CODIFICAÇÃO DE VÍDEO USANDO PLANOS DE BITS GENERALIZADOS

Rogério Caetano

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Gelson Vieira Mendonça, Ph.D.

Prof. Abraham Alcaim, Ph.D.

Prof. Marcos Craizer, D.Sc

Prof. Ricardo Lopes de Queiroz, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2004

CAETANO, ROGÉRIO

Codificação de Vídeo Usando Planos de Bits Generalizados [Rio de Janeiro] 2004

xv,180 pp 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Elétrica, 2004)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1.Codificação de vídeo

2.Matching Pursuits

3.Planos de Bits

4.Dicionários Super-completos

5.Dicionários Não-separáveis

6.Controle da Taxa de Bits

I.COPPE/UFRJ II.Título (série)

Agradecimentos

Os meus agradecimentos são para todos aqueles que foram compreensivos e pacientes comigo durante o difícil período de tempo em que elaborei esta tese. Mais particularmente, posso citar:

Meus pais, André e Zuleika, que durante toda a minha vida me apoiaram e tiveram grande responsabilidade no processo de formação da pessoa que sou hoje.

Meu irmão Ronaldo e minhas irmãs Rosemary e Rosângela, que sempre me apoiaram nas horas mais difíceis.

Minha esposa, Marlucia, que teve que agüentar inúmeros momentos de extremo cansaço, sendo sempre paciente, atenciosa e carinhosa, me apoiando nas mais diversas situações.

Meu orientador e amigo, Eduardo, por toda a sua paciência, incentivo e suporte durante todo o período desta tese. E por ter me mostrado que devemos sempre acreditar em nós mesmos e que com um trabalho árduo e sério somos capazes de alcançar os nossos sonhos.

Aos amigos Luiz Wagner, Décio Fonini, Fábio Freeland, Lisandro Lovisolo, Cássio Barbosa, Leonardo Fonteles e a todos os meus amigos, cuja lista completa necessitaria de mais um volume como este, mas que, tenho certeza, estão conscientes de como foram muito importantes não só neste momento, mas em toda a minha vida.

A todas estas pessoas, muito obrigado, esperando que possa retribuir à altura tudo o que fizeram por mim.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

CODIFICAÇÃO DE VÍDEO USANDO PLANOS DE BITS GENERALIZADOS

Rogério Caetano

Abril/2004

Orientador: Eduardo Antônio Barros da Silva

Programa: Engenharia Elétrica

Entre os métodos de codificação de vídeo que apresentam os melhores desempenhos estão aqueles que são baseados no algoritmo matching pursuits. Nesta Tese é proposto um método alternativo ao algoritmo matching pursuits, baseado na noção de decomposição em planos de bits generalizados. A estrutura da decomposição em planos de bits generalizados é tal que a quantização está intrinsecamente embutida no processo de decomposição, e não existe necessidade de se especificar qualquer parâmetro de quantização. Além disso, foi apresentado um teorema que estabelece um limitante teórico para as características taxa-distorção de tais decomposições generalizadas. A eficiência do método proposto é testada usando como ponto de partida o codificador de vídeo clássico utilizado por Neff e Zakhor. Os resultados obtidos são promissores, apresentando, sem qualquer suposição a priori sobre o comportamento taxa-distorção dos quadros codificados ou qualquer aumento na complexidade computacional, uma significativa melhora sobre o codificador de vídeo clássico que utiliza o algoritmo matching pursuits. Foram também investigados diversos dicionários super-completos, incluindo dicionários não-separáveis, formados por regiões de dicionários separáveis. Confirmando previsões teóricas, estes dicionários proporcionaram melhoras no desempenho dos codificadores.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

VIDEO CODING USING GENERALIZED BIT PLANES

Rogério Caetano

April/2004

Advisor: Eduardo Antônio Barros da Silva

Department: Electrical Engineering

Among the best performing video coding methods are those based on the matching pursuits algorithm. In this Thesis it is proposed a novel method for performing matching pursuits quantization, based on the notion of decomposition in generalized bit-planes. The structure of such decompositions is such that once the decomposition is carried out, it is already quantized, and there is no need to set up any quantization parameters. It does so by generating a decomposition that is readily organized in bit-planes. It provides an elegant solution to the trade-off between quantization of coefficients and number of passes in the matching pursuits algorithm. In addition, a theorem is presented, setting bounds for the R-D performance of such generalized decompositions. The effectiveness of the proposed method is tested using the framework of Neff and Zakhor's matching pursuits video encoder. The results obtained are promising, presenting, without any ad-hoc assumptions about the R-D behavior of the coded frames or any increase in computational complexity, a significant improvement over the classical matching pursuits video coders. Also, to improve the performance of the proposed video encoder it is investigated different overcomplete dictionaries, including non-separable ones, formed by rotations of separable structures. Confirming the theoretical analysis, the use of these dictionaries lead to improvements in coding performance.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Nos últimos anos a quantidade de informações visuais que devem ser processadas e armazenadas tem aumentado significativamente. Elas são usadas em aplicações que vão desde sensoriamento remoto até o entretenimento, incluindo medicina, imagens de satélites, visão de robôs e inspeção industrial. Videofones e videoconferências já são uma realidade e já temos hoje acesso à televisão digital em nossos lares. Sistemas Multimídia, que se apóiam bastante no uso de imagens digitais, estão penetrando o mercado consumidor rapidamente.

Neste contexto, as técnicas de compressão de imagens e vídeo, que visam reduzir a quantidade de bits necessária para a representação das informações visuais, e ao mesmo tempo manter uma qualidade aceitável, são essenciais para a exeqüibilidade de tais sistemas. Estas técnicas somente são viáveis porque as informações visuais, representadas na forma de imagens digitais, são bastante redundantes, e esta redundância pode ser explorada para comprimi-las, isto é, reduzir a quantidade de bits necessária para a sua representação [1].

Dentre as técnicas de compressão de vídeo, aquelas baseadas no Algoritmo Matching Pursuits proposto por Mallat e Zhang em [2] tem apresentado excelente resultados [3, 4, 5, 6, 7, 8, 9]. Neste algoritmo, a imagem residual é decomposta em um dicionário super-completo, gerando uma seqüência de pares os quais especificam os átomos usados e seus coeficientes correspondentes. Um dado compromisso taxa$\times$distorção é alcançado especificando-se o número de átomos e os quantizadores dos coeficientes destes átomos. Diversas estratégias têm sido propostas a fim de se obter bons compromissos taxa$\times$distorção [10, 11, 7].

Nesta Tese é proposto um novo método de decomposição que é equivalente a fazer a quantização nos "matching pursuits"baseada na noção de planos de bits generalizados. A estrutura de tais decomposições é tal que a decomposição e a quantização estão embutidas na mesma operação, e não existe necessidade de se especificar qualquer parâmetro de quantização. Tais decomposições, diferentemente das tradicionais, produzem apenas uma seqüência de índices. Isto gera uma decomposição que pode ser organizada em planos de bits. Elas apresentam uma solução elegante para o compromisso existente entre a quantização dos coeficientes e o número de passos do algoritmo matching pursuits. Na verdade, a decomposição em planos de bits generalizados pode ser considerada como uma generalização de qualquer decomposição em um dicionário seguida por uma representação dos coeficientes em formato binário usando precisão finita.

Visto que o algoritmo proposto nesta Tese mapeia a imagem residual em uma seqüência de índices, é obtido um preciso controle da taxa de bits gasta com cada átomo adicionado, e conseqüentemente, a decomposição pode ser parada sempre que a taxa de bits desejada seja encontrada. Usando este fato, são propostas duas estratégias de controle de taxa [12, 13, 14, 15, 16, 17, 18, 19]. Uma, usa um procedimento que divide, de forma precisa, a taxa de bits da seqüência igualmente entre todos os seus quadros, e outra utiliza um esquema que emprega otimização de Lagrange a fim de solucionar o problema de alocação de bits ótima [7]. No primeiro esquema proposto tem a vantagem de alocar precisamente a taxa desejada para o quadro, e conseqüentemente obter a taxa de bits alvo. No segundo é obtida a alocação ótima no senso taxa-distorção.

Além disso, são investigados diversos outros dicionários super-completos, incluindo dicionários não-separáveis [20, 6, 21, 22, 23], formados por regiões de dicionários separáveis. Confirmando previsões teóricas, estes dicionários proporcionam melhoras no desempenho dos codificadores.

Esta Tese está organizada como a seguir: a primeira parte está escrita em português e contém a introdução, o texto principal nos capítulos 2 até o capítulo 6 e uma conclusão. O conteúdo estes capítulos está reproduzido com mais detalhes em 11 apêndices.

O capítulo 2 apresenta o algoritmo matching pursuits e suas principais características.

No capítulo 3 é proposto um novo algoritmo (O Algoritmo Matching Pursuits em Planos de Bits Generalizados – Algoritmo MPGBP), que faz um tipo de decomposição

adaptativa, no qual um sinal é decomposto em planos de bits generalizados. Além disso, limitantes para as características taxa-distorção do algoritmo proposto são derivados analiticamente.

No capítulo 4 é usado o algoritmo MPGBP para codificar seqüências de vídeo usando um dicionário super-completo formado de funções de Gabor.

No capítulo 5 são propostas duas estratégias de controle de taxa a fim de melhorar o desempenho do codificador de vídeo MPGBP.

No capítulo 6 é investigado o desempenho do codificador de vídeo MPGBP usando dicionários super-completos separáveis e não-separáveis na codificação de diversas seqüências de vídeo.

O capítulo 7 mostra as conclusões e apresenta as sugestões para trabalhos futuros.

Capítulo 2

Decomposição de Sinais

Neste capítulo nós descrevemos as principais características do algoritmo matching pursuits proposto por Mallat e Zhang em [2]. Este algoritmo decompõe sinais iterativamente em um conjunto de funções que são selecionadas de um dicionário redundante. Em cada iteração, um procedimento de procura exaustiva é usado para escolher a estrutura do dicionário que melhor representa o sinal. A projeção do sinal nesta estrutura é subtraída do mesmo, e a diferença entre ambos é passada para a próxima iteração como um resíduo. As funções são selecionadas a fim de obter um bom casamento com as estruturas do sinal. Maiores detalhes sobre o algoritmo matching pursuits descrito neste capítulo, encontram-se no apêndice B.

2.1 O Algoritmo Matching Pursuits

Mallat and Zhang propuseram em [2] o algoritmo matching pursuits, o qual decompõe um sinal $\mathbf{x}$ iterativamente em passos, onde em cada passo, a estrutura (função pertencente a um dicionário redundante) que fornece a maior redução da energia do sinal ou átomo (o melhor casamento) é escolhida e subtraída do sinal. O sinal residual é passado para o próximo passo e o processo é repetido. O melhor casamento é dado pelo átomo com maior produto interno com o sinal.

No algoritmo Matching Pursuits nós usualmente decompomos um sinal $\mathbf{x}$ de dimensão N em um dicionário redundante $\mathcal{D} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_M\}$, $\|\mathbf{g}_i\| = 1$, $\forall i$. Os $\mathbf{g}_i$ são em geral chamados de átomos. O dicionário é chamado de redundante porque, em geral, $M > N$.

O sinal $\mathbf{x}$ é então aproximado em P passos como [2]

$$\mathbf{x} \approx \sum_{n=1}^{P} p_n \mathbf{g}_{\gamma_n} \tag{2.1}$$

Os pares (p_n, γ_n) são obtidos pelo algoritmo 1 abaixo

Algoritmo 1

❶ Começar com $\mathbf{w} = \mathbf{x}$, $n = 1$.

❷ Repetir até que um critério de parada seja encontrado

(a) Escolher $\gamma_n \in \{1, \ldots, M\}$ tal que

$$\mathbf{w} \cdot \mathbf{g}_{\gamma_n} = \max_{1 \le j \le M} \{\mathbf{w} \cdot \mathbf{g}_j\}.$$

(b) Escolher $p_n = < \mathbf{w}, \mathbf{g}_{\gamma_n} >$.

(c) Substituir $\mathbf{w}$ por $\mathbf{w} - p_n \mathbf{g}_{\gamma_n}$.

(d) Incrementar n.

❸ Parar.

É importante notar que decomposições feitas usando o Algoritmo Matching Pursuits são diferentes daquelas feitas usando algoritmos baseados em transformadas, onde nós temos que codificar todos os coeficientes da transformada. Em outras palavras, os algoritmos baseados em transformadas, tal como a DCT, tem que usar sempre todas as estruturas do dicionário durante a decomposição do sinal. No Algoritmo Matching Pursuits, este necessita codificar somente as projeções das estruturas escolhidas e os índices destas estruturas. Assim, o Algoritmo Matching Pursuits faz um tipo de decomposição adaptativa, visto que, para cada sinal, nós podemos usar um diferente conjunto de estruturas de todo o dicionário. O Algoritmo Matching Pursuits pode também fazer uma decomposição compacta do sinal. Isto se deve ao fato de que como um dicionário super-completo

é usado, nós podemos escolher as estruturas do dicionário as quais são mais similares às estruturas do sinal que nós desejamos codificar.

Mais detalhes deste algoritmo podem ser encontrados em [2] e no apêndice B, onde é mostrado que a energia do sinal residual decresce monotonicamente conforme o número de passos P é incrementado, e tende à zero quando P tende para o infinito.

2.2 Conclusões

Neste capítulo nos descrevemos o algoritmo matching pursuits proposto por Mallat e Zhang em [2]. As principais características deste algoritmo também foram comentadas. No apêndice B ele será descrito em mais detalhes.

Capítulo 3

Decomposição em Planos de Bits Generalizados

Neste capítulo nós propomos um novo algoritmo para fazer uma decomposição do tipo matching pursuits. O algoritmo proposto faz a decomposição de um sinal em planos de bits generalizados, com cada plano de bits sendo composto de um conjunto de átomos. Diferentemente do que ocorre no algoritmo matching pursuits clássico, no algoritmo proposto não existem coeficientes para serem quantizados, isto é, somente um conjunto de índices, correspondentes aos átomos de cada plano de bits, necessitam ser transmitidos. Isto fornece uma solução elegante para o problema da quantização dos coeficientes do algoritmo matching pursuits. É importante notar que maiores detalhes sobre o algoritmo de decomposição em planos de bits generalizados proposto neste capítulo, assim como uma derivação analítica de um limitante teórico para as suas características taxa-distorção encontram-se no apêndice C.

3.1 Decomposição do sinal em planos de bits generalizados

Suponhamos que $\mathbf{x}$ é um sinal N-dimensional que pode ser decomposto em um dicionário super-completo $\mathcal{D} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_M\}$, $\|\mathbf{g}_i\| = 1, \forall i$ como

$$\mathbf{x} = \sum_{n=1}^{M} c_n \mathbf{g}_n \tag{3.1}$$

Sem perda de generalidade, nós assumimos que $\|\mathbf{x}\| \leq 1$ e consideramos que o dicionário $\mathcal{D}$ seja completo, de modo que uma expansão em M termos possa representar $\mathbf{x}$ com distorção zero. Além disso, como $\|\mathbf{g}_i\| = 1$, existe uma expansão na forma da equação (3.1) tal que $|c_n| \leq 1$.

Visto que $|c_n| \leq 1$, nós podemos escrever a representação binária para c_n como

$$c_n = s_n \sum_{j=1}^{\infty} 2^{-j} b_{j,n} \tag{3.2}$$

Onde $s_n \in \{-1, 1\}$, e $b_{j,n} \in \{0, 1\}$.

Substituindo a representação binária de c_n, obtida com a equação (3.2), na equação (3.1), nós obtemos

$$\begin{aligned} \mathbf{x} &= \sum_{n=1}^{M} s_n \sum_{j=1}^{\infty} 2^{-j} b_{j,n} \mathbf{g}_n = \sum_{j=1}^{\infty} 2^{-j} \sum_{n=1}^{M} b_{j,n} s_n \mathbf{g}_n \\ &= \sum_{j=1}^{\infty} 2^{-j} \sum_{n=1}^{M} b_{j,n} \overline{\mathbf{g}}_n \end{aligned} \tag{3.3}$$

Como $s_n \in \{-1, 1\}$, temos $\overline{\mathbf{g}}_n = s_n \mathbf{g}_n \in \overline{\mathcal{D}} = \{\pm\mathbf{g}_1, \pm\mathbf{g}_2, \ldots, \pm\mathbf{g}_M\}$. Então, se definirmos o índice $i_{j,l}$ de modo que, $b_{j,i_{j,l}} = 1$ para $l \in \{1, 2, \ldots, L_j\}$, $L_j \leq M$, e $b_{j,i_{j,l}} = 0$ nos outros casos, a equação (3.3) pode ser expressa como

$$\mathbf{x} = \sum_{j=1}^{\infty} 2^{-j} \sum_{l=1}^{L_j} \overline{\mathbf{g}}_{i_{j,l}} \tag{3.4}$$

Note que a equação (3.4) pode ser considerada como uma decomposição do sinal $\mathbf{x}$ em planos de bits generalizados. O plano de bits j é composto pelas funções $\overline{\mathbf{g}}_{i_{j,l}}$ para $l = 1, \ldots, L_j$. Em [24] foi proposto um algoritmo convergente para achar tais decomposições, na mesma filosofia do algoritmo matching pursuits. De fato, o algoritmo proposto em [24] acha decomposições da seguinte forma:

$$\mathbf{x} = \sum_{j=1}^{\infty} \alpha^j \sum_{l=1}^{L_j} \overline{\mathbf{g}}_{i_{j,l}} \tag{3.5}$$

Estas decomposições são mais genéricas do que aquelas obtidas com a equação (3.4), visto que o termo 2^{-j} foi substituído por α^j, para $0 < \alpha < 1$. Em [24] foram derivadas as condições de convergência de tal algoritmo. Estas condições impõem que $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{3}$, onde $\Theta(\overline{\mathcal{D}})$ é o maior ângulo entre um sinal qualquer $\mathbf{x} \in \mathbb{R}^N$ e o vetor do dicionário $\overline{\mathcal{D}}$ mais próximo. Contudo, mesmo para sinais de dimensão moderada (por exemplo,

$N \leq 64$), os dicionários que podem fornecer $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{3}$ seriam aqueles que têm cardinalidades muito grandes. Isto levaria a uma decomposição ineficiente na perspectiva taxa-distorção, visto que um maior número de bits é necessário para codificar os índices $i_{j,l}$.

Neste capítulo nós propomos o algoritmo MPGBP (Matching Pursuits Generalized Bit-Planes Algorithm) para achar tais decomposições, o qual é convergente sempre que temos $0 < \alpha < 1$ e $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{2}$. A vantagem deste algoritmo é que $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{2}$ é somente uma restrição moderada, sendo satisfeita sempre que $\mathcal{D}$ é completo [2]. Note que, neste algoritmo a decomposição é executada pela adição de um átomo $\overline{\mathbf{g}}_{i_{j,l}}$ de cada vez, até que uma taxa e/ou um critério de distorção sejam encontrados.

Para um dicionário $\mathcal{C} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$, $\|\mathbf{v}_i\| = 1$, $\forall i$, o algoritmo MPGBP está reproduzido a seguir (os sinais de entrada são normalizados tal que $\|\mathbf{x}\| \leq 1$):

Algoritmo MPGBP

❶ Começar com $\mathbf{w} = \mathbf{x}$, $m = 1$ e para um dado α.

❷ Repetir até um critério de parada seja encontrado

(a) Escolher $r_m \in \{1, \ldots, M\}$ tal que

$$\mathbf{w} \cdot \mathbf{v}_{r_m} = \max_{1 \leq j \leq M} \{\mathbf{w} \cdot \mathbf{v}_j\}.$$

(b) Escolher $k_m = \left\lceil \frac{\ln(\mathbf{w} \cdot \mathbf{v}_{r_m})}{\ln(\alpha)} \right\rceil$.
onde $\lceil \cdot \rceil$ é o menor inteiro maior ou igual ao argumento.

(c) Substituir $\mathbf{w}$ por $\mathbf{w} - \alpha^{k_m} \mathbf{v}_{r_m}$.

(d) Incrementar m.

❸ Parar.

Note que o algoritmo MPGBP aproxima $\mathbf{x}$ em P ($P \leq M$) passos como

$$\mathbf{x}^{(P)} = \sum_{m=1}^{P} \alpha^{k_m} \mathbf{v}_{r_m} \tag{3.6}$$

Agora, se definirmos L_j como sendo o número de valores de m tal que $k_m = j$, nós podemos renomear os índices r_m como $i_{j,l}$ para $l = 1, \ldots, L_j$. Conseqüentemente, se fazemos o dicionário $\mathcal{C}$ no algoritmo igual a $\overline{\mathcal{D}}$, então temos que a equação (3.6) é equivalente à equação (3.5) para $P \to \infty$.

Pode-se dizer que o algoritmo MPGBP é convergente se $\lim_{P \to \infty} \mathbf{x}^{(P)} = \mathbf{x}$. Neste contexto, esta convergência é garantida pelo teorema 1 (A prova deste teorema pode ser encontrada no apêndice H).

Teorema 1: *Seja* $\mathbf{x} \in \mathbb{R}^N$, $\|\mathbf{x}\| \leq 1$, *tal que este seja aproximado pelo algoritmo MPGBP usando um dicionário* $\mathcal{C}$ *com* P *passos, gerando* $\mathbf{x}^{(P)}$ *como na equação (3.6), e seja* $\Theta(\mathcal{C})$ *o maior ângulo entre um sinal qualquer* $\mathbf{y} \in \mathbb{R}^N$ *e o átomo mais próximo do dicionário* $\mathcal{C}$. *Nós temos que* $\|\mathbf{r}^{(P)}\| = \|\mathbf{x} - \mathbf{x}^{(P)}\| \leq \beta_c^P$, *onde* $\beta_c = \sqrt{1 - (2\alpha - \alpha^2)\cos^2(\Theta(\mathcal{C}))} < 1$ *para todo* $0 < \alpha < 1$ *e* $0 \leq \Theta(\mathcal{C}) < \frac{\pi}{2}$.

As seguintes características do algoritmo MPGBP merecem ser ressaltadas:

(i) O algoritmo MPGBP executa uma decomposição tal que, para cada átomo adicionado, a distorção na aproximação de $\mathbf{x}$ decresce por pelo menos β, onde $\beta < 1$. Assim, quando o número de passos $P \to \infty$, $\|\mathbf{r}^{(P)}\| \to 0$, isto é, o algoritmo MPGBP é convergente.

(ii) A representação de saída do algoritmo MPGBP é dada por somente uma seqüência de pares de índices (k_m, r_m), $m = 1, 2, \ldots, P$. Isto implica que não existe a necessidade da quantização de nenhum coeficiente como no algoritmo matching pursuits clássico (veja equação (3.1)). Em outras palavras, pode ser dizer que o algoritmo MPGBP desempenha a decomposição e a quantização ao mesmo tempo. Assim, ele representa uma solução elegante para o problema de quantização inerente ao algoritmo matching pursuits clássico.

(iii) A decomposição obtida pode ser organizada em planos de bits como na equação (3.5). Isto pode ser feito observando que, na equação (3.6), os índices r_m para os valores de m tal que $k_m = j$ correspondem aos átomos que formam o plano de bits j.

(iv) O número de átomos usados na decomposição pode ser especificado arbitrariamente e cada átomo corresponde a um par (k_m, r_m). Isto permite um controle preciso da taxa, visto que a decomposição pode ser interrompida quando a taxa desejada for alcançada.

(v) Note que o Teorema 1 estabelece que uma decomposição feita como nas equações (3.5) and (3.6) pode ser executada para todo $0 < \alpha < 1$ e para todo dicionário completo. Conseqüentemente, decomposições feitas como nas equações (3.5) and (3.6) são generalizações de qualquer decomposição mais uma codificação em bit-planes, isto é, qualquer decomposição em um dicionário seguida por uma codificação em bit-planes é um caso particular das equações (3.5) and (3.6).

3.2 Limitante Teórico para as Características Taxa-Distorção

Nesta seção nós derivamos um limitante teórico para as características taxa-distorção das decomposições propostas na seção 3.1. A partir do teorema 1, pode ser visto que o sinal residual $\mathbf{r}$ decresce sua magnitude por pelo menos β_c em cada passo, isto é,

$$\frac{\|\mathbf{r}^{(P)}\|}{\|\mathbf{r}^{(P-1)}\|} \leqslant \sqrt{1-(2\alpha-\alpha^2)\cos^2(\Theta(\mathcal{C}))} = \beta_c < 1 \tag{3.7}$$

para todo $0 < \alpha < 1$ e $0 \leq \Theta(\mathcal{C}) < \frac{\pi}{2}$.

Então, após P passos, e assumindo que $\|\mathbf{x}\| \leqslant 1$, a distorção média quadrática por coeficiente, D, é limitada por

$$D \leqslant \frac{\beta_c^{2P}}{N} \tag{3.8}$$

onde N é a dimensão do dicionário $\mathcal{C}$. Reorganizando a equação (3.8) a fim de obter o número máximo de passos P que garante uma distorção máxima por coeficiente igual a D, temos

$$P \leqslant \frac{1}{2}\frac{\log_2 ND}{\log_2 \beta_c} \tag{3.9}$$

Se o dicionário $\mathcal{C}$ tem cardinalidade q e gastam-se no máximo b_c bits para codificar cada índice k_m, a taxa por coeficiente é limitada por

$$R \leqslant \frac{1}{N}\left[P \log_2 q + P b_e\right] \tag{3.10}$$

É importante notar que o número de bits b_e usados para calcular a taxa na equação (3.10) tem uma forte dependência com o valor máximo do índice k_m. Por exemplo, usando um código natural, $b_e \geqslant \log_2 k_m$. No apêndice C, nós mostramos que o número mínimo de bits ($b_{e_{min}}$) que pode ser usado para codificar cada índice k_m, com uma distorção D, é dado por

$$b_{e_{min}} = \log_2 \left[\frac{\log_2 \left(\frac{ND\alpha^2 \cos^2(\Theta(\mathcal{C}))}{\beta_c^2} \right)}{2 \log_2 \alpha} \right] \tag{3.11}$$

A partir das equações (3.10) e (3.11) e usando o valor de P obtido na equação (3.9), a taxa R pode ser expressa como

$$R \leqslant \frac{1}{2N} \frac{\log_2 (ND)}{\log_2 \beta_c} \log_2 \left[\frac{q}{2 \log_2 \alpha} \log_2 \left(\frac{ND\alpha^2 \cos^2(\Theta(\mathcal{C}))}{\beta_c^2} \right) \right] \tag{3.12}$$

Analisando a equação (3.12) nota-se que, se a cardinalidade q for aumentada, a taxa tende a ser aumentada também. Como os parâmetros N, q e $\Theta(\mathcal{C})$ têm uma forte dependência com o dicionário usado, se mantivermos a dimensão N constante e aumentarmos a sua cardinalidade q, nós diminuiremos o valor de $\Theta(\mathcal{C})$. Conseqüentemente, se o aumento da taxa causado pelo aumento da cardinalidade for compensado por uma redução significativa de $\Theta(\mathcal{C})$, então pode-se ter uma redução da taxa total gasta (veja o apêndice C para maiores detalhes).

3.3 Análise das Características Taxa-Distorção do Algoritmo MPGBP Usando Diferentes Dicionários

3.3.1 Reticulados Regulares

Nesta seção nós analisamos o características taxa-distorção do algoritmo MPGBP usando parâmetros de reticulados regulares [25, 26]. A tabela 3.1 mostra os parâmetros usados de cada reticulado regular.

Tabela 3.1: Parâmetros dos reticulados regulares com melhor distribuição de hiper-esferas para as dimensões N=4, 8, 16 e 24.

Tipo de Reticulado ($\mathcal{L}_N$)	índice da camada	dimensão N($\mathcal{L}_N$)	cardinalidade q($\mathcal{L}_N$)	$\Theta(\mathcal{L}_k)$
D_4	1	4	24	45°
D_4	1+2	4	48	32°
E_8	1	8	240	45°
E_8	2	8	2160	45°
E_8	3	8	6720	35°
E_8	1+2	8	2400	32°
E_8	1+2+3	8	9120	29°
Λ_{16} ou L16	16	2	4320	55°
Λ_{24} ou L24	24	1	196560	45°

A figura 3.1 mostra a curva da taxa versus α quando a equação (3.12) é usada com os parâmetros dos reticulados regulares D_4 camada 1, D_4 camada 1+2, E_8 camada 1+2+3, Λ_{16} camada 2 e Λ_{24} camada 1 e para uma distorção igual a 10^{-4}. A partir desta figura podemos ver que existe um α ótimo para cada reticulado regular em uma dada distorção.

A figura 3.2 mostra as curvas R-D para diferentes camadas do reticulado regular E_8. Nós podemos ver a partir desta figura que, para reticulados com a mesma dimensão, sempre que $\Theta(\mathcal{L}_i)$ é reduzido o compromisso taxa$\times$distorção tende a melhorar. Note que foram usados os valores ótimos de α de cada reticulado. Isto também pode ser observado analisando as curvas R-D do dicionário D_4 na figura 3.3.

Na figura 3.3 pode-se observar que para alguns casos em que aumentamos a dimensão N do dicionário não obtivemos um melhor compromisso taxa-distorção (veja curvas Λ_{16} e E_8 camada 1+2+3). Pode-se observar também que o desempenho dos dicionários dependem da distorção (as curvas R-D de Λ_{16} camada 2 cruza a curva de D_4 camada 1+2 e E_8 camada 1+2+3). A partir de tais observações podemos concluir que o desempenho é altamente dependente do reticulado regular usado, e não tem relação clara com a dimensão N.

3.3.2 Dicionário Ortogonal

Na seção 3.3.1 nós usamos parâmetros de reticulados regulares para analisar as características taxa-distorção do algoritmo MPGBP. Apesar de serem considerados bons

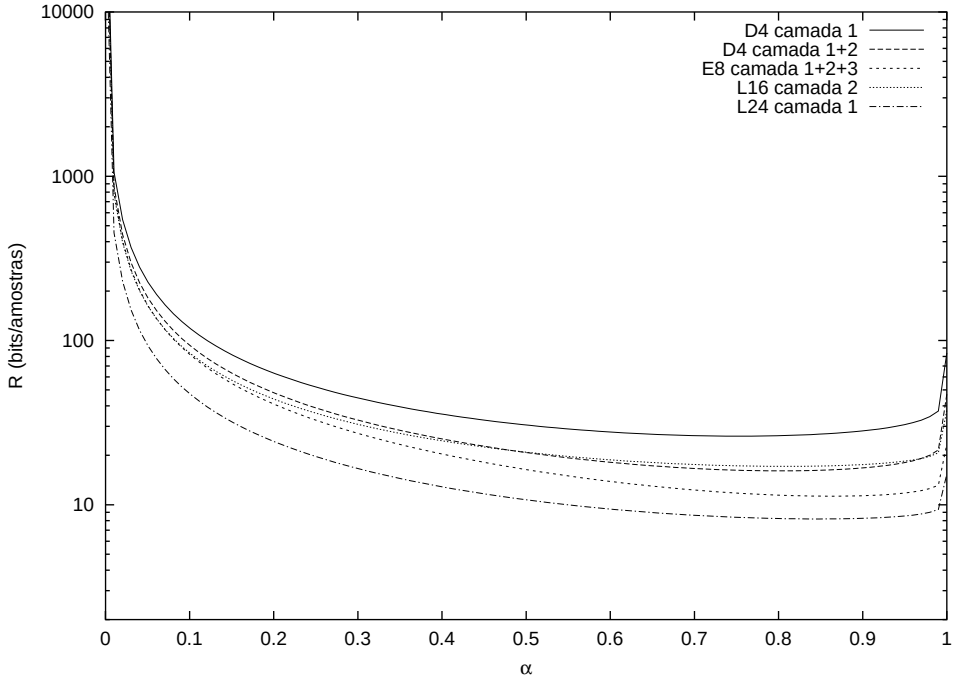


Figura 3.1: Variação da taxa com parâmetro α para diversos reticulados regulares.

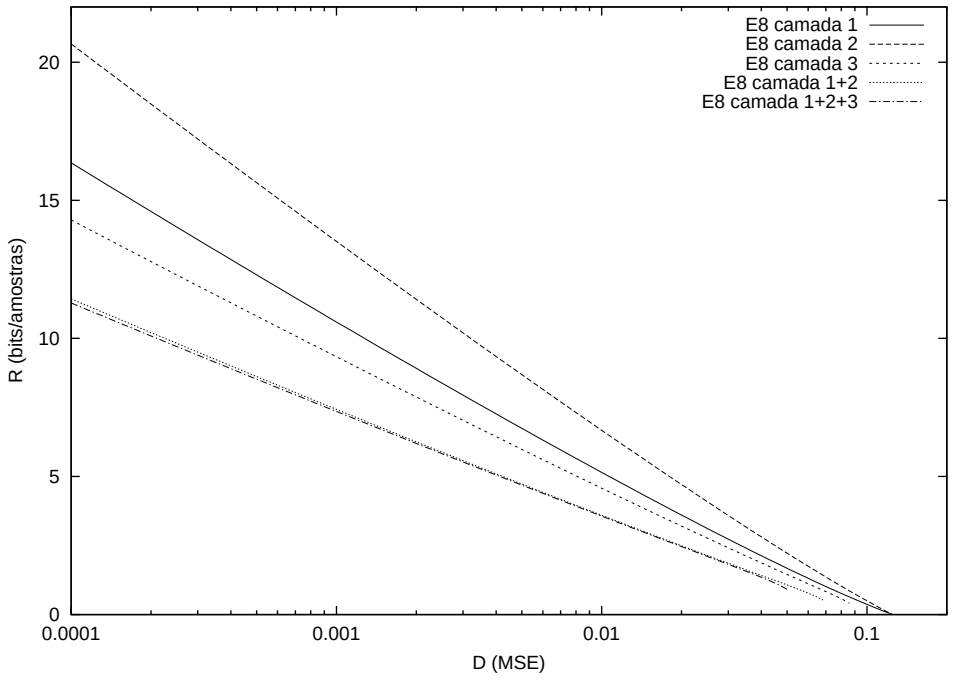


Figura 3.2: Curvas R-D para o reticulado regular E_8.

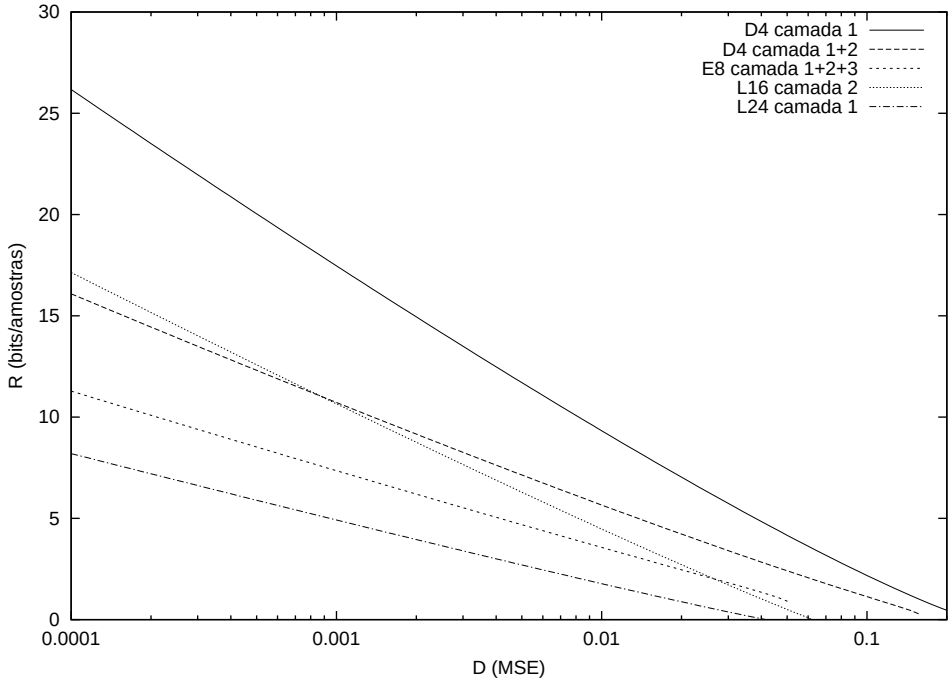


Figura 3.3: Curvas R-D para diferentes reticulados regulares.

dicionários, os reticulados regulares não nos fornecem uma visão completa sobre o comportamento R-D do algoritmo MPGBP. Isto é porque nós não sabemos as relações entre $\Theta(\mathcal{C})$, a dimensão $\mathrm{N}(\mathcal{C})$ e a cardinalidade $\mathrm{q}(\mathcal{C})$ quando estes reticulados são usados. Conseqüentemente não se pode fazer uma análise satisfatória de como a variação destes parâmetros influenciam no desempenho de um reticulado regular em particular.

A fim de ser possível analisar as variações de $\Theta(\mathcal{C})$, $\mathrm{N}(\mathcal{C})$ e $\mathrm{q}(\mathcal{C})$, primeiramente definimos um dicionário ortogonal simples $\mathcal{U}_{\text{ort}}$ como

$$\mathcal{U}_{\text{ort}} = \{1,0,\cdots,0\} \cup \{0,1,0\cdots,0\} \cup \{0,\cdots,1,0\} \cup \{0,\cdots,0,1\} \tag{3.13}$$

Então, usa-se este dicionário $\mathcal{U}_{\text{ort}}$ para formar o dicionário $\mathcal{U}$ que será utilizado em nossa análise. O dicionário $\mathcal{U}$ é formado por

$$\begin{aligned} \mathcal{U} &= \mathcal{U}_{\text{ort}} \cup -\mathcal{U}_{\text{ort}} && (3.14)\\ &= \{\pm 1,0,\cdots,0\} \cup \{0,\pm 1,0\cdots,0\} \cup \{0,\cdots,\pm 1,0\} \cup \{0,\cdots,0,\pm 1\} && (3.15) \end{aligned}$$

No caso da análise R-D do algoritmo MPGBP, a vantagem deste dicionário sobre os reticulados regulares, é que as relações entre a dimensão $\mathrm{N}(\mathcal{U})$, a cardinalidade

$q(\mathcal{U})$ e o ângulo $\Theta(\mathcal{U})$ podem ser facilmente obtidas. Para este dicionário temos uma cardinalidade $q(N) = 2N$ e um ângulo $\Theta(N) = \arccos\left(\frac{1}{\sqrt{N}}\right)$.

Usando estas relações pode-se analisar como a variação da dimensão N influencia no desempenho R-D do algoritmo MPGBP. Na figura 3.4 pode-se observar que, de forma similar ao obtido com os reticulados regulares, para cada dimensão, existe um α que leva a uma taxa mínima. Contudo, a figura 3.5 mostra que para cada α, existe uma dimensão que leva a uma taxa máxima, gerando assim um ponto de cela, isto é, temos um α ótimo e uma dimensão péssima (veja figura 3.6).

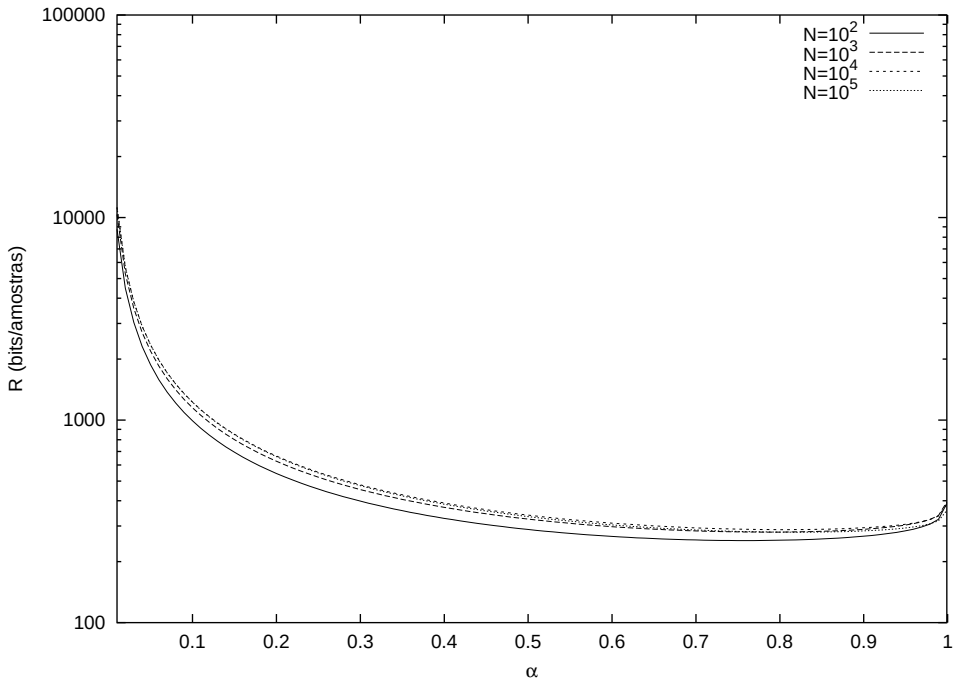


Figura 3.4: Taxa gasta versus α para o dicionário $\mathcal{U}$.

3.4 Conclusões

Neste capítulo foi proposto um novo algoritmo para fazer uma decomposição baseada no algoritmo de "matching pursuits". Ao invés de gerar como saída uma seqüência de pares de números, os quais são compostos dos índices dos átomos e de seus correspondentes coeficientes, como no "algoritmo matching"pursuits clássico, o algoritmo MPGBP gera apenas uma seqüência de índices. Estes índices podem ser arrumados como planos de bits generalizados. No algoritmo proposto neste capítulo não precisamos conhecer nada sobre o compromisso entre o número de átomos usados e a quantização dos coefi-

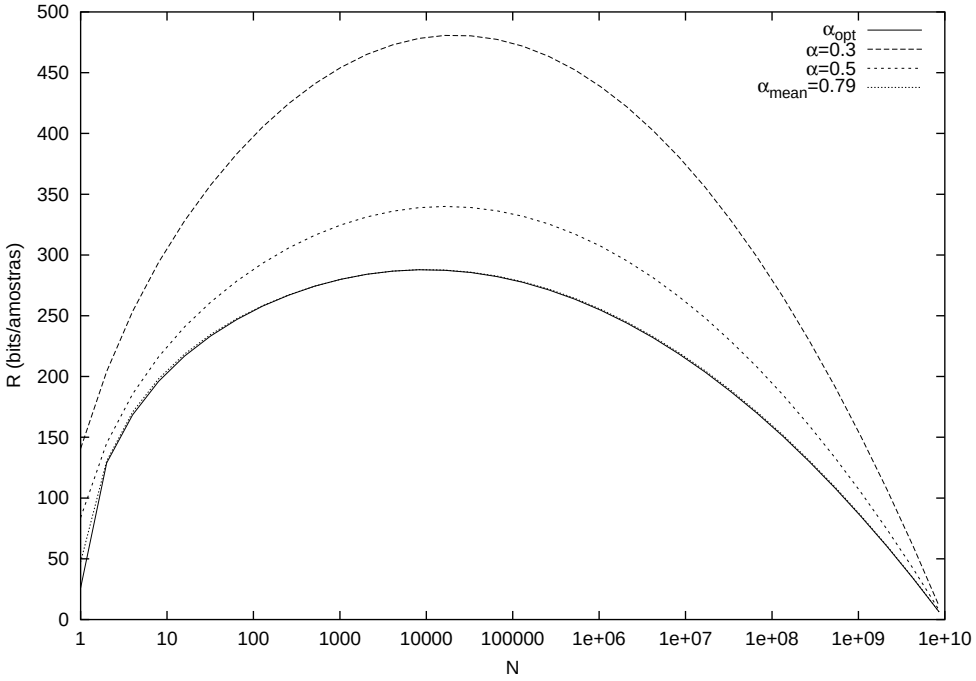


Figura 3.5: Taxa gasta versus a dimensão para o dicionário $\mathcal{U}$ usando diversos valores de α.

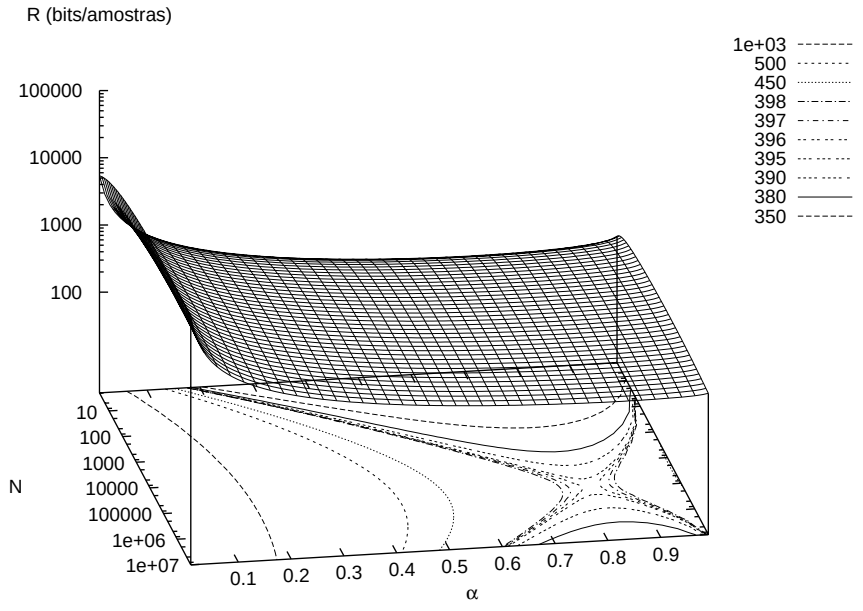


Figura 3.6: Superfície e curvas de nível para a taxa gasta usando o dicionário $\mathcal{U}$ para uma distorção $D = 10^{-1\frac{3}{4}}$.

cientes. Nós mostramos que o algoritmo proposto é uma generalização da decomposição em um dicionário seguida por uma quantização escalar. Além disso, foi apresentado um teorema que estabelece um limitante teórico para as características taxa-distorção de tais decomposições generalizadas.

Nós analisamos o desempenho taxa-distorção do algoritmo MPGBP usando parâmetros de reticulados regulares e de um dicionário ortogonal. Os resultados mostraram que a redução do valor de $\Theta(\mathcal{C})$ leva a um melhor compromisso taxa-distorção, e conseqüentemente, a uma redução na taxa de bits do processo de codificação, encorajando-nos a investigar dicionários com bons compromissos entre a cardinalidade $\mathrm{q}(\mathcal{C})$ e $\Theta(\mathcal{C})$.

Capítulo 4

Aplicações em Codificação de Vídeo

Neste capítulo nós utilizamos o algoritmo "Matching Pursuits"em Planos de Bits Generalizados (MPGBP) proposto no capítulo 3 para codificar seqüências de vídeo usando um dicionário super-completo formado de funções de Gabor tomando como base o codificador de vídeo proposto por Neff e Zakhor em [3, 4]. Os resultados experimentais mostram que o codificador de vídeo MPGBP apresenta um desempenho consistentemente superior ao do algoritmo "matching pursuits"clássico. Maiores detalhes sobre os codificadores de vídeo MP clássico e MPGBP, assim como a apresentação de todos os resultados obtidos encontram-se no apêndice D.

4.1 Um codificador de vídeo eficiente usando o algoritmo matching pursuits

Os métodos mais populares de codificação de vídeo são aqueles baseados em transformadas de blocos, onde a redundância dentro destes blocos é explorada. Neste contexto os algoritmos baseados na transformada Discreta Cosseno (DCT) são os que apresentam o melhor desempenho. Apesar de serem mais eficientes, os métodos baseados na DCT introduzem indesejáveis efeitos de blocagem e ringing, especialmente em baixas taxas de bits. Nos últimos anos, vários métodos não baseados em blocos, como aqueles que usam transformadas wavelet [27, 28, 29] e matching pursuits [2, 3, 5, 10], têm surgido para reduzir estes efeitos indesejáveis. Nesta seção nós descrevemos um codificador de vídeo eficiente, o qual foi baseado no trabalho proposto por Mallat em [2] e usa o algoritmo de matching pursuits. Este codificador de vídeo foi proposto por Neff e Zakhor

em [3, 4]. Este usa a mesma estrutura (compensação/estimação de movimento, fluxo de bits, tabelas de Huffman, etc) usada em [30], exceto pelo processo de codificação das imagens residuais. A principal diferença é que em [30] a imagem residual é dividida em blocos 8×8 onde a DCT é aplicada, enquanto que em [3] o algoritmo matching pursuits é empregado para decompor a imagem residual em uma soma ponderada de funções de Gabor. Ao invés de codificar os coeficientes da DCT, as posições no bloco 8×8 e o passo de quantização, o codificador de vídeo MP proposto em [3], codifica os índices das funções de Gabor e suas posições na imagem residual (os quais denominamos átomos) assim como os produtos internos obtidos pela projeção destas funções na imagem residual. É importante notar que o algoritmo matching pursuits é aplicado na imagem residual como um todo, e conseqüentemente não é baseado em blocos.

