie, Lee Ming-Chieh, Ribas-Corbera J., „Windows Media Video 9: overview and applications”, Signal Processing: Image Communication, Volume 19, Issue 9, pp.851-875, October 2004.

[Sull91a] Sullivan G.J., Baker R.L., „Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks”, IEEE Global Telecommun. Conf. (GLOBECOM), pp. 85-90, December 1991.

[Sull91b] Sullivan G.J., Baker R.L., „Motion compensation for video compression using control grid interpolation”, IEEE Intl. Conf. on Acoust., Speech, Signal Proc. (ICASSP), Toronto, Canada, pp. 2713-2716, May 1991.

[Taki74] Taki Y., Hatori M., Tanaka S., „Interframe coding that follows the motion”, Institute of Electronics and Commun. Eng. of Japan Annual Conv. (IECEJ), pp. 1263, July 1974.

[Tekalp95] Tekalp A., „Digital video processing”, Prentice-Hall PIR, Upper Saddle River, 1995.

[TM5] MPEG-2, Test Model 5, Doc. ISO/IEC JTC1/SC29 WG11/93-225b. Test Model Editing Committee, April 1993.

[TML4] Bjøntegaard G., „H.26L Test Model Long Term Number 4 (TML-4) draft0”, ITU-Document Q15-J-72, Osaka Meeting, May 2000

[TML8] Bjøntegaard G., „H.26L Test Model Long Term Number 8 (TML-8) [draft]”, ITU-T Q.6/SG16 (VCEG) Document, June 2001.

[TMN5] ITU Telecommunication Standardization Sector LBC-95, „Video codec test model TMN5”, available at: http://www.nta.no/brukere/DVC/.

[Vander01] Vandergheynst P., Frossard P., „Adaptive entropy-constrained matching pursuit quantization”, International Conference on Image Processing, Vol. 2, pp. 423-426, October 2001.

[VC-1] „Proposed SMPTE standard of television: VC-1 compressed video bitstream format and decoding process”, SMPTE Draft Standard for Television, SMPTE 421M, August 2005.

[Vett92] Vetterli M., Herley C., „Wavelets and filter banks: Theory and design”, IEEE Trans. Acoust. Speech Signal Proc., Vol. 40, No. 9, pp. 2207-2232, 1992.

[Vett94] Vetterli M., Kalker T., „Matching pursuit for compression and application to motion compensated video coding”, IEEE Image Processing Conf., pp. 725-729, 1994.

[Vlee98] Vleeschouwer C., Macq B., „New dictionaries for matching pursuit video coding”, IEEE Int. Conf. Image Processing, Vol. 9, pp. 764-768, 1998.

[VM8] Video Group, „Text of ISO/IEC 14496-2 MPEG-4 Video VM-Version 8.0”, ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Associated Audio MPEG 97/W1796, Stockholm, Sweden, July 1997.

[Wieg99] Wiegand T., Steinbach E., Girod B., „Long-term memory prediction using affine motion compensation”, IEEE International Conference on Image Processing, Vol. 1, pp.51-55, 1999.

[Wieg01] Wiegand T., Girod B., „Multi-frame motion-compensated prediction for video transmission”, Kluwer Academic Publishers, September 2001.

[Wieg03] Wiegand T., Sullivan G. J., Bjøntegaard G., Luthra A., „Overview of the H.264/AVC video coding standard”, IEEE Trans. on Circuits and Syst. for Video Tech., Vol. 13, no. 7, pp. 560-576, July 2003.

[Wien03] Wien M., „Variable block-size transform for H.264/AVC”, IEEE Trans. on CSVT, Vo1.13, No.7, pp. 604-619, 2003.

[Xin04] Xin Jun, Vetro A., Sun Huifang, „Converting DCT coefficients to H.264/AVC transform coefficients”, Mitsubishi Electric Research Laboratory, TR-2004-058, June 2004.

[Zhu00] Zhu C., Ma S.S., „A new diamond search algorithm for fast block matching motion estimation”, IEEE Trans. Image Processing, Vol.. 9, pp. 287-290, February 2000.