No apêndice B foi descrito o algoritmo matching pursuits e suas principais características para um dicionário unidimensional (1-D). Visto que nós usaremos este algoritmo para codificar imagens residuais no espaço discreto bi-dimensional (2-D), o dicionário usado tem que ser constituído de átomos 2-D. Além disso, como o algoritmo MP requer o cálculo do produto interno de todos os átomos deste dicionário com a imagem residual, é necessário que este algoritmo tenha um conjunto limitado de átomos, cada um dos quais com um tamanho finito, a fim de evitar um atraso demasiado no processo de codificação. Assim, com o objetivo de reduzir este tempo de procura, usou-se funções separáveis [31] para formar o dicionário. O dicionário usado consiste de uma coleção super-completa de funções de Gabor 2-D separáveis. Elas são versões escaladas, deslocadas e moduladas de uma janela Gaussiana. Definimos uma janela $g(t)$ como

$$g(t) = \sqrt[4]{2} e^{-\pi t^2} \tag{4.1}$$

A função de Gabor 1-D discreta de tamanho L pode então ser escrita como

$$g_{\vec{\alpha}}(i) = K_{\vec{\alpha}} \cdot g\left(\frac{i - \frac{L}{2} + 1}{s}\right) \cdot \cos\left(\frac{2\pi\xi(i - \frac{L}{2} + 1)}{L} + \phi\right) \tag{4.2}$$

onde $\vec{\alpha} = (s, \xi, \phi)$ compreende um escalamento positivo, uma modulação em freqüência e um deslocamento de fase, respectivamente, $i \in \{0, 1, \cdots, L-1\}$, e $K_{\vec{\alpha}}$ é uma constante de normalização, escolhida tal que $\|g_{\vec{\alpha}}\| = 1$. Além disso, nós definimos $\mathcal{B}$ como sendo o conjunto de todos os (s, ξ, ϕ) possíveis.

Usando uma combinação das funções de Gabor 1-D definidas na equação (4.2)

para construir um conjunto de funções de Gabor 2-D separáveis, pode-se formar o dicionário 2-D como a seguir

$$G_{\vec{\alpha},\vec{\beta}}(i,j) = g_{\vec{\alpha}}(i)g_{\vec{\beta}}(j) \tag{4.3}$$

onde $i,j \in \{0,1,\cdots,L-1\}$ e $\vec{\alpha},\vec{\beta} \in \mathcal{B}$.

A notação acima assume que todas as funções 1-D tem o mesmo tamanho L. Contudo, na prática, cada função 1-D pode ter o seu próprio tamanho, $L_{\vec{\alpha}}$. Isto foi feito a fim de aumentar a velocidade na procura.

Como o uso de um conjunto $\mathcal{B}$ consistindo de todos os valores possíveis de $\vec{\alpha}$ é computacionalmente proibitivo, nós temos que escolher um subconjunto limitado $\mathcal{B}'$. O dicionário reduzido $\mathcal{B}'$ foi construído da seguinte forma [3, 4]: um subconjunto maior de $\mathcal{B}$ contendo valores igualmente espaçados dos três parâmetros (s,ξ,ϕ) foi usado para definir um dicionário 2-D. Um conjunto de imagens residuais de uma seqüência de treinamento foi decomposta neste dicionário usando o algoritmo MP. Os vinte valores (s,ξ,ϕ) mais freqüentemente usados pelo algoritmo MP foram selecionados para formar o conjunto reduzido $\mathcal{B}'$.

Como já foi dito o codificador de vídeo MP proposto por Neff e Zakhor em [3] baseia-se no codificador de vídeo MPEG4 especificado em [30], com a diferença que, ao invés de usar a DCT para reduzir as redundâncias da imagem residual, se usa um dicionário super-completo formado por funções de Gabor. Inicialmente, se acha a função do dicionário que fornece a maior redução na energia da imagem residual. A escolha desta função é feita através o maior produto interno p_n obtido das projeções de todas as funções do dicionário na imagem residual. Contudo, como a procura completa sobre toda a imagem residual, usando estas funções, requer um grande esforço computacional, é feita uma pré-varredura da imagem residual a fim de encontrar o bloco de maior energia. A localização do bloco de maior energia é usada como uma estimativa inicial para a procura do produto interno. O produto interno é calculado em uma área 16×16 ao redor deste ponto. Após isto, o produto interno p_n é quantizado com um passo de quantização QP, e a função do dicionário escolhida (pesada pelo produto interno quantizado) é subtraída da imagem residual. Então, codifica-se o índice desta função e a sua posição na imagem residual, assim como o produto interno quantizado. Isto é feito usando-se tabelas de códigos de Huffman [32]. Este processo é repetido até que um número especificado de

átomos seja codificado.

4.2 Um codificador de vídeo usando o algoritmo MPGBP

Essencialmente o algoritmo MPGBP só substitui a decomposição e a estratégia de quantização empregadas em [3], usando exatamente o mesmo dicionário $\mathcal{D}$, assim como o mesmo procedimento de codificação de átomos. Os átomos são obtidos e codificados da mesma forma que com o codificador de vídeo MP proposto por Neff e Zakhor. Contudo, ao invés de codificar os valores quantizados do produto interno p_n (veja equação (2.1)), codifica-se os índices k_n (veja equação (3.6)) do plano de bits correspondente ao átomo codificado. É importante notar que o algoritmo MPGBP gera somente uma seqüência de índices, isto é, ao invés do produto interno do algoritmo MP, nós transmitimos somente o índice k_n correspondente ao expoente de α (veja equação 3.6). Além disso, como os índices k_n codificados no algoritmo MPGBP e os produtos internos quantizados p_n codificados no algoritmo MP tem características completamente diferentes, nós propomos um novo esquema para codificar tais índices usando um codificador aritmético adaptativo modificado (veja seção D.2.1 para mais detalhes).

4.3 Resultados experimentais

Nós codificamos as seqüências Coast-guard, Container, Hall-monitor, Foreman, Mother-and-daughter and Silent-voice (veja o apêndice K) com 300 quadros QCIF em 30 quadros/segundos, sub-amostradas no tempo por um fator de 4 (taxas menores que 20kbps) e 3 (todas as outras taxas) para gerar 7,5 quadros/segundo e 10 quadros/segundo, respectivamente. A codificação foi feita somente na componente de luminância para taxas de bits variando na faixa de 10-100kbps.

A tabela 4.1 compara o PSNR do codificador de vídeo matching pursuits original [3, 5] com o codificador de vídeo MPGBP para diversas taxas e seqüências de vídeo usando $\alpha = 0,56$. Nós podemos ver a partir desta tabela que o uso do esquema de planos de bits generalizados melhora consistentemente o desempenho do codificador matching pursuits original [3] para praticamente todas as taxas e seqüências testadas. Isto pode ser confirmado nas figuras 4.1 e 4.2 onde mostramos a variação do PSNR por quadro da

seqüência Mother na taxa de 64kbps e a variação do PSNR médio da seqüência com a taxa, respectivamente. Note que esta melhora tende a aumentar com o aumento da taxa de bits. Na verdade, nossos resultados são comparáveis com os melhores na literatura, os quais são obtidos usando sofisticadas estratégias adaptativas [10].

Tabela 4.1: Comparação, em termos de PSNR em dB, entre as duas implementações matching pursuits.

Seq + Taxa (kbps)	MPGBP	MP [3, 5]	MPGBP-MP
Container10	32,47	32,46	0,01
Mother10	33,27	33,31	-0,04
Hall10	33,30	33,26	0,04
Silent10	29,04	28,96	0,08
Coast10	24,32	24,27	0,05
Container24	34,64	34,43	0,21
Mother24	36,30	36,17	0,13
Hall24	36,52	36,12	0,40
Silent24	32,65	32,72	-0,07
Coast24	27,74	27,63	0,11
Container48	36,91	36,41	0,50
Mother48	39,11	38,45	0,66
Hall48	39,13	37,94	1,19
Silent48	36,25	35,97	0,28
Coast48	30,24	30,22	0,02
Container64	37,92	37,16	0,76
Mother64	40,40	39,31	1,09
Hall64	39,97	38,86	1,11
Silent64	37,82	37,28	0,54
Coast64	31,29	31,22	0,07
Mother96	42,23	40,98	1,25
Foreman96	35,55	35,35	0,20

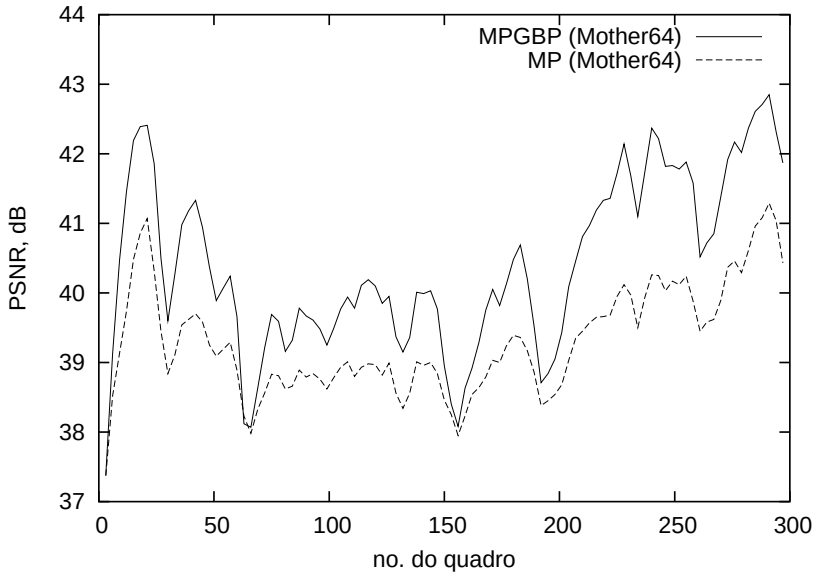


Figura 4.1: Variação do PSNR com os quadros da seqüência Mother na taxa de 64kbps.

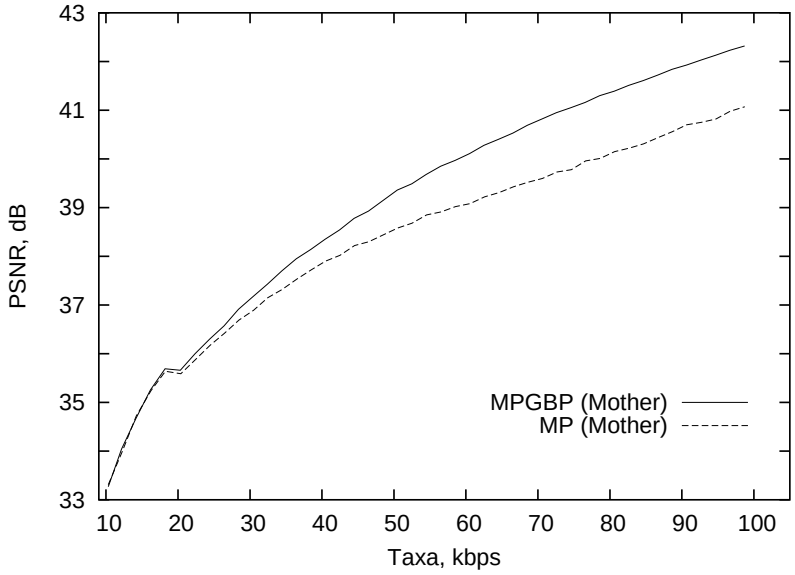


Figura 4.2: Variação do PSNR médio da seqüência com a taxa para a seqüência Mother.

4.4 Conclusões

Neste capítulo nós implementamos um codificador de vídeo usando o algoritmo MPGBP em substituição da decomposição e da quantização matching pursuits clássicas. Nosso codificador de vídeo foi usado com diferentes tipos de seqüências e em uma grande variedade de taxas de bits, produzindo resultados consistentes. Os resultados obtidos são muito promissores, indicando uma significante melhora sobre o algoritmo MP clássico [3, 4, 5].

Capítulo 5

Métodos de Controle de Taxa para o Codificador de Vídeo MPGBP

No capítulo 4 utilizou-se uma estratégia de controle de taxa na qual se tentou dividir a taxa de bits da seqüência igualmente entre todos os seus quadros. Nesta estratégia usa-se a taxa média de bits gasta por átomo codificado para fazer a estimação do número de átomos que devem ser empregados na decomposição de cada quadro da seqüência de vídeo. Contudo, este procedimento tende a gerar uma alocação de bits imprecisa para cada quadro da seqüência, e conseqüentemente, tem-se que a taxa alvo normalmente não é alcançada.

Neste capítulo nós propomos duas estratégias de controle de taxa para o codificador de vídeo MPGBP. Uma, na qual a taxa de bits da seqüência é dividida de forma precisa igualmente entre todos os seus quadros e outra que emprega a otimização Lagrangeana para resolver o problema da alocação ótima de bits para cada quadro da seqüência. Maiores detalhes sobre os métodos de controle de taxa propostos neste capítulo, assim como a apresentação de todos os resultados obtidos encontram-se no apêndice E.

5.1 Alocação de taxa fixa para o algoritmo MPGBP

A estratégia de controle de taxa usada no capítulo 4, a qual tenta dividir a taxa de bits da seqüência de vídeo igualmente entre todos os seus quadros, é baseada na estimação do número de átomos necessários para produzir a taxa desejada para cada quadro da seqüência. Contudo, como os quantizadores dos produtos internos dos átomos são des-

conhecidos a priori, este esquema de estimação tende a produzir um controle impreciso da taxa de bits. Nesta seção nós apresentamos um procedimento que permite um controle preciso da taxa de bits para o codificador de vídeo MPGBP. Este controle preciso da taxa baseia-se no fato de que o algoritmo MPGBP não realizar uma quantização explícita - ele simplesmente mapeia o sinal de entrada em uma seqüência de índices (veja apêndice C), fornecendo uma solução elegante para o problema de quantização inerente ao algoritmo matching pursuits clássico.

O procedimento usado para fazer uma alocação precisa de bits é a de se calcular a taxa gasta para cada novo átomo codificado até que a taxa de bits desejada para o quadro seja alcançada. Isto é possível porque o algoritmo MPGBP pode parar a decomposição do sinal sempre que a taxa de bits alvo é alcançada (veja seção C.1). Contudo, a posição dos átomos dentro de um mesmo macrobloco (16×16) é codificada diferencialmente de acordo com o padrão de varredura mostrado na figura D.6. Assim, para se ter um cálculo preciso da taxa de bits gasta nós devemos recalcular, para cada átomo adicionado, a taxa gasta para codificar a posição diferencial de todos os átomos do macrobloco.

Note que a alocação precisa de taxa obtida com o esquema proposto permite uma comparação justa entre os codificadores de vídeo, especialmente para baixas taxas de bits onde pequenas diferenças na taxa podem ter uma forte influência no desempenho do codificador de vídeo. Uma outra vantagem de se utilizar o esquema de alocação proposto, é que como este procedimento, pode ser usado um buffer menor para armazenar as variações de taxa. Isto pode ser visto na figura 5.1 onde é mostrada, para o método proposto nesta seção (Alocação Fixa) e para o método usado no capítulo 5 (Alocação Aproximada), a variação da taxa de bits gasta com cada quadro da seqüência Mother, esta codificada com 64kbps.

Na figura 5.2 nós podemos observar que o desempenho do codificador de vídeo MPGBP usando o esquema de alocação fixa proposto nesta seção é comparável ao obtido com o estado da arte dos codificadores de vídeo baseados no algoritmo matching pursuits [7]. Note que foram utilizados 60 quadros da seqüência de vídeo Foreman a 30 quadros/segundo.

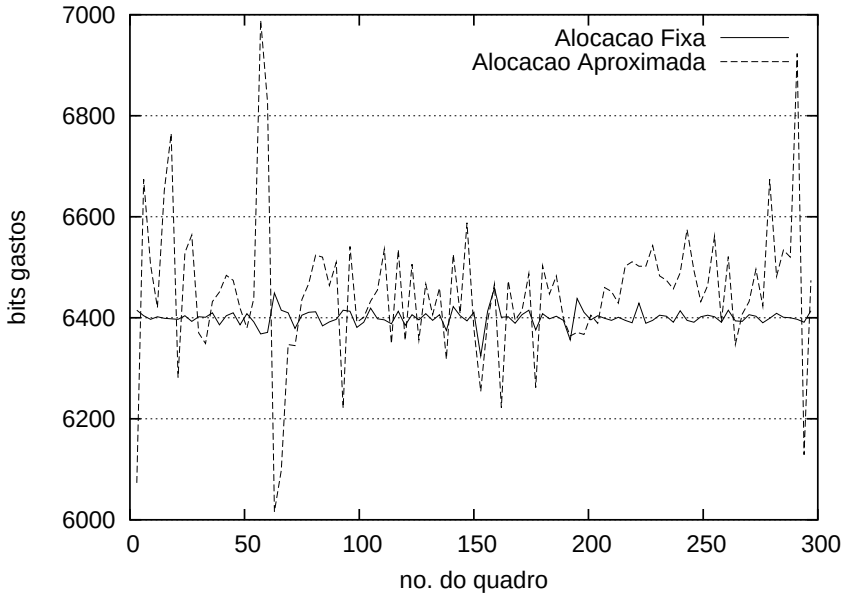


Figura 5.1: Variação dos bits gastos com cada quadro da seqüência Mother à taxa de 64kbps com 10 quadros/segundos.

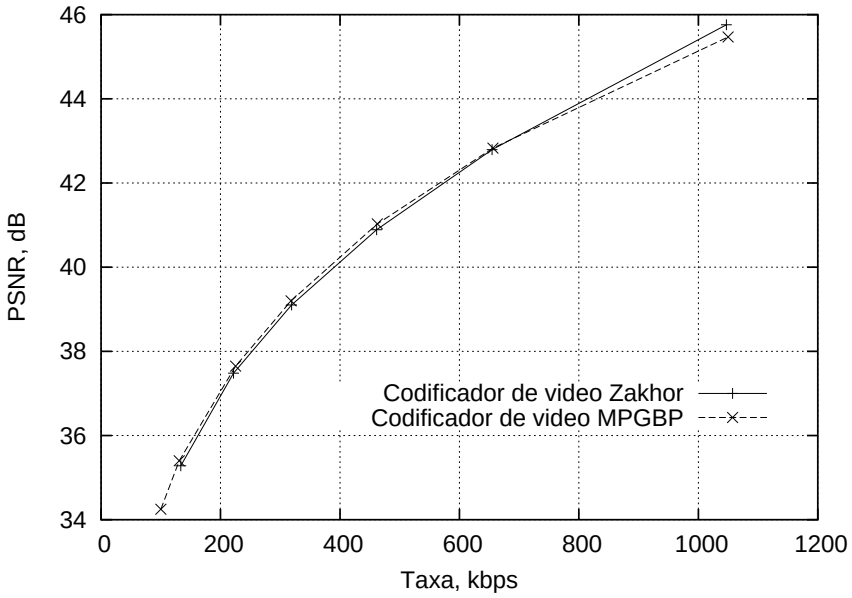


Figura 5.2: O desempenho dos codificadores MPGBP e Zakhor para a seqüência de vídeo Foreman em alta taxa de bits com 30 quadros/segundo.

5.2 Estratégia Lagrangeana de controle de taxa para o algoritmo MPGBP

Em aplicações de codificação de vídeo a solução trivial do problema de alocação da taxa é se dividir a taxa de bits total da seqüência de vídeo igualmente entre todos os seus

quadros. Apesar de sua simplicidade, esta solução normalmente não conduz aos melhores desempenhos taxa-distorção. Contudo, usando-se métodos de otimização Lagrangeana para achar a quantidade de bits que deve ser alocada para cada quadro da seqüência, pode-se alcançar valores de PSNR maiores, e conseqüentemente uma maior qualidade da imagem com o menor custo. Desta forma, nesta seção nós propomos um algoritmo de alocação de bits usando otimização de Lagrangeana.

Considerando que a função taxa-distorção (R-D) é convexa, o problema de alocação da taxa de bits pode ser estabelecido como:

$$\text{minimizar } J(\lambda) = D + \lambda R \tag{5.1}$$

onde a função de custo $J(\lambda)$ é minimizada para um dado valor do multiplicador de Lagrange λ. É importante notar que cada solução da equação (5.1) para um dado valor de λ corresponde a um compromisso taxa-distorção ótimo [33, 34] e para cada λ nós temos uma taxa diferente.

Visto que o algoritmo MPGBP garante que a distorção D sempre decresce (veja teorema 1 na seção C.1), após o i-ésimo átomo selecionado, nós teremos a variação de distorção ($\Delta D_i = D_i - D_{i-1}$) com valor negativo e a variação de taxa ($\Delta R_i = R_i - R_{i-1}$) com valor positivo para cada átomo codificado. Assim, nós podemos reescrever a equação(5.1) baseada nas contribuições incrementais ΔD e ΔR como

$$\Delta J(\lambda) = \Delta D + \lambda \Delta R \tag{5.2}$$

É importante notar que é vantajoso adicionar átomos para a decomposição enquanto o custo decresce ($\Delta J(\lambda) < 0$). Portanto, a solução para a equação (5.1) é obtida quando $\Delta J(\lambda) = 0$ [7]. Além disso, como no algoritmo MPGBP a decomposição e a quantização são desempenhadas juntas, sem qualquer suposição sobre os quantizadores, pode-se calcular precisamente os parâmetros ΔD_i e ΔR_i para cada átomo adicionado, e assim obter um critério de parada que leva a uma alocação de taxa ótima.

Usando o critério de parada definido acima ($\Delta J(\lambda) = 0$) é proposta uma estratégia de controle de taxa para o codificador de vídeo MPGBP. Neste esquema de alocação de bits a decomposição é interrompida quando a função custo J_i do i-ésimo átomo para de decrescer. É importante notar que esta estratégia de alocação de taxa assume que o quadro tenha uma curva R-D convexa. Contudo, para o codificador de vídeo MPGBP esta

curva não é estritamente convexa, visto que, devido à codificação diferencial das posições dos átomos, ΔR_i pode não ser sempre positivo. Para contornar este problema, é feita a alocação de bits usando o fecho convexo da curva R-D ao invés da própria curva R-D. No apêndice E é explicado com mais detalhes o problema da convexidade da curva R-D e é proposto um algoritmo de alocação de taxa baseada na otimização Lagrangeana. Este algoritmo está reproduzido a seguir.

Algoritmo de controle de taxa ótimo

Para um dado λ e usando um conjunto inicial de $N = N_o$ átomos

❶ Nós começamos com $J_{corrente} = J_o$, $J_{anterior} = \infty$, $k = k_{anterior} = 0$, e $p = 0$;

❷ Repetir enquanto $J_{corrente} < J_{anterior}$

a) Para i=k até N+k,

Calcular $\tilde{\lambda}_i = -\dfrac{D_i - D_k}{R_i - R_k}$;

b) Escolher $m \in \{1, \ldots, N\}$ tal que

$\tilde{\lambda}_m = \max\{\tilde{\lambda}_i\}$;

c) Calcular $J_m = D_m + \lambda R_m$;

d) Fazer $k_{anterior} = k$, $J_{anterior} = J_{corrente}$, $k = k + m$, $J_{corrente} = J_m$ e $\tilde{\lambda}_p = \tilde{\lambda}_m$;

e) Para i=0 até p,

Se $\tilde{\lambda}_{p-1} < \tilde{\lambda}_p$ fazer $N = N + 1$ e ir para o passo ❶;

f) Fazer $p = p + 1$;

❸ Codificar o quadro com $k_{anterior}$ átomos;

❹ Parar.

A figura 5.3 compara o desempenho do codificador de vídeo MPGBP que usa o esquema de alocação de taxa proposto nesta seção (Alocação Ótima) com o algoritmo de alocação fixa de taxa proposto na seção 5.1 (Alocação Fixa). Esta figura mostra a variação de PSNR com os quadros da seqüência Mother na taxa de 51,58 kbps obtida usando um $\lambda = 0,0006$. Pode-se notar que apesar de não melhorar significativamente o desempenho do codificador de vídeo, o algoritmo proposto produz uma menor variação entre os valores de PSNR dos quadros, e conseqüentemente uma melhor qualidade da seqüência de vídeo codificada.

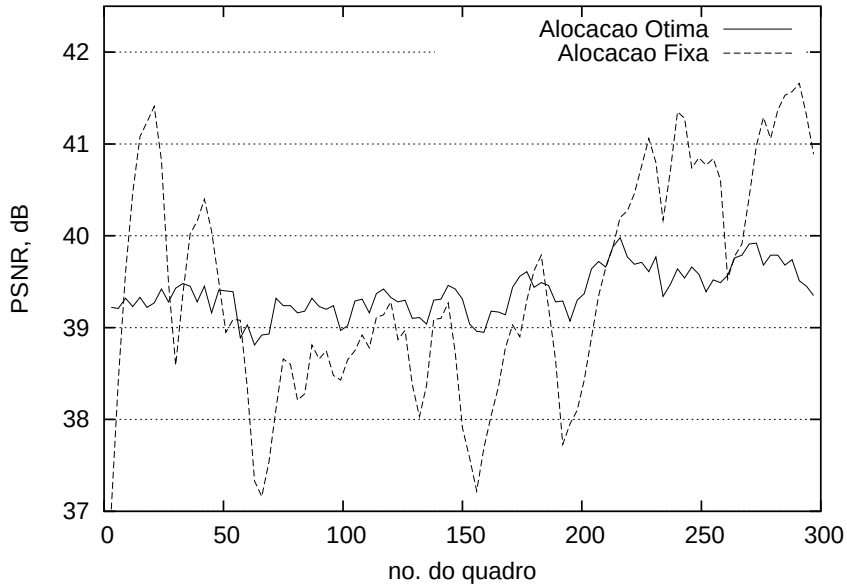


Figura 5.3: Variação do PSNR com os quadros da seqüência Mother na taxa de 51,58 kbps.

5.3 Conclusões

Neste capítulo nós propomos duas estratégias de controle de taxa para o codificador de vídeo MPGBP. Na primeira nós dividimos a taxa da seqüência igualmente entre todos os seus quadros usando o esquema preciso de alocação de bits proposto na seção 5.1. Este esquema tem a vantagem de alcançar precisamente a taxa de bits almejada para a seqüência assim como necessitar de um buffer menor para conter as variações da taxa. Os resultados mostram que este esquema pode ser usado para fornecer uma comparação mais justa entre os desempenhos dos codificadores de vídeo, visto que, com eles

obtemos precisamente a taxa desejada. O outro algoritmo utiliza otimização Lagrangeana para achar quantos bits devem ser alocados para cada quadro da seqüência a fim de que seja obtida a maior qualidade da imagem com o menor custo. Neste caso os resultados mostram que este esquema fornece somente uma pequena melhoria sobre o esquema de alocação fixa de bits. Apesar desta pequena melhoria no desempenho objetivo, este algoritmo produz uma variação de PSNR menor entre os quadros da seqüência de vídeo, e conseqüentemente uma melhor qualidade visual da seqüência de vídeo codificada.

Capítulo 6

Dicionários para o Algoritmo MPGBP

No capítulo 4 nós analisamos o desempenho do algoritmo MPGBP em aplicações de codificação de vídeo, comparando o seu desempenho com o do codificador matching pursuits clássico proposto por Neff e Zakhor em [3, 4]. Os resultados obtidos mostraram que o desempenho do codificador MPGBP é consistentemente superior ao do codificador matching pursuits clássico. Contudo, como estes resultados foram obtidos usando, para ambos os algoritmos, o dicionário proposto em [3], uma questão surge: existe um outro dicionário que produz um desempenho melhor?

A partir da análise teórica do desempenho taxa-distorção dos dicionários selecionados no apêndice C, nós conjecturamos que a redução do ângulo $\Theta(\mathcal{C})$ pode levar a um melhor compromisso taxa-distorção, desde que a cardinalidade $q(\mathcal{C})$ do dicionário $\mathcal{C}$ não aumente demasiadamente. Esta redução pode ser obtida usando uma melhor distribuição dos vetores do dicionário ou então adicionando vetores no dicionário original a fim de que estes preencham apropriadamente as regiões "vazias"do espaço de vetores. Note que o primeiro método preserva a cardinalidade e a dimensão do dicionário. Por outro lado, no segundo método, a cardinalidade é aumentada, e existe um compromisso entre esta cardinalidade e o ângulo $\Theta(\mathcal{C})$.

Neste capítulo nós investigamos o desempenho do codificador de vídeo MPGBP usando outros dicionários super-completos. É importante notar que maiores detalhes sobre a obtenção dos dicionários propostos neste capítulo, assim como a apresentação de todos os resultados obtidos encontram-se no apêndice F.

6.1 Dicionários derivados do treinamento com seqüências típicas de vídeo

Nesta seção nós propomos uma forma de gerar um dicionário redundante $\mathcal{C}$ que tenha um ângulo $\Theta(\mathcal{C})$ menor que o do dicionário original proposto por Neff e Zakhor [3, 4]. O dicionário proposto é obtido pela inclusão de novos vetores no dicionário original. A inclusão destes vetores tem por objetivo preencher os buracos que o dicionário original tem no espaço de vetores. Nós definimos como buracos as regiões do R^N onde o vetor do dicionário $\mathcal{C}$ mais próximo tem $\theta_n > \theta_c \leq \Theta(\mathcal{C})$. É importante notar que a inclusão destes vetores reduzem o ângulo $\Theta(\mathcal{C})$ do dicionário $\mathcal{C}$.

A figura 6.1 mostra uma ilustração deste processo para o R^2. Na figura 6.1a nós podemos ver um buraco localizado na região que tem ângulo máximo $\theta_1 > \theta_c$. Colocando-se um vetor neste buraco forma-se um novo dicionário com um vetor a mais e com menor θ médio. Um novo buraco com um ângulo $\theta_2 > \theta_c$ pode então ser achado em outra posição (veja figura 6.1b) e assim um outro vetor pode ser colocado neste novo buraco. Este processo pode ser repetido até que se tenha $\theta_n < \theta_c$ ou uma quantidade especificada de átomos seja alcançada. Note que a cada passo tem-se $\theta_n < \theta_{n-1}$. Em outras palavras, se um vetor $\mathbf{X}$ está em um buraco e $\mathbf{u_j}$ é o vetor do dicionário que está mais próximo, nós temos que

$$\theta(\mathbf{X}, \mathbf{u_j}) = \arccos \frac{< \mathbf{X}, \mathbf{u_j} >}{||\mathbf{X}|| \, ||\mathbf{u_j}||} = \arccos \frac{< \mathbf{X}, \mathbf{u_j} >}{||\mathbf{X}||} \geq \theta_c \tag{6.1}$$

O procedimento usado para selecionar novos vetores para serem adicionados ao dicionário original é descrito a seguir. Primeiramente, nós usamos o codificador de vídeo MPGBP (descrito no capítulo 4) para codificar diferentes seqüências de vídeo com diversos α e taxas de bits. Durante este processo de codificação (chamado de passo de treinamento [1]), as regiões 2-D da imagem residual, na posição e com o tamanho igual ao do respectivo átomo codificado, que geraram ângulos maiores que um determinado limiar (θ_c) foram consideradas como vetores candidatos a inclusão no dicionário. É importante notar que, se nós já colocamos um vetor em um buraco, é melhor colocar o próximo vetor em um outro buraco, obtendo-se assim uma melhor distribuição de vetores no espaço.

[1] As seqüências de vídeo usadas durante o passo de treinamento foram diferentes das seqüências usadas para avaliar o desempenho do codificador de vídeo MPGBP.

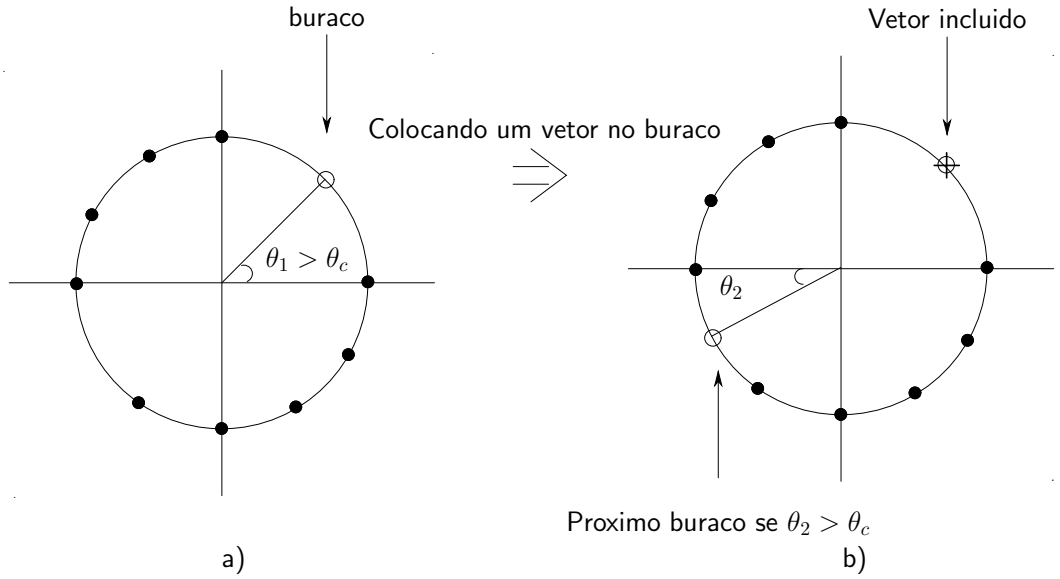


Figura 6.1: Ilustração da redução de $\Theta(\mathcal{C})$ pela colocação de um vetor em um buraco do dicionário $\mathcal{C}$ no R^2.

Para isso, nós calculamos os ângulos entre todos os vetores candidatos, dentre estes, os dois vetores que tem o menor ângulo são considerados como estando no mesmo buraco. Sendo assim, um destes dois vetores deve ser eliminado do conjunto de candidatos. Para isso, nós calculamos os ângulos entre estes dois vetores e todos os vetores do dicionário original. Dentre os dois vetores considerados no mesmo buraco aquele que apresentar o menor ângulo é eliminado do conjunto de candidatos. Este processo continua até que o conjunto de vetores candidatos seja reduzido ao número de vetores que desejamos adicionar no dicionário original.

Os resultados obtidos com o codificador de vídeo MPGBP usando os dicionários formados conforme o processo descrito acima, podem ser encontrados no apêndice F. Tais resultados mostram que a reduções de $\Theta(\mathcal{C})$ e $\overline{\Theta}$ obtidas não foram suficientes para melhorar o desempenho do codificador de vídeo MPGBP. Isto é, o procedimento de colocar vetores nos buracos não diminuiu suficientemente estes ângulos, o que consequentemente não compensou o aumento na taxa de bits obtido com o aumento da cardinalidade.

6.2 Dicionários derivados de expressões analíticas

Na seção F.1 do apêndice F mostra-se que reduções significativas nos ângulos $\Theta(\mathcal{C})$ e $\overline{\Theta}$ somente podem ser alcançadas com dicionários com uma boa distribuição de

seus vetores no espaço e que tenham estruturas não-separáveis. Nesta seção nós investigamos o desempenho do codificador de vídeo MPGBP usando dicionários não-separáveis derivados de expressões analíticas. Os dicionários propostos nesta seção foram gerados a partir de rotações, escalamentos e amostragem de funções 2-D contínuas obtidas de expressões analíticas.

Note que quando adicionamos ao dicionário original estruturas não-separáveis formadas de versões rodadas de suas estruturas separáveis, nós estamos tentando preencher os buracos do espaço de vetores, localizados próximos a detalhes diagonais. Estamos assim considerando que os detalhes diagonais estão próximos de um buraco do dicionário separável, visto que estes detalhes tendem a ter um pequeno produto interno com as funções separáveis. Além disso, adicionando estruturas rodadas (não-separáveis) em um dicionário separável, nós podemos capturar diversos contornos e arestas, os quais não são eficientemente representados usando somente estruturas separáveis. Isto pode ser visto na figura 6.2, onde nós temos uma ilustração de uma aresta curva representada usando estruturas separáveis ou não-separáveis. Note que, usando um dicionário separável (figura 6.2a) mais estruturas são necessárias para representar a aresta do que quando se usa um dicionário não-separável.

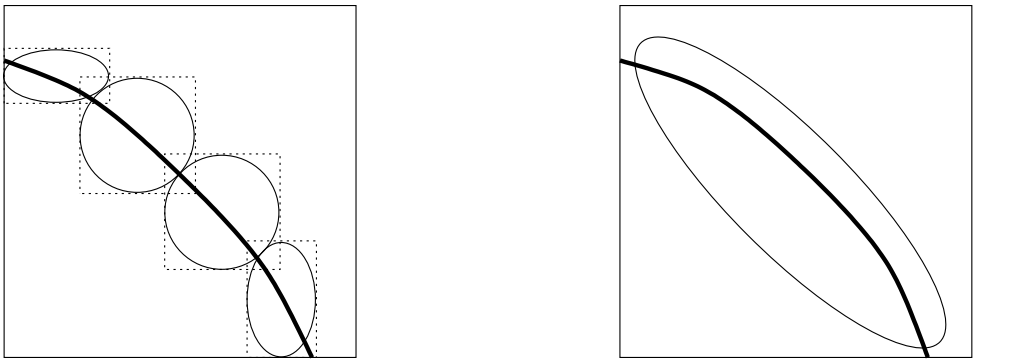


Figura 6.2: Representação de uma aresta curva usando estruturas separáveis ou não-separáveis.

Na tabela 6.1 nós comparamos o PSNR, e os valores de $\Theta(\mathcal{C})$ e $\overline{\Theta}$ obtidos usando o codificador de vídeo MPGBP com o dicionário separável proposto por Neff e Zakhor (referenciado como "N") e o dicionário não-separável gerado por rotações das estruturas separáveis do dicionário original (referenciado como "NR") e que cuja cardinalidade é 8090. A partir desta tabela nós podemos ver que o dicionário proposto produz, em geral,

uma redução razoável dos ângulos $\Theta(\mathcal{C})$ e $\overline{\Theta}$, compensando assim o aumento na taxa obtido quando a cardinalidade do dicionário é aumentada. É importante notar que quanto maior são as reduções de $\Theta(\mathcal{C})$ e $\overline{\Theta}$, melhor é o desempenho do codificador de vídeo usando o dicionário NR. Os resultados mostram que sempre que obtemos uma redução de $\overline{\Theta}$ maior que 1°, nós obtemos um melhora no desempenho do codificador de vídeo MPGBP. Na figura 6.3 nós mostramos que esta melhora tende a aumentar com o aumento da taxa. É importante notar que estes resultados confirmam a conjectura feita, baseada na análise teórica apresentada no apêndice C, que pela redução dos ângulos $\Theta(\mathcal{C})$ e $\overline{\Theta}$, mesmo com o aumento da cardinalidade, nós podemos obter um melhor compromisso taxa-distorção.

Tabela 6.1: Comparação, em termos de PSNR em dB, entre o codificador de vídeo MPGBP usando o dicionário original e o dicionário proposto formado por rotações.

Seq + Rate	N (Neff)			NR (Neff + Rotation)			Difference (NR - N)		
	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}$	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}$	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}$
Mother24	36,19	86,17º	82,29º	36,53	85,69º	81,03º	0,34	- 0,48º	- 1,26º
Weather24	31,75	87,52º	82,61º	32,25	86,62º	81,27º	0,50	- 0,90º	- 1,34º
Hall24	36,41	87,28º	81,62º	36,22	86,64º	80,68º	- 0,19	- 0,64º	- 0,94º
Silent24	32,59	87,08º	80,19º	32,74	86,29º	78,90º	0,15	- 0,79º	- 1,29º
Coast24	27,71	86,79º	81,45º	27,65	85,87º	80,51º	- 0,06	- 0,92º	- 0,94º
Mother64	40,38	87,19º	83,51º	40,72	86,51º	82,35º	0,34	- 0,68º	- 1,16º
Weather64	37,56	88,26º	83,60º	38,39	87,20º	82,34º	0,83	- 1,06º	- 1,26º
Hall64	39,87	87,69º	83,01º	39,85	86,97º	82,13º	- 0,02	- 0,72º	- 0,88º
Silent64	37,72	87,81º	82,25º	37,96	86,92º	81,14º	0,24	- 0,89º	- 1,11º
Foreman64	33,45	86,80º	80,77º	33,93	85,94º	79,25º	0,48	- 0,86º	- 1,52º
Coast64	31,33	87,32º	82,92º	31,36	86,29º	82,01º	0,03	- 1,03º	- 0,91º
Foreman96	35,52	87,07º	81,60º	36,01	86,34º	80,30º	0,49	- 0,73º	- 1,30º
Weather96	40,38	88,48º	84,00º	41,41	87,30º	82,72º	1,03	-1,18º	- 1,28º

6.3 Conclusões

Neste capítulo nós investigamos a desempenho do codificador de vídeo MPGBP usando diferentes dicionários super-completos. Nós usamos duas estratégias para escolher as estruturas dos dicionários, uma adicionando vetores no dicionário original pro-

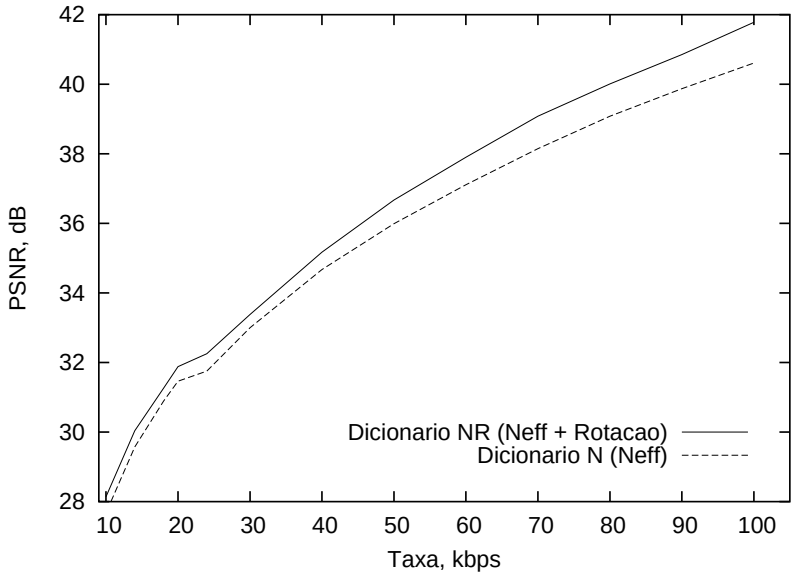


Figura 6.3: Variação do PSNR com a taxa para a seqüência Weather.

posto por Neff e Zakhor e um outra usando uma nova distribuição de vetores do dicionário. Os resultados obtidos pelo dicionário formado de versões rodadas das estruturas separáveis do dicionário original confirmam a conjectura, feita baseada na análise teórica no apêndice C, que reduzindo os ângulos $\Theta(\mathcal{C})$ e $\overline{\Theta}$, apesar do aumento na cardinalidade, pode ser obtido um melhor compromisso taxa-distorção.

Capítulo 7

Conclusão

Esta Tese introduziu uma classe de métodos de compressão de vídeo, baseados na decomposição em planos de bits generalizados, usados tomando como base o algoritmo "matching pursuits".

Inicialmente é apresentado, no capítulo 2, o algoritmo matching pursuits e suas principais características.

A seguir, no capítulo 3, a teoria de decomposição de sinais em planos de bits generalizados foi apresentada e um novo algoritmo (algoritmo MPGBP) para fazer decomposições usando o algoritmo de "matching pursuits"baseadas em tal teoria foi proposto. O algoritmo proposto tem a vantagem de não precisar especificar qualquer relação entre o número de átomos usados e a quantização dos seus coeficientes, dando uma solução elegante ao problema de quantização inerente ao algoritmo "matching pursuits"tradicional. Neste capítulo nós também analisamos o limitante teórico para as características taxa-distorção do algoritmo proposto usando parâmetros de reticulados regulares, assim como de um dicionário formado pela união de um dicionário ortogonal e seus simétricos em relação à origem. Os resultados mostraram que a redução do valor de $\Theta(\mathcal{C})$ de tais dicionários pode levar a um melhor compromisso taxa$\times$distorção, e conseqüentemente, uma redução da taxa de bits gasta no processo de codificação.

A fim de avaliar o desempenho do algoritmo MPGBP em codificação de vídeo, no capítulo 4, foi implementado um codificador de vídeo usando como base o algoritmo proposto por Neff e Zakhor em [3, 4]. O codificador de vídeo MPGBP foi testado com diferentes tipos seqüências e para uma grande variedade de taxas de bits, produzindo resultados consistentes. Estes resultados indicam uma significante melhora sobre o al-

goritmo "matching pursuits"clássico usado em codificação de vídeo [3, 4]. O algoritmo MPGBP também fornece um desempenho comparável com os obtidos usando algoritmos mais sofisticados, como por exemplo, o apresentado em [10, 7].

No capítulo 5, foram propostas duas estratégias de controle de taxa para melhorar o desempenho do codificador de vídeo MPGBP. Uma usou um procedimento de controle de taxa que divide de forma precisa a taxa de bits da seqüência igualmente entre todos os seus quadros; outra usou um esquema de controle de taxa que emprega otimização Lagrangiana a fim de resolver o problema de alocação ótima. O esquema fixo pode ser usado para fornecer uma comparação justa entre os desempenho dos codificadores de vídeo, visto que com eles podemos obter precisamente a taxa desejada. Seu desempenho é comparável com aqueles obtidos usando o estado-da-arte em codificadores de vídeo baseados no algoritmo de "matching pursuits" [7]. O esquema baseado em otimização Lagrangeana apresenta somente uma pequena melhoria no desempenho objetivo obtido sobre o de taxa fixa, apesar da menor variação de PSNR obtida os quadros da seqüência de vídeo, e conseqüentemente uma melhor qualidade subjetiva para a seqüência de vídeo codificada.

No capítulo 6 nós investigamos o desempenho do codificador de vídeo MPGBP usando diferentes dicionários super-completos a fim de se avaliar os efeitos destes dicionários no desempenho de tal codificador. Os resultados mostram que usando um dicionário não-separável formado de versões rodadas das estruturas separáveis do dicionário usado original por Neff e Zakhor obtemos uma significativa melhora no desempenho do codificador de vídeo. Isto confirma a conjectura, feita baseada na análise teórica do apêndice C, que reduzindo os ângulos $\Theta(\mathcal{C})$ e $\overline{\Theta}$ do dicionário $\mathcal{C}$, apesar do aumento da cardinalidade, pode ser obtido um melhor compromisso taxa$\times$distorção.

Como sugestão para trabalhos futuros, deve ser investigado o desempenho do algoritmo MPGBP codificando imagens estáticas. A fim de se fazer isto, devem ser investigados algoritmos baseados em "matching pursuits"que sejam diferentes ao de Neff e Zakhor [35, 36, 37, 38, 39, 40, 41, 42], que são apropriados apenas para uso em codificação de vídeo. Neste contexto, devem ser também investigados dicionários que fornecem uma melhor representação para as imagens estáticas [3, 4].

Como uma outra sugestão para trabalhos futuros, o algoritmo MPGBP poderia ser aplicado em codificação de vídeo usando dicionários mais sofisticados, como os baseados

em ridgelets [43, 44, 45, 46, 47, 48], curvelets [49, 50, 51] e contourlets [52, 53].

Resumindo, pode-se concluir que o algoritmo MPGBP fornece um esquema eficiente para codificação de vídeo, e seu desempenho encoraja a continuação desta linha de pesquisas.

Apêndice A

Introduction

Systems that use visual information, like multimedia systems, digital television, video phone, and Internet image and video transmission, just to mention some, have been increasingly used nowadays. As a consequence, in the last years, the amount of visual information that needs to be processed and stored has had a significant increase. In order to efficiently use such kind of information, image and video compression is mandatory. This Thesis deals with a class of video compression methods.

The Matching Pursuits Algorithm proposed by Mallat and Zhang in [2] has been employed successfully in video compression [3, 4, 5, 6, 7, 8, 9]. In them, the motion compensated frame difference is decomposed over an overcomplete dictionary, generating a sequence of pairs specifying the atoms used and their corresponding coefficients. A particular rate $\times$ distortion (R-D) trade-off is achieved by specifying both the number of atoms and the quantizers for their coefficients. Several strategies have been proposed in order to achieve good trade-offs [10, 11, 7].

Here, we propose a novel method that is equivalent to performing matching pursuits quantization based on the notion of decomposition in generalized bit-planes. The structure of such decompositions is such that once the decomposition is carried out, quantization is already done, and there is no need to set up any quantization parameters. It does so by generating a decomposition that is readily organized in bit-planes. Such decompositions, unlike the traditional ones, generate just a sequence of indexes. In other words, it can be said that the matching pursuits algorithm based on generalized bit-planes decomposition performs both decomposition and quantization at the same time. Thus, it presents an elegant solution for the trade-off between quantization of coefficients and the

number of passes in the matching pursuits algorithm. We show that they can be regarded as a generalization of any decomposition on a dictionary followed by coefficient representation in finite precision using the binary number format. In addition, we state a theorem that sets bounds for their R-D performance.

Since the matching pursuits algorithm that uses generalized bit-planes decompositions maps the frame difference into a sequence of indexes, we can obtain a precise control of the rate spent for each atom added, and consequently, the decomposition can be stopped whenever the bit-budget is exhausted. Based on this fact, we propose two rate-control strategies [12, 13, 14, 15, 16, 17, 18, 19] in order to improve the performance of the video encoder that uses such algorithm. One uses a rate-control procedure that divides precisely the bit-rate of the sequence equally among all its frames. The other uses a rate-control scheme that employs Lagrangean optimization in order to solve the optimal bit allocation problem [7]. In the former we have the advantage of precisely allocating the desired bit-rate for the frame, and consequently obtaining the desired rate budget. In the latter we can allocate the bit-rate in order to obtain the optimum R-D trade-off.

Also, to improve the performance of the proposed video encoder we investigate different overcomplete dictionaries [20, 6, 21, 22, 23], including non-separable ones, formed by rotations of separable structures. Confirming the theoretical analysis, the use of these dictionaries lead to improvements in coding performance.

This Thesis is organized as follows: the first part is written in Portuguese and contains the introduction, the main text in chapters 2 through chapter 6 and a conclusion. The content of these chapters is reproduced, with added detail, in 11 appendices.

Appendix B, presents the matching pursuits algorithm and its main characteristics.

In appendix C, we propose a novel algorithm (The Matching Pursuits Generalized Bit-Planes Algorithm – MPGBP Algorithm), that performs a kind of adaptive decompositions in which a signal is decomposed into generalized bit-planes. Also, we analyze the theoretical Rate Distortion bounds of the proposed algorithm.

In appendix D, we use the MPGBP Algorithm to encode video sequences using an overcomplete dictionary formed by Gabor functions, showing experimental results.

In appendix E, we propose two rate-control strategies for improving the performance of the MPGBP video encoder. One uses a fixed bit-rate allocation and the other uses Lagrangean optimization. Experimental results are also presented.

In appendix F, we investigate the performance of the MPGBP video encoder using different separable and non-separable overcomplete dictionaries, along with experimental results.

Finally, appendix G contains the conclusions along with future research directions.

Apêndice B

Signal Decompositions

In this appendix we show a review of the signal representation and also we describe the main characteristics of the Matching Pursuits Algorithm proposed by Mallat and Zhang in [2]. This algorithm decomposes signals iteratively into a linear expansion of waveforms that are selected from a redundant dictionary of functions. In each iteration, an exhaustive search is used to choose the best dictionary structure element to represent the signal. The projection of this element is subtracted from the signal, and the remaining signal energy is passed to the next iteration as a residual. It is important to note that these waveforms are selected in order to obtain the best match with the signal structures.

B.1 Signal Representations

A signal $\mathbf{x}$ of dimension N can be represented using a complete dictionary [2] $\mathcal{B} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}$, $\|\mathbf{f}_i\| = 1, \forall i$ by a linear combination, finite or infinite, of signals $\mathbf{f}_i$ (called atoms or structures) by the following equation [54, 55]

$$\mathbf{x} = \sum_{i=1}^{M} a_i \mathbf{f}_i \tag{B.1}$$

where the coefficients a_i form a binary vector $\mathbf{a}$.

Note that the dictionary $\mathcal{B}$ may contain more elements then necessary to span the space, that is, M may be larger than the dimension N of the signal space. The key point is how to obtain the coefficients a_i and the atoms or structures $\mathbf{f}_i$. In general, for the decomposition represented by equation (B.1) be capable of identifying different characteristics

of the signal, the dictionary must contain a large number of structures $\mathbf{f}_i$. In fact, the number of structures in the dictionary should be as large as the number of characteristics that we want to identify. Dictionaries that have more structures than the dimension of the signal are referred to as overcomplete dictionaries, and the decompositions of a signal in that dictionaries are called redundant.

B.1.1 Basis

In equation (B.1), if the vectors $\mathbf{f}_i$ form a basis of $\mathcal{B}$, the binary vector $\mathbf{a}$ of dimension N will be the unique solution for the decomposition $\mathbf{x}$. Then, we have the binary vector $\mathbf{a}$ representing the signal $\mathbf{x}$ using the basis $\mathcal{B}$.

A basis is orthogonal when the inner product between any two vectors $\mathbf{f}_i$ of the basis is zero, $< \mathbf{f}_l, \mathbf{f}_k >= 0 \; \forall k \neq l$. This implies that an orthogonal basis in the Euclidean space is formed by vectors with angles of $90°$ between them. When an orthogonal basis has the inner product between its vectors satisfying $< \mathbf{f}_l, \mathbf{f}_k >= \delta(l-k) \; \forall (k,l)$ this basis is known as orthonormal.

A basis is called biorthogonal when we have the inner product between its vectors and the ones of its dual basis given by:

$$< \mathbf{f}_l, \tilde{\mathbf{f}}_k >= \delta(k-l) \tag{B.2}$$

where $\mathbf{f}_l$ are the vectors of the basis Γ and $\tilde{\mathbf{f}}_k$ are the ones of the dual basis $\tilde{\Gamma}$. It is important to note that the basis Γ and its dual $\tilde{\Gamma}$ are both non-orthogonal.

Algorithms that perform decompositions as shown in equation (B.1), using complete dictionaries, as the Discrete Cosine Transforms (DCT) [56], Wavelet Transforms [57, 58, 59, 60, 61, 62, 63, 64, 65, 55], Fourier Transform [66, 67, 68], among others, are widely used nowadays. However, these decompositions are not flexible enough to provide a good representation for signals that have components whose localizations in time and frequency vary widely. For example, the Discrete Wavelet Transform (DWT) has high frequency basis functions well located in time and poorly located in frequency, and low frequency basis functions well located in frequency and poorly located in time. All the DCT basis functions have fixed time and frequency resolutions [55]. One way to deal with this problem is to employ an overcomplete dictionary, in which the number of elements is greater than dimension of the signal space. In this way one can have elements

representing several compromises between time and frequency resolutions, providing a great deal of flexibility for signal decompositions.

B.1.2 Frames

A frame [55, 69, 70, 54, 71] is a family of vectors $\{\phi_n\}$ that characterizes any signal $\mathbf{x}$ from its inner products $\{< \mathbf{x}, \phi_n >\}$. In a frame there is in general more structures than in a basis, what tends to produce better representations of a signal.

The dictionary $\mathcal{C} = \{\phi_1, \cdots, \phi_M\}$ in a Hilbert space $\mathcal{H}$ represents a frame if we can find an $A > 0$ and a $B < \infty$ such that for any signal $\mathbf{x} \in \mathcal{H}$ we have

$$A||\mathbf{x}||^2 \leq \sum_n | < \mathbf{x}, \phi_n > |^2 \leq B||\mathbf{x}||^2 \tag{B.3}$$

where $< \mathbf{x}, \phi_n >$ represents de inner product between the signal $\mathbf{x}$ and a structure of the dictionary $\mathcal{C}$. Note that M is the cardinality of the dictionary $\mathcal{C}$ and N is the dimension of signal space, and we can have $M \geq N$. The frame becomes a basis for $M = N$.

The constants A and B are the *frame bounds* and when we have $A = B$ the frame is referred to as *tight*. For a *tight frame* we have:

$$||\mathbf{x}||^2 = A^{-1} \sum_n || < \phi_n, \mathbf{x} > ||^2 \text{ and} \tag{B.4}$$

$$\mathbf{x} = A^{-1} \sum_n < \phi_n, \mathbf{x} > \phi_n \tag{B.5}$$

If $\phi_n = 1$ for all n, then the constant A gives the redundancy ratio of the frame. For example, if $A = 2$ there are twice as many vectors as needed to span the space $\mathcal{H}$. The expansion in equation (B.5) is not unique because the vectors in the frame are linearity dependent. In fact, the linear dependence means that $\sum n\alpha_n\phi_n = 0$ has a nontrivial solution (some $\alpha_n \neq 0$) so $\mathbf{x} = \sum_n (A^{-1} < \phi_n, \mathbf{x} > +\alpha_n)\phi_n$ is also a valid expansion.

B.2 Adaptive Representation of Signals

Adaptive representations [2, 54, 55, 72, 73] combine the concepts redundancy and compactness [73] by choosing from a large dictionary which atoms are used to represent a signal. This choice is signal dependent, hence the term adaptive. These representations

have been used in a wide range of applications, from coding and compression to signal enhancement, de-noising, pattern recognition and understanding of signals [74, 75, 76, 77].

Note that, one of the best things of adaptive decomposition is that a signal no longer needs to be represented in terms of a single orthonormal basis set [78]. This can be achieved by decomposing a signal onto a set of waveforms. This set of waveforms is often referred to as a dictionary.

A dictionary is a collection of parameterized waveforms $\mathcal{D} = (\phi_\gamma : \gamma \in \Gamma)$. These waveforms ϕ_γ are discrete-time signals of length n called atoms [79]. The parameter γ is dependent upon the dictionary type, e.g. if using a frequency dictionary, then γ is the indexing frequency. Usually dictionaries can either be complete or overcomplete, in which case they contain exactly N atoms, or more than N atoms, respectively. There are also continuous dictionaries, which contain an infinity of atoms, and under-complete dictionaries for special purposes, which contain fewer than N atoms [79].

Using a overcomplete dictionary the decomposition represented by equation (B.1) becomes non-unique. This is so because some elements in the overcomplete dictionary have representations in terms of other elements. Non-uniqueness gives us the possibility of adaptation, i.e., of choosing among many representations one that is most suited to our purposes. We are motivated by the aim of achieving simultaneously the following goals:

- *Sparsity*. We should obtaining the sparsest possible representation of the object – the one with the fewest significant coefficients.

- *Super-resolution*. We should obtain a resolution of sparse objects that is much higher-resolution that that possible with traditional non-adaptive approaches.

An important constrain, which is perhaps in conflict with both the goals is the *Speed*.

Several methods have been proposed for obtaining signal representations in over-complete dictionaries. These range from general approaches, like the Method of Frames [80], Basis Pursuits [79], and Matching Pursuits [2], to clever schemes derived for specialized dictionaries, like the method of Best Orthogonal Basis [81, 82]. These methods are briefly explained below.

B.2.1 Method of Frames (MOF)

The Method of Frames was one of the first decomposition techniques to be suggested by Daubechies [80] in representing the signal using an overdetermined dictionary. Generally, Method of frames does not carry out sparse decomposition. Given a discrete dictionary of **p** waveforms (each of length n) those are collected as columns of an $n \times p$ matrix, Φ, the decomposition problem is:

$$\Phi \mathbf{a} = \mathbf{x} \tag{B.6}$$

where **a** is vector of coefficients.

The Method of Frames picks out among all solutions [69] of equation B.6, one whose coefficients have the minimum L^2 norm:

$$\min \|\mathbf{a}\|_2 \text{ subject to } \Phi \mathbf{a} = \mathbf{x} \tag{B.7}$$

The solution of this problem is unique and it is obtained by using a conjugate gradient method [83] to solve the equations (B.6) and (B.7). This technique has two problems [79, 84]. Firstly, the Method of Frames is not sparsity preserving. It has a tendency to use all the basis functions non-orthogonal to the signal yielding a very non-sparse representation. If the signal can be represented by a minimal set of the dictionary elements, then the coefficients found by Method of Frames are likely to be more numerous than this minimal set. Secondly, the Method of Frames is resolution-limited. It does not allow object to be reconstructed with features sharper than those allowed by the analysis and synthesis operators.

B.2.2 Best Orthogonal Basis (BOB)

Best Orthogonal Basis, proposed by Coifman and Wickerhauser [81, 82] is a technique, which seeks to find a best basis out of an orthogonal set of vectors relative to a given signal. Thus, overall information cost is optimized. This method uses a library of orthogonal waveforms that has a natural dyadic tree structure. Utilizing this type of structured dictionary makes it easy to construct orthogonal bases by an $O(N \log N)$ search algorithm.

Given a library as a tree structure, the best basis of a signal $\mathbf{x}$ is found by traversing the tree and selecting nodes that correspond to a minimization of the entropy function. The union of these nodes correspond to the best basis [81]. Shannon's entropy function is used as the selection criteria.

In other words, the Best Orthogonal Basis technique attempts to solve $\Phi\mathbf{a} = \mathbf{x}$, by picking $\mathbf{a}$ whose coefficients have the least entropy. Configuration for this method is as follows:

$$\min \sum_{k} \mathbf{a}_k \log \frac{1}{\mathbf{a}_k} \text{ subject to } \Phi\mathbf{a} = \mathbf{x} \tag{B.8}$$

The procedures of this Best Orthogonal Basis technique are as follows:

1. Firstly, it started by constructing a large dictionary, which consists of elements that are orthogonal. These elements are later structured into a tree of sub-collection of the basis functions.

2. The inner product of the data with each element is the computed.

3. The entropy of collections of the resulting coefficients is also computed.

4. The different combinations of the least entropy are retained.

5. Finally, the whole procedures repeat itself until the minimum entropy is obtained.

One of the advantages of using this algorithm is that it brings a near to optimal sparsity representation in some cases [78]. In a way, it allows the decomposing classes of operators to make their structure sparse for fast computation. This resulted in a great interest for the development of the forward-scattering techniques.

However, there are disadvantages with this technique as well [78]. It does not work with dictionaries consisting of disparate, non-orthogonal functions. For example, BOB will not work with dictionaries that are created from scattering functions. However, dictionaries that comprise of elements, which exhibit the characteristics of the signal components and satisfy the structure does work well with BOB. One very good example of such dictionaries is those dictionaries that are composed of wavelets.

B.2.3 Basis Pursuits (BP)

The principle of Basis Pursuits technique, proposed by Chen and Donoho [79] is to find a representation of a signal whose coefficients have minimal L_1 norm. Formally, one solves the problem:

$$\min \|\mathbf{a}\|_1 \text{ subject to } \Phi\mathbf{a} = \mathbf{x} \tag{B.9}$$

From one point of view, equation (B.9) is very similar to equation (B.7) of the Method of Frames. We are simply replacing the L_2 norm in equation (B.7) with the L_1 norm. However, this apparently slight change has major consequences. The Method of Frames leads to a quadratic optimization problem with linear equality constraints, and so involves essentially just the solution of a system of linear equations. The minimization is found in the first derivative where the minimum can be easily found. In contrast, Basis Pursuits requires the solution of a convex optimization problem with inequality constraints, which involves considerably more effort and sophistication [78]. Note that, the solution of equation (B.9) can be obtained by solving an equivalent linear program [85, 86, 87].

Because of the non-differentiability of the L_1 norm, Basis Pursuits leads to decompositions that can have very different properties from the Method of Frames. Basis Pursuits decompositions can be much sparser. Because Basis Pursuits always delivers a decomposition on an optimal basis and not necessarily on an orthogonal basis, it seems better than the Best Orthogonal Basis method resolving non-orthogonal structures. However the cost to achieve this is at expense of greater computational complexity.

B.2.4 Matching Pursuits (MP)

Mallat and Zhang proposed in [2] the matching pursuits algorithm, that decomposes a signal $\mathbf{x}$ iteratively in passes, where in each pass, the waveform (from a redundant dictionary) that gives the largest reduction on the energy of the signal (the best match) is chosen and subtracted from the signal. The residual signal is passed on to the next pass and the process is repeated. The best match is given by the waveform with largest inner product with the signal.

Using a matching pursuits we can decompose a signal $\mathbf{x} \in \mathcal{H}$ of dimension N,

where $\mathcal{H}$ is a Hilbert space, using a redundant dictionary $\mathcal{D} = \{\mathbf{g}_\gamma\}_{\gamma \in \Gamma}$ of M vectors (the $\mathbf{g}_\gamma$ are in general referred to as atoms), and unit norm ($\|\mathbf{g}_\gamma\| = 1, \forall\gamma$). The dictionary is said to be redundant because, in general, $M > N$. A matching pursuits begins by projecting $\mathbf{x}$ on a vector $\mathbf{g}_{\gamma_1} \in \mathcal{D}$ and computing the reside $\mathbf{Rx}$:

$$\mathbf{x} = < \mathbf{x}, \mathbf{g}_{\gamma_1} > \mathbf{g}_{\gamma_1} + \mathbf{Rx} \tag{B.10}$$

where $\mathbf{Rx}$ is the residual vector after approximating $\mathbf{x}$ in the direction of $\mathbf{g}_{\gamma_1}$. Since $\mathbf{Rx}$ is orthogonal to $\mathbf{g}_{\gamma_1}$ (see figure B.1), the modulus of $\mathbf{x}$ is:

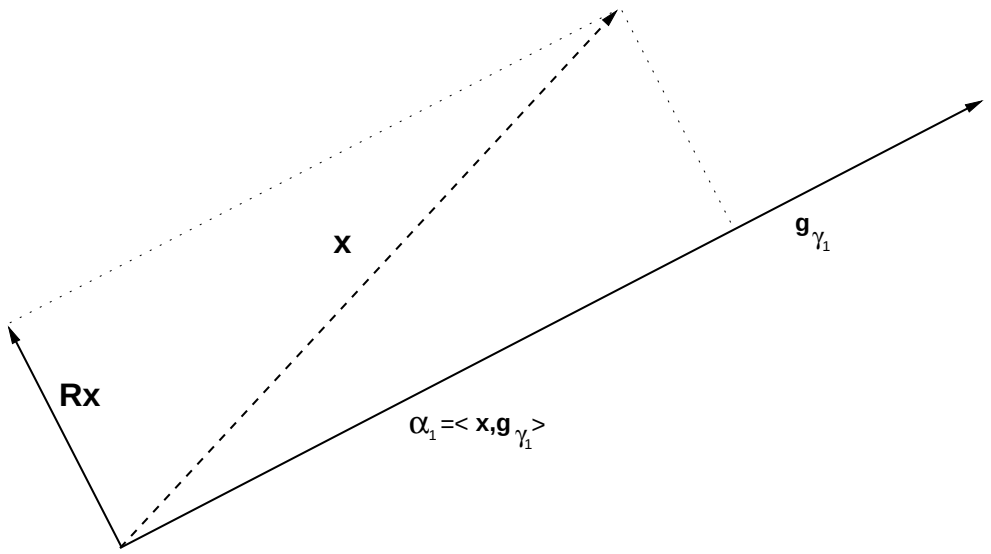


Figure B.1: Representation of a step of the matching pursuits algorithm.

$$\|\mathbf{x}\|^2 = |< \mathbf{x}, \mathbf{g}_{\gamma_1} >|^2 + |\mathbf{Rx}|^2 \tag{B.11}$$

As the term that must be minimized is the error $\|\mathbf{Rx}\|^2 = \|\mathbf{x}\|^2 - |< \mathbf{x}, \mathbf{g}_{\gamma_1} >|^2$, the $\mathbf{g}_{\gamma_1} \in \mathcal{D}$ to be chosen is the one that maximizes $|< \mathbf{x}, \mathbf{g}_{\gamma_1} >|$. In some cases it is not computationally efficient to find the optimal solution, and a suboptimal solution is computed instead:

$$|< \mathbf{x}, \mathbf{g}_{\gamma_1} >| \geq \alpha \sup_{\gamma \in \Gamma} |< \mathbf{x}, \mathbf{g}_\gamma >| \tag{B.12}$$

where $\alpha \in (0, 1]$ is an optimality factor which is 1 when the optimal solution has been chosen. This sub-optimality factor α depends on the searching method used to find the solution.

A matching pursuits is an iterative algorithm that sub-decomposes the residue $\mathbf{Rx}$ by projecting it on a vector of $\mathcal{D}$ that matches $\mathbf{Rx}$ (almost) at best. If we consider $\mathbf{R}^1\mathbf{x} = \mathbf{x}$ and we suppose that the n^{th} order residue $\mathbf{R}^n\mathbf{x}$ ($n \geq 1$) has been computed, the next iteration will chose $\mathbf{g}_{\gamma_n} \in \mathcal{D}$ such that:

$$| < \mathbf{R}^n, \mathbf{g}_{\gamma_n} >> | \geq \alpha \sup_{\gamma \in \Gamma} | < \mathbf{R}^n\mathbf{x}, \mathbf{g}_\gamma > | \tag{B.13}$$

With this choice $\mathbf{R}^n\mathbf{x}$ is projected on $\mathbf{g}_{\gamma_n}$ and decomposed as follows:

$$\mathbf{R}^n\mathbf{x} = < \mathbf{R}^n\mathbf{x}, \mathbf{g}_{\gamma_n} > \mathbf{g}_{\gamma_n} + \mathbf{R}^{n+1}\mathbf{x} \tag{B.14}$$

$\mathbf{R^{n+1}x}$ and $\mathbf{g}_{\gamma_n}$ are orthogonal, so the quadratic module of the previous equation is:

$$\|\mathbf{R}^n\mathbf{x}\|^2 = | < \mathbf{R}^n\mathbf{x}, \mathbf{g}_{\gamma_n} > |^2 + \|\mathbf{R}^{n+1}\mathbf{x}\|^2 \tag{B.15}$$

From equation (B.14) we can see that the decomposition of $\mathbf{x}$ is given by:

$$\mathbf{x} = \sum_{n=1}^{m} < \mathbf{R}^n\mathbf{x}, \mathbf{g}_{\gamma_n} > \mathbf{g}_{\gamma_n} + \mathbf{R}^{m+1}\mathbf{x} \tag{B.16}$$

and with the same principle we can also deduce from equation (B.15) that the L^2 norm of the signal $\mathbf{x}$ is:

$$\|\mathbf{x}\|^2 = \sum_{n=1}^{m} | < \mathbf{R}^n\mathbf{x}, \mathbf{g}_{\gamma_n} > |^2 + \|\mathbf{R}^{m+1}\mathbf{x}\|^2 \tag{B.17}$$

where $\|\mathbf{R}^{m+1}\mathbf{x}\|$ converges exponentially to 0 when m tends to infinity and M is finite (see [88] for a proof).

B.3 The Matching Pursuits Algorithm

In section B.2.4 we have shown that a signal $\mathbf{x}$ can be decomposed using an iterative method [2, 55], known as matching pursuits. This one is a greedy algorithm [88] that decomposes any signal into a linear expansion of waveforms that are selected from a redundant dictionary. These waveforms are iteratively chosen to best match the signal structures, producing a suboptimal expansion. Vectors are selected one by one from the dictionary, while optimizing the signal approximation (in terms of energy) at each step.

Even though the expansion is linear, it gives a non-linear signal decomposition, because the set of functions depends on the signal coded. The Matching Pursuits method has already found applications in medicine [89], audio [90, 91], speech [92] and image [35, 93] and video coding [3, 4, 9, 21] (though in video coding it is usually used to code the motion estimation error [5]). Most of the applications of Matching Pursuits use Gabor functions or symmetric dictionaries. Some use also orthogonalized MP [88] to be able to have a zero estimation error, though this is not very used because it increases too much the computational cost. In this section we show a practical algorithm to implement the matching pursuits method as well as several properties of this one.

B.3.1 The Algorithm

From equation (B.16) we can obtain an approximation of the signal $\mathbf{x}$ in P passes [2]

$$\mathbf{x} \approx \sum_{n=1}^{P} p_n \mathbf{g}_{\gamma_n} \tag{B.18}$$

where $p_n = < \mathbf{R}^n \mathbf{x}, \mathbf{g}_{\gamma_n} >$ are the inner products.

In the matching pursuits we have to compute only the pairs (p_n, γ_n). It can be done by the algorithm below

MP Algorithm

❶ Start with $\mathbf{w} = \mathbf{x}$, $n = 1$.

❷ Repeat until a stop criterion is met

(a) Choose $\gamma_n \in \{1, \ldots, M\}$ such that

$$\mathbf{w} \cdot \mathbf{g}_{\gamma_n} = \max_{1 \le j \le M} \{\mathbf{w} \cdot \mathbf{g}_j\}.$$

(b) Choose $p_n = < \mathbf{w}, \mathbf{g}_{\gamma_n} >$.

(c) Replace $\mathbf{w}$ by $\mathbf{w} - p_n \mathbf{g}_{\gamma_n}$.

(d) Increment n.

❸ Stop.

It is important to note that the matching pursuits coefficients p_n computed by the MP Algorithm must be quantized for being used in compression applications [11, 10, 7, 94].

B.3.2 Properties of Matching Pursuits

Matching Pursuits has two kinds of properties: properties that depends on the dictionary (so the Matching Pursuits decomposition only has them if the chosen dictionary has them as well) and the ones that are independent on the dictionary (so any Matching Pursuits decomposition has them no matter which set of functions has been used to perform the decomposition).

The main properties derived directly from the matching pursuits algorithm (so, independent from the dictionary) are:

- *Energy conservation:* In Matching Pursuits, when dealing with an infinite decomposition series, the energy in the transformed domain and the energy in the space domain is the same, as can be deduced from equation (B.17). As:

$$\lim_{M \to \infty} \mathbf{R}^M \mathbf{x} = 0 \tag{B.19}$$

(because of the exponential decrease of the coefficients and the error), when $M \to \infty$ equation (B.17) turns to:

$$||\mathbf{x}||^2 = \sum_{m=1}^{\infty} | < \mathbf{R}^m \mathbf{x}, \mathbf{g}_{\gamma_m} > |^2 \tag{B.20}$$

which mimics Parseval's equality for Fourier series.

- *Invertible*: A complete Pursuits recovers a perfect version of the signal **x**:

$$\mathbf{x} = \sum_{m=1}^{\infty} < \mathbf{R}^m \mathbf{x}, \mathbf{g}_{\gamma_m} > \mathbf{g}_{\gamma_m} \tag{B.21}$$

  Thus the signal **x** may be reconstructed from its Matching Pursuits coefficients, but if the decomposition is finite in general the reconstruction of the coded signal will not be perfect, there will be a reconstruction error given by $\mathbf{R}^{M+1}\mathbf{x}$, where M is the number of coefficients used by the decomposition.

- *Non-linearity*: The fact of having a non-linear dictionary gives to matching pursuits the two following characteristics, which are very appreciated when performing coding:

  - *Robustness to quantization*: Robustness to quantization comes from the fact the decomposition is overcomplete. Because of overcompleteness, the transformed domain space has dimension P, higher than the dimension N of the original signal. When quantizing the transformed domain, error quantization is spread allover the P dimensions of the transformed domain. But when applying the inverse transform, some of the information in the transformed domain (the one that belongs to the dimensions that do not exist in the original space) is lost, and so part of the quantization error performed will not at all affect the quality of the decoded signal.
  - *Exponential decrease of the error*: This implies a great decrease in the first coefficients, and thus a fast initial approximation. After a certain number of coefficients, the Matching Pursuits error decrease is no too fast, and a change

of dictionary or of coding method may be worthy. This exponential decreasing of the error will only happen if the signal has finite dimension, because in infinite dimension the condition of the unit sphere being dense in the working space becomes false.

The Matching Pursuits may have other properties, depending on the properties of the set of functions in the dictionary used to decompose the signal. In many signal processing application, like image processing, it may be interesting to have certain invariance to image transformations, as translation, rotation and dilation (for example, in pattern recognition). Matching Pursuits will have such invariances if the dictionary used to decompose a signal has also them:

- *Translation Invariance*: A dictionary $\mathcal{D}$ is translation invariant if for any $\mathbf{g}_\gamma[n] \in \mathcal{D}$ and any $k = [k_x, k_y] \in [0 \ldots N-1, 0 \ldots N-1]$ then $\mathbf{g}_\gamma[n-k] \in \mathcal{D}$. If matching pursuits is computed in a translation invariant dictionary, then its decomposition will be translation invariant. Given the decomposition of $\mathbf{x}$ in $\mathcal{D}$,

$$\mathbf{x}[n] = \sum_{m=1}^{M} < \mathbf{R}^m \mathbf{x}, \mathbf{g}_{\gamma_m} > \mathbf{g}_{\gamma_m}[n] + \mathbf{R}^{M+1}\mathbf{x}[n] \tag{B.22}$$

  it is easy to verify [88] that the matching pursuit of $\mathbf{x}_k[n] = \mathbf{x}[n-k]$ selects a translation by k of the same vectors $\mathbf{g}_{\gamma_m}$ with the same decomposition coefficients:

$$\mathbf{x}_k[n] = \sum_{m=1}^{M} < \mathbf{R}^m \mathbf{x}, \mathbf{g}_{\gamma_m} > \mathbf{g}_{\gamma_m}[n-k] + \mathbf{R}^{M+1}\mathbf{x}_k[n] \tag{B.23}$$

- *Rotation invariance*: By analogy, a rotation invariant matching pursuit can be obtained by using a rotation invariant dictionary $\mathcal{D}$. A dictionary is rotation invariant if for any $\mathbf{g}_\gamma[n] \in \mathcal{D}$ and any $\theta \in [0, 2\pi]$ then $\mathbf{g}_\gamma[r_\theta \cdot n] \in \mathcal{D}$, where r_θ is the rotation operator given by the matrix:

$$r_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{B.24}$$

  Given the decomposition of $\mathbf{x}$ in $\mathcal{D}$,

$$\mathbf{x}[n] = \sum_{m=1}^{M} < \mathbf{R}^m \mathbf{x}, \mathbf{g}_{\gamma_m} > \mathbf{g}_{\gamma_m}[n] + \mathbf{R}^{M+1}\mathbf{x}[n] \tag{B.25}$$

one can verify that the matching pursuit if $\mathbf{x}_\theta[n] = \mathbf{x}[r_\theta \cdot n]$ selects a rotation by θ of the same vectors $\mathbf{g}_{\gamma_m}$ with the same decomposition coefficients:

$$\mathbf{x}_\theta[n] = \sum_{m=1}^{M} < \mathbf{R}^m \mathbf{x}, \mathbf{g}_{\gamma_m} > \mathbf{g}_{\gamma_m}[r_\theta \cdot n] + \mathbf{R}^{M+1}\mathbf{x}_\theta[n] \tag{B.26}$$

This makes Matching Pursuit a useful technique to rotate images, because the only extra calculation to be done is to modify the index of the reconstruction atoms for computing the coded image, instead of applying the rotation matrix to every pixel of the image.

- *Dilation invariance*: As in the previous two cases, Matching Pursuit is dilation invariant if the dictionary functions used by the pursuit is dilation invariant. A dictionary $\mathcal{D}$ is dilation invariant when for any $\mathbf{g}_\gamma[n] \in \mathcal{D}$ and any $s \in [0, s_{max}]$ then $\mathbf{g}_\gamma\left[\frac{n}{s}\right]$ will select a dilation by s of the same vectors $\mathbf{g}_{\gamma_m}$ with the same decomposition coefficients:

$$\mathbf{x}_s[n] = \sum_{m=1}^{M} < \mathbf{R}^m \mathbf{x}, \mathbf{g}_{\gamma_m} > \mathbf{g}_{\gamma_m}\left[\frac{n}{s}\right] + \mathbf{R}^{M+1}\mathbf{x}_s[n] \tag{B.27}$$

It is important to note that decompositions performed using the Matching Pursuits Algorithm are different from the ones performed by the algorithms that use transforms, where we must encode all the transform coefficients. In the Matching Pursuits Algorithm, one needs to encode only the projections of the chosen waveforms and the indexes of these ones. Thus, the Matching Pursuits Algorithm performs a kind of adaptive decomposition, since, for each signal, we can use a different set of the waveforms from in the whole dictionary. On the other hand, the algorithms based on transforms, like DCT, must use always all the waveforms of the dictionary during the decomposition of the signal. The Matching Pursuits Algorithm can also perform a compact decomposition of the signal. This is due to the fact that as an overcomplete dictionary is used, we can choose the waveforms of this dictionary which are most similar to the structures of signal that we want to encode.

B.4 Conclusions

In this appendix we have shown the main characteristics of the signal representation, including a brief review of several adaptive signal decomposition methods. Also we have described the Matching Pursuits Algorithm proposed by Mallat and Zhang in [2]. The main characteristics of this algorithm have also been commented.

Apêndice C

Generalized Bit-Plane Decomposition

In this appendix we propose a novel algorithm to perform an MP-like decomposition in which a signal is decomposed in generalized bit-planes, each bit-plane being composed by a set of atoms. In it, unlike the classical Matching Pursuits, there are no coefficients to be quantized. That is, only the atoms corresponding to each generalized bit-plane need to be transmitted. It provides an elegant solution to the coefficient quantization problem in the MP algorithm, and presents improvements over the existing MP-based encoders.

This appendix is organized as follows: Section C.1 outlines the theory of signal decomposition in generalized bit-planes, i.e., the base of the proposed algorithm. In section C.2 we analyze the theoretical bounds and the rate-distortion characteristics of this algorithm. Section C.3 presents a theoretical analysis of the rate-distortion performance using parameters of both a regular lattice and a dictionary formed by the union of an orthogonal dictionary and its symmetry in relation to the origin. The conclusions can be found in section C.4.

C.1 Signal Decomposition in Generalized Bit-Planes

Suppose $\mathbf{x}$ is an N-dimensional signal that can be decomposed in a redundant dictionary $\mathcal{D} = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_M\}$, $\|\mathbf{g}_i\| = 1, \forall i$ as

$$\mathbf{x} = \sum_{n=1}^{M} c_n \mathbf{g}_n \tag{C.1}$$

Without loss of generality, we are assuming that $\|\mathbf{x}\| \leq 1$. Also, note that we are considering that the dictionary $\mathcal{D}$ is complete, so that an expansion in M terms can represent $\mathbf{x}$ with zero distortion. In addition, since $\|\mathbf{g}_i\| = 1$, there is an expansion in the form of equation (C.1) such that $|c_n| \leq 1$.

Since $|c_n| \leq 1$, we can write the binary representation for c_n as $c_n = s_n \sum_{j=1}^{\infty} 2^{-j} b_{j,n}$. $s_n \in \{-1, 1\}$ is the sign of c_n, and $b_{j,n} \in \{0, 1\}$. Replacing this value of c_n in equation (C.1) we have that

$$\begin{aligned} \mathbf{x} &= \sum_{n=1}^{M} s_n \sum_{j=1}^{\infty} 2^{-j} b_{j,n} \mathbf{g}_n = \sum_{j=1}^{\infty} 2^{-j} \sum_{n=1}^{M} b_{j,n} s_n \mathbf{g}_n \\ &= \sum_{j=1}^{\infty} 2^{-j} \sum_{n=1}^{M} b_{j,n} \overline{\mathbf{g}}_n \end{aligned} \tag{C.2}$$

Note that since $s_n \in \{-1, 1\}$, then $\overline{\mathbf{g}}_n = s_n \mathbf{g}_n \in \overline{\mathcal{D}} = \{\pm\mathbf{g}_1, \pm\mathbf{g}_2, \ldots, \pm\mathbf{g}_M\}$. Now, defining the indexes $i_{j,l}$ such that, for $l \in \{1, 2, \ldots, L_j\}$, $L_j \leq M$, $b_{j,i_{j,l}} = 1$, and zero elsewhere, the summation in equation (C.2) can be expressed as

$$\mathbf{x} = \sum_{j=1}^{\infty} 2^{-j} \sum_{l=1}^{L_j} \overline{\mathbf{g}}_{i_{j,l}} \tag{C.3}$$

Equation (C.3) can be regarded as a generalized bit-plane decomposition of the signal $\mathbf{x}$. The bit-plane j is composed by the functions $\overline{\mathbf{g}}_{i_{j,l}}$ for $l = 1, \ldots, L_j$. In [24] a convergent algorithm for finding such decompositions has been proposed, in the same philosophy of the MP algorithm. In fact, the algorithm proposed in [24] finds decompositions of the following form

$$\mathbf{x} = \sum_{j=1}^{\infty} \alpha^j \sum_{l=1}^{L_j} \overline{\mathbf{g}}_{i_{j,l}} \tag{C.4}$$

These decompositions are more general then the one in equation (C.3), since the term 2^{-j} has been replaced by α^j, for $0 < \alpha < 1$. We refer to α as the *approximation scaling factor*.

In [24] there have been derived conditions for the algorithm to be convergent (that is, for any signal $\mathbf{x}$ be approximated with arbitrary precision by adding a sufficient number of terms to the summations). These conditions impose that $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{3}$, where $\Theta(\overline{\mathcal{D}})$ is the largest angle between any signal $\mathbf{x} \in \mathbb{R}^N$ and the closest atom in dictionary $\overline{\mathcal{D}}$. However, even for signals of moderate dimension (e.g., $N \geq 64$), the dictionaries that could provide $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{3}$ would have very large cardinality. This would lead to inefficient

decompositions from an R-D perspective, since a large number of bits would be needed to encode the indexes $i_{j,l}$.

In this appendix we propose a novel algorithm (MPGBP Algorithm- Matching Pursuits Generalized Bit-Planes Algorithm) for finding such decompositions, that is convergent whenever $0 < \alpha < 1$ and $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{2}$. The advantage of this algorithm is that $\Theta(\overline{\mathcal{D}}) \leq \frac{\pi}{2}$ is only a very mild restriction, being satisfied whenever $\mathcal{D}$ is complete [2]. Since it implements a decomposition as in equation (C.4), this algorithm is a generalization of any decomposition in basis with the approximation being performed in bit-planes [24]. In the MPGBP algorithm a greedy decomposition is carried out by adding one atom $\overline{\mathbf{g}}_{i_{j,l}}$ at a time, until a rate and/or distortion criterion is met. Given a dictionary $\mathcal{C} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$, $\|\mathbf{v}_i\| = 1$, $\forall i$, the algorithm is as follows (the input signals are normalized so that $\|\mathbf{x}\| \leq 1$):

MPGBP Algorithm

❶ Start with $\mathbf{w} = \mathbf{x}$, $m = 1$, for a given α.

❷ Repeat until a stop criterion is met

(a) Choose $r_m \in \{1, \ldots, M\}$ such that

$$\mathbf{w} \cdot \mathbf{v}_{r_m} = \max_{1 \leq j \leq M} \{\mathbf{w} \cdot \mathbf{v}_j\}.$$

(b) Choose $k_m = \left\lceil \frac{\ln(\mathbf{w} \cdot \mathbf{v}_{r_m})}{\ln(\alpha)} \right\rceil$.
where $\lceil \cdot \rceil$ is the ceiling operator, i.e., returns the smallest integer larger than or equal to the argument.

(c) Replace $\mathbf{w}$ by $\mathbf{w} - \alpha^{k_m} \mathbf{v}_{r_m}$.

(d) Increment m.

❸ Stop.

Note that the MPGBP algorithm approximates $\mathbf{x}$ in P ($P \leq M$) passes as

$$\mathbf{x}^{(P)} = \sum_{m=1}^{P} \alpha^{k_m} \mathbf{v}_{r_m} \tag{C.5}$$

If we define L_j as the number of values m such that $k_m = j$, we can rename the corresponding indexes r_m as $i_{j,l}$ for $l = 1, \ldots, L_j$. Therefore, if we make the dictionary $\mathcal{C}$ in MPGBP algorithm equal to $\overline{\mathcal{D}}$, then equation (C.5) is equivalent to equation (C.4) for $P \to \infty$.

We can say that the MPGBP algorithm is convergent if $\lim_{P \to \infty} \mathbf{x}^{(P)} = \mathbf{x}$. In that sense, its convergence is guaranteed by Theorem 1 (the proof can be found in appendix H).

Theorem 1: *Be* $\mathbf{x} \in \mathbb{R}^N$, $\|\mathbf{x}\| \leq 1$, *such that it is approximated by MPGBP algorithm using a dictionary* $\mathcal{C}$ *with* P *passes, generating* $\mathbf{x}^{(P)}$ *as in equation (C.5), and be* $\Theta(\mathcal{C})$ *the largest angle between any signal* $\mathbf{y} \in \mathbb{R}^N$ *and the closest atom in dictionary* $\mathcal{C}$. *We have that* $\|\mathbf{r}^{(P)}\| = \|\mathbf{x} - \mathbf{x}^{(P)}\| \leq \beta_c^P$, *where* $\beta_c = \sqrt{1 - (2\alpha - \alpha^2)\cos^2(\Theta(\mathcal{C}))} < 1$ *for every* $0 < \alpha < 1$ *and* $0 \leq \Theta(\mathcal{C}) < \frac{\pi}{2}$.

The following points regarding the MPGBP algorithm should be highlighted:

(i) This algorithm performs a decomposition such that, for every atom added, the distortion in the approximation of $\mathbf{x}$ decreases by at least β, where $\beta < 1$. Thus, when the number of passes $P \to \infty$, $\|\mathbf{r}^{(P)}\| \to 0$, that is, the MPGBP algorithm is convergent.

(ii) The representation output by MPGBP algorithm is given by just a sequence of pairs of indexes (k_m, r_m), $m = 1, 2, \ldots, P$. This implies that there is no need for coefficients quantization as in the classical MP algorithm (see equation (C.1) and the discussion that follows). In other words, it can be said that the MPGBP algorithm performs both the decomposition and quantization at the same time. Thus, it presents an elegant solution to the coefficient quantization problem inherent to the classical MP algorithm.

(iii) The decomposition obtained can be organized in bit-planes as in equation (C.4). This can be done by noting that, in equation (C.5), the indexes r_m for the values of m such that $k_m = j$ correspond to the atoms comprising bit-plane j.

(iv) The number of atoms used in the decomposition can be set arbitrarily and each atom corresponds to a pair (k_m, r_m). This permits a precise rate control, since the decomposition can be stopped when the bit-budget is exhausted. This feature can be useful in more sophisticated R-D schemes.

(v) Note that Theorem 1 states that a decomposition as in equations (C.4) and (C.5) can be performed for every $0 < \alpha < 1$ and every complete dictionary. Therefore, decompositions as in equations (C.4) and (C.5) are generalizations of any decomposition plus bit-plane encoding paradigm, that is, any decomposition in a dictionary followed by bit-plane encoding is a particular case of equations (C.4) and (C.5).

C.2 Theoretical Bound on Rate-Distortion Performance

In this section we compute a theoretical bound on the rate-distortion performance of the decompositions that we have proposed in section C.1 (see the MPGBP algorithm). We can see, from theorem 1, that the signal residual $\mathbf{r}$ decreases its magnitude by at least β_c in each pass, that is,

$$\frac{\|\mathbf{r}^{(P)}\|}{\|\mathbf{r}^{(P-1)}\|} \leqslant \sqrt{1-(2\alpha-\alpha^2)\cos^2(\Theta(\mathcal{C}))} = \beta_c < 1 \tag{C.6}$$

for every $0 < \alpha < 1$ and $0 \leq \Theta(\mathcal{C}) < \frac{\pi}{2}$.

Then, after P passes, and assuming that $\|\mathbf{x}\| \leqslant 1$, the mean square distortion per coefficient, D, is bounded by

$$D \leqslant \frac{\beta_c^{2P}}{N} \tag{C.7}$$

where N is the dimension of the dictionary $\mathcal{C}$. Note that, because the overall distortion per vector, ND, decreases in every pass (theorem 1), then , since $\|\mathbf{x}\| \leqslant 1$, $ND \leqslant 1$, $\forall P$. We can reorganize equation (C.7) so as to obtain the maximum number of passes P that guarantees a maximum distortion per coefficient equal to D

$$P \leqslant \frac{1}{2}\frac{\log_2 ND}{\log_2 \beta_c} \tag{C.8}$$

If the dictionary $\mathcal{C}$ has cardinality q and at most b_c bits are spent for coding each index k_m, the rate per coefficient is bounded by (actually, if entropy coding is used, it can be much smaller, depending on the statistics of k_m and r_m).

$$R \leqslant \frac{1}{N}\left[P \log_2 q + P b_e\right] \tag{C.9}$$

It is important to note that the number of bits b_e used to calculate the rate in equation (C.9) has a strong dependence on the maximum value of the index k_m index. For example, using a natural code, $b_e \geqslant \log_2 k_m$. We can see from figure C.1 that the maximum k_m value corresponds to the minimum value of y. Also from this figure, we can see that if y is minimum, then D is minimum. Since the error in the approximation decreases by at least β_c in each pass, we have that the segments $\overline{ab}$ and $\overline{bd}$ in figure C.1 are larger than or equal to $\frac{\sqrt{ND}}{\beta}$ and $\sqrt{ND}$, respectively. Since we are interested in the minimum y and $\beta \leqslant \beta_c$, then if we make $\beta = \beta_c$, the estimative that we obtain for y is smaller than or equal to the minimum, and is therefore conservative.

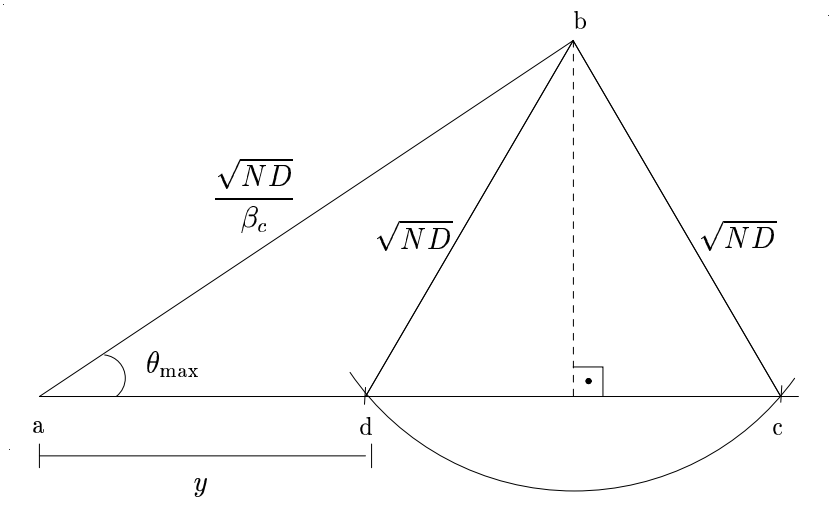


Figure C.1: Computation of y_{min}.

From figure C.1 we obtain

$$\left(\sqrt{ND}\right)^2 = \left(\frac{\sqrt{ND}}{\beta_c}\right)^2 + y^2 - 2y\frac{\sqrt{ND}}{\beta_c}\cos\left(\Theta(\mathcal{C})\right) \tag{C.10}$$

and consequently

$$y^2 - 2y\frac{\sqrt{ND}}{\beta_c}\cos\left(\Theta(\mathcal{C})\right) - ND\left(1 - \frac{1}{\beta_c^2}\right) = 0 \tag{C.11}$$

If we use the β_c value obtained from theorem 1 ($\beta_c = \sqrt{1-(2\alpha-\alpha^2)\cos^2\left(\Theta(\mathcal{C})\right)}$), the smallest y is given by (the largest one is equal to the segment $\overline{ac}$)

$$
\begin{aligned}
y_{min} &= \frac{\sqrt{ND}}{\beta_c}\left(\cos(\Theta(\mathcal{C})) - \sqrt{\beta_c^2 - \sin^2(\Theta(\mathcal{C}))}\right) \\
&= \frac{\sqrt{ND}}{\beta_c}\left(\cos(\Theta(\mathcal{C})) - \sqrt{\cos^2(\Theta(\mathcal{C}))(1-\alpha)^2}\right) \\
&= \frac{\sqrt{ND}}{\beta_c}(\cos(\Theta(\mathcal{C})) - \cos(\Theta(\mathcal{C}))(1-\alpha)) \\
&= \frac{\sqrt{ND}}{\beta_c}\alpha\cos(\Theta(\mathcal{C})) \qquad \text{(C.12)}
\end{aligned}
$$

Considering that the maximum number of bits for representing k_m corresponds to $(k_m)_{max}$ such that $\alpha^{(k_m)_{max}} \leqslant y_{min}$, then we have

$$
(k_m)_{max} \geqslant \frac{\log_2\left(\frac{\sqrt{ND}\,\alpha\cos(\Theta(\mathcal{C}))}{\beta_c}\right)}{\log_2\alpha} \qquad \text{(C.13)}
$$

From equation (C.13), since $b_e = \log_2[(k_m)_{max}]$, the minimum of bits b_e ($b_{e_{min}}$) that can be used to code each index k_m for distortion D is given by

$$
b_{e_{min}} = \log_2\left\lceil \frac{\log_2\left(\frac{ND\alpha^2\cos^2(\Theta(\mathcal{C}))}{\beta_c^2}\right)}{2\log_2\alpha}\right\rceil \qquad \text{(C.14)}
$$

From equations (C.9) and (C.14) we have that

$$
\begin{aligned}
R &\leqslant \frac{1}{N}P\left\{\log_2 q + \log_2\left\lceil \frac{\log_2\left(\frac{ND\alpha^2\cos^2(\Theta(\mathcal{C}))}{\beta_c^2}\right)}{2\log_2\alpha}\right\rceil\right\} \\
&\leqslant \frac{1}{N}P\log_2\left[\frac{q}{2\log_2\alpha}\log_2\left(\frac{ND\alpha^2\cos^2(\Theta(\mathcal{C}))}{\beta_c^2}\right)\right] \qquad \text{(C.15)}
\end{aligned}
$$

Now, if we use the value of P obtained in equation (C.8), the rate can be expressed as

$$
R \leqslant \frac{1}{2N}\frac{\log_2(ND)}{\log_2\beta_c}\log_2\left[\frac{q}{2\log_2\alpha}\log_2\left(\frac{ND\alpha^2\cos^2(\Theta(\mathcal{C}))}{\beta_c^2}\right)\right] \qquad \text{(C.16)}
$$

It is important to note that both the dimension N and the cardinality q are dependent on the dictionary used. In fact, we have that the triple (Θ, N, q) depends on the

dictionary $\mathcal{C}$. Then, in order to be precise we should use the notation $(\Theta(\mathcal{C}), N(\mathcal{C}), q(\mathcal{C}))$ in our formulation. However, we use $(\Theta(\mathcal{C}), N, q)$ in order to not make it cumbersome.