[Zhu02] Zhu C., Lin X., Chau L.P., „Hexagon-based search pattern for fast block motion estimation”, IEEE Trans. Circuits Syst. Video Technology, Vol. 12, pp. 349-355, May 2002.

[Ziyad98] Ziyad N., Gilmore E., Chouikha M., „Dictionary approaches to image compression and reconstruction”, IASTED International Conf. On Signal and Image Processing, Las Vegas, 12-16 September 1998.

Appendix

Theorem 1

Let $(\mathbb{R}^n, d)$ be a metric space with distance defined as follows:

$$d(x,y) = \sum_{i=1}^{n} (x_i - y_i)^2 \ .$$

For any $x, y \in \mathbb{R}^n$, the distance $d(x, ay)$ is minimal, if and only if,

$$a = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} y_i^2} \ .$$

This theorem is well known, and the proof can be found in any book of linear algebra.

Proof :

Note, that :

$$d(x,ay) = \sum_{i=1}^{n} (x_i - ay_i)^2 = a^2 \sum_{i=1}^{n} y_i^2 - 2a \sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} x_i^2 \ .$$

For chosen $x, y \in \mathbb{R}^n$, the distance $d(x, ay)$ can be treated as polynomial of variable a.

Let $W_{(x,y)}(a) = d(x,ay)$. Since,

$$W'_{(x,y)}(a) = 2a \sum_{i=1}^{n} y_i^2 - 2 \sum_{i=1}^{n} x_i y_i$$

and

$$W_{(x,y)}(\pm\infty) \to +\infty$$

then minimum of $W_{(x,y)}(a)$ is at

$$a = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} y_i^2} \ .$$

Q.E.D.

****
****

We say that a is an expansion coefficient of vector $g \in \mathbb{R}^n$ for vector $f \in \mathbb{R}^n$, if :

$$\forall b \in \mathbb{R} \qquad d(f, bg) \geq d(f, ag).$$

The Theorem 1 allows for calculation of an expansion coefficient for any vectors $f, g \in \mathbb{R}^n$. In addition, if vector g is normalised i.e. $\|g\| = 1$, then the value of expansion coefficient can be calculated in the easier way, this is :

$$a = \sum fg = \langle f, g \rangle.$$

Theorem 2

Let $(\mathbb{R}^n, d)$ be a metric space defined as in the Theorem 1.

Let $diff(x,y)$ be a measure of representation i.e. how two vectors can be close together:

$$diff(x,y) = \min\{d(x,by); b \in \mathbb{R}\} = d(x,ay),$$

where a is an expansion coefficient of y for x.

For given $u \in \mathbb{R}^n$ and $\{v_i \in \mathbb{R}^n\}_1^M$ such that $\|v_i\| = 1$, and for expansion coefficients of v_i for u, i.e. $\{a_i \in \mathbb{R}\}_1^M$

$$\min\{diff(u,v_i); i = 1..M\} = diff(u,v_l) \quad \Leftrightarrow \quad |a_l| = \max\{|a_1|,|a_2|,..,.|a_M|\}.$$

Proof:

In accordance with the Theorem 1, it is known that for any v_i:

$$diff(u,v_i) = d(u,a_i v_i),$$

where $a_i = \dfrac{\sum uv_i}{\sum v_i^2} = \sum uv_i$.

Then,

$$diff(u,v_i) = \sum u^2 + a_i^2 \sum v_i^2 - 2a_i \sum uv_i .$$

Taking into consideration the definition of a_i and $\|v_i\| = 1$,

$$diff(u,v_i) = \|u\|^2 - a_i^2 .$$

Note, that for given $u \in \mathbb{R}^n$, the energy of $\|u\|$ is constant, furthermore $dif(u,v_i) \geq 0$, therefore the minimal value of $diff(\cdot,\cdot)$ for a given set $\{v_i\}_1^M$ is for:

$|a_i| = \max\{|a_1|,|a_2|,..,.|a_M|\}$.

Q.E.D.