Suppose we have different dictionaries $(\mathcal{C}_1, \mathcal{C}_2, \cdots)$ of same dimension and cardinality, but with different $\Theta(\mathcal{C}_i)$ values. We can analyze how the reduction of $\Theta(\mathcal{C}_i)$ values influences the rate spent when equation (C.16) is used. Figure C.2 shows the curves of rate spent versus α for 7 dictionaries $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_7$ with different values for $\Theta(\mathcal{C}_i)$. In order to obtain these curves we use for the dimension N, the cardinality q and the distortion D, the values 10, 10^4 and 10^{-2}, respectively. We can see from this figure that there is an optimum α for each $\Theta(\mathcal{C}_i)$ value. Also, for a particular α value the rate spent is smaller when we use a dictionary with a smaller value for $\Theta(\mathcal{C}_i)$. This can also be observed in figure C.3, that shows the surfaces of the rate spent when we have the distortion D and the α parameter varying for two dictionaries with different $\Theta(\mathcal{C}_i)$ values.

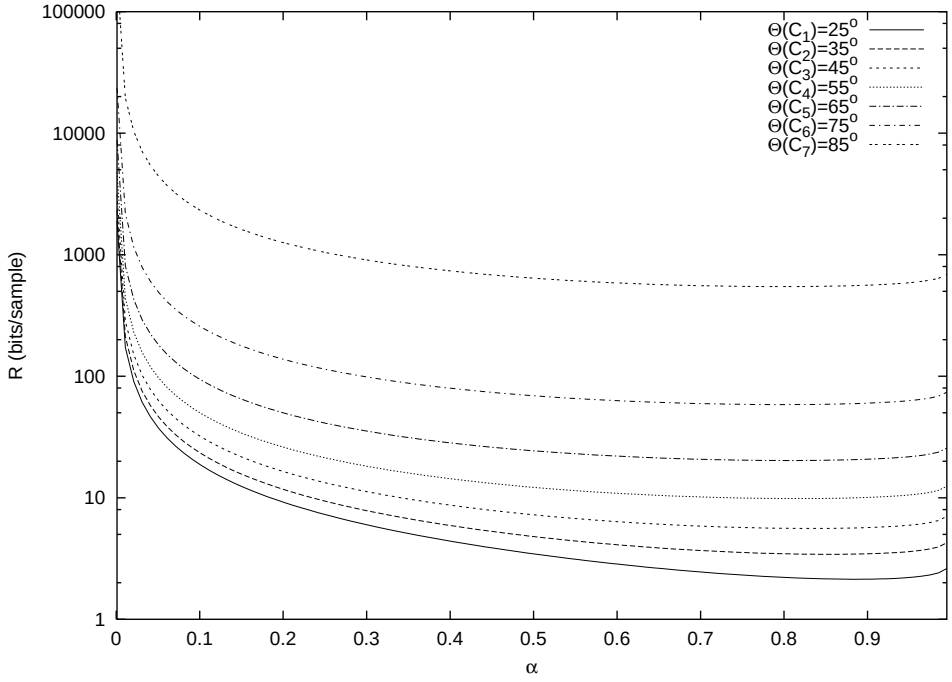


Figure C.2: Variation of the rate with α parameter for dictionaries with different $\Theta(\mathcal{C}_i)$ values.

In figure C.4 we can see the rate$\times$distortion curves for 7 dictionaries $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_7$, with the dimension $N = 10$, for an $\alpha = 0.7$ and a cardinality q equal to 10^3, with different $\Theta(\mathcal{C}_i)$ values. This figure shows that dictionaries with smaller $\Theta(\mathcal{C}_i)$ produce a smaller rate for all distortion levels.

Analyzing equation (C.16) is possible to see that if the cardinality q is increased,

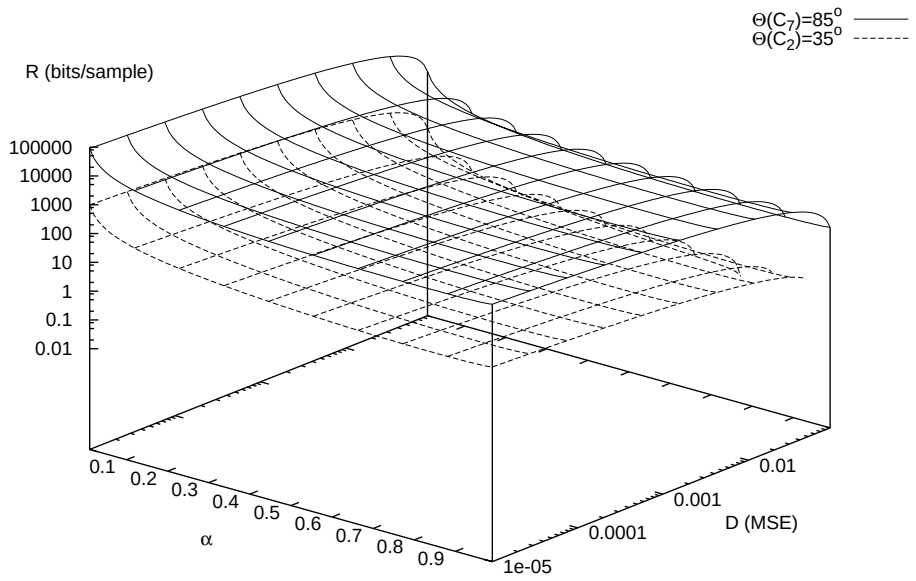


Figure C.3: Surface that shows the variation of the rate with α parameter and the distortion for two dictionaries with different $\Theta(\mathcal{C}_i)$ values.

the rate tends to increase too. As it was pointed out above, the parameters N, q and $\Theta(\mathcal{C})$ have a strong dependence on the particular dictionary used. In general, for a given dictionary with a good distribution of points (that is, tending to be uniformly distributed), if the dimension N is held constant and its cardinality q is increasing, we have $\Theta(\mathcal{C})$ decreasing. Therefore, if the rate increase caused by the increasing of the cardinality is compensated by a significant reduction of $\Theta(\mathcal{C})$, then there can be an overall rate reduction.

C.3 Rate-Distortion Analysis Using Different Dictionaries

C.3.1 Regular Lattices

A regular lattice is a discrete set of points in the k-dimensional Euclidean space $\mathcal{R}^k$ that can be generated by the integer linear combination of a given set of basis vectors. A k-dimensional lattice L_k is defined as a subset of real space $\mathcal{R}^k$, such that:

$$L_k = \{\vec{y} \in \mathcal{R}^k : \vec{y} = a_1\vec{u}_1 + a_2\vec{u}_2 + \cdots + a_k\vec{u}_k\} \tag{C.17}$$

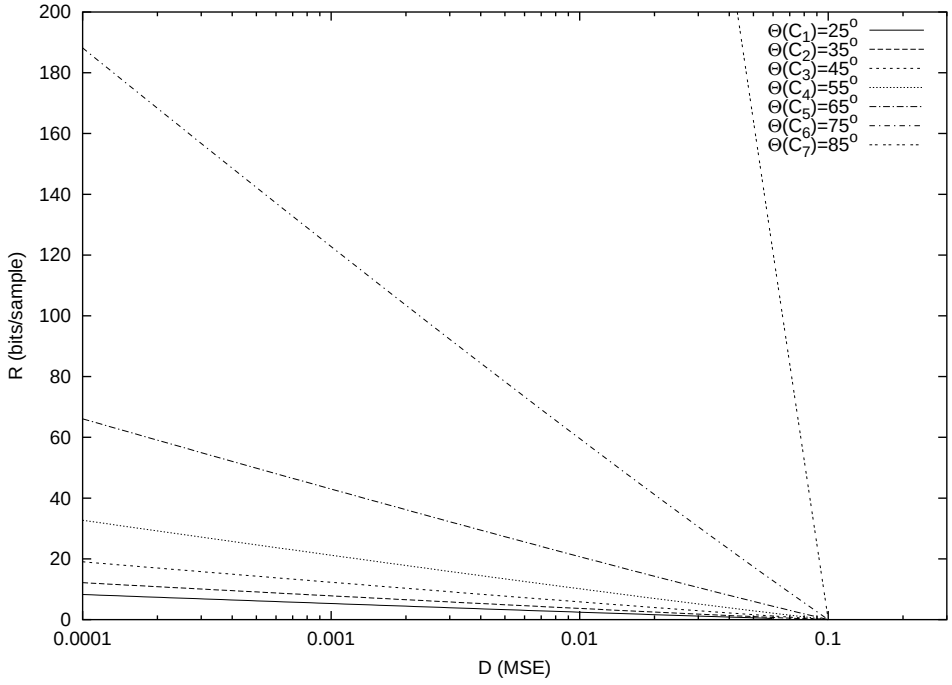


Figure C.4: R-D curves for dictionaries with different $\Theta(\mathcal{C}_i)$ values.

where $\{\vec{u}_i\}$ is a set of linearly independent vectors that span L_k, called basis vectors of lattice L_k, and $\{a_i\}$ is the set of integers which specify a particular point in lattice L_k, known as coefficients of the basis vectors.

Regular lattices have been originally investigated in the context of sphere packing. Sphere packing is concerned with the densest way of arranging k-dimensional, identical, non-overlapping spheres in the real k-dimensional space [25, 26].

An important category of lattices are the root lattices, namely $Z_k(k > 1)$, $A_k(k > 1)$, $D_k(k > 3)$, $E_k(k = 6, 7, 8)$ [95, 96, 97, 98, 99, 100], the Barnes-Wall Λ_{16} [101] and the Leech Λ_{24} [25, 102], which have been shown to offer the best known lattice packing of their space [25]. All these regular lattices have an infinite number of vectors.

C.3.1.1 The integer lattice Z_k

The integer or cubic lattice $Z_k(k > 1)$ is defined as the set of k-dimensional vectors with all integer components $Z_k = \{\vec{y} = (y_1 y_2 \cdots y_k) : y_i \in \mathcal{Z}\}$ where $\mathcal{Z}$ is the set of integer numbers. Lattice Z_k gives the simplest structure of points in $\mathcal{R}^k$, and most regular lattices can be generated from Z_k.

C.3.1.2 The lattice D_k

The k-dimensional lattice $D_k (k > 3)$ is defined by spanning the integer lattice Z_k and retaining those points $\vec{y}$ in Z_k which have coordinates with an even sum:

$$D_k = \{\vec{y} : y_i \in \mathcal{Z} \wedge \sum_{i=1}^{k} y_i = 0 (\mathrm{mod} 2)\} \tag{C.18}$$

D_k is the "back-bone"of other more complex lattices that give the most dense packing at high dimensions, namely the Gosset E_8 and Barnes-Wall Λ_{16}.

C.3.1.3 The lattice E_8

The most dense lattices in $k = 6, 7$ and 8 dimensions are the members of the $E_k (k = 6, 7, 8)$ family. Among them the Gosset E_8 is particularly useful due to its symmetrical structure. It is defined as the union of two subset of points, the lattice D_8 and the coset $\{D_8 + \frac{\vec{1}}{2}\}$:

$$E_8 = D_8 \cup \left\{ D_8 + \frac{\vec{1}}{2} \right\}, \text{ where } \frac{\vec{1}}{2} = \left(\frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \frac{1}{2} \right) \tag{C.19}$$

C.3.1.4 The lattice Λ_{16}

The Barnes-Wall lattice Λ_{16} is the most dense lattice at $k = 16$ dimensions. Λ_{16} can be conveniently defined as the union of 32 cosets of the lattice $2D_{16}$. The scaled lattice $2D_{16}$ is the set of even coordinate points in Z_{16} such that the sum of the coordinates is a multiple of 4. Thus, Λ_{16} is defined as:

$$\Lambda_{16} = \bigcup_{i=1}^{32} \{\vec{c}_i + 2D_{16}\} \tag{C.20}$$

where the coset representatives $\vec{c}_i$ are codewords of the rows of the Hadamard matrix H_{16} and its complimentary $\bar{H}_{16}$ after changing 1's to 0's and -1's to 1's. Therefore, Λ_{16} can be decomposed into 32 subsets of points based in the Hadamard matrix rows:

$$\begin{aligned}\Lambda_{16} = \{2D_{16} + (0000000000000000)\} \ \cup \ \{2D_{16} + (1111111111111111)\} \cup \\ \{2D_{16} + (0101010101010101)\} \ \cup \ \{2D_{16} + (1010101010101010)\} \cup \\ \{2D_{16} + (0011001100110011)\} \ \cup \ \{2D_{16} + (1100110011001100)\} \cup \\ \vdots \\ \{2D_{16} + (0110100110010110)\} \ \cup \ \{2D_{16} + (1001011001101001)\} \end{aligned} \quad \text{(C.21)}$$

C.3.1.5 The lattice Λ_{24}

The lattice Λ_{24} was discovered by Leech in 1965 [102]. This lattice is generated by all vectors of the form [25]:

$$\frac{1}{\sqrt{8}}\left(\mp 3, \pm 1^{23}\right) \quad \text{(C.22)}$$

where the ∓ 3 may be in any position, and the upper signs are taken on a "$\mathcal{C} - set$", i.e. the set of coordinates where a codeword of the binary Golay code [103] $\mathcal{C}_{24}$ is 1.

C.3.1.6 Shells of a Regular Lattice

All the lattices defined above have an infinite number of vectors. However, here we are interested only in a finite number of orientation code vectors because the process must lead to a compact representation of vectors. Therefore, we have to make use of the concept of shells of a regular lattice. In general, the points of a given regular lattice are distributed on the surface of successive, concentric, k-dimensional hyper-shells centered at the origin, so that all lattice points at the same shell have the same l_r-norm. Hence, the m^{th} shell S_m of a given lattice L_k is the set of all L_k-points at the same distance from the origin, $r(L_k, m)$:

$$S_m(L_k) : \{\vec{y} \in L_k : \|\vec{y}\|_r = r(L_k, m)\} \quad \text{(C.23)}$$

where $\|\vec{y}\|_r = \left[\sum_{i=1}^{k} |y_i|^r\right]^{\frac{1}{r}}$ is the l_r norm of $\vec{y}$.

The shells have pyramidal shape for $r = 1$ (l_1-norm) and spherical shape for $r = 2$ (l_2-norm). The exact number of L_k-lattice point at any shell, for the most important regular lattices, can be calculated by using the theta functions[104] or the recently developed

$N\mu$ functions[105] for spherical and pyramidal shells, respectively. In this analysis, only the case of $r = 2$ (Euclidean distance) will be considered.

C.3.1.7 Analyzing the Rate-Distortion Performance with Regular Lattices

In this section we show the rate-distortion performance of the MPGBP algorithm using regular lattices. Table C.1 shows the parameters $N(\mathcal{L}_k)$ (dimension), $q(\mathcal{L}_k)$ (cardinality) and $\Theta(\mathcal{L}_k)$ of the each regular lattice used.

Table C.1: Parameters of regular lattices with best known packing for dimensions N=4, 8, 16 and 24.

Dictionary ($\mathcal{L}_N$)	shell index	dimension N($\mathcal{L}_N$)	cardinality q($\mathcal{L}_N$)	$\Theta(\mathcal{L}_k)$
D_4	1	4	24	45°
D_4	1+2	4	48	32°
E_8	1	8	240	45°
E_8	2	8	2160	45°
E_8	3	8	6720	35°
E_8	1+2	8	2400	32°
E_8	1+2+3	8	9120	29°
Λ_{16} or L16	16	2	4320	55°
Λ_{24} or L24	24	1	196560	45°

Figure C.5 shows the curves of rate versus α when equation (C.16) is used with the regular lattices D_4 shell 1, D_4 shell 1+2, E_8 shell 1+2+3, Λ_{16} shell 2 and Λ_{24} shell 1 and for a distortion equal to 10^{-4}. From this figure we can observe that for a given distortion there is an optimum α for each regular lattice. In order to obtain a better analysis of the rate spent for different distortions, we show in figure C.6 the rate surfaces with their contour plots for α values in the range $[0.0001, 0.9999]$ and distortions between 10^{-2} and 10^{-12}, for these regular lattices (these contour plots can also be seen in figure C.7). From these figures we can see that each regular lattice used has a different optimum α, that is dependent on the distortion. Despite this distortion dependence, the optimum α values do not present significant changes with the distortion. This is shown in figure C.8. This implies that we can use an average optimum α for each regular lattice for all distortions. These average optimum α values can be found in table C.2.

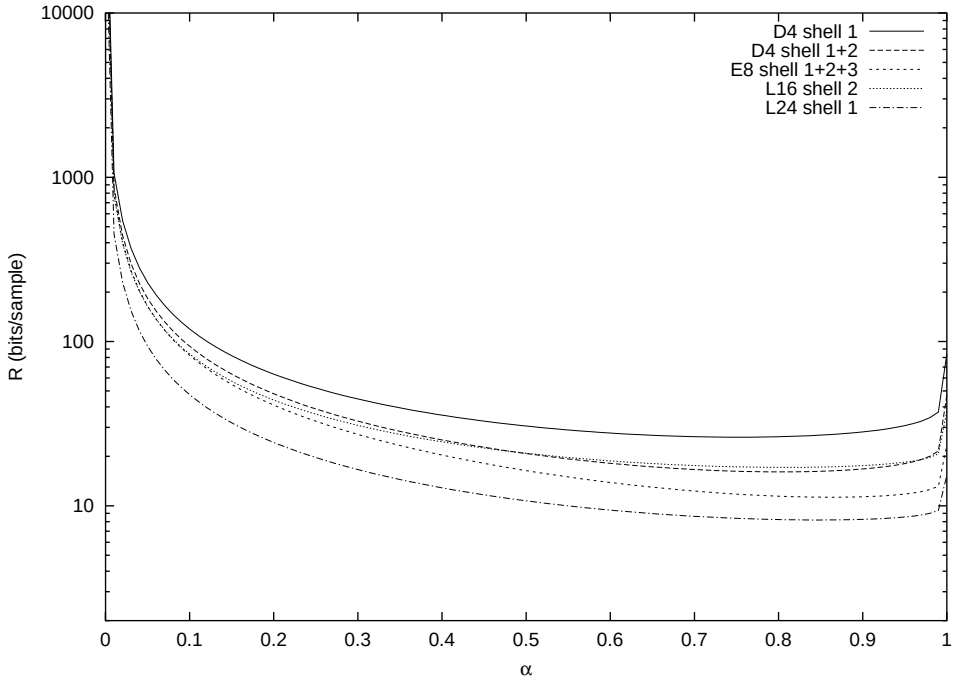


Figure C.5: Variation of the rate spent with the α parameter for regular lattices.

Table C.2: The average optimum α values for regular lattices.

Dictionary ($\mathcal{L}_N$)	shell index	average optimum α
D_4	1	0.75
D_4	1+2	0.81
E_8	1	0.80
E_8	2	0.83
E_8	3	0.85
E_8	1+2	0.85
E_8	1+2+3	0.87
Λ_{16}	2	0.82
Λ_{24}	1	0.86

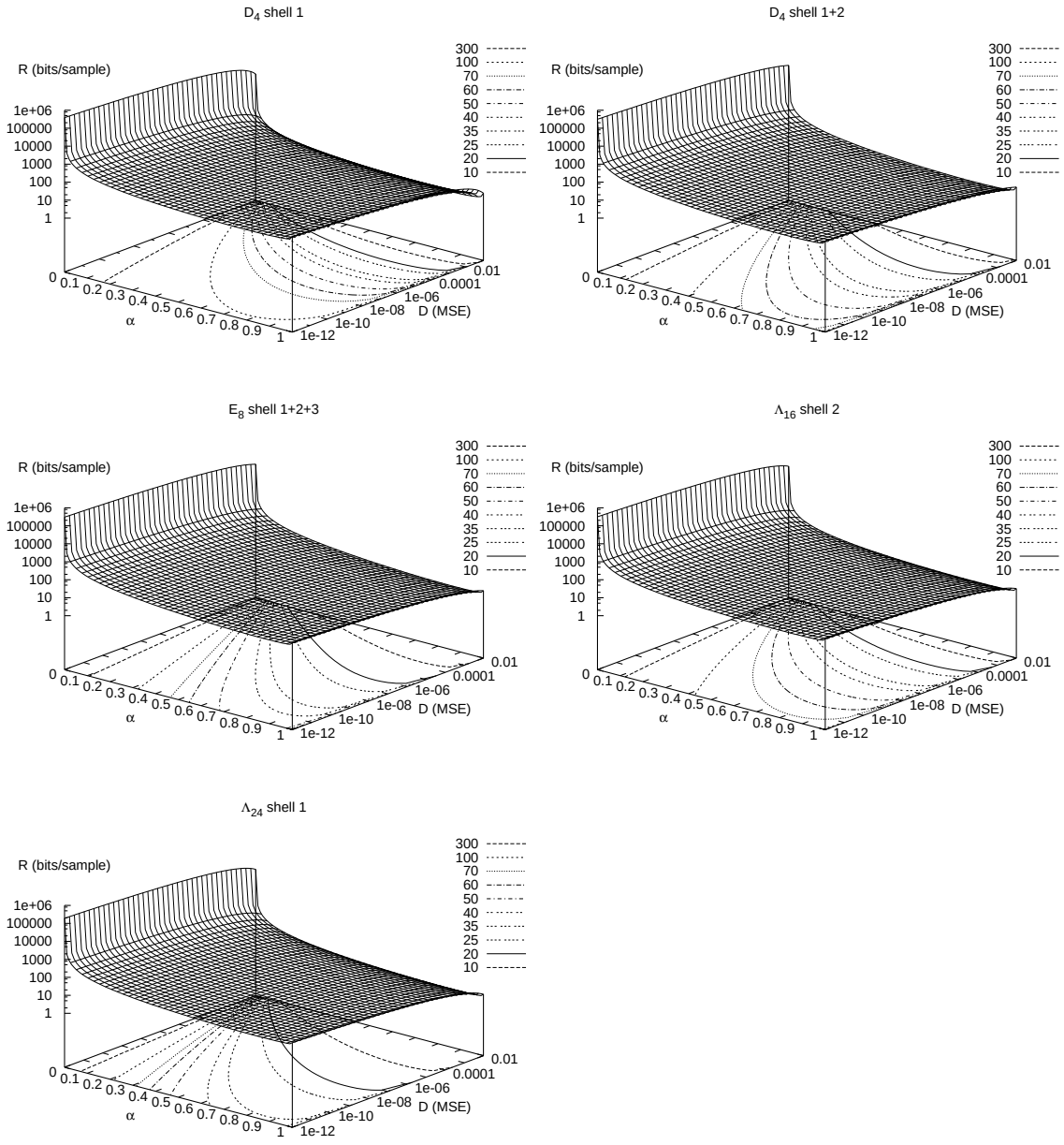


Figure C.6: The rate spent and its contour plots surface for regular lattices.

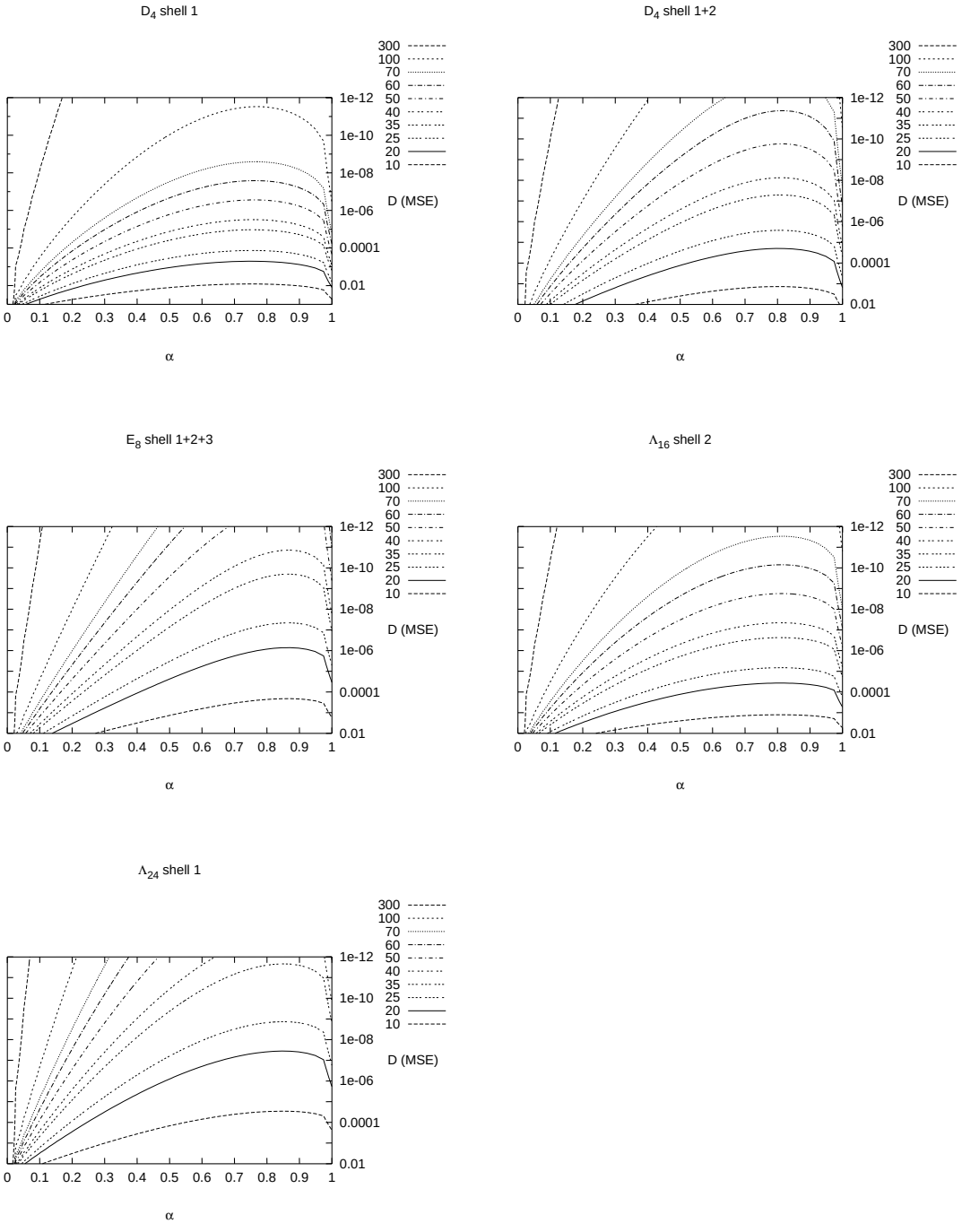


Figure C.7: Contour Plots of the rate spent surfaces for regular lattices.

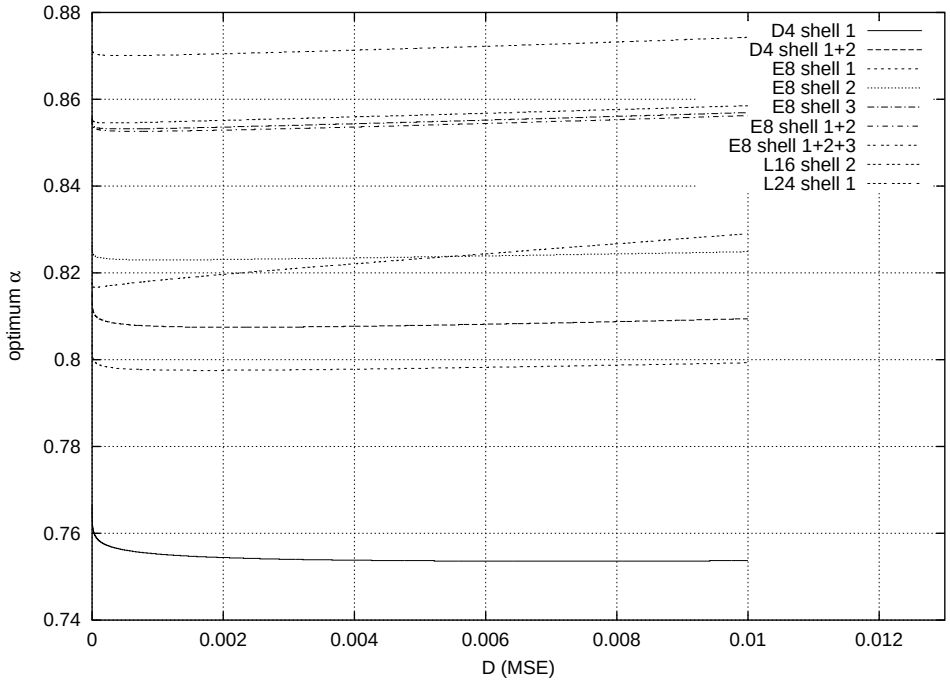


Figure C.8: Variation of optimum α versus distortion for regular lattices.

Figure C.9 depicts the R-D curves for different shells of the regular lattice E_8. We can see from this figure that, for dictionaries with the same dimension, whenever $\Theta(\mathcal{L}_i)$ is decreased the rate$\times$distortion trade-off tends to get better. Note that the optimum α is used for each dictionary. We can also see this by analyzing the R-D curves of D_4 dictionaries in figure C.10.

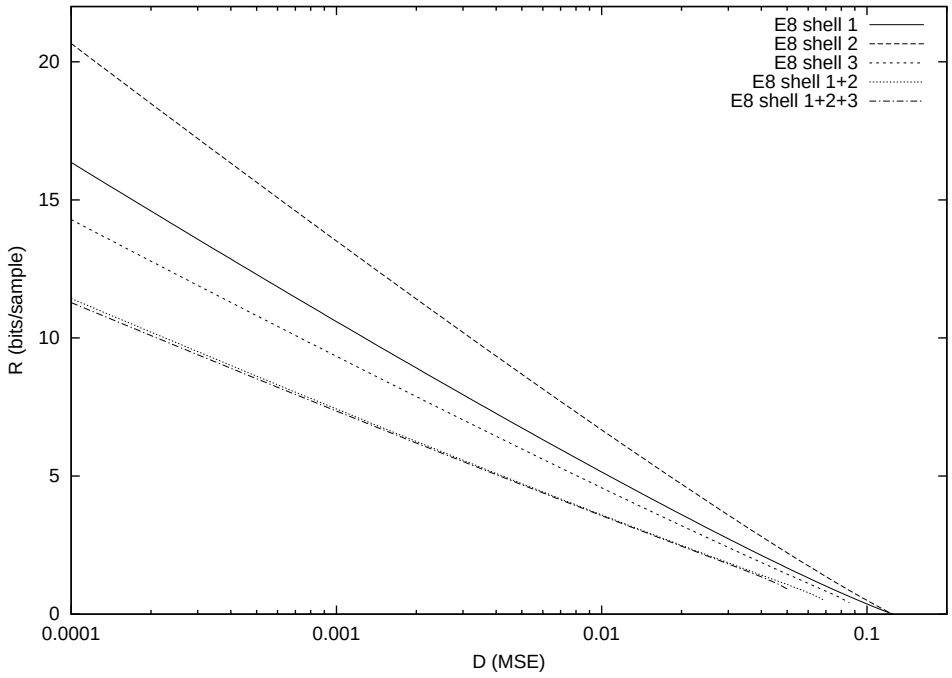


Figure C.9: R-D curves for E_8 regular lattices.

In figure C.10 we can also see the R-D curves for different regular lattices. We can note that in some cases even if the dimension N of the dictionaries is increased it does not lead to a better R-D trade-off (see curves Λ_{16} and E_8 shell 1+2+3). Also, the dictionaries performances depend on the particular distortion level (the R-D curve of Λ_{16} shell 2 crosses the curves of D_4 shell 1+2 and E_8 shell 1+2+3). From this we can see that performance is heavily dependent on the lattice used, and has no noticeable relation to the dimension N.

C.3.2 Orthogonal Dictionary

In section C.3.1 we have used the regular lattices to perform the Rate-Distortion analysis of the MPGBP algorithm. In spite of being good dictionaries, regular lattices do not give us a complete insight about the R-D behaviour of the MPGBP algorithm.

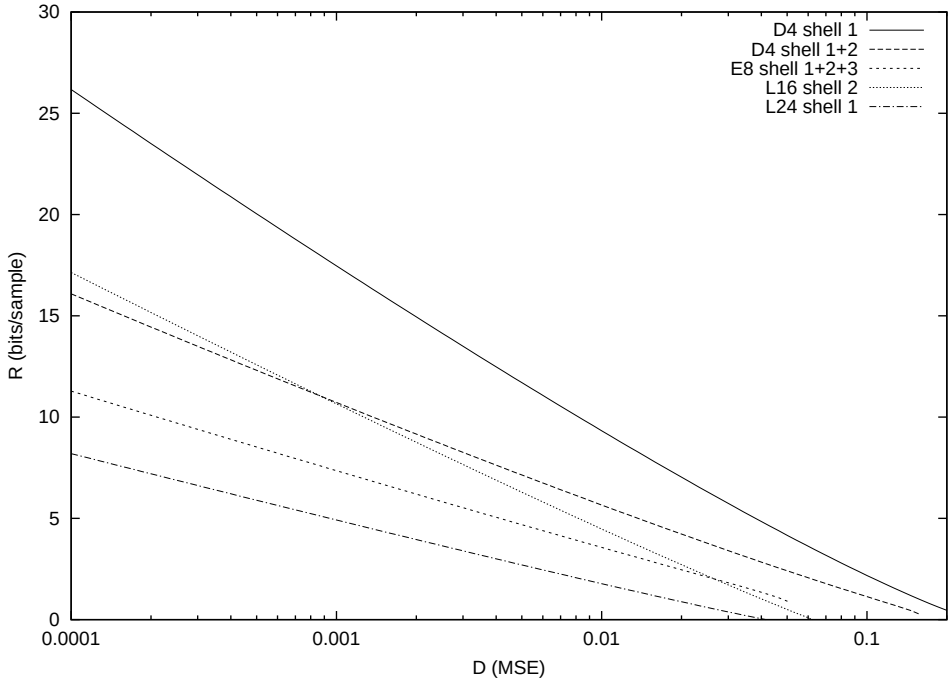


Figure C.10: R-D curves for different regular lattices.

This is because we do not know the relations among $\Theta(\mathcal{C})$, the dimension $N(\mathcal{C})$ and the cardinality $q(\mathcal{C})$ when these lattices are used. Therefore we can not perform a thorough analysis of how the variation of these parameters influences the performance of a specific regular lattice.

In order to be possible to analyze the variation of the $\Theta(\mathcal{C})$, $N(\mathcal{C})$ and $q(\mathcal{C})$, in the first place we define the simple orthogonal dictionary $\mathcal{U}_{\text{ort}}$ as

$$\mathcal{U}_{\text{ort}} = \{1,0,\cdots,0\} \cup \{0,1,0\cdots,0\} \cup \{0,\cdots,1,0\} \cup \{0,\cdots,0,1\} \tag{C.24}$$

Then, we use this dictionary $\mathcal{U}_{\text{ort}}$ to form the dictionary $\mathcal{U}$ that will be used in our analysis. The dictionary $\mathcal{U}$ is formed by

$$\mathcal{U} = \mathcal{U}_{\text{ort}} \cup -\mathcal{U}_{\text{ort}} \tag{C.25}$$

$$= \{\pm 1,0,\cdots,0\} \cup \{0,\pm 1,0\cdots,0\} \cup \{0,\cdots,\pm 1,0\} \cup \{0,\cdots,0,\pm 1\} \tag{C.26}$$

In the case of the R-D analysis of the MPGBP algorithm, the advantage of this dictionary over the regular lattices, is that the relations among the dimension $N(\mathcal{U})$ can easily be obtained, the cardinality $q(\mathcal{U})$ and $\Theta(\mathcal{U})$. In fact, for this dictionary we have a cardinality $q(N) = 2N$ and a $\Theta(N) = \arccos\left(\frac{1}{\sqrt{N}}\right)$ (see appendix I).

Using these relations we can analyze how the variation of the dimension N influences the R-D performance of the MPGBP algorithm. First of all, we show in figure C.11 the rate surface for this dictionary for dimension $N = 10^2$. We can see from this figure that its behaviour is similar to the one obtained with the regular lattices shown in figure C.6, that is, there is an optimum α for each distortion. Figures C.12 and C.13 show the curves of the optimum α versus distortion for the dimensions $N = 10^2$ and $N = 10^4$, respectively. We can note that the optimum α decreases whenever the distortion increases. However, the variation of this optimum α is too small and does not interfere significantly in the rate spent. Then, similar to the case of regular latices, we can use, for each dimension an average optimum α for all distortions.

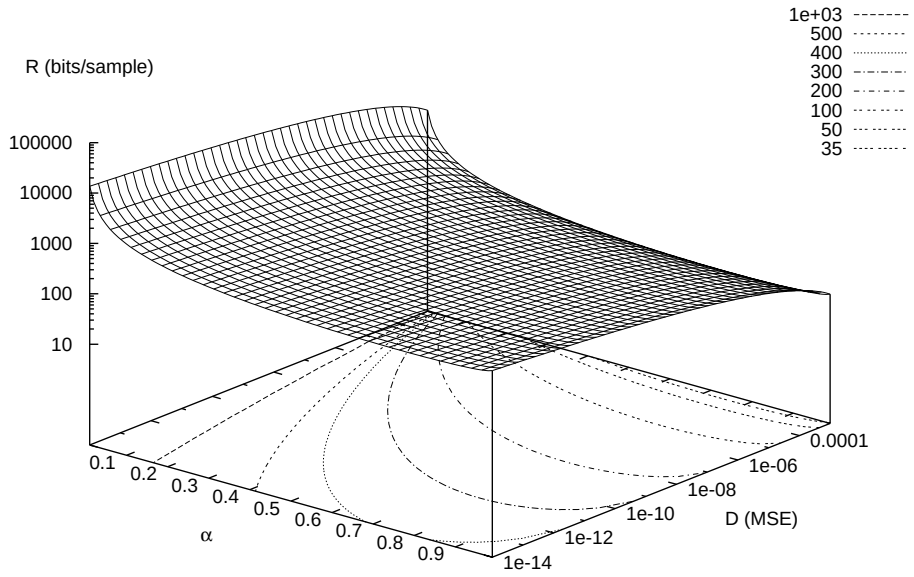


Figure C.11: The rate surface and its contour plots for the dictionary $\mathcal{U}$ with $N = 10^2$.

In order to observe the behaviour of the rate spent with the variation of both the dimension of this dictionary and the α values when the MPGBP algorithm is used, we show in figures C.14 and C.15 the surfaces of the rate and its contour plots for the distortions $D = 10^{-12}$ and $D = 10^{-3}$, respectively. In these figures, the contour plots show us that these surfaces do not have a minimum, but a saddle point instead. This can be easily observed comparing figures C.16 and C.17, where we plot the rate versus α and rate versus dimension, respectively, for a distortion $D = 10^{-10}$. In figure C.17 we plot the rate $\times$ dimension for several α values. The curve labeled α_{opt}, in this figure, was obtained

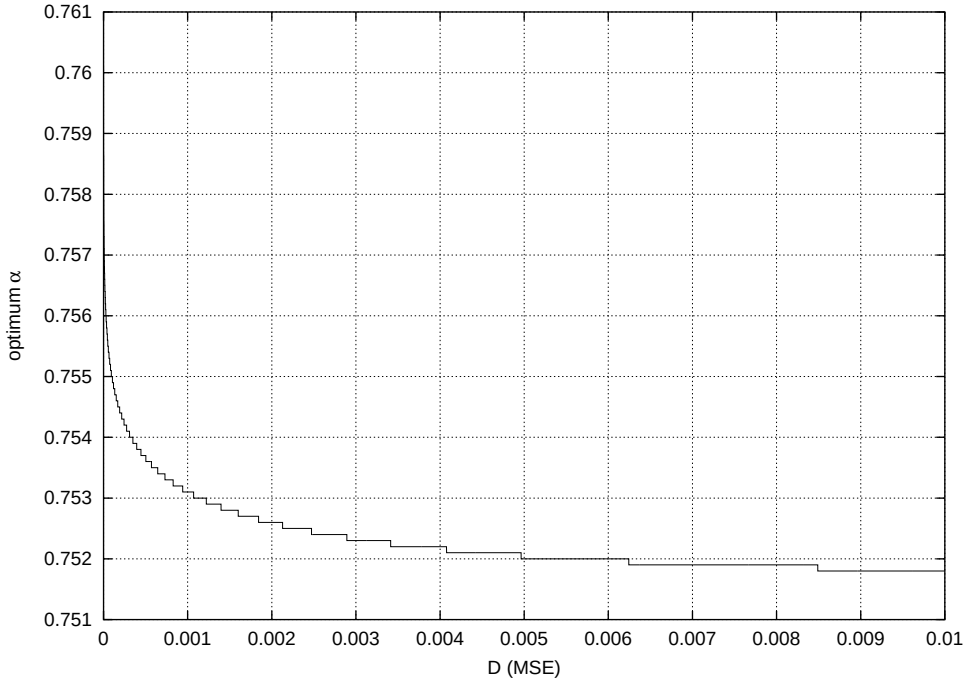


Figure C.12: Optimum α versus distortion for the dictionary $\mathcal{U}$ with $N = 10^3$.

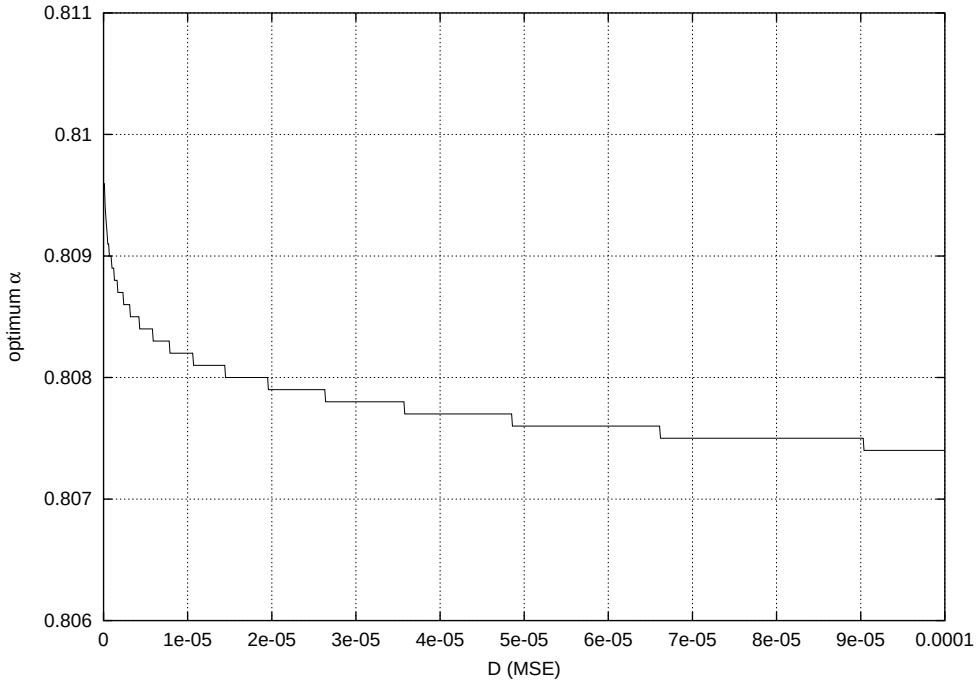


Figure C.13: Optimum α versus distortion for the dictionary $\mathcal{U}$ with $N = 10^4$.

using the optimum α for each dimension. From figure C.16 we can see that, for a given dimension, there is an α that leads to a minimal rate. However, figure C.17 shows that for each α, there is a dimension that leads to a maximum rate; this generates the saddle point shown in figures C.14 and C.15. Also, for this dictionary the best dimension to be used in practice, is $N = 2$. This is so because only extremely large dimensions (e.g. $N > 10^{10}$), give a rate less than the one for the dimension $N = 2$.

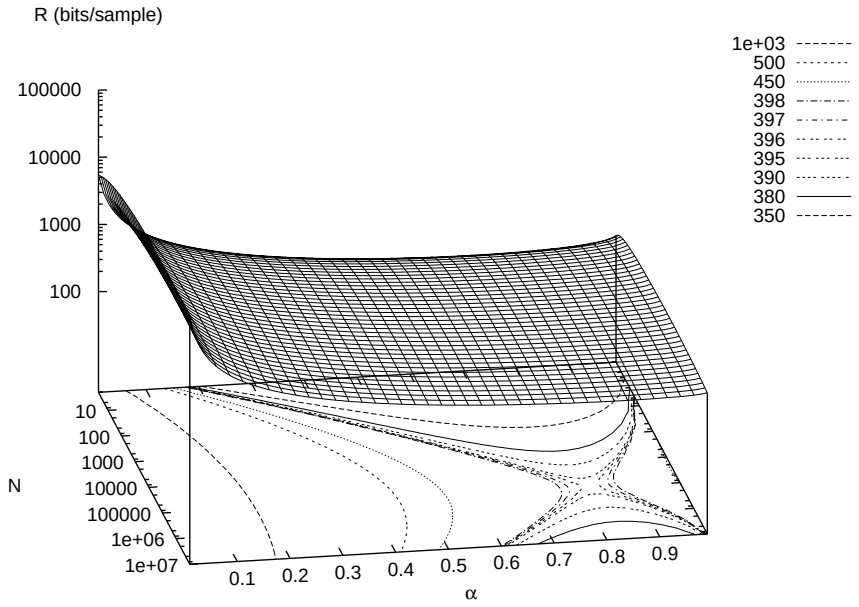


Figure C.14: Surface and contour plots of the rate spent for the distortion $D = 10^{-12}$ using the dictionary $\mathcal{U}$.

In figure C.18 we plot the optimum α values versus dimension for distortions $D = 10^{-6}$, $D = 10^{-9}$ and $D = 10^{-12}$. From this figure we can observe that there is an optimum α for each dimension N, and that the optimum α tends to increase with the dimension.

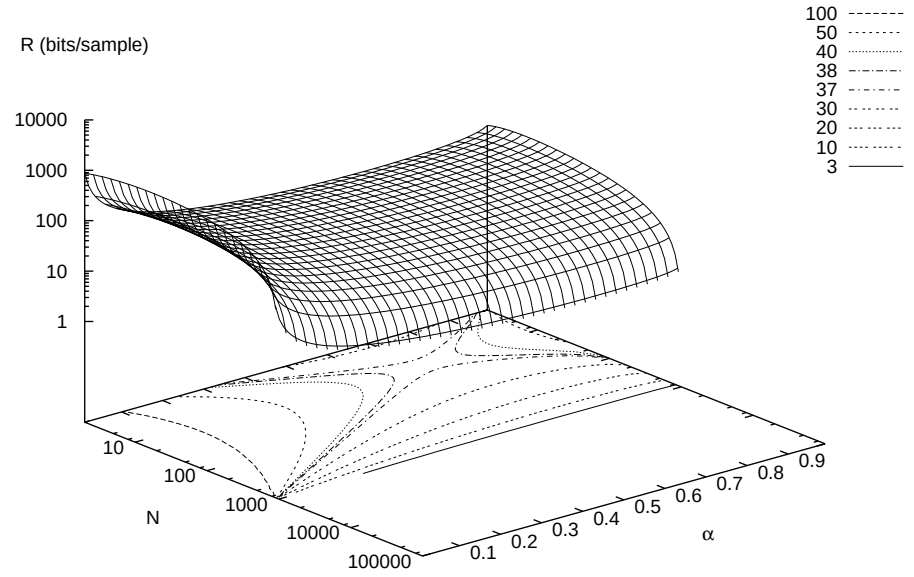


Figure C.15: Surface and contour plots of the rate spent for the distortion $D = 10^{-3}$ for the dictionary $\mathcal{U}$.

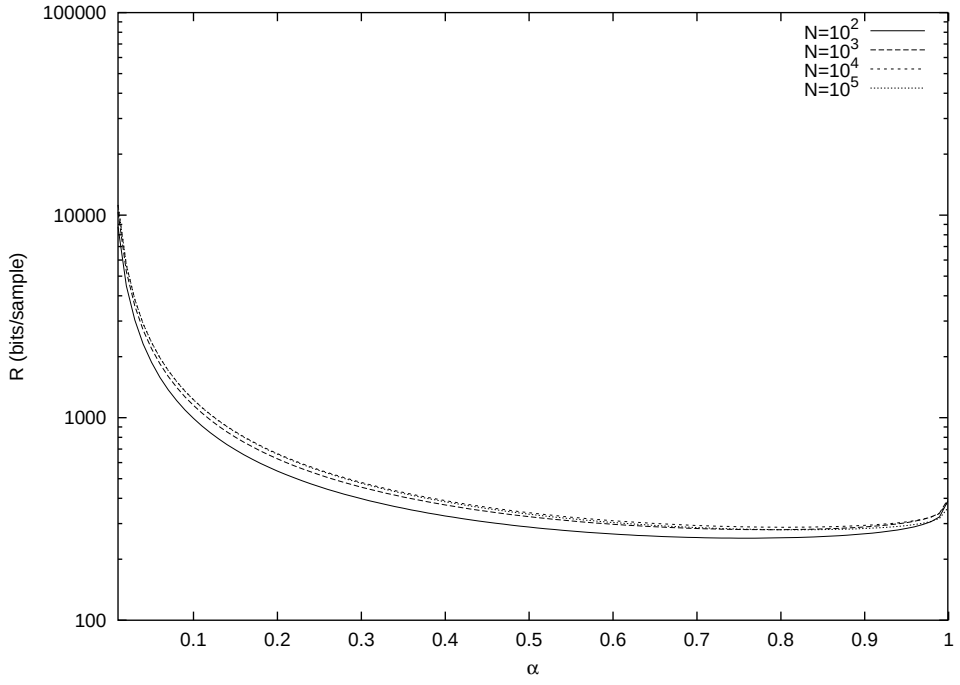


Figure C.16: Rate spent versus α for the dictionary $\mathcal{U}$.

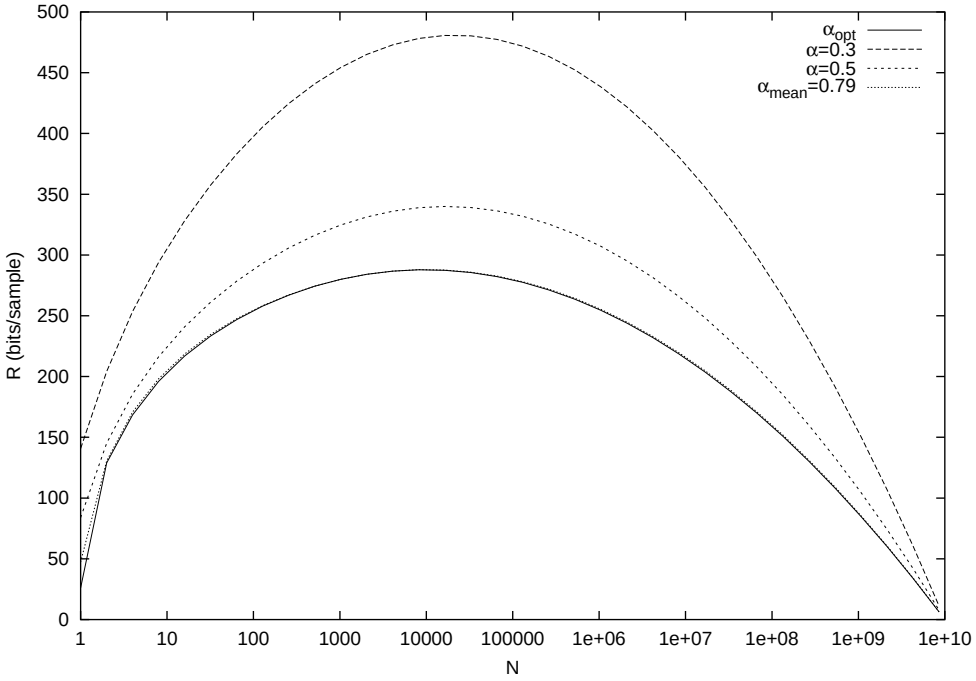


Figure C.17: Rate spent versus dimension for the dictionary $\mathcal{U}$ using different α values.

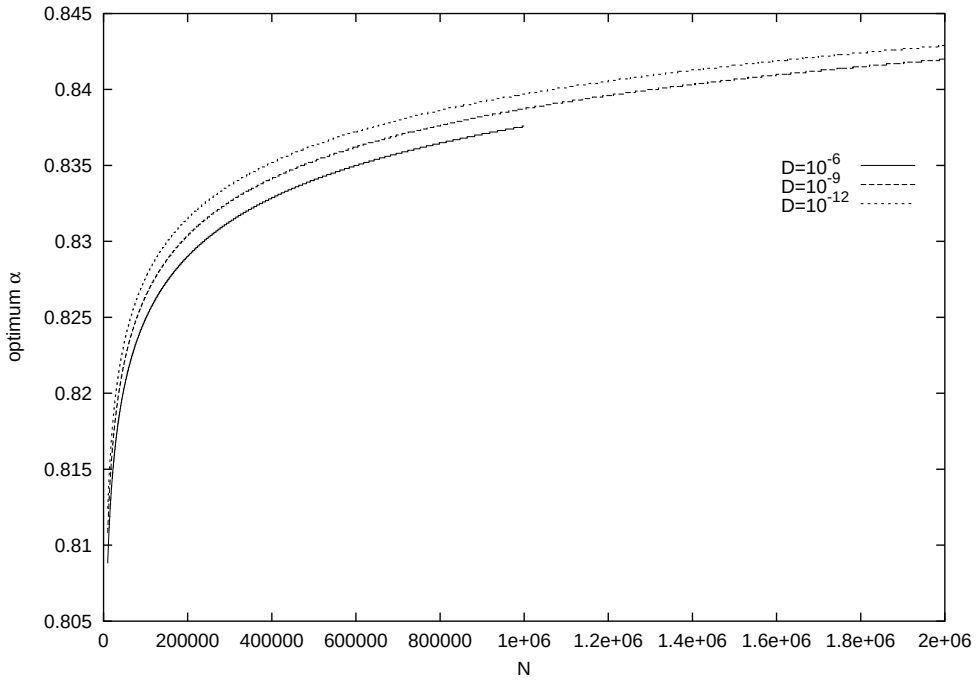


Figure C.18: Plot of the optimum α versus dimension for the dictionary $\mathcal{U}$ for some distortions.

C.4 Conclusions

In this appendix we have proposed a novel algorithm for performing matching pursuits decomposition. Instead of generating a sequence of pairs comprising atoms indexes and corresponding coefficients, as in the classical MP algorithm, it generates just a sequence of indexes. These indexes can be arranged as generalized bit-planes. The proposed algorithm has the advantage of obviating the need for setting up arbitrary trade-offs between number of atoms used and coefficient quantization. We have shown that the proposed algorithm corresponds to a generalization of the usual decomposition on a dictionary or basis followed by uniform scalar quantization. Also, we have proved a theorem setting a bound for the distortion obtainable for a decomposition in generalized bit-planes using a given number of atoms.

We have analyzed the Rate-Distortion performance of the MPGBP algorithm both for regular lattices and for a dictionary formed by the union of an orthogonal dictionary and its symmetric in relation to the origin. The results show that the reduction of the $\Theta(\mathcal{C})$ value leads to a better rate$\times$distortion trade-off, and consequently, a reduction of the bit-rate spent in the encoding process. These results encourage us to do further investigation on dictionaries with good trade-offs between the cardinality $\mathrm{q}(\mathcal{C})$ and $\Theta(\mathcal{C})$.

In the next appendix, the behaviour of this algorithm is evaluated in video encoding framework, by replacing the matching pursuits algorithm in Neff and Zakhor's video coder [3].

Apêndice D

Video Coding Applications

In this appendix we use the Matching Pursuits Generalized Bit-Planes Algorithm proposed in appendix C to code video sequences using an overcomplete dictionary formed by Gabor basis functions in the same framework of Neff and Zakhor's matching pursuits video encoder [3]. In section D.1 we describe the main characteristics of the Neff and Zakhor's matching pursuits video encoder, what includes the dictionary set used and the frame encoding procedure. Section D.2 shows the changes that must be done so as to use the MPGBP algorithm in the framework described in section D.1, as well as the experimental results. The conclusions are shown in section D.3.

D.1 An Efficient Video Coder Using The Matching Pursuits Algorithm

The most popular video coding methods nowadays are those based on block transforms, where the redundancy inside these blocks are explored. In this context the algorithms based on the Discrete Cosine Transform (DCT) present the best performances. In spite of being fast and efficient, the block-based DCT methods introduce undesirable blocking and ringing artifacts, especially at low bit-rates. In the last years, many non-block-based methods, like those using the wavelet transform [27, 28, 29] and matching pursuits [2, 3, 10, 5, 8], have emerged in order to reduce these undesirable effects.

In this section we describe an efficient video encoder that is based on the original work proposed by Mallat in [2] (the matching pursuits algorithm) and that uses matching pursuits. The matching pursuits video encoder described here was proposed by Neff and

Zakhor in [3] for low bit-rates. This video encoder uses the same framework (motion compensation/estimation, bit-stream, Huffman tables, etc) as in MPEG-4 [30], except for the encoding process of the residual images (frame differences). The main difference is that in MPEG-4 the residual image is divided in 8×8 blocks where the DCT is applied, while in [3] a greedy matching technique is used to decompose the residual image into a weighted summation of Gabor functions. Instead of coding the DCT coefficients, their positions in the 8×8 block and the quantization step size, the matching pursuits video encoder proposed in [3], encodes the indexes of the Gabor functions and their positions in the residual image (named atom) as well as the inner products obtained by the projection of these Gabor functions on the residual image. It is important to note that the matching pursuits algorithm is applied in the whole residual image, and therefore it is not block-based. In figures D.1 and D.2 we can see the block diagrams of the matching pursuits video coder and decoder, respectively.

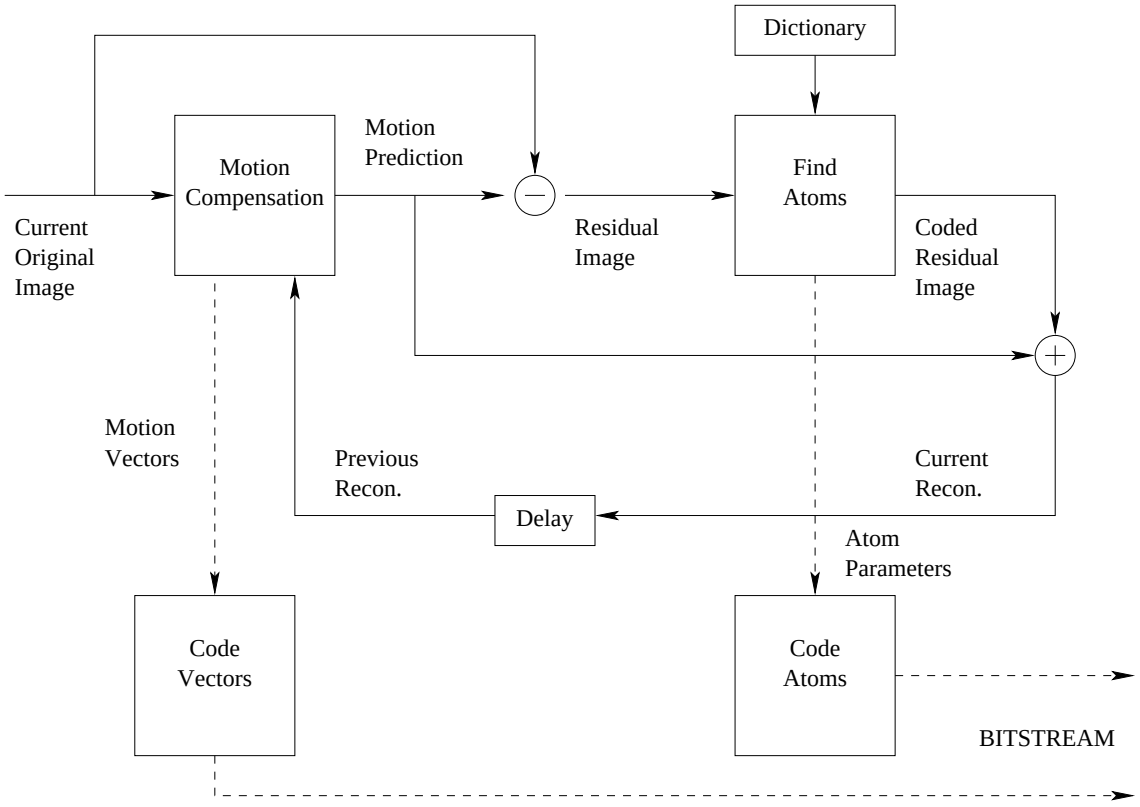


Figure D.1: Block diagram of the matching pursuits video coder.

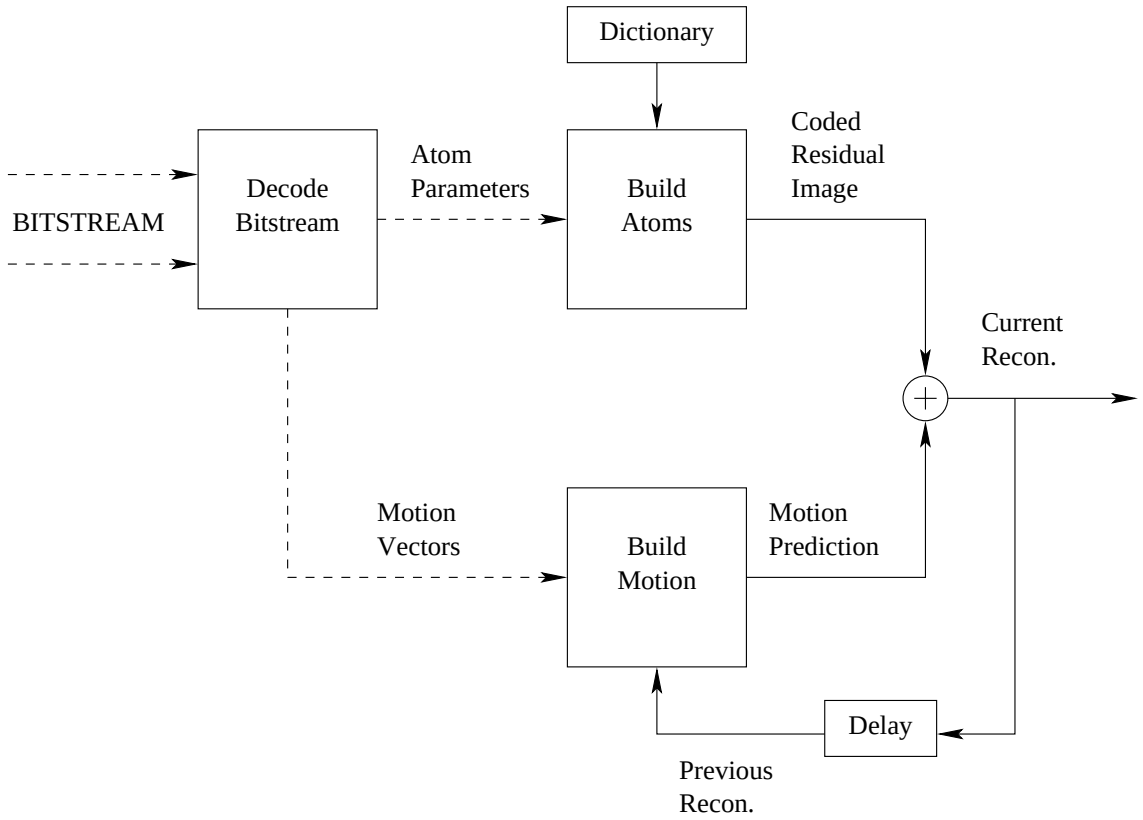


Figure D.2: Block diagram of the matching pursuits video decoder.

D.1.1 Dictionary Set

In appendix B it was described the matching pursuits algorithm and its main characteristics for a one-dimensional (1-D) dictionary. Since we will use this algorithm to encode frame differences in a discrete two-dimensional (2-D) space, the dictionary used must consist of discrete 2-D basis functions. Also, since the matching pursuits algorithm requires exhaustive inner-product searches in this dictionary, this must be limited to a reasonable number of basis elements, each of which with a finite spatial extent in order to avoid excessive delays in the encoding process. In order to reduce the search time even further, it was required that each basis function be separable [31].

The dictionary used consists of an overcomplete collection of 2-D separable Gabor functions. They are scaled, shifted and modulated versions of a Gaussian window function. Let's define a prototype window $g(t)$ as

$$g(t) = \sqrt[4]{2}e^{-\pi t^2} \tag{D.1}$$

We can write 1-D discrete Gabor functions of length L as

$$g_{\vec{\alpha}}(i) = K_{\vec{\alpha}} \cdot g\left(\frac{i - \frac{L}{2} + 1}{s}\right) \cdot \cos\left(\frac{2\pi\xi(i - \frac{L}{2} + 1)}{L} + \phi\right) \tag{D.2}$$

where $\vec{\alpha} = (s, \xi, \phi)$ is a triple consisting of a positive scale, a modulation frequency and a phase shift, respectively, $i \in \{0, 1, \cdots, L-1\}$, and $K_{\vec{\alpha}}$ is a normalizing constant, chosen such that $\|g_{\vec{\alpha}}\| = 1$. We define $\mathcal{B}$ as the set of all triples (s, ξ, ϕ).

Now, we can use the combination of the 1-D Gabor functions defined in equation (D.2) to construct a set of 2-D separable Gabor functions. This 2-D dictionary is formed as follows

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

where $i, j \in \{0, 1, \cdots, L-1\}$ and $\vec{\alpha}, \vec{\beta} \in \mathcal{B}$.

The above notation assumes that all the 1-D functions have the same size L. However, in practice, each 1-D function may have its own associated size, $L_{\vec{\alpha}}$. It was done in order to increase the search speed, since smaller basis functions are no longer padded with zeros to a fixed size. This is equivalent to the resulting 2-D functions having an arbitrary rectangular extent.

Since the use of a set $\mathcal{B}$ consisting of all triples $\vec{\alpha}$ is computationally prohibitive, we must choose among them a limited subset $\mathcal{B}'$. The reduced dictionary set $\mathcal{B}'$ was constructed in the following manner: a large subset of $\mathcal{B}$ containing evenly spaced instances of the three parameters (s, ξ, ϕ) was used to define a 2-D dictionary. A set of motion residual images from a training sequence was decomposed on this dictionary using the matching pursuits algorithm. The twenty triples which were most often selected by the matching pursuits algorithm were retained in the reduced set, denoted $\mathcal{B}'$. This set of triples can be found in table D.1 [3, 32].

D.1.2 Encoding Procedure

The matching pursuits video encoder proposed by Neff and Zakhor in [3] uses the same framework as the MPEG4 video encoder specified in [30]. In this section, we describe the main differences between the MP and the MPEG4 video encoders. The most important difference is in the encoding process of the residual images. Instead of using the DCT to reduce the redundancies in the residual images, we use an overcomplete

Table D.1: Dictionary triples used, $\vec{\alpha} = (s, \xi, \phi)$.

Scale(s)	Freq(ξ)	Phase (ϕ)	Size ($L_{\vec{\alpha}}$)
1	0	0	1
3	0	0	5
5	0	0	9
7	0	0	11
9	0	0	15
12	0	0	21
14	0	0	23
17	0	0	29
20	0	0	35
1.4	1	$\pi/2$	3
5	1	$\pi/2$	9
12	1	$\pi/2$	21
16	1	$\pi/2$	27
20	1	$\pi/2$	35
4	2	0	7
4	3	0	7
8	3	0	13
4	4	0	5
4	2	$\pi/4$	7
4	4	$\pi/4$	7

dictionary formed by Gabor functions to do it. Initially, we find the Gabor function that gives the greatest energy reduction in the residual image. The choice of this function is done by choosing the largest inner product p_n obtained from the projection of all Gabor functions of the dictionary (see section D.1.1) on the residual image. After this, p_n is quantized with a fixed step size QP, and the chosen Gabor function (weighted by the quantized inner product) is subtracted from the residual image. Then, we encode the index of this Gabor function and its position in the residual image (named atom) as well as the quantized inner product using Huffman tables. This process is repeated until we have a particular number of atoms coded. The following sections show the details of this process.

D.1.2.1 Finding Atoms

In order to choose the Gabor function that gives the largest energy reduction on the residual image, we have to examine each Gabor function of the dictionary at all possible integer-pixel locations in the image and compute all of the resulting inner products. However, since a full search using these functions over the entire residual image requires a large amount of computing time, one has to make some assumptions about the residual image to be coded. Specifically, we assume that the image is sparse, and contains pockets of energy at locations where the motion prediction model was inadequate. If this is true, we can "pre-scan"the image for high-energy pockets. The location of such pockets can be used as an initial estimate for the inner-product search. The residual image to be coded is first divided into blocks (16×16 for the luminance component and 8×8 for the two chrominance components), and the sum of the squares of all pixel intensities is computed for each block. The center of the block with the largest weighted energy value is adopted as an initial estimate for the inner product search. The dictionary is then exhaustively matched in a 16×16 window around the initial estimate.

Note that the use of a separable dictionary (see section D.1.1) allows for very large savings in computational time. For example, suppose we ignore separability. In this case the inner-product between the 2-D functions of the dictionary, defined in equation (D.3), and the frame difference requires $L_h L_v$ multiply-accumulate operations, where L_h and L_v are the associated sizes of the horizontal ($\vec{\alpha}$) and vertical ($\vec{\beta}$) functions, respectively. Finding a single atom requires the inner-product to be computed for all non-separable

functions at each location in $S \times S$ search window. The total number of operations, according to [3] is thus

$$T_{\text{non-sep}} = S^2 \sum_{h=0}^{M_{h'}-1} \sum_{v=0}^{M_{h'}-1} L_h L_v \tag{D.4}$$

where $M_{h'}$ is the number of 1-D functions used to generate the 2-D dictionary.

If we now assume a separable dictionary, one can make significant savings in their computational complexity, a single 2-D inner-product can be computed taking L_h vertical 1-D inner-products, each of length L_v, and then following with a single horizontal inner-product of length L_h as we show in figure D.3. The atom search requires us to exhaustively compute the inner-product at each location using all combinations of horizontal and vertical 1-D functions.

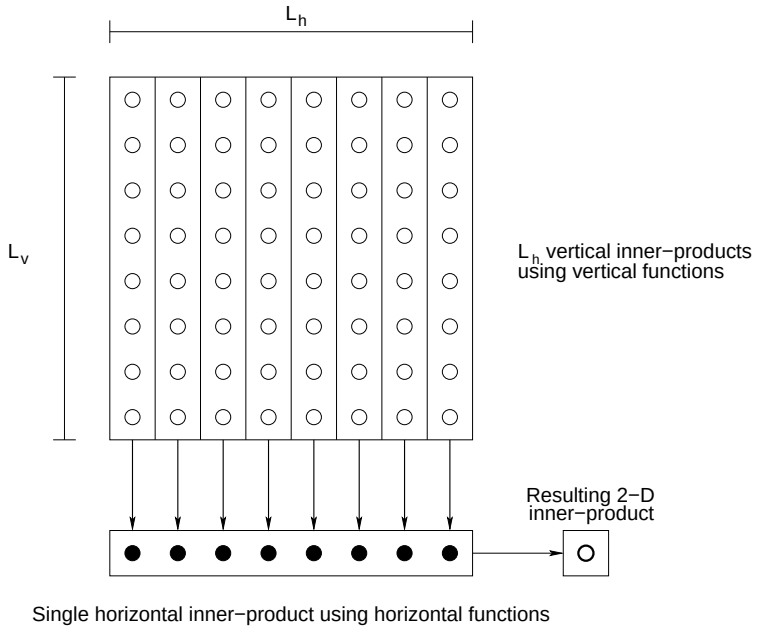


Figure D.3: Separable computation of a 2-D inner-product.

It is thus natural to pre-compute the necessary vertical 1-D inner-products with a particular vertical function, and then cycle though the horizontal 1-D inner-products using all possible horizontal functions. The fast algorithm to search an atom is described below (see figure D.4):

For each $v \in \{0 \cdots M_{b'-1}\}$

{

Compute the pre-filter matrix for the vertical 1-D function g_v

For each $h \in \{0 \cdots M_{b'-1}\}$ *and for each position in the* $S \times S$ *search window*

{

Filter using 1-D horizontal function g_h *to complete the 2-D inner-product computation*

}

}

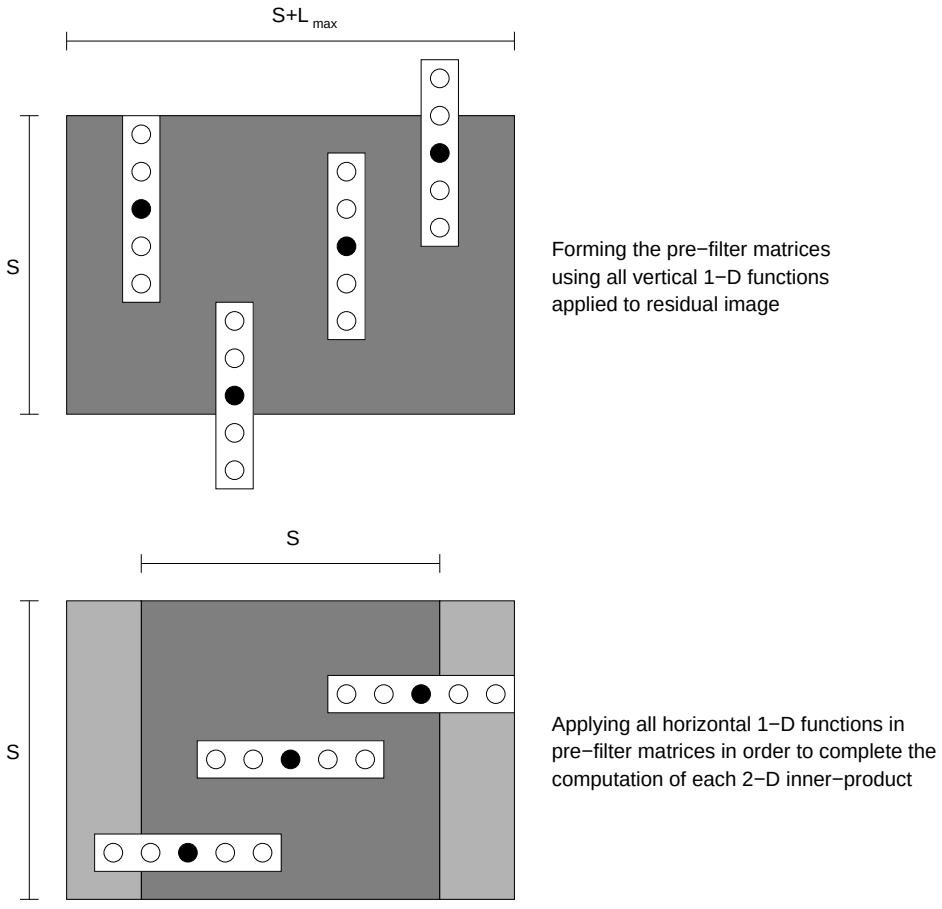


Figure D.4: Illustration of the separable inner-product search algorithm.

The operation count for finding a single atom is thus, according to [3]

$$T_{sep} = \sum_{v=0}^{M_{y'}-1} (L_v S(S + L_{max})) + S^2 \sum_{h=0}^{M_{y'}-1} L_h \qquad (D.5)$$

where L_{max} is the size of the largest 1-D basis function.

Using the parameters of table D.1, a search window of 16×16 ($S = 16$) and equations (D.4) and (D.5), we get 21.5 and 1.7 million multiply-accumulate operations for the non-separable and the separable 2-D functions, respectively.

Also, note that as the luminance blocks are bigger than the chrominance blocks, the largest energy measure from the latter ones is weighted by 4 before comparing with the largest energy block found in the luminance component.

D.1.2.2 Inner Product Quantization

After the choice of the Gabor function to be used to decompose the residual image, we need to quantize the inner product p_n. We used a uniform quantizer with dead zone with step size $QP = 32$. Although the procedure of using this step size for performing the quantization is good for low bit-rates, it can become a problem for high bit-rates. At high bit-rates, after coding a lot of atoms, the energy of the residual signal tends to be very small, precluding further refinement of the coded image. In order to avoid it, we shrink the dead zone, thus allowing smaller reconstructions levels. This dead zone shrinkage is performed by the addition of new bins during coding. We need to send a flag at the frame level to specify how many of the new bins were used. For three added bins ($3QP/32$, $3QP/16$ and $3QP/8$), the flag takes a value in set $\{0, 1, 2, 3\}$, for an overhead of 2 bits/frame (see figure D.5).

D.1.2.3 Coding Atoms Parameters

After the residual image is decomposed, the encoder is left with a list of the chosen atoms, which must be coded and transmitted to the receiver. Each atom is defined by four parameters which describe the position (x, y) and the indexes of the chosen Gabor functions. Also, we must encode the quantized inner products related to each atom. Since the atoms can appear in any position in the frame and it is desirable that the transmission errors are confined to a limited area, it was decided to encode all atoms that are in the same macroblock (16×16 pixels) together. The atoms of each macroblock are encoded

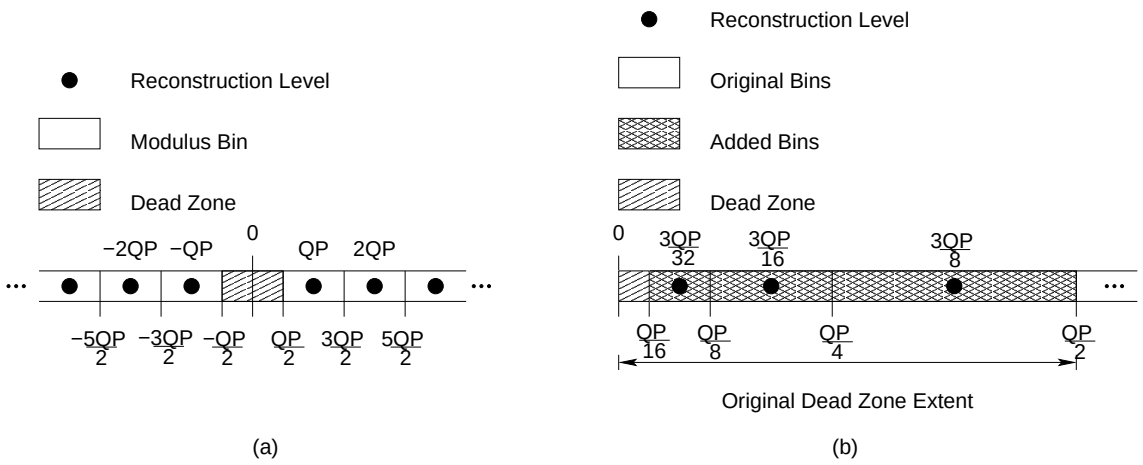


Figure D.5: (a) Simple uniform quantizer with fixed step size $QP = 32$. (b) Modification of the fixed quantizer design. Original dead zone has been divided into additional quantization bins.

according to the scanning pattern shown in figure D.6. Note that all the atoms parameters are coded using Huffman codes [32].

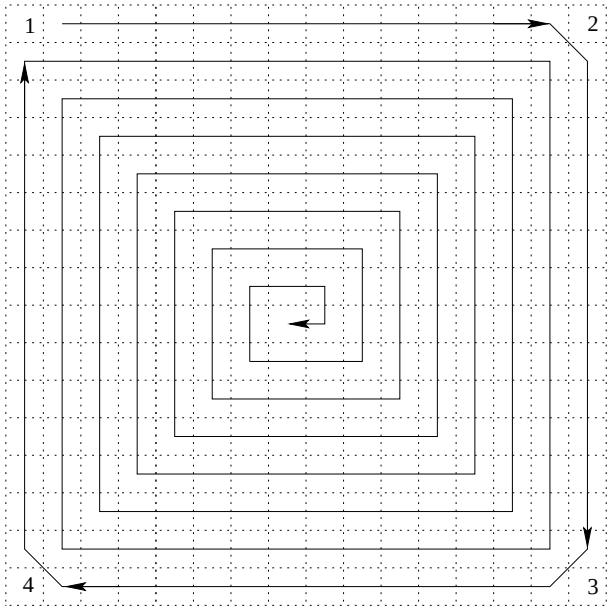


Figure D.6: Scan order for the atoms in a macroblock. First pixels 1, 2, 3 and 4 are coded.

D.1.2.4 Rate Control

The bit-rate allocation in the matching pursuits video encoder is performed using a simple solution: the bit-rate of the sequence is divided equally among all its frames. However, since the rate control is performed by controlling the number of atoms N used

for coding each residual image, this number has to be estimated for each frame according to

$$N = \frac{\text{Bit-rate allocated for the frame}}{\sharp \text{ of bits/atom in prev. frame}} \tag{D.6}$$

D.2 A Video Encode Using the MPGBP Algorithm

In this section the effectiveness of the MPGBP Algorithm will be evaluated by employing it in the framework of Neff and Zakhor's Matching Pursuits video encoder [3] described in section D.1. Essentially, the MPGBP Algorithm will replace the decomposition and quantization strategy employed in [3], using exactly the same dictionary $\mathcal{D}$, as well as the same atom encoding procedure. The atoms are obtained and encoded in the same way as the atoms in the Neff and Zakhor's matching pursuits video encoding. On the other hand, instead of encoding the value of the quantized inner product p_n (see equation (B.18)), the index k_m (see equation (C.5)) of the bit plane corresponding to the atom is encoded. It is important to note that the MPGBP algorithm generates just a sequence of indexes, that is, instead of the inner product of the MP algorithm, we transmit just an index k_m corresponding to exponent of α (see equation C.5) Also, since the indexes k_m encoded in the MPGBP algorithm and the quantized inner products p_n encoded in the MP algorithm have different characteristics, we propose a new scheme for coding these indexes using a modified adaptive arithmetic coder. Although the atom search in the MPGBP algorithm is performed in the same way as in the MP algorithm, this search in the MPGBP algorithm is influenced by the support region in which the inner products are computed. We will discuss this region next.

D.2.1 Modified Adaptive Arithmetic Coder

Unfortunately, the Huffman codes [32] used for encoding the quantized inner products p_n, in the MP algorithm, cannot be used to encode the indexes k_m of the bit planes in the MPGBP algorithm. This is due to the fact that these codes were designed to encode symbols, i.e., modulus of the quantized inner products. These symbols which have their values decreasing with the energy of the residual image. The bit planes indexes, differently from these symbols, tend to have their values increased as the energy of the

residual image decreases. Then, in order to encode the bit planes indexes, we propose a new coding scheme using a modified adaptive arithmetic coder. The modification to the adaptive arithmetic coder [106] is motivated because the maximum bit plane index is related with the level of refinement desired. In general, we do not know when coding starts the value of this maximum index. Also, this index is dependent on both the bit-rate and the parameter α used. In figures D.7 and D.8 we can see the histograms of the indexes k_m for different α values for the sequence Mother-and-daughter at rates 24kbps and 64kbps, respectively. From these figures we can see that the values of k_m are in general smaller than a few tens, except for α values very close to one.

Since we do not know a priori what is the maximum value that k_m can assume, we have to perform a slight modification to the arithmetic encoder in [106]. The initial number of possible indexes k_m is set to two (k_m=1 and k_m=2) plus a escape code. If we need to transmit k_m=3 we first transmit the escape code to indicate an increase in the number of symbols and then transmit the code for k_m=3. At this point the possible symbols are k_m=1,2,3 plus a escape code. The same process is repeated for each new value of k_m that is out of the current range. Also, note that, in the subsequent frame, the initial number of possible values of k_m is the same as the one at the end of the previous frame. Note that, since use α values much smaller than this upper limit, we will have to send just a few escape codes in most of cases. This does not interfere significantly in the rate spent. It was verified experimentally that, as long as the initial number of symbols is small enough, the quantity of escape codes does not significantly influence the algorithm performance.

D.2.2 Support Region for Inner Product Computation

Since the MPGBP algorithm assumes that the norm of the input signal is $\|\mathbf{x}\| \leq 1$, we need to compute, for each video frame, the largest norm of the macroblocks, X_{max}. It is important to note that, for each macroblock, as in [3], we search for the closest atom by centering every atom at every pixel of the macroblock. This implies that the atoms searched for in a macroblock B_i superpose the neighboring macroblocks. Then, effectively, it is as if the dimensions of the signal we are decomposing are not those of macroblock B_i, but also including the pixels of neighboring macroblocks superposed by the atoms used to decompose B_i. Referring to figure D.9, since the luminance mac-

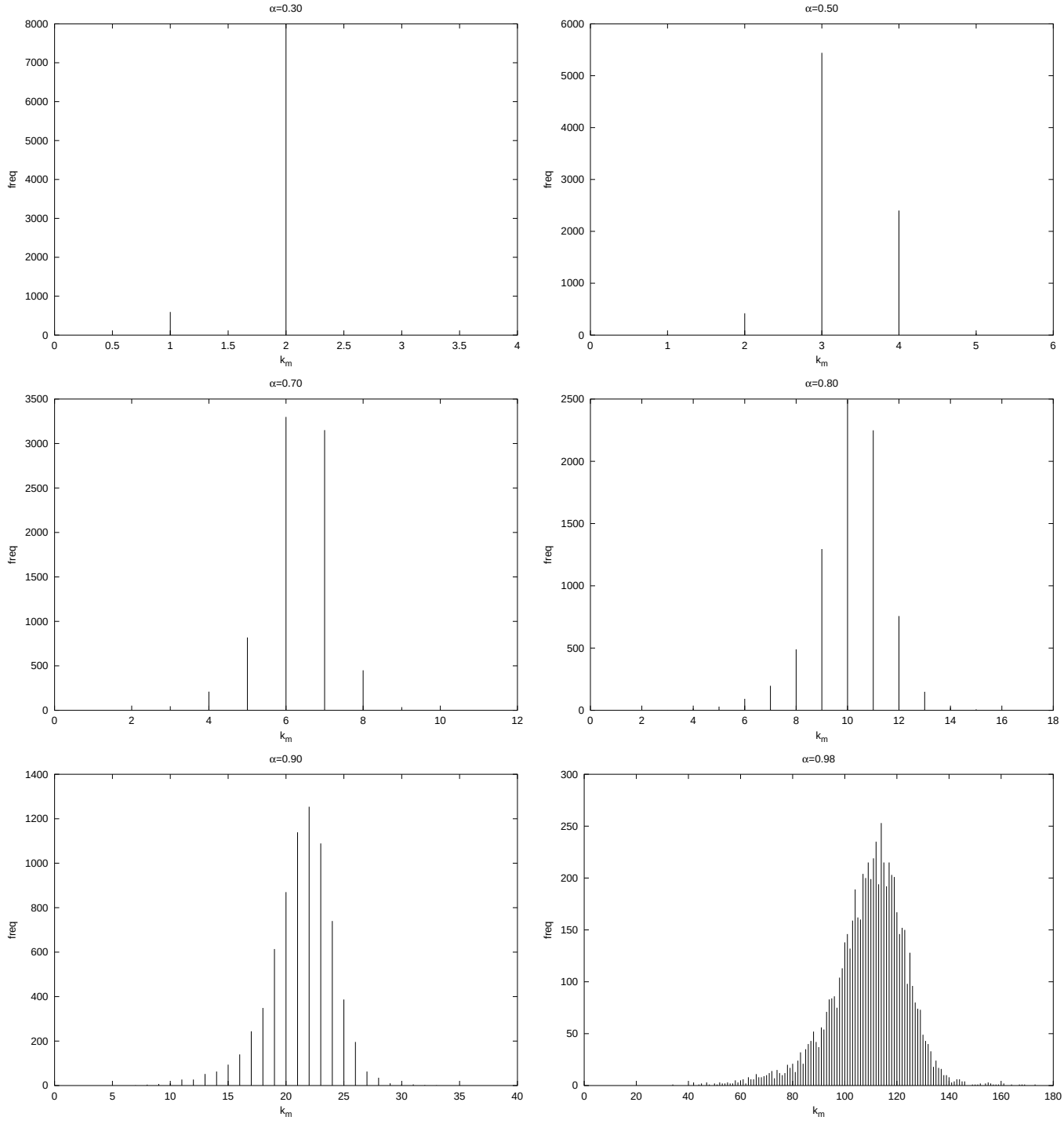


Figure D.7: Histograms of the k_m indexes for different alpha values of the Mother sequence at 24kbps.

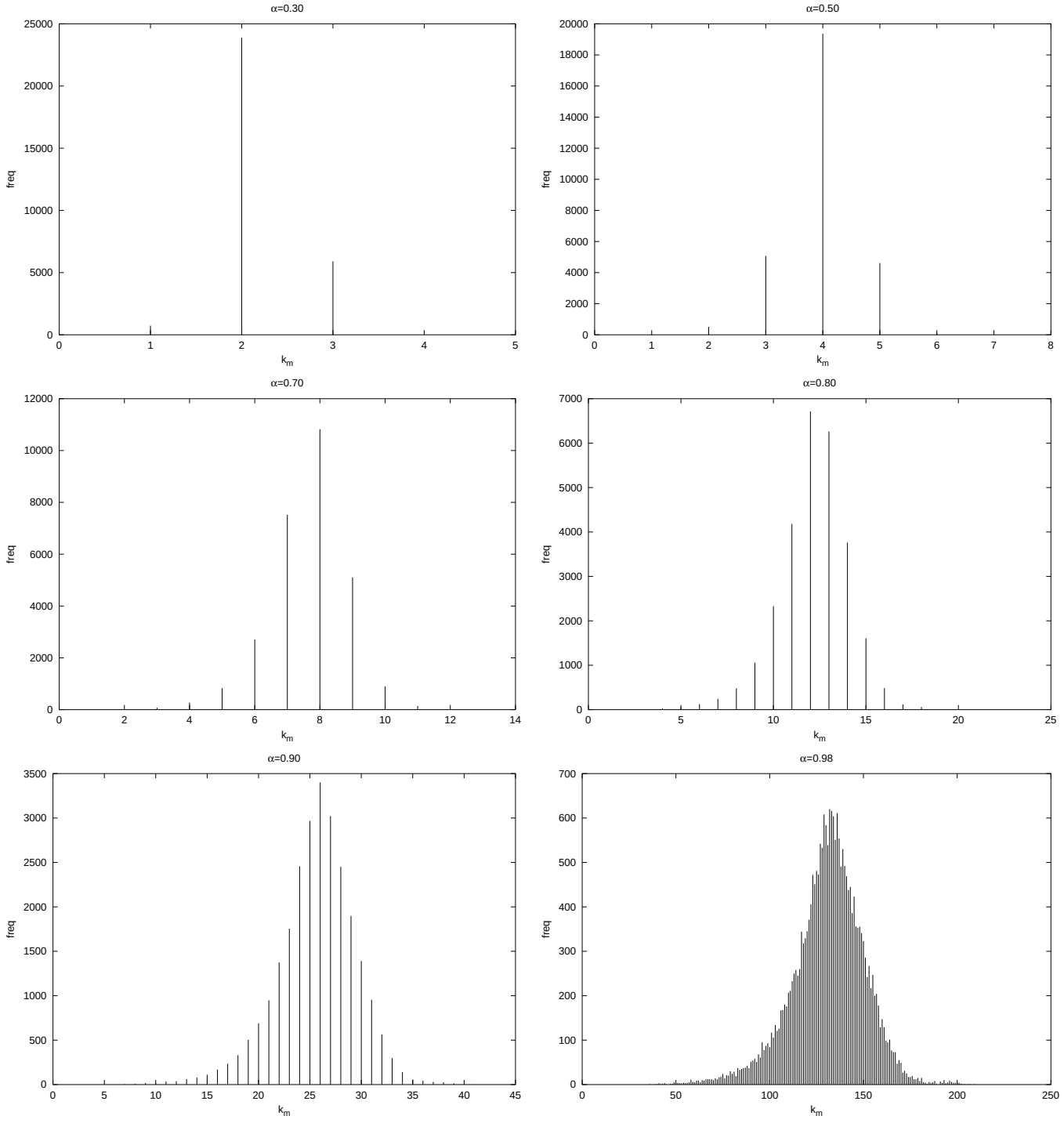


Figure D.8: Histogram of the k_m indexes for different alpha values of the Mother sequence at 64kbps.

roblocks are of dimensions 16×16, the value of X_{max} is computed for a region of $(15 + n_{max}) \times (15 + n_{max})$ around the macroblock ($n_{max} \times n_{max}$ is the support of the atom having largest support). In our case, $n_{max} = 35$, and X_{max} is computed considering 50×50 windows centered in every macroblock (see figure D.9).

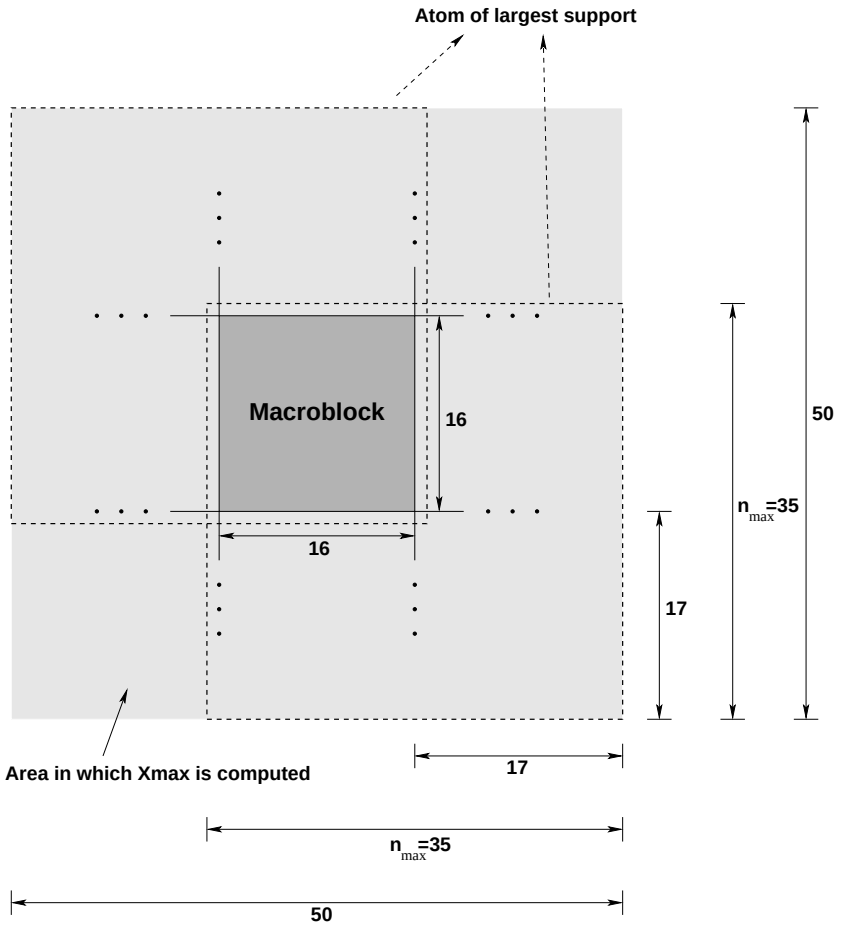


Figure D.9: Illustration of the area in which X_{max} is computed.

D.2.3 Experimental Results

We have coded the sequences Coast-guard, Container, Hall-monitor, Foreman, Mother-and-daughter and Silent-voice (see appendix K) with 300 QCIF frames at 30 frames/s, sub-sampled in time by factors of 4 (rates under 20kbps) and 3 (all other rates) to generate 7.5 frames/s and 10 frames/s, respectively. Coding was performed only on the luminance component for bit-rates ranging from 10 to 100kbps.