Theorem 3

Let $s(\cdot,\cdot)$ be a similarity of vectors defined as:

$$s(f_1, f_2) = \frac{\left|\left\langle f_1, f_2 \right\rangle\right|}{\|f_1\|\|f_2\|}.$$

For any normalized vectors $f_1, f_2 \in \mathcal{H}$, such that:

$$f_2(j) = f_1(j) + d(j) \quad ,$$

$$s(f_1, f_2) = 1 - \frac{1}{2}\sum d(j)^2 .$$

Proof :

Note, that vectors similarity can be simplified to

$$s(f_1, f_2) = \left\langle f_1, f_2 \right\rangle,$$

since $\|f_1\| = 1$ and $\|f_2\| = 1$.

Then,

$$s(f_1, f_2) = \left\langle f_1, f_2 \right\rangle = \sum f_1(j)\big(f_1(j) + d(j)\big) = 1 + \sum f_1(j)d(j).$$

Note that,

$$\|f_2\|^2 = \sum \big(f_1(j) + d(j)\big)^2 = 1 \Rightarrow \sum f_1^{\ 2} + 2\sum f_1(j)d(j) + \sum d(j)^2 = 1$$

$$2\sum f_1(j)d(j) + \sum d(j)^2 = 0 \quad \Rightarrow \quad \sum f_1(j)d(j) = -\frac{1}{2}\sum d(j)^2$$

Therfore,

$$s(f_1, f_2) = 1 - \frac{1}{2}\sum d(j)^2 .$$

Q.E.D.

Theorem 4

For the sake of simplicity, let us consider 2-D function space $\mathcal{H} = \{f : X \times Y \to \mathbb{R}\}$. In addition, let $\mathcal{H}_{S\&N} \subset \mathcal{H}$ contains all separable and normalised functions from the defined function space $\mathcal{H}$. Let $f \in \mathcal{H}$, $s \in \mathcal{H}_{S\&N}$, $s(i,j) = \alpha(i)\beta(j),\ \|\alpha\| = 1,\ \|\beta\| = 1$.

Let:

$$\alpha'(i) = K\sum_j f(i,j)\beta(j),$$

where $K \in \mathbb{R}$ normalizes function α' , this is $\|\alpha'\| = 1$.

The new function $s'(i,j) = \alpha'(i)\beta(j)$ satisfies the inequality:

$$diff(f,s) \geq diff(f,s').$$

Proof:

In accordance with definition, the above thesis is equivalent to $|\langle f,s\rangle| \leq |\langle f,s'\rangle|$.

Let,

$$s(i,j) = \alpha(i)\beta(j)\ ,$$

where :

$\alpha \in (X \to \mathbb{R})$ and $\|\alpha\| = 1$,

$\beta \in (Y \to \mathbb{R})$ and $\|\beta\| = 1$.

Then,

$$\langle f,s\rangle = \sum_i \sum_j f(i,j)\alpha(i)\beta(j) = \sum_i \alpha(i)\sum_j f(i,j)\beta(j).$$

Let

$$\alpha'(i) = K\sum_j f(i,j)\beta(j),$$

where $K \in \mathbb{R}$ normalizes function α' this is $\|\alpha'\| = 1$.

Then, we can create the new function $s'(i,j) = \alpha'(i)\beta(j)$ for which:

$$|\langle f,s'\rangle| = \left|\sum_i \sum_j f(i,j)\alpha'(i)\beta(j)\right| = \left|\sum_i \alpha'(i)\sum_j f(i,j)\beta(j)\right| = \left|\frac{1}{K}\sum_i \alpha'(i)\alpha'(i)\right| = \left|\frac{1}{K}\right|\|\alpha'\|^2 = \left|\frac{1}{K}\right|.$$

Similarly,

$$|\langle f,s\rangle| = \left|\sum_i \alpha(i)\sum_j f(i,j)\beta(j)\right| = \left|\frac{1}{K}\sum_i \alpha(i)\alpha'(i)\right| = \left|\frac{1}{K}\langle \alpha,\alpha'\rangle\right| \leq \left|\frac{1}{K}\right|\|\alpha\|\|\alpha'\| = \left|\frac{1}{K}\right|.$$

Q.E.D.

****
****

Theorem 4 can be generalised to N-dimensional function space.

Theorem 4 allows creating sequence of separable functions that minimise distance between input function f and last calculated separable function. In other words, Theorem 4 allows us to find the best separable function under the $diff(\cdot,\cdot)$ criteria.