The value of α (see equation (C.5)) chosen at the beginning of the encoding process interferes with the number of vectors used to code each frame. Note that the closest

α is to 1, the more accurate is a representation of a number $y \in (0,1)$ as α^{k_m}, $k_m \in \mathbb{N}$. Smaller values of α lead to smaller values of k_m (see figures D.7 and D.8), but also to a worse approximation in each pass; this leads to a larger number of vectors in order to guarantee a given distortion. Likewise, larger values of α lead to larger values of k_m (see again figures D.7 and D.8) but to a smaller number of vectors. We can see then that there is a trade-off among the value of α, the number of vectors and the range of values of k_m. Therefore, the value of α can potentially affect the rate $\times$ distortion characteristics of the encoder. This trade-off can be seen in figures D.10 and D.11. There, we show the variation of the average number of bits spent for both specifying the atoms indexes r_m and coefficients k_m (see equation (C.5)) against α. The straight lines represent these values for the MP algorithm (γ_n and p_n in equation (B.18)). In this figure, we can see that increasing α, the bits spent to code the atoms indexes tend to decrease, while the bits spent to code the coefficients k_m tend to increase. In addition, for values of α under 0.65, approximately, the MPGBP algorithm spends less bits for encoding the k_m than the MP algorithm for encoding the projection p_n; Likewise, it spends more bits for encoding the atoms indexes than the MP algorithm. It was verified experimentally that this result holds for all bit-rates and video sequences used. Indeed, in appendix A, we have shown analytically that the distortion tends to decrease as α increases (see figure H.2).

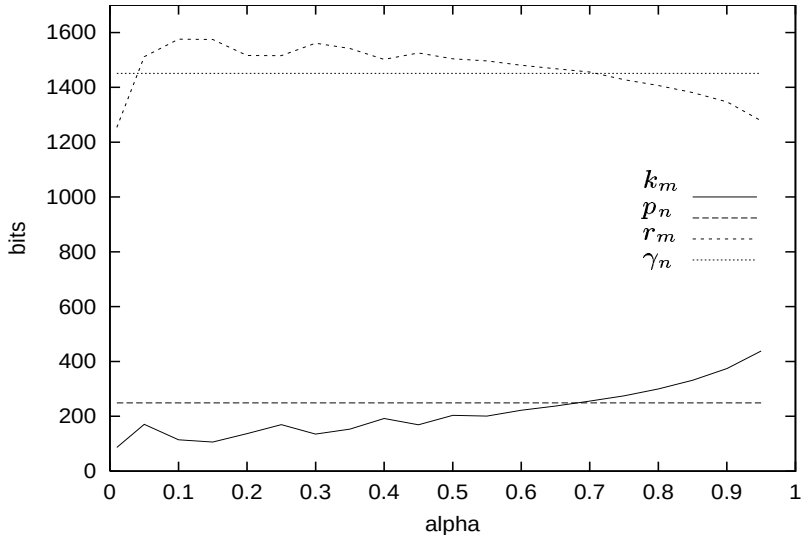


Figure D.10: Average bits spent for k_m, p_n, r_m and γ_n against α for Mother sequence for 24kbps.

In figures D.12 and D.13 we can see the variation of the average PSNR with α

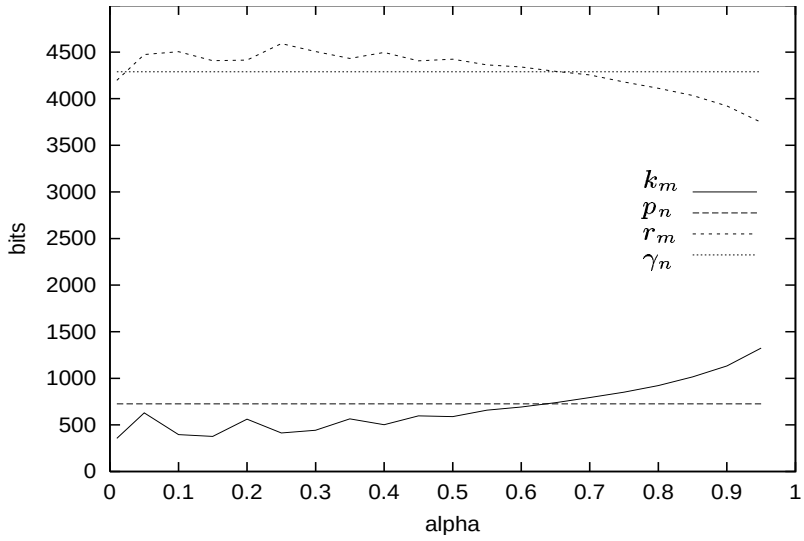


Figure D.11: Average bits spent for k_m, p_n, r_m and γ_n against α for Coast sequence for 64kbps.

for rates 24kbps and 48kbps, respectively. We verify that the variation of α does not interfere significantly with the results, except when α is close to one or below 0.4. In this case, there is a significant drop in performance. Thus, α in the range $[0.40, 0.85]$ is a good choice. In our experiments, we have used $\alpha = 0.56$ for all cases. It is interesting to note that despite the variation in the number of bits spent with the atom indexes r_m and coefficients k_m, the average peak signal to noise ratio (PSNR) of the sequences is approximately constant for $\alpha \in [0.4, 0.85]$.

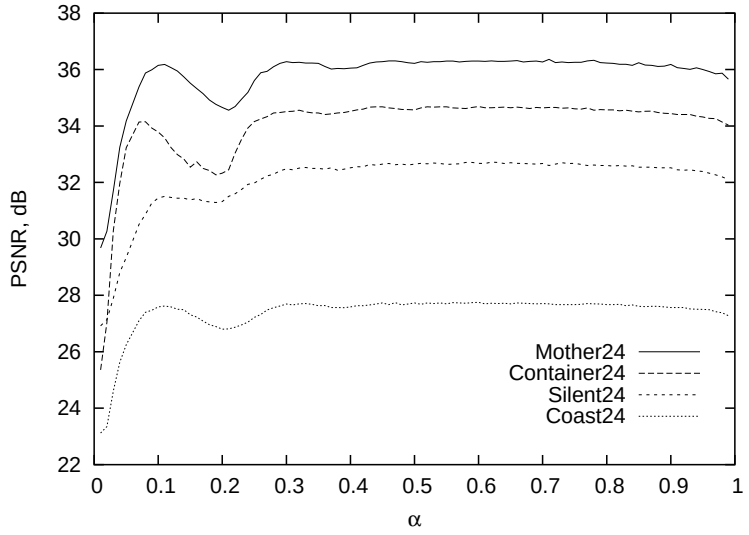


Figure D.12: Variation of the average PSNR with α parameter for Mother, Silent, Container and Coast sequences for 24kbps.

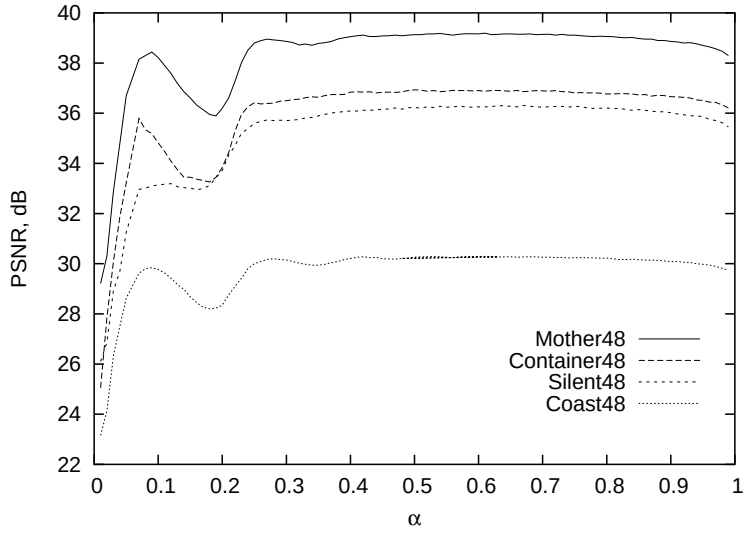


Figure D.13: Variation of the average PSNR with α parameter for Mother, Silent, Container and Coast sequences for 48kbps.

Table D.2 compares the PSNR of the original matching pursuits video encoder (MP) [3, 5] with our adaptation using generalized bit-planes (MPGBP) for several rates and video sequences. In this table, the first column shows the sequences and the bit-rate used, the second column shows the PSNR for MPGBP algorithm, and the third column shows the PSNR for the MP algorithm. The fourth column shows the improvement of the MPGBP over MP algorithm. We can see from this table that MPGBP is always better than the original matching pursuits algorithms [3, 5] for different bit-rates and video sequences. In figures D.14 and D.15 we can confirm this better performance of the MPGBP over the MP video encoder for each frame of the sequences Mother and Hall at the rates 64kbps and 24kbps, respectively.

In table D.3 it is shown the PSNR values and the bit-rates for the MP and the MPGBP encoders. Several sequences were coded using a constant number of atoms per frame. It was done in order to compare the performance without any influence of the rate control method used. Note that, for a given number of atoms, the MPGBP encoder generally leads to an increase in the PSNR values as well as to a decrease in the rate.

Figures D.16 and D.17 show the variation of the average PSNR with rate for both implementations of the matching pursuits encoders for Mother and Silent sequences, respectively. We can see from these figures that the use of the generalized bit-planes scheme consistently improves the performance of the matching pursuits encoder from [3] for all rates. Note that the higher the bit-rate, the higher the improvement. Indeed, our results are

Table D.2: Comparison, in terms of PSNR (dB), between the two matching pursuits implementations.

Seq + Rate	MPGBP	MP [3, 5]	MPGBP - MP
Container10	32.47	32.46	0.01
Mother10	33.27	33.31	-0.04
Hall10	33.30	33.26	0.04
Silent10	29.04	28.96	0.08
Coast10	24.32	24.27	0.05
Container24	34.64	34.43	0.21
Mother24	36.30	36.17	0.13
Hall24	36.52	36.12	0.40
Silent24	32.65	32.72	-0.07
Coast24	27.74	27.63	0.11
Container48	36.91	36.41	0.50
Mother48	39.11	38.45	0.66
Hall48	39.13	37.94	1.19
Silent48	36.25	35.97	0.28
Coast48	30.24	30.22	0.02
Container64	37.92	37.16	0.76
Mother64	40.40	39.31	1.09
Hall64	39.97	38.86	1.11
Silent64	37.82	37.28	0.54
Coast64	31.29	31.22	0.07
Mother96	42.23	40.98	1.25
Foreman96	35.55	35.35	0.20

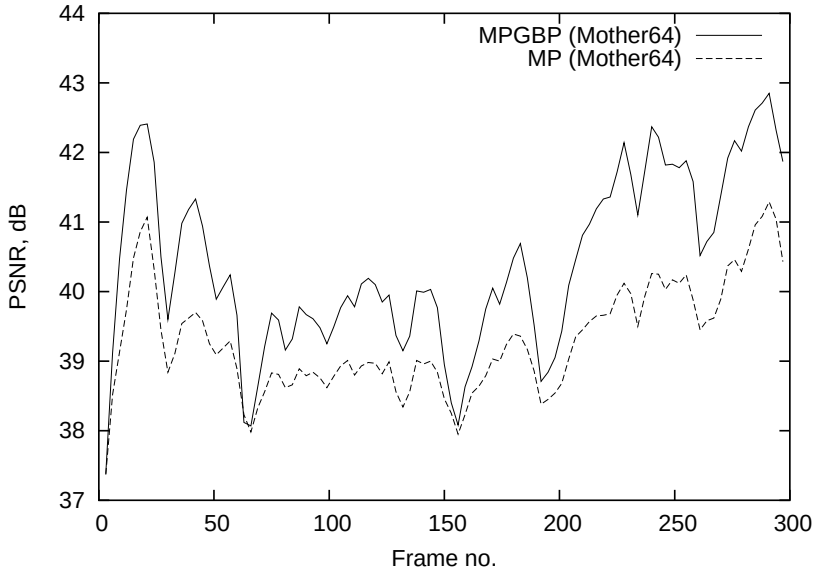


Figure D.14: Variation of the PSNR with the frames for Mother sequence at 64kbps.

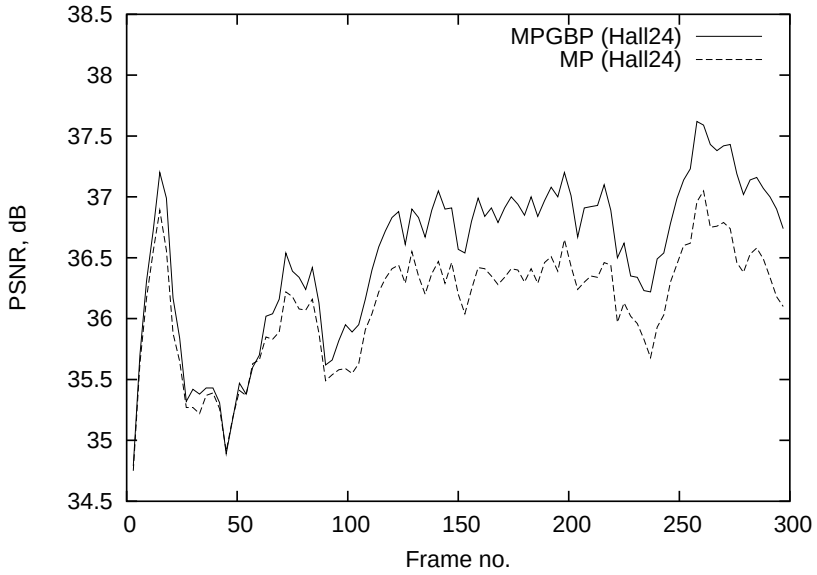


Figure D.15: Variation of the PSNR with the frames for Hall sequence at 24kbps.

Table D.3: Comparison, in terms of PSNR (dB) and bit-rate (kbps), between the two matching pursuits implementations, for a constant number of atoms per frame.

		100 atoms		300 atoms		500 atoms	
		PSNR	rate	PSNR	rate	PSNR	rate
Mother	MP	36.56	26902	39.26	63389	41.18	101005
	MPGBP	36.59	26544	40.42	64445	42.46	101736
Hall	MP	35.94	23489	38.60	61048	40.31	100192
	MPGBP	36.28	23432	39.80	63188	41.27	101326
Silent	MP	33.83	31350	37.62	71066	39.50	110309
	MPGBP	33.93	31310	38.22	70657	40.74	109001
Coast	MP	28.93	32999	31.61	71584	33.24	109631
	MPGBP	28.88	32064	31.67	70721	33.33	107985
Container	MP	34.16	22367	36.88	59255	38.38	97993
	MPGBP	34.34	22208	37.62	60504	39.44	97307

comparable to the best found in the literature, that have been obtained using sophisticated adaptive strategies [10]. The knee on the curves around 20kbps is due to the increase of the frame rate from 7.5 fps to 10 fps.

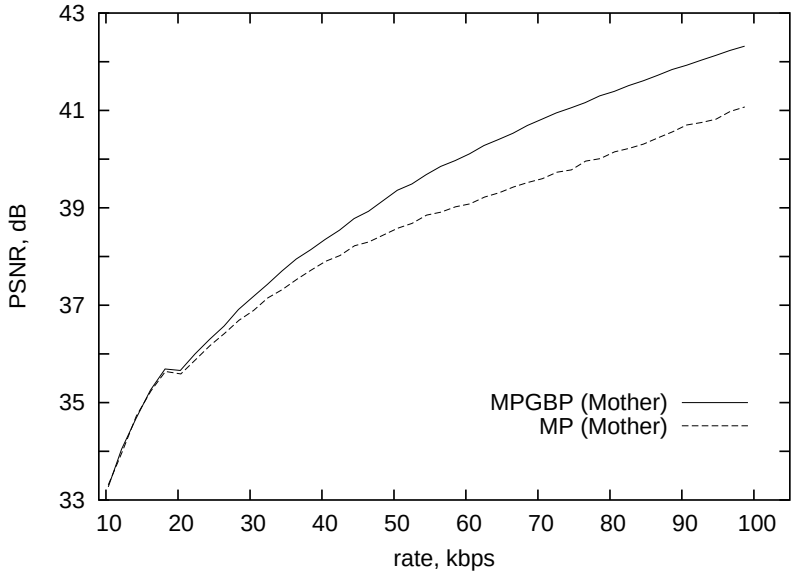


Figure D.16: Variation of the average PSNR with rate for Mother sequence.

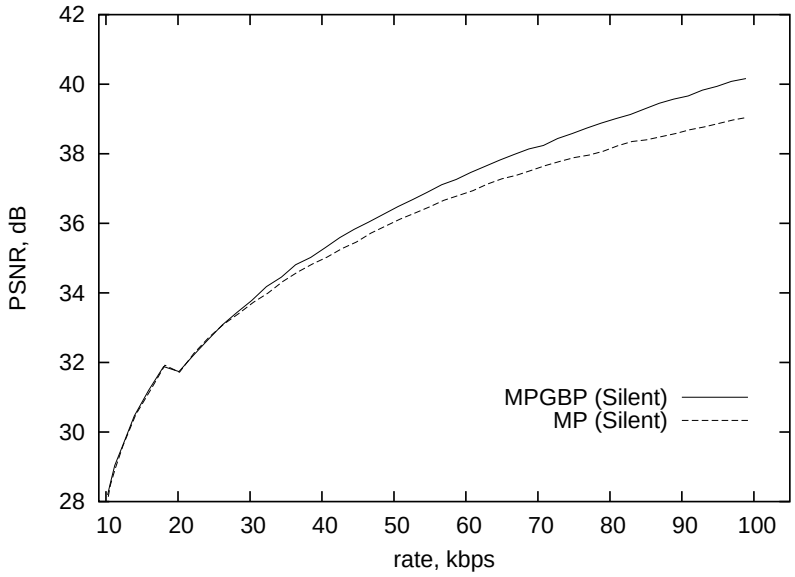


Figure D.17: Variation of the average PSNR with rate for Silent sequence.

It is straightforward extend the coding method to chrominance components. In this case, we must compute, for each video frame, the largest norm X_{max} for both chrominance components. Since the chrominance blocks are 8×8 their X_{max} values are computed considering 42×42 windows centered in every block. In table D.4 we can compare the

performances among a DCT-based video encoder and both matching pursuits video encoders, the MPGBP and MP video encoder. Note that the performance of the MPGBP video encoder is always better than the performance of the DCT and the MP video encoders for the all components.

Table D.4: Comparison, in terms of PSNR (dB) for Y, Cb and Cr, among a DCT-based video encoder and the two matching pursuits implementations.

Seq + Rate		MPGBP	MP [3]	DCT [30]	MPGBP - MP	MPGBP - DCT
Container10	Y	32.36	32.36	29.88	0.00	2.48
	Cb	38.60	38.59	37.00	0.01	1.60
	Cr	38.30	38.19	36.53	0.11	1.77
Mother10	Y	33.23	33.29	32.64	-0.06	0.59
	Cb	39.08	39.03	38.73	0.05	0.35
	Cr	39.62	39.74	39.65	-0.12	-0.03
Hall10	Y	33.22	33.18	30.30	0.04	2.92
	Cb	37.66	37.49	36.52	0.17	1.14
	Cr	39.73	39.85	39.63	-0.12	0.10
Container24	Y	34.48	34.32	33.38	0.16	1.10
	Cb	40.47	40.30	39.45	0.17	1.02
	Cr	40.13	40.12	38.63	0.01	1.50
Mother24	Y	36.14	36.04	35.27	0.10	0.87
	Cb	41.44	41.22	40.13	0.22	1.31
	Cr	41.80	41.81	40.91	-0.01	0.89
Coast48	Y	30.23	30.19	29.42	0.04	0.81
	Cb	40.42	40.46	40.00	-0.04	0.42
	Cr	42.64	42.80	41.90	-0.16	0.74
Foreman48	Y	31.52	31.47	31.14	0.05	0.38
	Cb	37.93	38.02	37.22	-0.09	0.71
	Cr	38.53	38.43	37.39	0.10	1.14

D.3 Conclusions

In this appendix we have described the implementation of a matching pursuits video encoder using the proposed algorithm (MPGBP algorithm) to replace the classical matching pursuits decomposition and quantization algorithm. Our video encoder was used for different sequences and bit-rates, yielding consistent results. The results obtained are very promising, indicating a significant improvement over the classical video-MP algorithm [3].

Apêndice E

Rate-Control Methods for the MPGBP video encoder

The rate-control strategy used in appendix D was to estimate, for each frame, the number of atoms that should be used in order to divide the sequence bit-rate equally among all its frames. However, this estimation procedure performs imprecise bit-rate allocation for each frame of the sequence. Therefore, the rate budget is not usually achieved. In this appendix we propose two rate-control strategies for the MPGBP video encoder. One uses a rate-control procedure that divides precisely the bit-rate of the sequence equally among all its frames and the other uses a rate-control scheme that employs Lagrangean optimization in order to solve the optimal bit allocation problem. In the former we have the advantage of allocating precisely a desired bit-rate for the frame, thus obtaining the desired rate budget. In the latter we can try to optimize to the bit-rate allocation in a R-D sense.

This appendix is organized as follows. In section E.1 we show a procedure that yields precise bit-rate allocation for the MPGBP video encoder. A fixed rate-control that uses this precise bit-rate allocation is also proposed. In section E.2 we propose a rate-control strategy using Lagrangean optimization for the MPGBP video encoder. The conclusions are shown in section E.3.

E.1 A fixed bit-rate allocation scheme for the MPGBP algorithm

The allocation scheme used in appendix D, that divides the bit-rate of the video sequence equally among all its frames is based on an estimation of the number of atoms that should give the desired rate. This estimation uses the average bit-rate spent per atom (see equation (D.6)). However, since inner product quantizers of the atoms are a priori unknown, this scheme tends to produce an imprecise bit-rate control. In this section we present a procedure that permits a precise bit-rate control for the MPGBP video encoder. We perform a precise rate-control relying on the fact that the MPGBP algorithm does not perform explicit quantization - it maps the input signal into a sequence of index (see appendix C), giving an elegant solution for the coefficient quantization problem inherent to the classical MP algorithm.

The procedure used to perform a precise bit-rate allocation is to compute the rate spent for each new encoded atom until the desired bit-rate is achieved. This is possible because the MPGBP algorithm can stop the decomposition whenever the bit-budget is exhausted (see section C.1). However, the position of the atoms in the same macroblock (16×16) is encoded differentially according to the scanning pattern shown in figure D.6. Thus, in order to have precise computation of the bit-rate we should recompute, for each encoded atom added, the bit-rate spent for encoding the differential position of all encoded atoms.

Table E.1 shows the performance of the MPGBP video encoder using both the bit-rate allocation scheme presented in section D.1.2 (referred to as "approximate allocation") and the scheme proposed in this section (referred to as "fixed allocation") with $\alpha = 0.56$. We can see from this table that using the proposed procedure we have a precise bit-rate control which is comparable to the approximate allocation scheme. Note that the precise bit-rate allocation obtained with the proposed scheme permits a fair comparison among video encoders, especially for low bit-rates where small bit-rate differences can have a strong impact. Another advantage of using the precise allocation scheme is that we need a smaller buffer to smooth out bit-rate variations. This can be seen in figures E.1 and E.2 where we show the variation of the bits spent within each frame of the sequences Mother and Hall at rates of 64 and 24kbps, respectively.

Table E.1: Comparison, in terms of PSNR and bit-rate, between the performance of MPGBP video encoder for both the approximate and the fixed allocation schemes.

Seq + Desired Rate	Approximate allocation		Fixed allocation	
	PSNR (dB)	bit-rate (kbps)	PSNR (dB)	bit-rate (kbps)
Container24	34.64	24.59	34.60	24.00
Mother24	36.30	24.39	36.31	24.00
Hall24	36.52	24.43	36.47	24.00
Silent24	32.65	24.14	32.65	24.00
Coast24	27.74	24.11	27.69	24.00
Container64	37.92	65.32	37.87	64.00
Mother64	40.40	64.51	40.36	64.00
Hall64	39.97	65.45	39.86	64.00
Silent64	37.82	64.72	37.71	64.00
Coast64	31.29	64.14	31.29	64.00

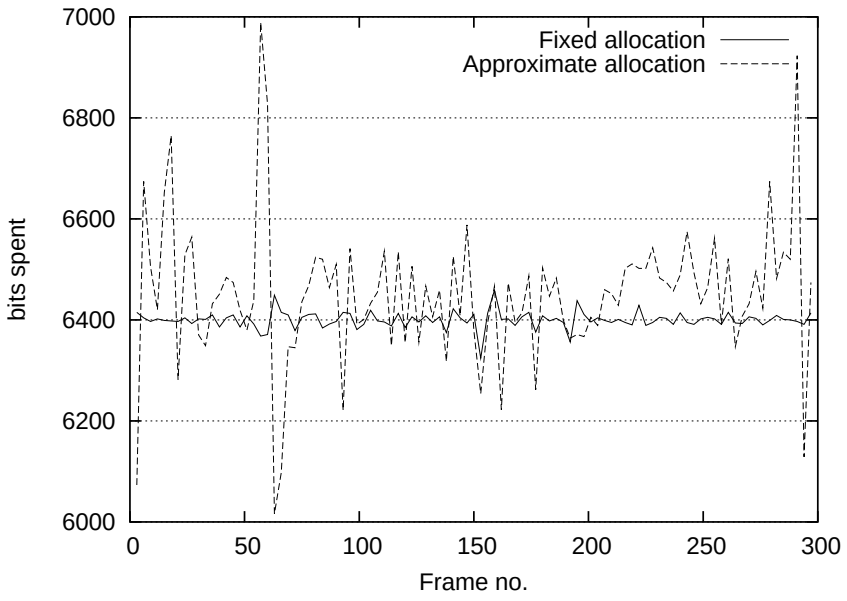


Figure E.1: Variation of the bit spent with each frame of the Mother sequence at a rate of 64kbps and 10fps.

In figure E.3 we compare the performance of the MPGBP video encoder for high bit-rates (using fixed bit-rate allocation) with the De Vleeschouwer and Zakhor's state of art MP video encoder [7] for 60 frames of the Foreman sequence at 30fps. We can see

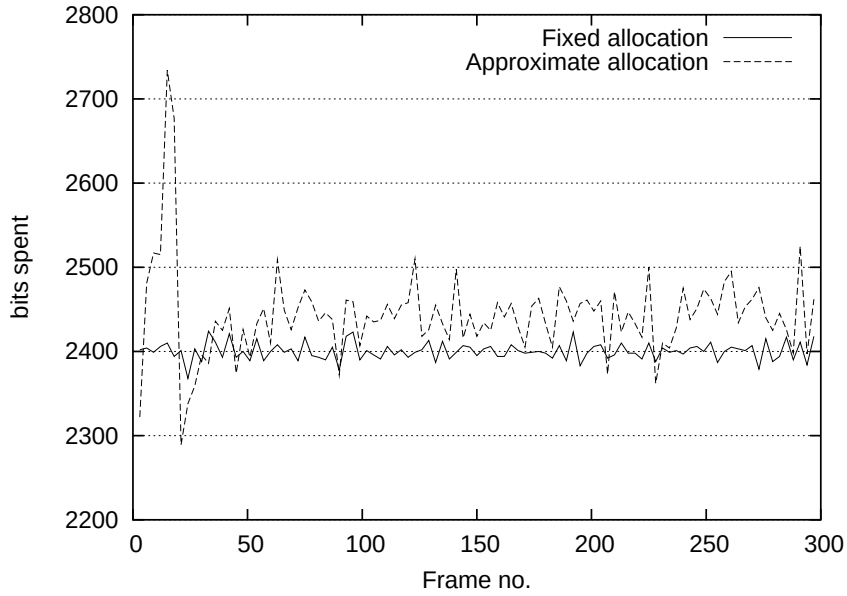


Figure E.2: Variation of the bit spent with each frames off the Hall sequence at rate 24kbps.

from this figure that our results are comparable to those in [7].

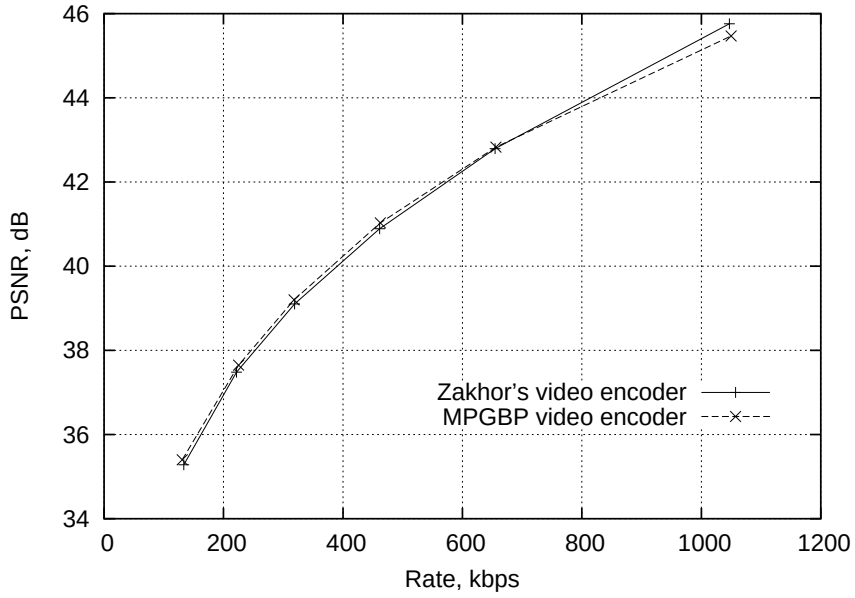


Figure E.3: The performance of the MPGBP and Zakhor's video encoders for Foreman sequence at high bit-rate with 30fps.

E.2 Lagrangean rate-control strategy for the MPGBP algorithm

The trivial solution for the bit-rate allocation problem in video coding applications is to divide the overall bit-rate of the entire video sequence equally among all its frames. In spite of being a very simple solution, it normally does not give the highest average signal-to-noise ratio of the entire sequence at a rate budget. However, we can use Lagrangean optimization to find out how many bits should be allocated to each frame in order to achieve the highest image quality at the lowest cost. In this section we propose an algorithm so as to perform this bit-rate allocation using Lagrangean optimization.

Considering that the rate-distortion (R-D) function is convex, the bit-rate allocation problem can be stated as:

$$\text{minimize } J(\lambda) = D + \lambda R \tag{E.1}$$

where the Lagrangean cost function $J(\lambda)$ is minimized for a given value of the Lagrange multiplier λ. It is important to note that each solution of equation (E.1) for a value of λ corresponds to an optimal rate-distortion trade-off [33, 34] and for each λ we have a different rate.

Since the MPGBP algorithm guarantees that the distortion D always decreases (see Theorem 1 in section C.1), we have that after the i^{th} selected atom, the distortion variation ($\Delta D_i = D_i - D_{i-1}$) is negative and rate variation ($\Delta R_i = R_i - R_{i-1}$) is positive for each encoded atom. As a consequence, we can rewrite equation (E.1) based on the incremental contributions ΔD and ΔR as

$$\Delta J(\lambda) = \Delta D + \lambda \Delta R \tag{E.2}$$

It is important to note that it is advantageous to add atoms to the decomposition while $\Delta J(\lambda) < 0$. Therefore, the solution for equation (E.1) is obtained when we have $\Delta J(\lambda) = 0$ [7]. Also, since in the MPGBP algorithm the decomposition and the quantization are performed together, without any assumptions about quantizers, we can compute precisely the ΔD_i and ΔR_i parameters for each atom added, and thus obtain a stopping criterion that leads to the optimal bit-rate allocation.

Using the stop criterion defined above ($\Delta J(\lambda) = 0$) we propose a rate-control

strategy for the MPGBP video encoder. In this allocation scheme we stop coding atoms when the Lagrangean cost function J_i of the i^{th} atom stops to decrease. It is important to note that this rate allocation strategy assumes the convexity of the R-D curve of the frame. However, for the MPGBP video encoder this curve is not strictly convex, since, due to the differential encoding of the atoms position, ΔR_i may not be always positive. This can be seen in figure E.4 where we show the R-D curve for the $21th$ frame of the Foreman sequence.

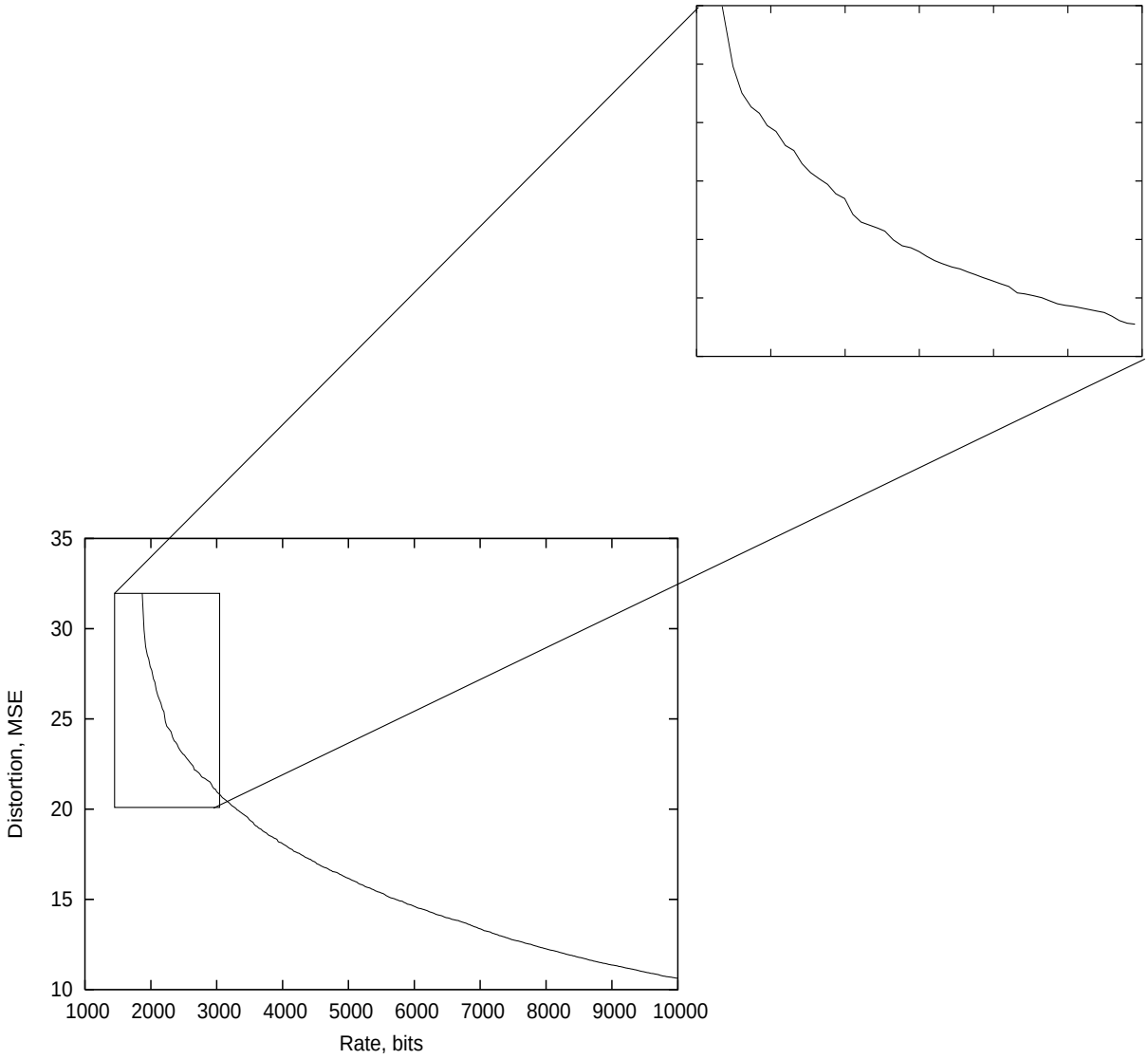


Figure E.4: A real R-D curve for the $21th$ frame of the Foreman sequence.

In order to deal with this problem, we perform bit-allocation using the convex hull of the R-D curve. This process is described as follows. We use sets of N atoms where for all of them we compute $\tilde{\lambda}_i = -\frac{D_i - D_1}{R_i - R_1}$ with D_i and R_i representing, respectively,

the distortion and the rate for the i^{th} atom. In order to find the convex hull, we should choose the maximum $\tilde{\lambda}_j$. After this, for $k > j$, we compute $\tilde{\lambda}_k = -\frac{D_k - D_j}{R_k - R_j}$ for the N atoms following the atom j, and find the maximum slope $\tilde{\lambda}_p$. We repeat this process for the whole region of interest of the R-D curve. Note that this procedure is guaranteed to find the convex hull for the R-D curve only for $N = \infty$ (a finite N value is guaranteed to find the convex hull just for part of N atoms of the R-D curve), we have to do a little modification in this one in order to use finite values of N. After choosing the maximum slope $\tilde{\lambda}_j$ of a set of N atoms we evaluate if the R-D represented by all chosen slopes is really convex. If this one is not convex we increase the N value and process is restarted. Figure E.5 shows an illustration of the convex hull of an R-D curve not convex.

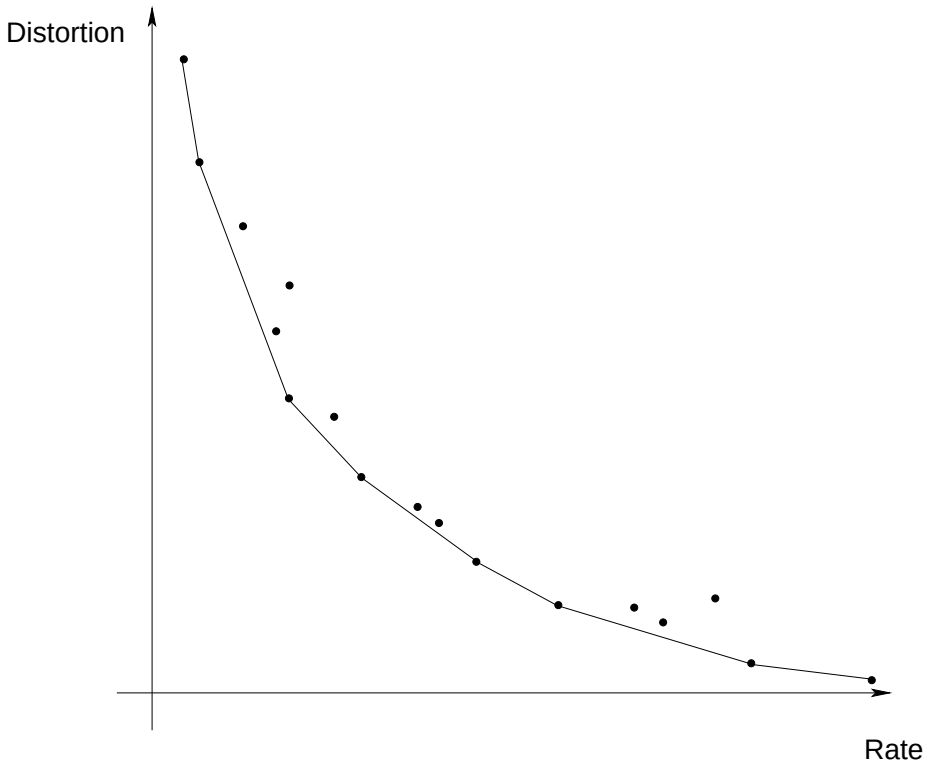


Figure E.5: Illustration of the convex hull.

The R-D optimized rate control algorithm is:

Optimum Rate-Control Algorithm

For a given λ and using a initial set of $N = N_o$ atoms

❶ We start with $J_{current} = J_o$, $J_{previous} = \infty$, $k = k_{previous} = 0$, and $p = 0$;

❷ Repeat while $J_{current} < J_{previous}$

(a) For i=k up to N+k,

Compute $\tilde{\lambda}_i = -\dfrac{D_i - D_k}{R_i - R_k}$;

(b) Choose $m \in \{1, \ldots, N\}$ such that

$\tilde{\lambda}_m = \max\{\tilde{\lambda}_i\}$;

(c) Compute $J_m = D_m + \lambda R_m$;

(d) Do $k_{previous} = k$, $J_{previous} = J_{current}$, $k = k + m$, $J_{current} = J_m$ and $\tilde{\lambda}_p = \tilde{\lambda}_m$;

(e) For i=0 up to p,

if $\tilde{\lambda}_{p-1} < \tilde{\lambda}_p$ do $N = N + 1$ and go to step ❶;

(f) Do $p = p + 1$;

❸ Encoding the frame with $k_{previous}$ atoms;

❹ Stop.

Table E.2 shows the performance of the MPGBP video encoder for several video sequences with 300 frames at 10fps. We used both the precise bit-rate allocation scheme proposed in section E.1 (labeled as "fixed allocation") and the one proposed in this section (labeled as "optimum allocation") with $\lambda = 0.0006$. Since the proposed algorithm performs an optimized bit-rate allocation using the R-D curve in the sense of having the lowest average distortion at the lowest cost, we compute the PSNR of the sequence as:

$$PSNR_1 = 10 \log_{10} \frac{255^2}{MSE_{seq}} \tag{E.3}$$

where MSE_{seq} is the average among all frames. Note that, in the literature, the most

popular measured is:

$$\overline{PSNR}_2 = \frac{1}{N}\sum_{i=1}^{N} PSNR_i \qquad (E.4)$$

From this table we can see that the optimum allocation strategy improves the performance of the MPGBP video encoder in all cases.

Table E.2: Comparison, in terms of PSNR (dB) and bit-rate (kbps), between the performance of MPGBP video encoder using both the fixed and the optimum allocation schemes.

Sequence	Rate	Fixed allocation		Optimum allocation			Optimum - Fixed	
		PSNR1	PSNR2	PSNR1	PSNR2	λ	$PSNR1_{diff}$	$PSNR2_{diff}$
Container	62.79	37.78	37.80	37.84	37.84	0.0006	0.06	0.04
Mother	51.58	39.27	39.43	39.38	39.38	0.0006	0.11	-0.05
Hall	45.50	38.83	38.88	38.97	38.98	0.0006	0.14	0.10
Silent	73.13	38.32	38.47	38.52	38.54	0.0006	0.20	0.07
Foreman	148.28	37.54	37.81	37.76	37.84	0.0006	0.22	0.03
Weather	68.18	37.40	37.97	37.95	37.98	0.0006	0.57	0.01
Container	135.98	40.88	40.91	40.92	40.92	0.0002	0.03	0.01
Mother	102.66	42.32	42.47	42.50	42.50	0.0002	0.18	0.03
Hall	109.97	41.53	41.56	41.64	41.64	0.0002	0.11	0.08
Silent	130.20	41.62	41.81	41.91	41.91	0.0002	0.29	0.10
Foreman	265.30	40.72	40.97	41.18	41.22	0.0002	0.46	0.25
Weather	120.02	41.16	42.06	42.26	42.30	0.0002	1.10	0.24

Figures E.6 and E.7 compare the MPGBP video encoder that uses the rate allocation scheme proposed in this section and the fixed algorithm proposed in section E.1. They show the variation of the PSNR among the frames for the sequence Mother at 51.58kbps obtained using $\lambda = 0.0006$ and for the sequence Silent at 130.20kbps obtained using $\lambda = 0.0002$, respectively. We can see from these figures that in spite of not improving significantly the performance (final average PSNR), the proposed algorithm produces a smaller distortion (or PSNR) variation among the frames. It is also worth noting that the PSNR improvement tends to increase with the bit-rate.

It is important to note that the rate spent using the optimum allocation algorithm is controlled by the value of λ. However, in order to achieve a precise bit-rate, one should find the corresponding λ. This can be done, for example, using the bisection algorithm [107]. However, this is a computationally demanding process. Consid-

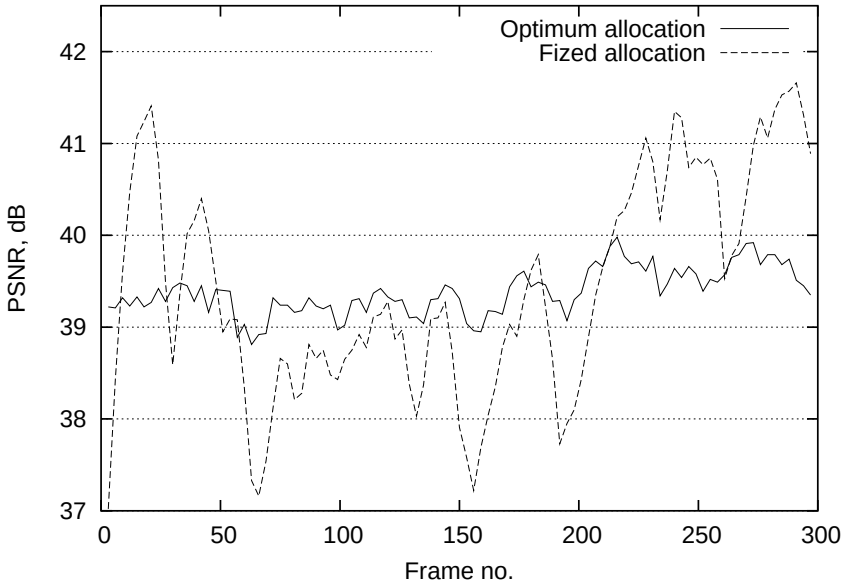


Figure E.6: Variation of the PSNR with the frames for Mother sequence at 51.58kbps.

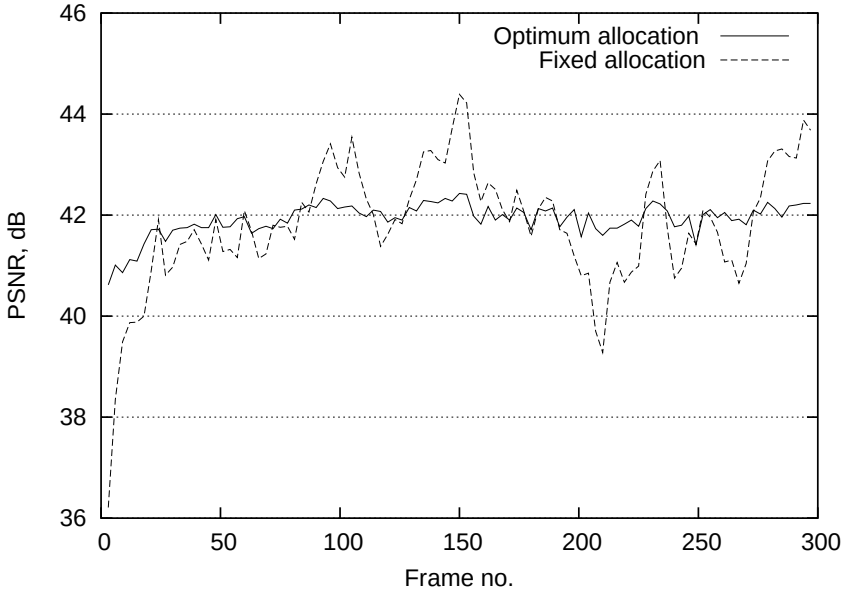


Figure E.7: Variation of the PSNR with the frames for silent sequence at 130.20kbps.

ering the order of the improvements obtained in table E.2, we conclude that the best complexity×performance trade-off is still given by the precise rate control (see section E.1).

E.3 Conclusions

In this appendix we propose two rate-control strategies for the MPGBP video encoder. In the first, we equally divide the sequence bit-rate among all its frames, using the precise bit-rate allocation scheme proposed in section E.1. This scheme has the advantage of precisely achieving the rate budget as well as requiring a small buffer to reduce the bit-rate variations. The results show that this scheme can be used to provide a fair comparison among the performance of video encoders, since one can precisely obtain the desired bit-rate. The other proposed algorithm uses Lagrangean optimization to find out how many bits should be allocated to each frame in order to achieve the highest image quality at the lowest cost. In this case the results show that this scheme only provides only a slight improvement over the fixed allocation. Despite this small improvement in objective performance, this algorithm produces a smaller PSNR variation among the frames of video sequence, and therefore a better subjective quality for the encoded video sequence.

Apêndice F

Different Dictionaries for the MPGBP Algorithm

In appendix D we have analyzed the performance of the MPGBP algorithm in video coding. We have made a comparison between the performance of this one and the original Matching Pursuits algorithm implemented using Neff and Zakhor's video encoder framework [3]. The results show that the performance of the MPGBP algorithm is consistently better than the original one. Since in that appendix, we have used both algorithms with the dictionary proposed in [3], a question emerges: is there another dictionary that produces a better performance?

From the theoretical analysis of the R-D performance of selected dictionaries made in appendix C, we have conjectured that the reduction of the $\Theta(\mathcal{C})$ value can lead to a better rate$\times$distortion trade-off, provided that $\mathrm{q}(\mathcal{C})$, the cardinality of the dictionary $\mathcal{C}$, does not increase too much. This $\Theta(\mathcal{C})$ reduction can be obtained using either a better distribution of its vectors, or extra vectors to appropriately fill the "empty"regions of the vector space (see figure F.1). Note that in the former we preserve both the cardinality $\mathrm{q}(\mathcal{C})$ and the dimension N of the dictionary $\mathcal{C}$. On the other hand, in the latter, the cardinality $\mathrm{q}(\mathcal{C})$ is increased, and there is a trade-off between this one and the $\Theta(\mathcal{C})$ value.

In this appendix we investigate the performance of the MPGBP video encoder using different overcomplete dictionaries in order to assess the effect of these dictionaries on the performance of a video encoder. Also, we identify good dictionaries, that is, with good trade-offs between the cardinality $\mathrm{q}(\mathcal{C})$ and $\Theta(\mathcal{C})$. It is important to note that the fixed rate control strategy proposed in section E.1 is used in all experiments of this

appendix.

This appendix is organized as follows. In section F.1 we have used extra added vectors, obtained by training, in order to reduce the $\Theta(\mathcal{C})$ value. We also investigate the impact of forcing the separability of the included vectors. In order to do so we have developed a measure of separability for two-dimensional discrete functions. In section F.2 we have used it in order to reduce $\Theta(\mathcal{C})$ non-separable dictionaries derived using analytical expressions. The conclusions are shown in section F.3.

F.1 Dictionaries derived by training with typical video sequences

In this section we propose a way of obtaining a dictionary with reduced $\Theta(\mathcal{C})$ starting from the original Neff and Zakhor's dictionary $\mathcal{C}$ [3]. This proposed dictionary can be obtained by adding extra vectors to the dictionary $\mathcal{C}$. Our idea is to build such a dictionary by placing new vectors in the original dictionary so as to fill its deep holes. By deep holes we mean a region in R^N in which the closest vector in the dictionary $\mathcal{C}$ has $\theta_n > \theta_c \leq \Theta(\mathcal{C})$. These new vectors reduce the angle $\Theta(\mathcal{C})$ of the dictionary $\mathcal{C}$.

Figure F.1 shows an illustration of this process for R^2. In figure F.1a we can see a deep hole located in the region that has the maximum $\theta_1 > \theta_c$ angle. By placing an extra vector in this deep hole we form a new dictionary with one more vector but smaller average θ. A new deep hole with an angle $\theta_2 > \theta_c$ can be found in another position (see figure F.1b) and another extra vector can be placed in this deep hole. This process can be repeated until we have either $\theta_n < \theta_c$ or a defined number of extra vectors has been reached. Note that in each pass we have $\theta_n < \theta_{n-1}$. In other words, if a vector $\mathbf{X}$ is in a region containing a deep hole and $\mathbf{u_j}$ is its nearest vector, we have that

$$\theta(\mathbf{X}, \mathbf{u_j}) = \arccos \frac{< \mathbf{X}, \mathbf{u_j} >}{||\mathbf{X}|| \, ||\mathbf{u_j}||} = \arccos \frac{< \mathbf{X}, \mathbf{u_j} >}{||\mathbf{X}||} \geq \theta_c \qquad \text{(F.1)}$$

The training step consists of considering as candidates for inclusion in the dictionary vectors $\mathbf{X_k}$ from the residual image of the training video sequences that satisfy equation (F.1).

As a first experiment we included just separable vectors. We decided to do so because as seen in section D.1.2.1 the implementation of the Matching Pursuits algorithm

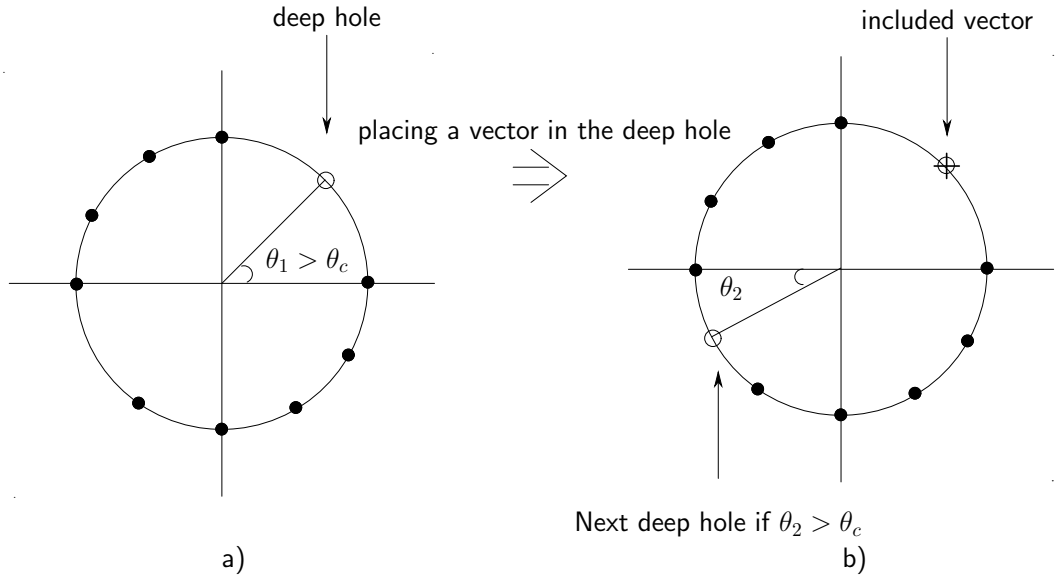


Figure F.1: Illustration of the $\Theta(\mathcal{C})$ reduction by placing of a vector in the hole of the dictionary $\mathcal{C}$ in the R^2.

using separable dictionaries is much less complex than the implementation using non-separable ones. In order to do so we had to derive a measure of separability of a vector, which is introduced in appendix J.

This measure of non-separability varies in the range $[0,1)$, having $\mathcal{S} = 1$ for the separable candidates and $\mathcal{S} < 1$ for the other ones. Figure F.2 shows the histogram of the measure of separability $\mathcal{S}$ computed for all candidates to be added to the original Neff and Zakhor's dictionary (see table D.1). From this histogram we can see that most of the candidates are non-separable. In fact, only less than 1% of the candidates are separable.

The procedure used to select from all the 2-D candidates the vectors to be added to the original Neff and Zakhor's dictionary is described as follows. First of all, we use the MPGBP video encoder, described in appendix D, to encode different video sequences with several α and the rate values. During this encoding process (training step [1]), the 2-D regions of the residual image at the position and with the same size of the respective encoded atoms that give angles greater than a particular threshold are consider as candidate vectors. From all these candidates we select only the ones which are separable. It was done computing the measure of non-separability $\mathcal{S}$ of all candidates and selecting the ones that have the this measure equal to one (separable candidates). The vectors that form

[1] The video sequences used during the training step were different from the sequences used for evaluating the performance of the MPGBP video encoder.

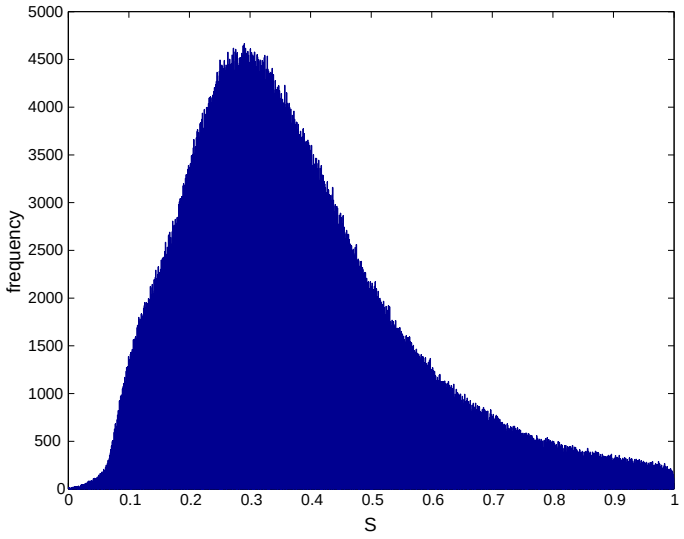


Figure F.2: Histogram of the measure of non-separability $\mathcal{S}$ for the candidates to be a new structure.

the 2-D separable candidates can be obtained by equations (J.6) and (J.7) (see appendix J for more details).

It is important to note that, if we already place an extra vector in a deep hole, it is better to place the next vector onto another deep hole so as to obtain a better distribution of the vectors in the space. Thus, after obtaining the set of vectors that form the selected separable candidates, we compute the angles among them. Then, the two vectors that provide the smallest angle among them are considered to be in the same deep hole. Thus, one vector of the two should be eliminated from the set. In order to do so, we compute the angles among the vectors of the selected couple and all vectors of the original Neff and Zakhor's dictionary. Then, the vector of the selected pair that has the smallest angle is eliminated from the set. This process continues until the set of vectors is reduced to the number of extra vectors desired.

At the beginning of this section we conjectured that the $\Theta(\mathcal{C})$ reduction can lead to a better rate$\times$distortion performance of the video encoder. In order to assess this, it is important to consider the reduction in the average distortion (β_{avg}), since we have $D^{(P)} \leq \beta D^{(P-1)}$ for each new encoded vector. Therefore, in order to facilitate the comparisons we use, in addition to the angle $\Theta(\mathcal{C})$, the angle $\overline{\Theta}$, which is the angle corresponding to β_{avg}, that is

$$\overline{\Theta} = \arccos\left(\sqrt{\frac{1-\beta_{avg}^2}{2\alpha-\alpha^2}}\right) \tag{F.2}$$

where β_{avg} is the average value of all β (see equation(C.6)).

Table F.1 shows the difference among the PSNR's, the $\Theta(\mathcal{C})$ and the $\overline{\Theta}$ angles for video encoders that use the MPGBP algorithm with different dictionaries. We have used the original Neff and Zakhor's dictionary (DicNeff) and two proposed dictionaries, one with 40 separable structures (20 new structures in addition to the 20 original ones) referred as Dic40, and other with 64 separable structures (44 new structures in addition to the 20 original ones) referred as Dic64. It is important to note that both proposed dictionaries were derived with $\theta_c = 50°$ (see equation F.1).

Table F.1: Comparison, in terms of PSNR (dB), $\Theta(\mathcal{C})$ and $\overline{\Theta}$, between the MPGBP video coder using the original Neff and Zakhor's dictionary and two proposed dictionaries formed by training of video sequences.

Seq + Rate	Dic40 - DicNeff			Dic64 - DicNeff		
	$PSNR_{diff}$	$\Theta(\mathcal{C})_{diff}$	$\overline{\Theta}_{diff}$	$PSNR_{diff}$	$\Theta(\mathcal{C})_{diff}$	$\overline{\Theta}_{diff}$
Container24	0.05	-0.93°	-0.80°	0.15	-1.22°	-1.03°
Mother24	-0.29	-0.51°	-0.26°	-0.26	-0.76°	-0.45°
Hall24	-0.47	-1.22°	-0.27°	-0.56	-1.36°	-0.54°
Silent24	-0.29	-0.86°	-0.29°	-0.27	-0.87°	-0.44°
Coast24	-0.16	-0.70°	-0.48°	-0.09	-0.92°	-0.69°
Container64	-0.01	-0.93°	-0.74°	0.04	-1.16°	-1.04°
Mother64	-0.44	-0.90°	-0.35°	-0.40	-1.40°	-0.58°
Hall64	-0.30	-1.17°	-0.44°	-0.40	-1.37°	-0.66°
Silent64	-0.55	-1.19°	-0.36°	-0.54	-1.78°	-0.54°
Coast64	-0.03	-1.00°	-0.65°	0.11	-1.26°	-0.93°

We can see from this table that the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ reductions obtained using the proposed dictionaries (Dic40 and Dic64), in this first experiment were not enough to improve the performance of the MPGBP video encoder. In other words, the procedure of placing vectors in the holes does not decrease enough the angles $\Theta(\mathcal{C})$ and $\overline{\Theta}$ and therefore it does not compensate for the increase in bit-rate advent from the increase of

the cardinality. This is probably due to the fact that we have added to the original Neff and Zakhor's dictionary only separable structures, which represent a very small part of the candidates (1% of the total candidates). In fact, most of the candidates to be new structures are non-separable (see figure F.2), and a significant $\overline{\Theta}$ reduction only could be achieved if these ones were used to increase the cardinality of the dictionary. Thus, in a second experiment, we use a similar procedure to the one above, with the difference that non-separable structures are allowed.

In order to form the non-separable dictionary, we should choose more structures than what we have chosen in the separable case. (for example, the separable dictionary Dic64 was formed with 64 separable structures, which yields 4096 2-D structures). However, the computation complexity of the above procedure increases proportionally with the number of desired structures for the dictionary. Thus, we have to change the procedure so as to keep the computational complexity with reasonable bounds. The difference in the procedure is that instead of selecting all desired structures at the same time we select sets of N structures so as to form the dictionary. Table F.2 shows the performance of the MPGBP video encoder using the original Neff and Zakhor's dictionary (labeled as "DicNeff") and the dictionary formed considering the non-separable structures as well as the 400 original ones (labeled as "DicNonSep"). The proposed dictionary has a cardinality of 8100 and was formed by sets of 300 structures (N=300). Note that the reductions in the angles $\Theta(\mathcal{C})$ and $\overline{\Theta}$ obtained with the proposed non-separable dictionary are not enough to compensate the increase of the bit-rate obtained with the increase of the cardinality. This might be due to the modification in the algorithm of selecting structures. In fact, the procedure of selecting sets of structures allows for more than one extra vector in the same deep hole. Thus, in spite of using non-separable structures in the dictionary, this one does not provide a good spatial vector distribution, and thus, it does not lead to an improvement in performance.

F.2 Dictionaries derived using analytical expressions

Since in section F.1 it was shown that significant $\Theta(\mathcal{C})$ and $\overline{\Theta}$ reductions could only be achieved with dictionaries with good distribution of its vectors in the space. Also, we have seen that they should be non-separable. In this section we investigate the per-

Table F.2: Comparison, in terms of PSNR (dB), between the two matching pursuits implementations using different dictionaries with MPGBP algorithm.

Seq + Rate	DicNonSep			DicNeff			DicNonSep - DicNeff		
	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}$	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}$	$PSNR_{diff}$	$\Theta(\mathcal{C})_{diff}$	$\overline{\Theta}_{diff}$
Container24	34.05	85.43°	81.41°	34.60	86.47°	81.84°	-0.55	-1.04°	-0.43°
Mother24	35.72	85.33°	80.14°	36.31	85.22°	80.44°	-0.59	-0.11°	-0.30°
Hall24	35.47	85.65°	79.40°	36.47	86.22°	79.71°	-1.00	-0.57°	-0.31°
Silent24	32.11	85.51°	77.66°	32.65	86.44°	78.06°	-0.54	-0.93°	-0.40°
Coast24	27.34	85.40°	79.17°	27.69	85.65°	79.46°	-0.55	-0.25°	-0.29°
Container64	37.12	85.86°	82.42°	37.87	86.79°	82.46°	-0.75	-0.93°	-0.04°
Mother64	39.43	85.60°	81.59°	40.36	85.98°	81.91°	-0.93	-0.38°	-0.32°
Hall64	39.05	86.10°	80.92°	39.86	86.74°	81.31°	-0.81	-0.64°	-0.39°
Silent64	36.75	86.01°	80.06°	37.71	87.19°	80.50°	-0.96	-1.18°	-0.44°
Coast64	30.68	85.80°	80.83°	31.29	86.71°	81.23°	-0.61	-0.91°	-0.40°

formance of the MPGBP video encoder using non-separable dictionaries derived using analytical expressions.

The non-separable dictionaries proposed in this section were generated from rotations, scaling and sampling of the continuous 2-D function obtained from their analytical expressions.

Figure F.3 depicts this process. In it can see that the new coordinates x and y used for the rotation, scaling and sampling operations can be generated using the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ by

$$\begin{bmatrix} x \\ y \end{bmatrix} = m\mathbf{v}_1 + n\mathbf{v}_2 \tag{F.3}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ are dependent of the rotation and scaling operations.

If we define a and b as the scaling factors in the horizontal and vertical directions, respectively, and γ as the rotation angle, the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ can be represented by

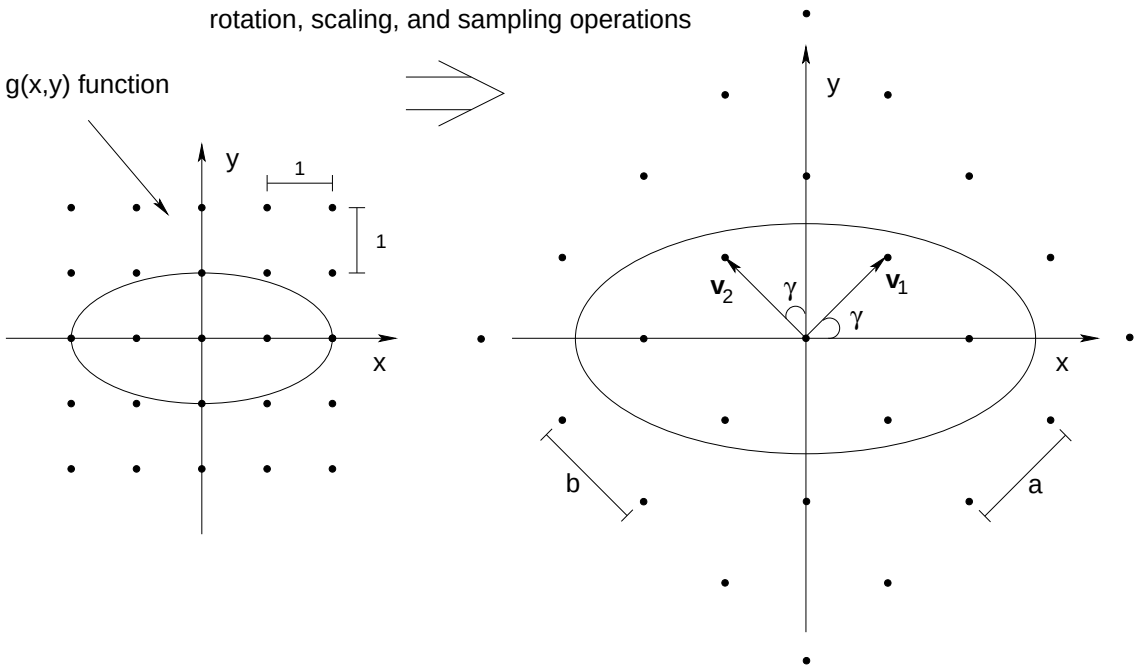


Figure F.3: Representation of the rotation, scaling and sampling operations of a continuous function $g(x, y)$.

$$\mathbf{v}_1 = \begin{bmatrix} a\cos\gamma \\ a\sin\gamma \end{bmatrix} \tag{F.4}$$

$$\mathbf{v}_2 = \begin{bmatrix} b\sin\gamma \\ -b\cos\gamma \end{bmatrix} \tag{F.5}$$

Using equations (F.3), (F.4) and (F.5) we can obtain the new coordinates x and y that generate the rotated and scaled samples of the continuous function $g(x, y)$ as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ma\cos\gamma + nb\sin\gamma \\ ma\sin\gamma - nb\cos\gamma \end{bmatrix} \tag{F.6}$$

It is important to note that the values of m and n in equation (F.6) are integers in the ranges $[-L_x/2, \cdots, L_x/2]$ and $[-L_y/2, \cdots, L_y/2]$, respectively, with L_x and L_y values obtained by

$$L_x = l_x\|\cos\gamma\| + l_y\|\sin\gamma\| \tag{F.7}$$

$$L_y = l_x\|\sin\gamma\| + l_y\|\cos\gamma\| \tag{F.8}$$

where l_x and l_y are the horizontal and vertical sizes of $g(x, y)$ ("non-rotated"function).

Since the structures of the proposed dictionary are generated from rotations of rectangular regions, we have to use the support region in figure F.4 instead of the one in figure D.9 to compute the largest norm of the macroblocks (X_{max}). Note that the proper computation of the X_{max} is important because the MPGBP algorithm assumes that the norm of the input signal is unitary. It is important to note that the new support region is dependent both on the largest rectangular region and on the angles of rotation that were used to calculate the R_{max}.

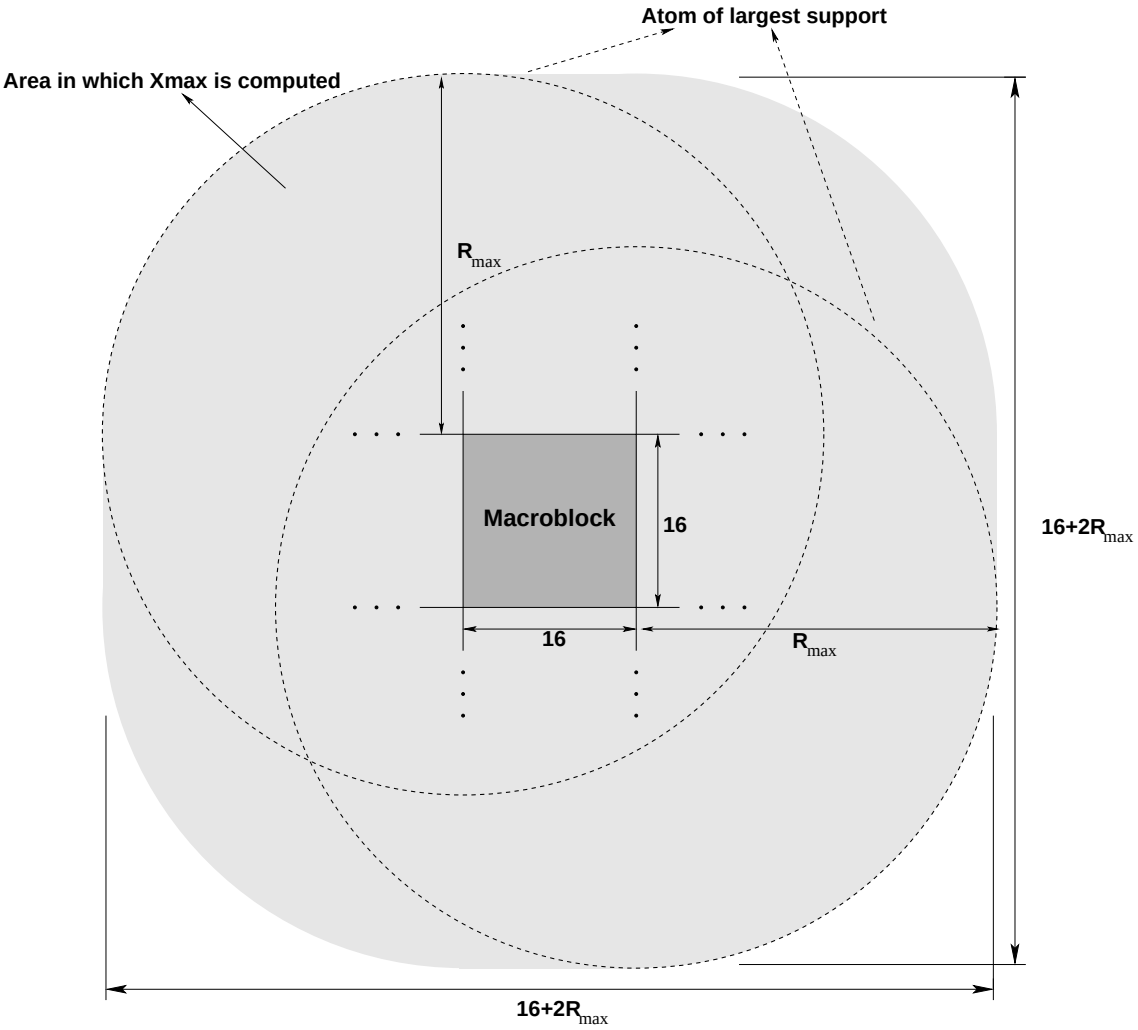


Figure F.4: Illustration of the area in which X_{max} is computed when the dictionary $Dicg_0$ is used.

Note that whenever we increase the support region defined to compute the inner product we are also increasing the dimension of the dictionary. This increase in dictionary's dimension tends to increase its $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles without significantly affecting the performance of the MPGBP video encoder. Then, in order to have a fair comparison

between the original and the proposed dictionaries, we should use the same support region for both. This forced us to adapt the algorithm in section D.2 to use a support region as in figure F.4. Table F.3 shows the performance of the MPGBP video encoder using both the circular and the original support regions with the fixed allocation rate-control scheme proposed in section E.1. For the circular region a $R_{max} = 51$ was used. From this table we can confirm that the use of the circular support region increase the angles $\Theta(\mathcal{C})$ and $\overline{\Theta}$ does not influence on the performance of the MPGBP video encoder.

Table F.3: Performance of the MPGBP video encoder using both the circular and the original support regions.

Seq + Rate	Circular support region			Original support region		
	PSNR (dB)	$\Theta(\mathcal{C})$	$\overline{\Theta}$	PSNR (dB)	$\Theta(\mathcal{C})$	$\overline{\Theta}$
Container24	34.58	87.04°	83.17°	34.60	86.47°	81.84°
Mother24	36.19	86.17°	82.29°	36.31	85.22°	80.44°
Hall24	36.41	87.28°	81.62°	36.47	86.22°	79.71°
Silent24	32.59	87.08°	80.18°	32.65	86.44°	78.06°
Coast24	27.71	86.79°	81.45°	27.69	85.65°	79.46°
Container64	37.83	87.70°	84.16°	37.87	86.79°	82.46°
Mother64	40.38	87.19°	83.51°	40.36	85.98°	81.91°
Hall64	39.87	87.69°	83.01°	39.86	86.74°	81.31°
Silent64	37.72	87.81°	82.25°	37.71	87.19°	80.50°
Coast64	31.33	87.32°	82.91°	31.29	86.71°	81.23°

In what follows we propose two types of dictionaries: one that is formed by rotations of the function defined in equations (F.9) and (F.10) and another that is generated from rotations of the Neff and Zakhor's structures. Both of them are defined by analytical expressions and use rotated structures generated from the corresponding "non-rotated"separable functions.

F.2.1 A dictionary formed by rotations and anisotropic scalings of a 2-D function

In this section we investigate the performance of the MPGBP video encoder using a non-separable dictionary formed from rotations and anisotropic scalings of a 2-D function [93]. We expect that using such a non-separable dictionary one can reduce the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles, what can lead to a better rate$\times$distortion trade-off.

The non-separable dictionary used in this section was proposed in [21]. This one is generated by rotations and anisotropic scaling of the function obtained as

$$g_{\vartheta}(x,y) = w_{\vartheta}(x) \cdot e^{-\left(x^2+y^2\right)} \tag{F.9}$$

where we have

$$w_{\vartheta}(x) = \begin{cases} 2-4\|x\|, & \text{for } 0 \leq \|x\| < 1 \\ -4+2\|x\|, & \text{otherwise} \end{cases} \tag{F.10}$$

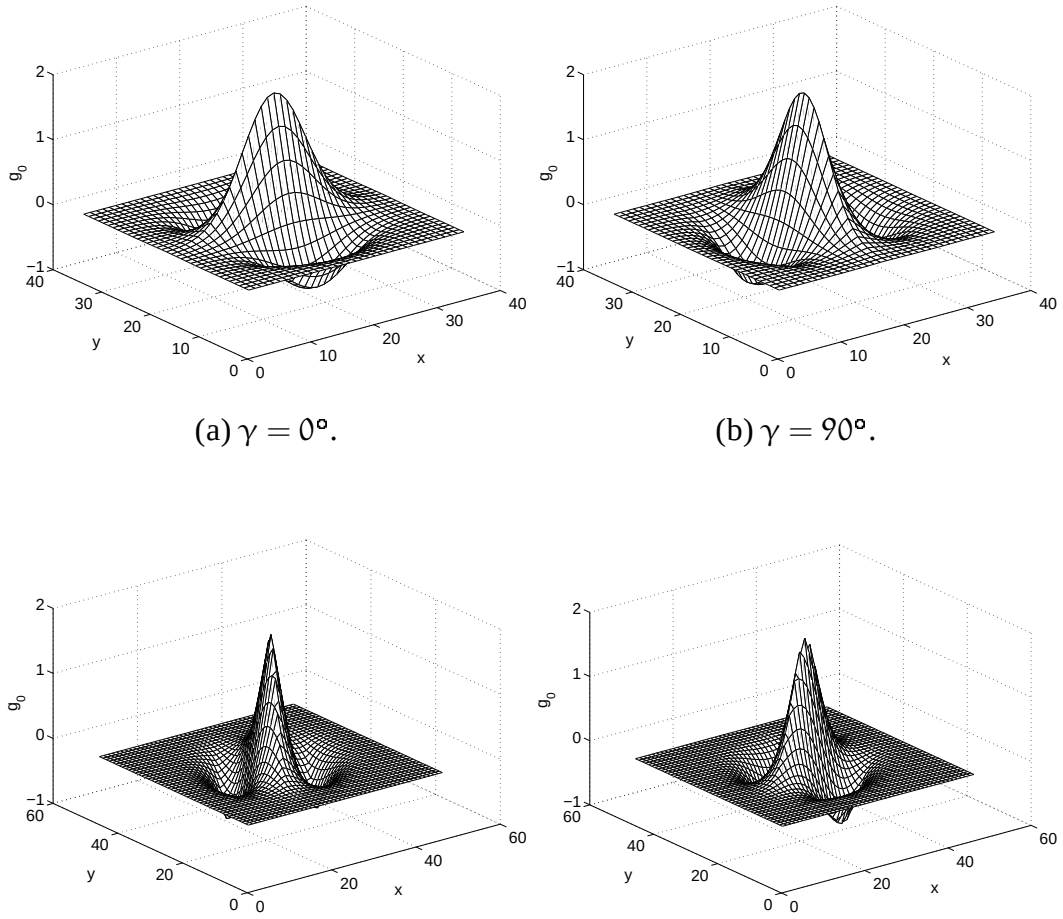
The rotation, scaling and sampling operations of the continuous function $g_{\vartheta}(x,y)$ were performed using equation (F.6) with the γ angles selected in the range $[0,\pi)$ by

$$\gamma = \frac{n \cdot \pi}{128} \tag{F.11}$$

where n is an integer.

Using function $g_{\vartheta}(x,y)$ defined in equation (F.9), we propose in this section two non-separable dictionaries, labeled as $\mathrm{Dicg}_{\vartheta}^{1}$ and $\mathrm{Dicg}_{\vartheta}^{2}$. The dictionary $\mathrm{Dicg}_{\vartheta}^{1}$ was formed using the $g_{\vartheta}(x,y)$ separable structures whose size is $l_x \times l_y$ ($l_x, l_y \in \{1,3,5,\cdots,25\}$), as well as the non-separable ones generated from rotations of approximately $1.4°$ ($n = 0,1,\cdots,127$) of their separable structures. The cardinality of this dictionary is $21,505$. The dictionary $\mathrm{Dicg}_{\vartheta}^{2}$ was formed from rotations of approximately $2.8°$ ($n = 0,1,\cdots,63$) of separable structures that have $l_x, l_y \in \{1,3,5,\cdots,35\}$. This dictionary has a cardinality of $20,673$. Note that we have used for the proposed dictionaries ($\mathrm{Dicg}_{\vartheta}^{1}$ and $\mathrm{Dicg}_{\vartheta}^{2}$) the support region shown in figure F.4 with R_{max} equal to 37 and 51, respectively.

Table F.4 shows the differences between the PSNR, $\Theta(\mathcal{C})$ and $\overline{\Theta}$ values obtained when the MPGBP video encoder uses either the original Neff and Zakhor's dictionary (DicNeff) or the dictionaries $\mathrm{Dicg}_{\vartheta}^{1}$ or $\mathrm{Dicg}_{\vartheta}^{2}$ for encoding several video sequences. Note that due to the increase of the computation complexity (the dictionaries are non-separable



(a) $\gamma = 0°$. (b) $\gamma = 90°$.

(c) $\gamma = 112.5°$. (d) $\gamma = 157.5°$.

Figure F.5: Plots of $g_0(x, y)$ function with several γ rotations.

and have a cardinality about $20,000$) we have used for all experiments on this table only the 100 first frames of each video sequence. From this table we can see that the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ reductions were not significant (less than $1°$) for both dictionaries, and consequently the performance of the video encoder that uses these dictionaries was worse than the one using the separable Neff and Zakhor's dictionary.

Table F.4: Comparison, in terms of PSNR (dB) and the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles, between the performance of MPGBP video coder using the original and the proposed dictionaries.

Seq + Rate	$Dicg_{\mathcal{O}}^{1}$ - DicNeff			$Dicg_{\mathcal{O}}^{2}$ - DicNeff		
	$PSNR_{diff}$	$\Theta(\mathcal{C})_{diff}$	$\overline{\Theta}_{diff}$	$PSNR_{diff}$	$\Theta(\mathcal{C})_{diff}$	$\overline{\Theta}_{diff}$
Container24	-1.02	-0.81°	-0.26°	-1.09	-0.24°	-0.17°
Mother24	-0.98	-0.15°	-0.32°	-0.78	0.23°	-0.44°
Hall24	-1.73	0.25°	0.10°	-1.52	-0.11°	-0.13°
Silent24	-1.15	-0.30°	-0.01°	-0.89	-0.28°	-0.35°
Coast24	-0.59	-0.36°	-0.17°	-0.52	-0.34°	-0.50°

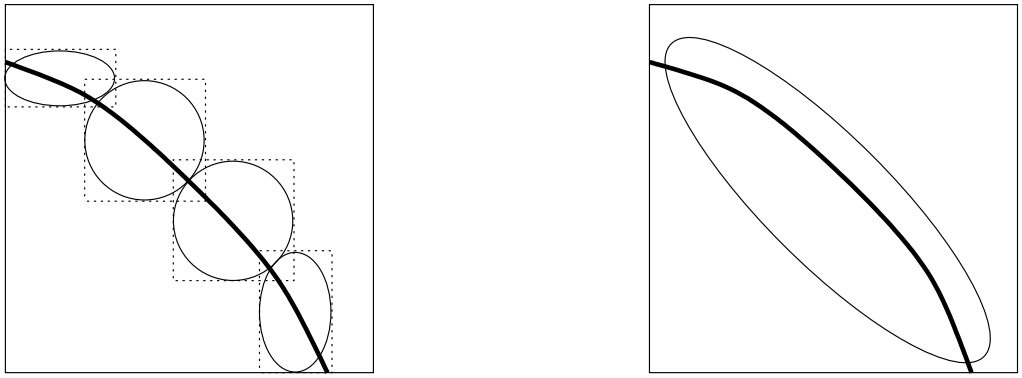
In spite of being non-separable, the dictionaries proposed in this section did not provided a reasonable reduction on the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles. In fact, the maximum reduction obtained was $0.80°$. This implies that rotations and anisotropic scalings of equation (F.9) do not generate functions with a good distribution in $\mathcal{R}^N$ (small $\Theta(\mathcal{C})$ and $\overline{\Theta}$).

F.2.2 A dictionary formed from rotations of the Neff and Zakhor's dictionary structures

In this section we have investigated the performance of the MPGBP video encoder using a non-separable dictionary formed from rotations of the Neff and Zakhor's dictionary structures. We expect that the non-separable structures thus generated reduce the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles of the original dictionary $\mathcal{C}$, leading to a better rate$\times$distortion trade-off. In fact, when we form a new dictionary by adding rotated versions of its separable structures, we are trying to fill the deep holes of the vector space, located near diagonal edges. Diagonal edges are considered be near deep holes of a separable dictionary since they tend to have low inner product with separable functions.

Another way of seeing this is by noting that, by adding rotated structures (non-

separable) in a separable dictionary, we can capture several contours and edges, which are not efficiently represented only by separable structures. This can be seen in figure F.6, where we have an illustration of curve edges represented using both separable and non-separable structures. Note that, using a separable dictionary (figure F.6a) more structures are needed for representing the edge than when we use a non-separable dictionary with structures in other directions (figure F.6b), besides the horizontal and vertical ones.



a) using separable structures

b) using non−separable structures

Figure F.6: Representation of a curve edge using separable or non-separable structures.

In order to generate the rotated structures for the proposed dictionary, we replace $(i - L/2 + 1)$ in equation (D.2) by the coordinates obtained from equation (F.6) with the scaling parameters a and b equal to one. This is so because the separable structures of the original Neff and Zakhor's dictionary already had their sizes appropriately scaled by the scaling parameter (s) in equation (D.2). Also, in this section we use a different approach for the choice of the rotation angles γ. It is based on the intuitive conjecture that structures of smaller support region do not need to be rotated by as many angles as the ones with larger support region in order to fill in the deep holes. Then, instead of rotating all 2-D structures by the same angles, as it was done in section F.2.1, we have used smaller angles for the larger structures and larger angles for the smaller ones [45]. We have chosen the rotation angles γ for each 2-D structure with size $l_x \times l_y$ as

$$\gamma = \arctan\left(\frac{2k}{L_{max} - 1}\right) \tag{F.12}$$

where k are integers in $\{-L_{max}/2, \cdots, L_{max}/2\}$ and L_{max} is the maximum between l_x and l_y.

Table F.5 verifies the conjecture above by showing the performance of the MPGBP

video encoder with different dictionaries formed using either the same rotations angles for all 2-D structures ($\gamma = n\pi$, where n is a real in $[0, \pi)$) or the rotations angles generated from equation (F.12). The dictionary labeled as $\mathrm{Dic1}$ was generated using rotations of approximately 2.5° of Neff and Zakhor's 2-D structures and has cardinality of 14764. The dictionary labeled as $\mathrm{Dic2}$ was formed using the rotations angles, according to equation (F.12). This dictionary has a cardinality of 8090 and was generated using equation (F.12). Also, we have used the same support region shown in figure F.4 with $R_{max} = 51$ for encoding the 300 first frames of the sequences Foreman at the rate 64kbps and Mother and Weather at the rate 24kbps. We can see from this table that despite having less structures (see equation (F.12)), the dictionary generated using rotation angles dependent of the structures sizes ($\mathrm{Dic2}$), obtains a performance similar to the one of the dictionary generated using the choice of the rotations angles independent of the structures sizes ($\mathrm{Dic1}$). Note that we obtain only a small reduction of $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles when the dictionary $\mathrm{Dic1}$ is used, suggesting that the dictionary $\mathrm{Dic2}$ has its vectors with a better distribution.

Table F.5: Performance of MPGBP video coder using dictionaries formed with different strategies for the γ angles choice.

Seq + Rate	Dic1			Dic2		
	PSNR (dB)	$\Theta(\mathcal{C})$	$\overline{\Theta}$	PSNR (dB)	$\Theta(\mathcal{C})$	$\overline{\Theta}$
Mother24	36.58	85.36°	80.91°	36.53	85.69°	81.03°
Weather24	32.20	86.47°	81.13°	32.25	86.62°	81.27°
Foreman64	33.86	86.06°	79.09°	33.93	85.94°	79.24°

In figure F.7 we can see the variation of the average PSNR with α for Mother, Container, Silent and Coast sequences at $24\mathrm{kbps}$ encoded with the MPGBP video encoder using a non-separable dictionary formed by rotations of the original Neff and Zakhor's dictionary. The rotations angles used were generated according equation (F.12). This dictionary has a cardinality of 8090 and uses the same support region shown in figure F.4 with $R_{max} = 51$ for encoding the 300 first of the video sequences. From this figure we can note that the performance of this version of the MPGBP video encoder varies with α like the one described in appendix D.

In table F.6 we compare the PSNR, $\Theta(\mathcal{C})$ and $\overline{\Theta}$ values obtained using the MPGBP

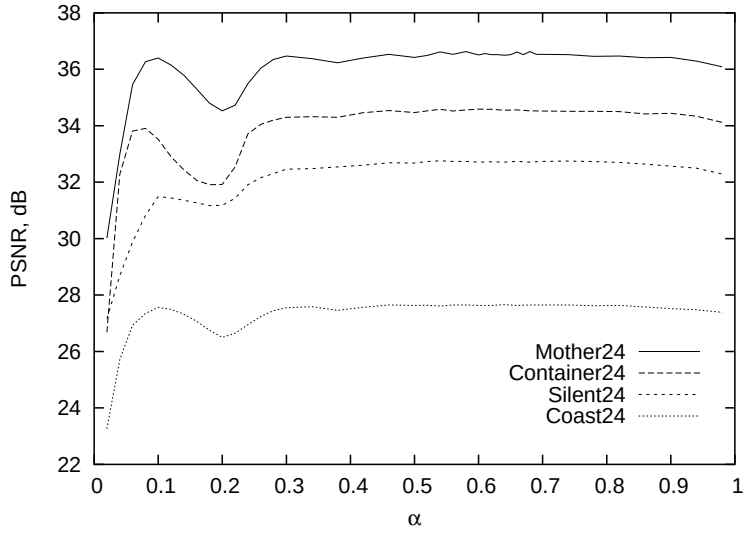


Figure F.7: Variation of the average PSNR with α parameter for several video sequences at 24kbps encoded using a dictionary generated by rotations.

video encoder with either the separable Neff and Zakhor's dictionary (labeled as "N") or the dictionary of rotations according equation (F.12) (labeled as "NR"). From this table we can see that the proposed dictionary provided, in general, reasonable $\Theta(\mathcal{C})$ and $\overline{\Theta}$ reductions. In fact, it compensates the increase in rate obtained when we increase the dictionary cardinality by decreasing $\Theta(\mathcal{C})$, thus, improving the performance of the video encoder. It is important to note that, the larger the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ reductions, the better is the performance of the MPGBP video encoder using the NR dictionary. The results show that whenever the $\overline{\Theta}$ reduction is grater than 1° we obtain an improvement in the performance of the MPGBP video encoder.

Figures F.8 and F.9 show the variation of the average PSNR with rate for the MPGBP video encoder using the the separable Neff and Zakhor's dictionary (labeled as "Dictionary N") or the dictionary of rotations according equation (F.12) (labeled as "Dictionary NR"). We can see from these figures that the use of the rotations in the original dictionary consistently improves the performance of MPGBP video encoder for all rates. Also, note that this improvement tends to increases with the bit rate. It is important to note that the results shown in table F.6 and figures F.8 and F.9 corroborate our conjecture, made based on the theoretical analysis in appendix C, that by reducing, the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles, even at the expense of increased cardinality, we can obtain better rate$\times$distortion trade-offs.

Table F.6: Comparison, in terms of PSNR (dB), between the MPGBP video encoder using the original Neff and Zakhor's dictionary and the proposed dictionary formed by rotations.

Seq + Rate	N (Neff)			NR (Neff + Rotation)			Difference (NR - N)		
	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}$	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}$	PSNR	$\Theta(\mathcal{C})$	$\overline{\Theta}$
Mother24	36.19	86.17º	82.29º	36.53	85.69º	81.03º	0.34	- 0.48º	- 1.26º
Weather24	31.75	87.52º	82.61º	32.25	86.62º	81.27º	0.50	- 0.90º	- 1.34º
Hall24	36.41	87.28º	81.62º	36.22	86.64º	80.68º	- 0.19	- 0.64º	- 0.94º
Silent24	32.59	87.08º	80.19º	32.74	86.29º	78.90º	0.15	- 0.79º	- 1.29º
Coast24	27.71	86.79º	81.45º	27.65	85.87º	80.51º	- 0.06	- 0.92º	- 0.94º
Mother64	40.38	87.19º	83.51º	40.72	86.51º	82.35º	0.34	- 0.68º	- 1.16º
Weather64	37.56	88.26º	83.60º	38.39	87.20º	82.34º	0.83	- 1.06º	- 1.26º
Hall64	39.87	87.69º	83.01º	39.85	86.97º	82.13º	- 0.02	- 0.72º	- 0.88º
Silent64	37.72	87.81º	82.25º	37.96	86.92º	81.14º	0.24	- 0.89º	- 1.11º
Foreman64	33.45	86.80º	80.77º	33.93	85.94º	79.25º	0.48	- 0.86º	- 1.52º
Coast64	31.33	87.32º	82.92º	31.36	86.29º	82.01º	0.03	- 1.03º	- 0.91º
Foreman96	35.52	87.07º	81.60º	36.01	86.34º	80.30º	0.49	- 0.73º	- 1.30º
Weather96	40.38	88.48º	84.00º	41.41	87.30º	82.72º	1.03	-1.18º	- 1.28º

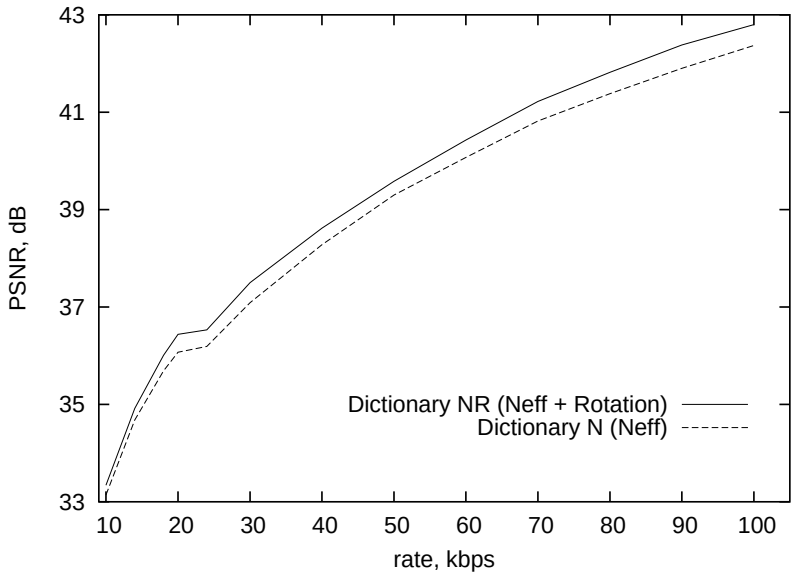


Figure F.8: Variation of the average PSNR with rate for Mother sequence.

In figure F.10 we compare the performance of the MPGBP video encoder for high bit-rates using both the separable Neff and Zakhor's dictionary (labeled as "Dictionary N") and the dictionary of rotations according to equation (F.12) (labeled as "Dictionary NR") with the De Vleeschouwer and Zakhor's state-of-the-art MP video encoder [7] for

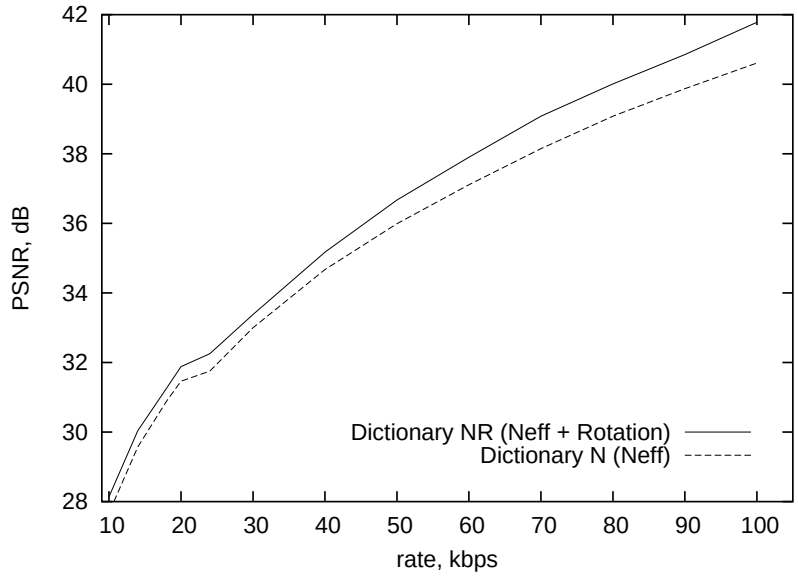


Figure F.9: Variation of the average PSNR with rate for Weather sequence.

60 frames of the Foreman sequence at 30fps. We can note from this figure that our results present a significant improvement over the ones in [7] (average gain of 1dB).

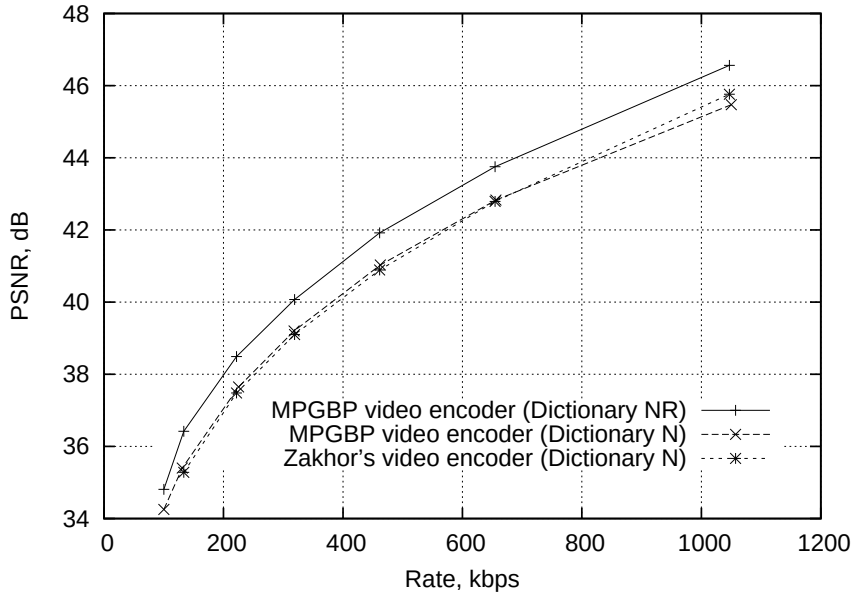


Figure F.10: The performance of the MPGBP video encoder using the original and the rotated dictionaries and the Zakhor's video encoder for Foreman sequence at high bit-rate with 30fps.

F.3 Conclusions

In this appendix we have investigated the performance of the MPGBP video encoder using different overcomplete dictionaries. We have used two strategies in order to choose the dictionary structures, one adding extra structures in the original Neff and Zakhor's dictionary and another using a new vector distribution of the dictionary. The results obtained by the dictionary formed from rotated versions of the separable Neff and Zakhor's dictionary structures are significantly superior to the ones obtained with the separable dictionary. One should note that an important aspect of the obtained results is that they corroborate the conjecture, made based on the theoretical analysis in appendix C, that by reducing the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles, even at the expense of increased cardinality, we can obtain better rate$\times$distortion trade-offs.

Apêndice G

Conclusions

This Thesis introduces a class of video compression methods based on a generalized bit-planes decomposition within a matching pursuits framework.

It begins by presenting, on appendix B, the matching pursuits algorithm and its main characteristics.

Next, in appendix C, the theory of signal decomposition in generalized bit-planes was presented and a novel algorithm (MPGBP algorithm) for performing matching pursuits decomposition based on this theory was proposed. The proposed algorithm has the advantage of obviating the need for setting up arbitrary trade-offs between number of atoms used and coefficient quantization, giving an easy solution for the quantization problem inherent to the traditional matching pursuits algorithm. In this appendix we also analyzed the theoretical bounds and the Rate-Distortion characteristics of the proposed algorithm using parameters of regular lattices as well as of a dictionary formed by the union of an orthogonal dictionary and its symmetric in relation to the origin. The results have shown that the reduction of the $\Theta(\mathcal{C})$ value of a dictionary $\mathcal{C}$ can lead to a better rate$\times$distortion trade-off, and, therefore to a reduction of the bit-rate spent in the encoding process.

In order to assess the performance of the MPGBP algorithm in video coding, in appendix D, it was implemented a video coder using the same framework of the Neff and Zakhor MP in [3, 4]. The MPGBP video encoder was used with different kinds of sequences and for a large variety of bit-rates, yielding consistent results. These results indicate a significant improvement over the classical video-MP algorithm [3, 4]. It also provides a performance comparable to that obtained by more sophisticated algorithms as,

for example, the ones in [10, 7].

In appendix E, it was proposed two rate-control strategies for improving the performance of the MPGBP video encoder. One used a rate-control procedure that divided precisely the bit-rate of the sequence equally among all its frames (fixed scheme) and the other used a rate-control scheme that employed Lagrangean optimization in order to solve the optimal bit allocation problem (Lagrangean scheme). The fixed scheme can be used to provide a fair comparison among the performance of video encoders, since one can obtain precisely a desired bit-rate. Its performance is comparable to those obtained using the state-of-the-art matching pursuits video encoder [7]. The scheme based on Lagrangean optimization presents just a small improvement in objective performance over the fixed rate one. Nevertheless it produces a smaller PSNR variation among the video sequence frames, and therefore yields a better subjective quality for the encoded video sequence.

In appendix F we investigated the performance of the MPGBP video encoder using different overcomplete dictionaries in order to assess the effect of these dictionaries on the performance of a video encoder. The results show that using a non-separable dictionary formed from rotated versions of the separable Neff and Zakhor's dictionary structures improves the performance of the MPGBP video encoder and corroborates the conjecture, that by reducing the $\Theta(\mathcal{C})$ and $\overline{\Theta}$ angles of a dictionary $\mathcal{C}$, even at the expense of an increase in cardinality, we can obtain better rate$\times$distortion trade-offs.

As suggestions for further work, it should be investigated the performance of the MPGBP algorithm in still images coding. In order to do so, we should investigate alternative frameworks [35, 36, 37, 38, 39, 40, 41, 42], to the one of Neff and Zakhor [3, 4]. In this context, we should also investigate dictionaries that give good representations for still images.

As another suggestion for further work, the MPGBP algorithm could be applied in video coding using more sophisticated dictionaries, as ones based on ridgelets [43, 44, 45, 46, 47, 48], curvelets [49, 50, 51], contourlets [52, 53].

Summarizing, one can conclude that the MPGBP algorithm provides an efficient scheme for video coding, and its characteristics deserve further research.

Apêndice H

Proof of Theorem 1

In this appendix we present the proof of theorem 1, stated in chapter C, that guarantees the convergence of the Matching Pursuits Generalized Bit planes (MPGBP) algorithm. We restate it here for convenience:

Theorem 1: *Be* $\mathbf{x} \in \mathbb{R}^N$, $\|\mathbf{x}\| \leq 1$, *such that it is approximated by MPGBP algorithm using a dictionary* $\mathcal{C}$ *with* P *steps, generating* $\mathbf{x}^{(P)}$ *as in equation (C.5), and be* $\Theta(\mathcal{C})$ *the largest angle between any signal* $\mathbf{y} \in \mathbb{R}^N$ *and the closest atom in dictionary* $\mathcal{C}$. *We have that* $\|\mathbf{r}^{(P)}\| = \|\mathbf{x} - \mathbf{x}^{(P)}\| \leq \beta_c^P$, *where* $\beta_c = \sqrt{1 - (2\alpha - \alpha^2)\cos^2(\Theta(\mathcal{C}))} < 1$ *for every* $0 < \alpha < 1$ *and* $0 \leq \Theta(\mathcal{C}) < \frac{\pi}{2}$.

Proof. We can represent the residual signal $\mathbf{r}^{(P)}$ in pass P as (see figure H.1)

$$\|\mathbf{r}^{(P)}\|^2 = \|\mathbf{r}^{(P-1)}\|^2 + \|\mathbf{t}\|^2 - 2\,\|\mathbf{r}^{(P-1)}\|\,\|\mathbf{t}\|\,\cos\theta \tag{H.1}$$

where θ is the angle between the residual $\mathbf{r}^{(P-1)}$ and its closest vector. Since we can assume to be using complete dictionaries, it can be said that $\theta \leq \Theta(\mathcal{C}) < \frac{\pi}{2}$.

Since $\alpha < 1$ then $\exists Q \in \mathbb{Z}$ such that (see figure H.1)

$$\alpha^{Q+1} \leq \|\mathbf{r}^{(P-1)}\|\cos\theta < \alpha^Q \tag{H.2}$$

Equation (H.2) can be reorganized as

$$\alpha\cos\theta < \frac{\alpha^{Q+1}}{\|\mathbf{r}^{(P-1)}\|} \leq \cos\theta \tag{H.3}$$

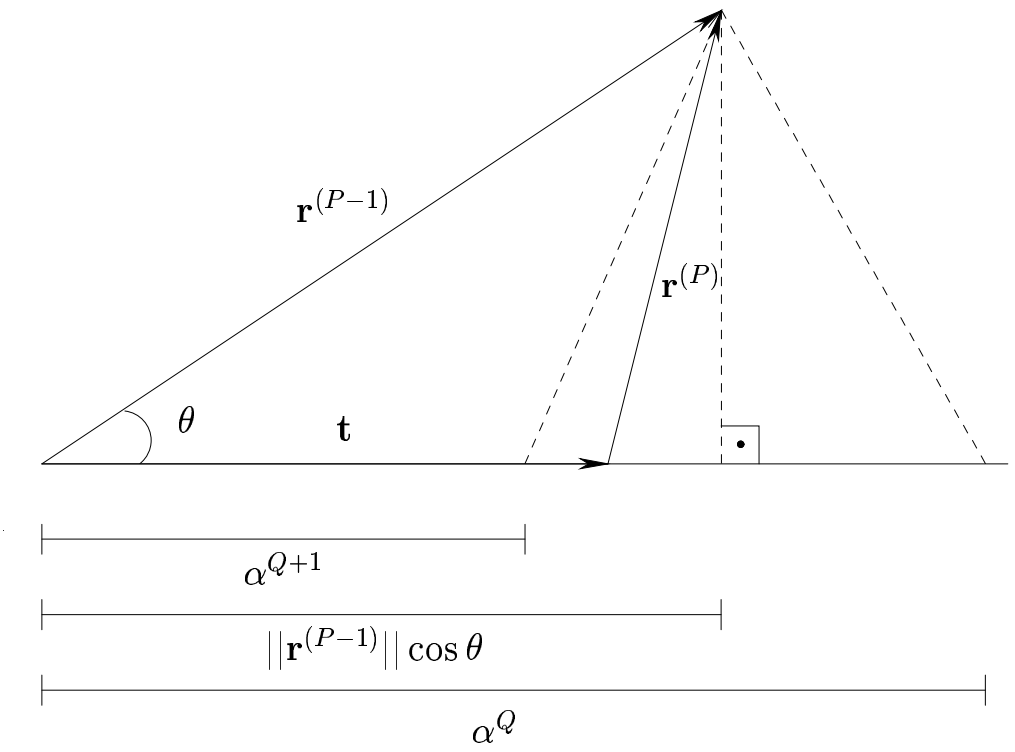


Figure H.1: Illustration of the signal approximation **x**.

According to the MPGBP algorithm (see page 62), we choose to **t** to be α^{Q+1}. In this case, the residual $\mathbf{r}^{(P)}$ becomes, from equation (H.1),

$$\|\mathbf{r}^{(P)}\|^2 = \|\mathbf{r}^{(P-1)}\|^2 + (\alpha^{Q+1})^2 - 2\,\|\mathbf{r}^{(P-1)}\|\;\alpha^{Q+1}\;\cos\theta \tag{H.4}$$

and consequently

$$\left(\frac{\|\mathbf{r}^{(P)}\|}{\|\mathbf{r}^{(P-1)}\|}\right)^2 = 1 + \left(\frac{\alpha^{Q+1}}{\|\mathbf{r}^{(P-1)}\|}\right)^2 - 2\,\frac{\alpha^{Q+1}}{\|\mathbf{r}^{(P-1)}\|}\;\cos\theta \tag{H.5}$$

From equations (H.3) and (H.5) we have that

$$\begin{aligned}\left(\frac{\|\mathbf{r}^{(P)}\|}{\|\mathbf{r}^{(P-1)}\|}\right)^2 &< 1 + \alpha^2\cos^2\theta - 2\alpha\cos^2\theta \\ &= 1 - \left(2\alpha - \alpha^2\right)\cos^2\theta = \beta^2(\theta)\end{aligned} \tag{H.6}$$

The equality occurs for $\dfrac{\alpha^{Q+1}}{\|\mathbf{r}^{(P-1)}\|} = \alpha\,\cos\theta$.

Since $0 < \alpha < 1$ and $|\cos\theta| \leq 1$, we have that the smallest $\beta^2(\theta)$ value (see equation (H.6)) is obtained when $\alpha \to 1$.

In figure H.2 we plot the value of $\beta^2(\theta)$ against α for $0 < \alpha < 1$.

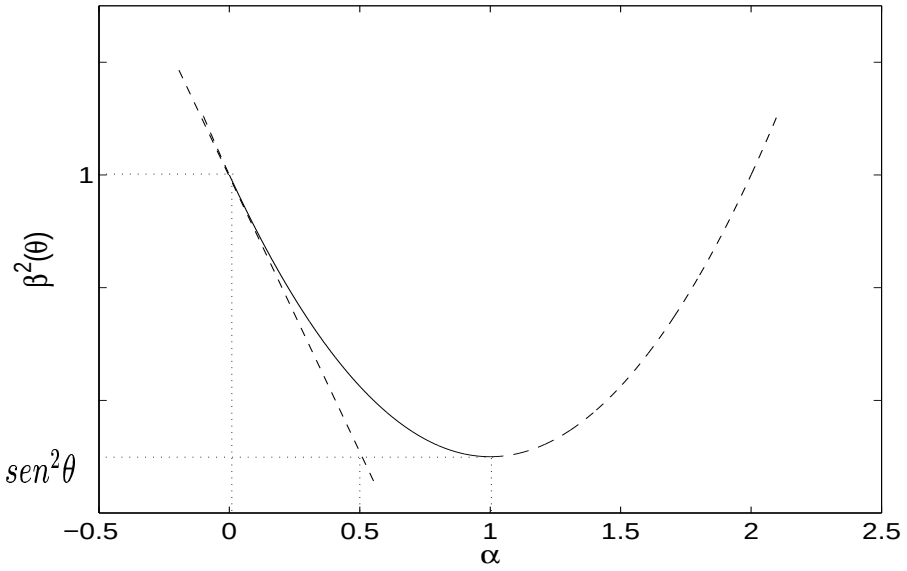


Figure H.2: Plot of $\beta^2(\theta)$ against α.

From equation (H.6), we have that the residual decreases its magnitude by at least $\beta(\theta)$ in each pass. Since $\theta \leq \Theta(\mathcal{C})$, and , for $\alpha \in (0,1)$, $(2\alpha - \alpha^2) > 0$, then we have that

$$\beta(\theta) \leq \sqrt{1 - (2\alpha - \alpha^2)\cos^2(\Theta(\mathcal{C}))} = \beta_c \tag{H.7}$$

Thus, since $\|\mathbf{x}\| \leq 1$, we can say that, after P passes, $\|\mathbf{r}^{(P)}\| \leq \beta_c^{(P)}$.

QED

Apêndice I

Derivation of $\Theta(\mathcal{U})$

In this appendix we present the derivation of the relations among the dimension $N(\mathcal{U})$, the cardinality $q(\mathcal{U})$ and $\Theta(\mathcal{U})$ for the dictionary $\mathcal{U}$ defined in section C.3.2. This dictionary was defined as

$$\mathcal{U} = \mathcal{U}_{\text{ort}} \cup -\mathcal{U}_{\text{ort}} \tag{I.1}$$

where we have $\mathcal{U}_{\text{ort}} = \{1,0,\cdots,0\} \cup \{0,1,0\cdots,0\} \cup \{0,\cdots,1,0\} \cup \{0,\cdots,0,1\}$.

Then, the dictionary $\mathcal{U}$ can be written as

$$\mathcal{U} = \{\pm 1,0,\cdots,0\} \cup \{0,\pm 1,0\cdots,0\} \cup \{0,\cdots,\pm 1,0\} \cup \{0,\cdots,0,\pm 1\} \tag{I.2}$$

Since the dictionary $\mathcal{U}$ is formed by the union of two orthogonal dictionaries of cardinality N, the cardinality of this dictionary is the sum of the cardinalities of both orthogonal dictionaries, and can be expressed by

$$q(\mathcal{U}) = N + N = 2N \tag{I.3}$$

Now, in order to obtain a relation for $\Theta(\mathcal{U})$, we will consider a vector $\mathbf{v} = (v_1, v_2, \cdots, v_N)$ with $|\mathbf{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$. The largest inner product of the vector v with a vector u from the dictionary $\mathcal{U}$ is given by $\max_j \{< \mathbf{v} \cdot \mathbf{u}_j >\} = \max_j \{|v_j|\}$. From this we can obtain the maximum value of $\cos(\Theta)$ given $\mathbf{v}$ as (note that $\|\mathbf{u}_j\| = 1$)

$$\cos(\Theta) = \frac{\max_j\{< \mathbf{v}, \mathbf{u}_j >\}}{|\mathbf{v}||\mathbf{u}_j|} \leq \frac{\max_j\{|v_j|\}}{\sqrt{\sum_{i=1}^{N} v_i^2}} = \frac{1}{\sqrt{\sum_{i=1}^{N} \frac{v_i^2}{\max_j\{|v_j|^2\}}}} \tag{I.4}$$

We have that for the $\mathbf{v}$ which gives the maximum Θ, then $\cos(\Theta)$ is minimum, and therefore $\left\{ \sum_{i=1}^{N} \frac{v_i^2}{\max_j\{|v_j|^2\}} \right\}$ is maximum.
Since

$$\frac{v_i^2}{\max\{|v_j|^2\}} \leq 1 \tag{I.5}$$

We have that

$$\sum_{i=1}^{N} \frac{v_i^2}{\max\{|v_j|^2\}} \leq N \tag{I.6}$$

From equations (I.4) and (I.6) we have that

$$\cos(\Theta) \geq \frac{1}{\sqrt{N}} \tag{I.7}$$

Since, for $v_i = \{1, \cdots, 1\}$ we have that

$$\cos(\Theta) = \frac{1}{\sqrt{N}} \tag{I.8}$$

Then, $\cos(\Theta(\mathcal{U})) = \frac{1}{\sqrt{N}}$

QED

Apêndice J

Measure of Non-Separability $\mathcal{S}$

We have that any separable matrix $\mathbf{X}_{N\times M}$ formed from the vectors $\mathbf{a}_{N\times 1}$ and $\mathbf{b}_{M\times 1}$ can be written as

$$\mathbf{X} = \mathbf{a}\mathbf{b}^{\top} = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} \begin{bmatrix} b_1 & \cdots & b_M \end{bmatrix} = \begin{bmatrix} a_1\mathbf{b}^{\top} \\ a_2\mathbf{b}^{\top} \\ \vdots \\ a_N\mathbf{b}^{\top} \end{bmatrix} \tag{J.1}$$

Using $\mathbf{X}$ defined by equation (J.1), we have

$$\mathbf{X}\mathbf{X}^{\top} = \begin{bmatrix} a_1\mathbf{b}^{\top} \\ a_2\mathbf{b}^{\top} \\ \vdots \\ a_N\mathbf{b}^{\top} \end{bmatrix} \begin{bmatrix} a_1\mathbf{b} & a_2\mathbf{b} & \cdots & a_N\mathbf{b} \end{bmatrix} \tag{J.2}$$

$$= \begin{bmatrix} a_1^2\mathbf{b}^{\top}\mathbf{b} & a_1a_2\mathbf{b}^{\top}\mathbf{b} & \cdots & a_1a_N\mathbf{b}^{\top}\mathbf{b} \\ a_2a_1\mathbf{b}^{\top}\mathbf{b} & a_2^2\mathbf{b}^{\top}\mathbf{b} & \cdots & a_2a_N\mathbf{b}^{\top}\mathbf{b} \\ \vdots & \ddots & \vdots & \vdots \\ a_Na_1\mathbf{b}^{\top}\mathbf{b} & a_Na_2\mathbf{b}^{\top}\mathbf{b} & \cdots & a_N^2\mathbf{b}^{\top}\mathbf{b} \end{bmatrix} \tag{J.3}$$

$$= \begin{bmatrix} a_1^2\|\mathbf{b}\|^2 & \cdots & a_1 a_N\|\mathbf{b}\|^2 \\ a_1 a_2\|\mathbf{b}\|^2 & \cdots & a_2 a_N\|\mathbf{b}\|^2 \\ \vdots & \ddots & \vdots \\ a_1 a_N\|\mathbf{b}\|^2 & \cdots & a_N^2\|\mathbf{b}\|^2 \end{bmatrix} \tag{J.4}$$

$$= \begin{bmatrix} k_1\mathbf{a} & \cdots & k_N\mathbf{a} \end{bmatrix} \tag{J.5}$$

where $k_i = a_i\|\mathbf{b}\|^2$.

If we normalize each column of $\mathbf{X}\mathbf{X}^T$ we obtain

$$\left(\mathbf{X}\mathbf{X}^T\right)_{\text{Norm}} = \begin{bmatrix} \frac{\mathbf{a}}{\|\mathbf{a}\|} & \frac{\mathbf{a}}{\|\mathbf{a}\|} & \cdots & \frac{\mathbf{a}}{\|\mathbf{a}\|} \end{bmatrix} \tag{J.6}$$

In similar way, noting that $\mathbf{X} = \mathbf{a}\mathbf{b}^T = \begin{bmatrix} b_1\mathbf{a} & b_2\mathbf{a} & \cdots & b_M\mathbf{a} \end{bmatrix}$ and normalizing each line of $\mathbf{X}^T\mathbf{X}$, we have

$$\left(\mathbf{X}^T\mathbf{X}\right)_{\text{Norm}} = \begin{bmatrix} \frac{\mathbf{b}}{\|\mathbf{b}\|} \\ \frac{\mathbf{b}}{\|\mathbf{b}\|} \\ \vdots \\ \frac{\mathbf{b}}{\|\mathbf{b}\|} \end{bmatrix} \tag{J.7}$$

From equations (J.6) and (J.7) we can obtain the normalized **a** and **b** that form the separable matrix **X**. Note that if **X** is separable, the rows and the columns in equations (J.6) and (J.7) are equal, respectively. Otherwise, if **X** is non-separable, we have at least two rows and columns of these equations with different values, respectively. For a non-separable **X**, these equations can be written as

$$\left(\mathbf{X}\mathbf{X}^T\right)_{\text{Norm}} = \begin{bmatrix} \tilde{\mathbf{a}}_1 & \cdots & \tilde{\mathbf{a}}_N \end{bmatrix} \tag{J.8}$$

$$\left(\mathbf{X}^T\mathbf{X}\right)_{\text{Norm}} = \begin{bmatrix} \tilde{\mathbf{b}}_1 \\ \vdots \\ \tilde{\mathbf{b}}_M \end{bmatrix} \tag{J.9}$$

where $\tilde{\mathbf{a}}_i$ and $\tilde{\mathbf{b}}_i$ are the values obtained in each rows and columns of equations (J.6) and (J.7), respectively.

Also, we have that

$$
\begin{aligned}
\left(\mathbf{X}\mathbf{X}^{\intercal}\right)_{\text{Norm}}^{\intercal}\left(\mathbf{X}\mathbf{X}^{\intercal}\right)_{\text{Norm}} &= \begin{bmatrix} \tilde{\mathbf{a}}_1 \\ \vdots \\ \tilde{\mathbf{a}}_N \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{a}}_1 & \cdots & \tilde{\mathbf{a}}_N \end{bmatrix} \\
&= \begin{bmatrix} 1 & \tilde{\mathbf{a}}_1^{\intercal}\tilde{\mathbf{a}}_2 & \cdots & \tilde{\mathbf{a}}_1^{\intercal}\tilde{\mathbf{a}}_N \\ \tilde{\mathbf{a}}_2^{\intercal}\tilde{\mathbf{a}}_1 & 1 & \tilde{\mathbf{a}}_2^{\intercal}\tilde{\mathbf{a}}_3 & \cdots \\ \vdots & \ddots & 1 & \ddots \\ \tilde{\mathbf{a}}_{N-1}^{\intercal}\tilde{\mathbf{a}}_1 & & \cdots & 1 \\ \tilde{\mathbf{a}}_N^{\intercal}\tilde{\mathbf{a}}_1 & & \cdots & \end{bmatrix}
\end{aligned} \tag{J.10}
$$

$$
\begin{aligned}
\left(\mathbf{X}^{\intercal}\mathbf{X}\right)_{\text{Norm}}\left(\mathbf{X}^{\intercal}\mathbf{X}\right)_{\text{Norm}}^{\intercal} &= \begin{bmatrix} \tilde{\mathbf{b}}_1 \\ \vdots \\ \tilde{\mathbf{b}}_M \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{b}}_1^{\intercal} & \cdots & \tilde{\mathbf{b}}_M^{\intercal} \end{bmatrix} \\
&= \begin{bmatrix} 1 & \tilde{\mathbf{b}}_1\tilde{\mathbf{b}}_2^{\intercal} & \cdots & \tilde{\mathbf{b}}_1\tilde{\mathbf{b}}_M^{\intercal} \\ \tilde{\mathbf{b}}_2\tilde{\mathbf{b}}_1^{\intercal} & 1 & \tilde{\mathbf{b}}_2\tilde{\mathbf{b}}_3^{\intercal} & \cdots \\ \vdots & \ddots & 1 & \ddots \\ \tilde{\mathbf{b}}_{M-1}\tilde{\mathbf{b}}_1^{\intercal} & & \cdots & 1 \\ \tilde{\mathbf{b}}_M\tilde{\mathbf{b}}_1^{\intercal} & & \cdots & \end{bmatrix}
\end{aligned} \tag{J.11}
$$

The separability of $\mathbf{X}$ can be evaluated computing the mean of the square for all coefficients above the main diagonal of the matrices showed in equations (J.10) and (J.11). It is the desired measure of non-separability $\mathcal{S}$, and can be computed by

$$
\mathcal{S} = \frac{\displaystyle\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(a_i^{\intercal}a_j)^2 + \sum_{i=1}^{M-1}\sum_{j=i+1}^{M}(b_i b_j^{\intercal})^2}{\dfrac{(N-1)N+(M-1)M}{2}} \tag{J.12}
$$

Since a separable matrix has its them vectors $\tilde{\mathbf{a}}_1 = \tilde{\mathbf{a}}_2 = \cdots = \tilde{\mathbf{a}}_N$ and $\tilde{\mathbf{b}}_1 = \tilde{\mathbf{b}}_2 = \cdots = \tilde{\mathbf{b}}_M$, we have all values above the main diagonal equal to one ($\tilde{\mathbf{a}}_i^{\intercal}\tilde{\mathbf{a}}_j = 1$ and $\tilde{\mathbf{b}}_i\tilde{\mathbf{b}}_j^{\intercal} = 1$). This produces $\mathcal{S} = 1$. A non-separable matrix does not have all $\tilde{\mathbf{a}}_i$ and/or all $\tilde{\mathbf{b}}_i$. It produces for some vectors $\tilde{\mathbf{a}}_i^{\intercal}\tilde{\mathbf{a}}_j < 1$ and/or $\tilde{\mathbf{b}}_i\tilde{\mathbf{b}}_j^{\intercal} < 1$, and consequently, $\mathcal{S}$ varying in the range $[0,1)$.

Apêndice K

Original Sequences

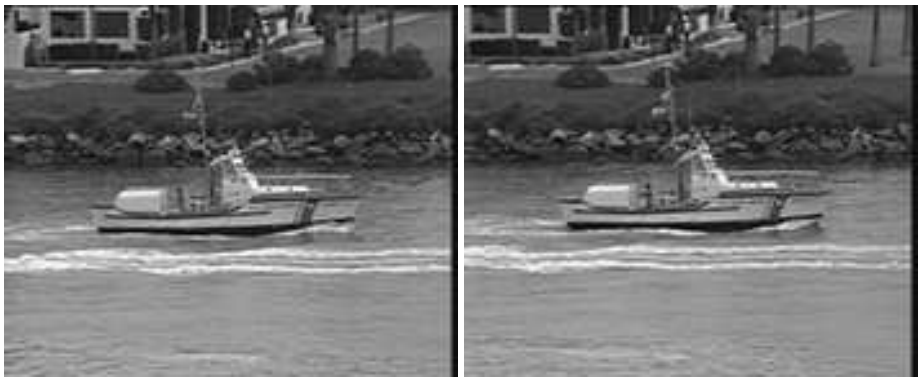
In this appendix we show the luminance of frames 000, 020, 040, 060, 080, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280 and 299 of the original sequences: Coast-guard, Container, Foreman, Hall-monitor, Mother-and-daughter, Silent-voice and Weather. These sequences have 300 QCIF frames (176×144) with 8bit/pixel.

Coast 000 Coast 020

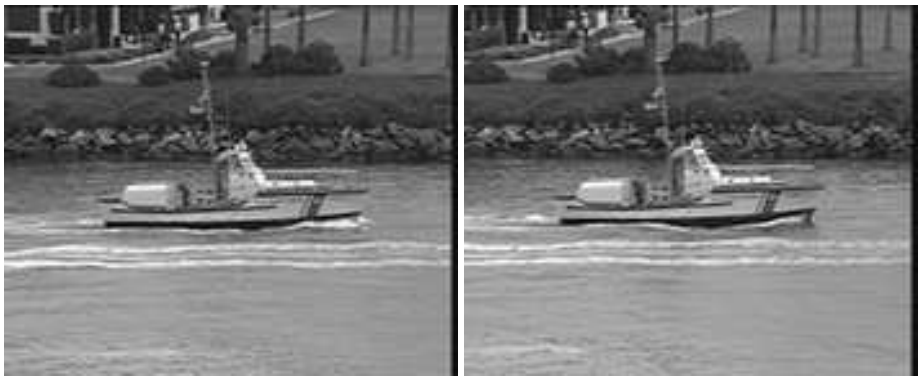
Coast 040 Coast 060

Coast 080 Coast 100

Figure K.1: Frames 000, 020, 040, 060, 080 and 100 of the original *Coast-guard* sequence.



Coast 120 Coast 140



Coast 160 Coast 180

Coast 200 Coast 220

Figure K.2: Frames 120, 140, 160, 180, 200 and 220 of the original *Coast-guard* sequence.

Coast 240 Coast 260

Coast 280 Coast 299

Figure K.3: Frames 240, 260, 280 and 299 of the original *Coast-guard* sequence.

Container 000 Container 020

Container 040 Container 060

Container 080 Container 100

Figure K.4: Frames 000, 020, 040, 060, 080 and 100 of the original *Container* sequence.

Container 120 Container 140

Container 160 Container 180

Container 200 Container 220

Figure K.5: Frames 120, 140, 160, 180, 200 and 220 of the original *Container* sequence.

Container 240 Container 260

Container 280 Container 299

Figure K.6: Frames 240, 260, 280 and 299 of the original *Container* sequence.

Foreman 000 Foreman 020

Foreman 040 Foreman 060

Foreman 080 Foreman 100

Figure K.7: Frames 000, 020, 040, 060, 080 and 100 of the original *Foreman* sequence.

Foreman 120 Foreman 140

Foreman 160 Foreman 180

Foreman 200 Foreman 220

Figure K.8: Frames 120, 140, 160, 180, 200 and 220 of the original *Foreman* sequence.

Foreman 240 Foreman 260

Foreman 280 Foreman 299

Figure K.9: Frames 240, 260, 280 and 299 of the original *Foreman* sequence.

Hall 000 Hall 020

Hall 040 Hall 060

Hall 080 Hall 100

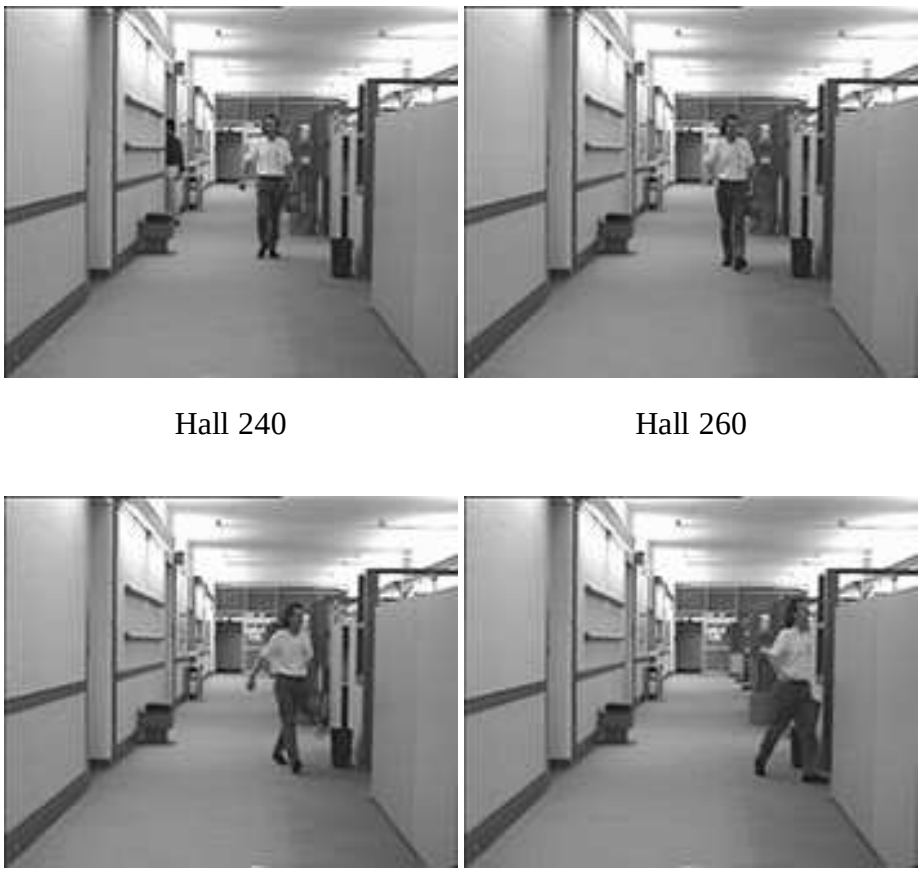
Figure K.10: Frames 000, 020, 040, 060, 080 and 100 of the original *Hall-monitor* sequence.

Hall 120 Hall 140

Hall 160 Hall 180

Hall 200 Hall 220

Figure K.11: Frames 120, 140, 160, 180, 200 and 220 of the original *Hall-monitor* sequence.



Hall 240 Hall 260

Hall 280 Hall 299

Figure K.12: Frames 240, 260, 280 and 299 of the original *Hall-monitor* sequence.

Mother 000 Mother 020

Mother 040 Mother 060

Mother 080 Mother 100

Figure K.13: Frames 000, 020, 040, 060, 080 and 100 of the original *Mother-and-daughter* sequence.

Mother 120 Mother 140

Mother 160 Mother 180

Mother 200 Mother 220

Figure K.14: Frames 120, 140, 160, 180, 200 and 220 of the original *Mother-and-daughter* sequence.

Mother 240 Mother 260

Mother 280 Mother 299

Figure K.15: Frames 240, 260, 280 and 299 of the original *Mother-and-daughter* sequence.

Silent 000 Silent 020

Silent 040 Silent 060

Silent 080 Silent 100

Figure K.16: Frames 000, 020, 040, 060, 080 and 100 of the original *Silent-voice* sequence.

Silent 120 Silent 140

Silent 160 Silent 180

Silent 200 Silent 220

Figure K.17: Frames 120, 140, 160, 180, 200 and 220 of the original *Silent-voice* sequence.

Silent 240 Silent 260

Silent 280 Silent 299

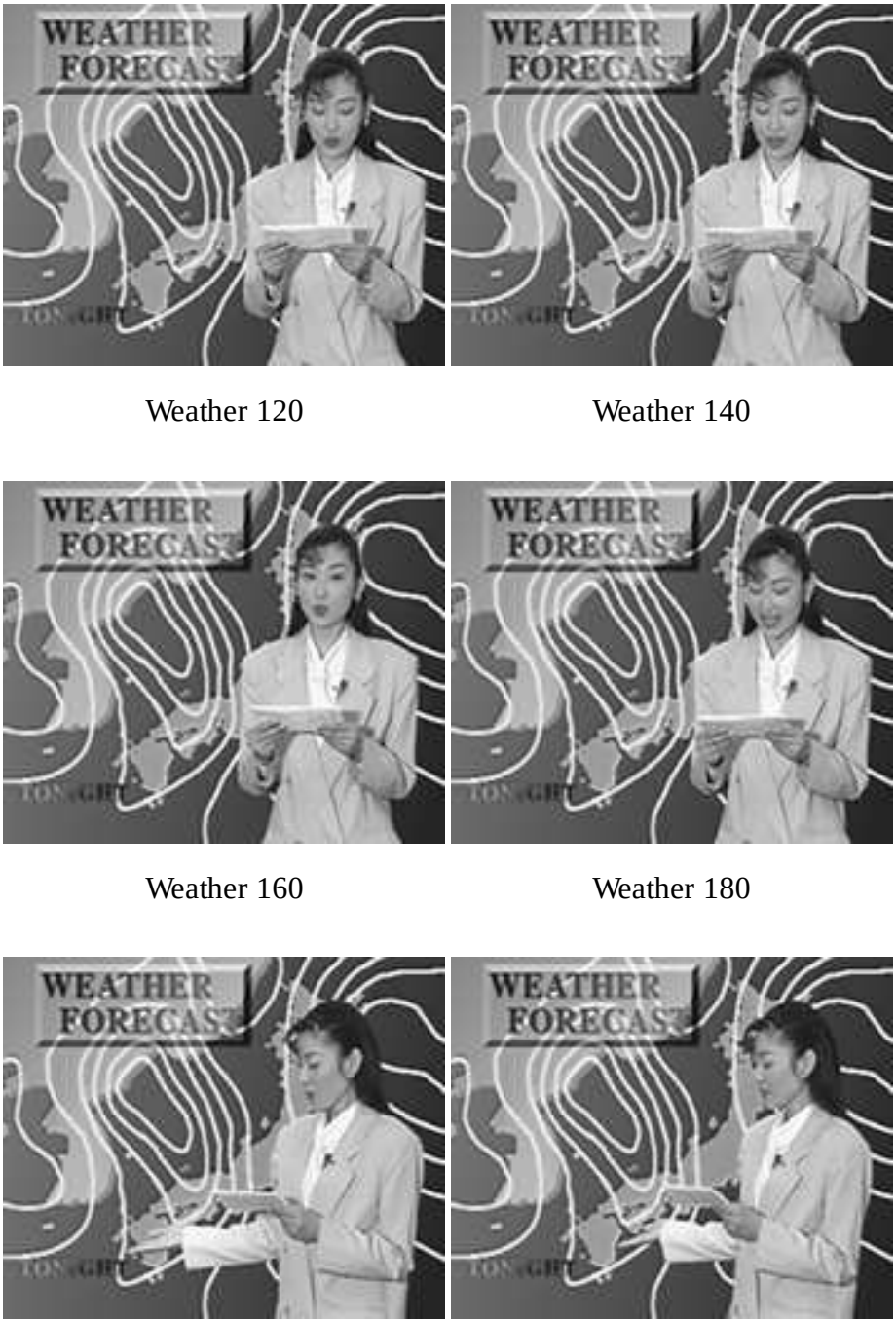
Figure K.18: Frames 240, 260, 280 and 299 of the original *Silent-voice* sequence.

Weather 000 Weather 020

Weather 040 Weather 060

Weather 080 Weather 100

Figure K.19: Frames 000, 020, 040, 060, 080 and 100 of the original *Weather* sequence.



Weather 120 Weather 140

Weather 160 Weather 180

Weather 200 Weather 220

Figure K.20: Frames 120, 140, 160, 180, 200 and 220 of the original *Weather* sequence.

Weather 240

Weather 260

Weather 280

Weather 299

Figure K.21: Frames 240, 260, 280 and 299 of the original *Weather* sequence.

Referências Bibliográficas

[1] NETRAVALI, A., HASKELL, B., *Digital Pictures. Representation and compression*. 2 ed. Plenum Press, 1995.

[2] MALLAT, S. G., ZHANG, Z., "Matching Pursuits with Time-Frequency Dictionaries", *IEEE Transactions on Signal Processing*, v. 41, n. 12, pp. 3397–3415, December 1993.

[3] NEFF, R., ZAKHOR, A., "Very Low Bit Rate Video Coding Based in Matching Pursuits", *IEEE Transactions Circuits and Systems*, v. 7, n. 1, pp. 158–171, February 1997.

[4] NEFF, R., ZAKHOR, A., "Adaptive Modulus Quantizer Design for Matching Pursuits Video Coding", *IEEE International Conference on Image Processing*, v. 2, pp. 81–85, 1999.

[5] AL-SHAYKH, O. K., MILOSLAVSKY, E., NOMURA, T., *et al.*, "Video Compression Using Matching Pursuits", *IEEE Transactions Circuits and Systems For Video Technology*, v. 9, n. 1, pp. 123–143, February 1999.

[6] NEFF, R., ZAKHOR, A., "Dictionary Approximation For Matching Pursuit Video Coding". In: *2000 IEEE International Conference on Image Processing*, v. 2, pp. 828–831, October 2000.

[7] VLEESCHOUWER, C. D., ZAKHOR, A., "In-Loop Atom Modulus Quantization for Matching Pursuits Video Coding and its Application to Video Coding", *IEEE Transactions on Image Processing*, v. 12, n. 10, pp. 1226–1242, October 2003.

[8] NEFF, R., NOMURA, T., ZAKHOR, A., "Decoder Complexity and Performance Comparison of Matching Pursuit and DCT-based MPEG-4 Video Codecs", *International Conference on Image Processing*, v. 1, pp. 783–787, October 1998.

[9] TANG, X., ZAKHOR, A., "Matching pursuits multiple description coding for wireless video", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 12, n. 6, pp. 566–575, June 2002.

[10] NEFF, R., ZAKHOR, A., "Modulus Quantization for Matching Pursuits Video Coding", *IEEE Transactions Circuits and Systems for Video Technology*, v. 10, pp. 895–912, 2000.

[11] VANDERGHEYNST, P., FROSSARD, P., "Adaptive Entropy-Constrained Matching Pursuits Quantization", *IEEE International Conference on Image Processing*, pp. 423–426, 2001.

[12] RAMCHANDRAN, K., ORTEGA, A., VETTERLI, M., "Bit Allocation for Dependent Quantization with Applications to Multiresolution and MPEG Video Coders", *IEEE Transactions on Image Processing*, v. 3, n. 5, pp. 533–545, September 1994.

[13] HSU, C.-Y., ORTEGA, A., REIBMAN, A., "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints", *IEEE J. Select. Areas Commun.*, v. 15, n. 6, pp. 1016–1027, August 1997.

[14] CHEN, J.-J., LIN, D., "Optimal bit allocation for coding of video signals over ATM networks", *IEEE J. Select. Areas Commun*, v. 15, pp. 1002–1015, August 1997.

[15] RAMCHANDRAN, K., ORTEGA, A., "Rate-Distortion Methods for Image and Video Compression", *IEEE Signal Processing*, v. 15, n. 6, pp. 23–50, November 1998.

[16] LIN, L.-J., ORTEGA, A., "Bit-Rate Control Using Piecewise Approximated Rate-Distortion Characteristcs", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 8, n. 4, pp. 446–459, August 1998.

[17] LIN, L.-J., ORTEGA, A., "Bit-rate control using piecewise approximated rate-distortion characteristics", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 9, pp. 446–459, August 1998.

[18] CAETANO, R., da SILVA, E. A. B., "Rate Control Strategy for Embedded Wavelet Video coders", *Electronics Letters*, v. 35, n. 21, pp. 1815–1817, October 1999.

[19] CAETANO, R., da SILVA, E. A. B., "A Bit Allocation Scheme for a Class of Embedded Wavelet Video Encoders", *Journal of Visual Communication and Image Representation*, v. 14, pp. 136–149, 2003.

[20] PENG, C.-K., HWANG, W.-L., HUANG, C.-L., "Matching pursuits low bit rate video coding with codebooks adaptation", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 1, pp. 408 – 411, June 2000.

[21] MOSCHETTI, F., GRANAI, L., VANDERGHEYNST, P., *et al.*, "New Dictionary and Fast Atom searching Method for Matching Pursuits Representation of Displaced Frame Difference". In: *2002 IEEE International Conference on Image Processing*, v. 3, pp. 685–688, October 2002.

[22] CZEREPIŃSKI, P., DAVIES, C., CANAGARAJAH, N., *et al.*, "Matching Pursuits Video Coding: Dictionaries and Fast Implementation", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 10, n. 7, pp. 1103–1115, October 2000.

[23] NEFF, R.; ZAKHOR, A., "Matching pursuit video coding. Part I. Dictionary approximation", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 12, pp. 13–26, January 2002.

[24] da SILVA, E. A. B., FONINI Jr., D. A., CRAIZER, M., "Successive approximation quantization for image compression", *IEEE Circuits and Systems Magazine*, v. 2, n. 3, pp. 20–45, 2002.

[25] CONWAY, J. H., SLOANE, N. J. A., *Sphere Packings, Lattices and Groups*. New York, Springer-Verlag, 1988.

[26] SLOANE, N. J. A., "Tables of Sphere Packings and Spherical Codes", *IEEE Transactions on Information Theory*, v. IT-27, pp. 327–338, 1981.

[27] ISO/IEC JTC1/SC29/WG1 (ITU/T SG28), "JPEG2000 Verification Model 5.3", 1999.

[28] SHAPIRO, J. M., "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 41, n. 12, pp. 3445–3462, December 1993.

[29] SAID, A., PEARLMAN, W. A., "A New, Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 6, n. 3, pp. 243–250, June 1996.

[30] ISO/IEC JTC1/SC29/WG11, "MPEG-4 Video Verification Model Version 11.0", March 1998.

[31] JAIN, A. K., *Fundamentals of Digital Image Processing*. Englewood Cliffs, New Jersey, Prentice Hall, 1989.

[32] CHEUNG, S.-C. S., ZAKHOR, A., "Matching Pursuits Experimental Video Codec". URL: `http://www-video.eecs.berkeley.edu/~cheungsc/work/docs/mpsoftare.pdf`.

[33] SHOHAM, Y., GERSHO, A., "Efficient Bit Allocation for an Arbitrary Set of Quantizes", *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 36, n. 9, pp. 1445–1453, September 1988.

[34] EVERETT III, H., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", *Operations Research*, v. 11, pp. 399–417, 1963.

[35] AIAZZI, B., ALPARONE, L., BARONTI, S., "Fuzzy Logic-Based Matching Pursuits for Lossless Predictive Coding of Still Images", *IEEE Transactions Fuzzy Systems*, v. 10, n. 4, pp. 473–483, August 2002.

[36] COTTER, S. F., RAO, B. D., "Application of Tree-Based Searches to Matching Pursuit", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 6, pp. 3933 – 3936, May 2001.

[37] MARUSIC, B., SKOCIR, P., TASIC, J., "A Matching Pursuit Enhanced Three-Dimensional Wavelet Transform Code", *10th Mediterranean Electrotechnical Conference*, v. 2, pp. 482 – 485, May 2000.

[38] de CARVALHO, M. B., da SILVA, E. A. B., FINAMORE, W. A., *et al.*, "Universal Multi-Scale Matching Pursuits Algorithm With Reduced Blocking Effect", *International Conference on Image Processing*, v. 3, pp. 8533–856, September 2000.

[39] HUI, L., WOLF, I., "Multiscale Matching Pursuit for Image Coding", *ISSPA '99. Proceedings of the Fifth International Symposium on Signal Processing and Its Applications*, v. 2, pp. 805–808, October 1999.

[40] RABIEE, H. R., KASHYAP, R. L., SAFAVIAN, S. R., "Scalable Subband Image Coding with Segmented Orthogonal Matching Pursuit", *International Conference on Image Processing*, v. 2, pp. 233–236, October 1998.

[41] HOROWITZ, M. J., NEUHOFF, D. L., "Image Coding by Matching Pursuit and Perceptual Pruning", *International Conference on Image Processing*, v. 3, pp. 654–657, October 1997.

[42] BERGEAUD, F., MALLAT, S., "Matching Pursuit of Images", *International Conference on Image Processing*, v. 1, pp. 53–56, October 1995.

[43] DONOHO, D. L., *Orthonormal Ridgelets and Linear Singularities*, Report, Stanford University, 1998.

[44] CANDÈS, E. J., DONOHO, D. L., "Ridgelets: a key to higher-dimensional intermittency?", *Phil. Trans. R. Soc. Lond. A.*, v. 357, pp. 2495–2509, June 1999.

[45] DONOHO, D. L., FLESIA, A. G., "Digital Ridgelet Transform based on True Ridge Functions", January 2002. URL: `http://www-stat.stanford.edu/~donoho/Reports/2001/DigRidgeTrue.pdf`.

[46] GRANAI, L., MOSCHETTI, F., VANDERGHEYNST, P., "Ridgelet Transform Applied to Motion Compensated Images", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, v. 3, pp. 381–384, April 2003.

[47] DO, M. N., VETTERLI, M., "The Finite Ridgelet Transform for Image Representation", *IEEE Transaction on Image Processing*, v. 12, pp. 16–28, January 2003.

[48] DO, M. N., VETTERLI, M., "Orthogonormal Finite Ridgelet Transform for Image Processing", *IEEE International Conference on Image Processing*, v. 2, pp. 367–370, September 2000.

[49] CANDÈS, E. J., DONOHO, D. L., "Curvelets - A Surprisingly Effective Nonadaptive Representation For Objects with Edges", *In A. Cohen, C. Rabut and L. L. Schumaker, editors, Curve and Surface Fitting, Saint-Malo, Vanderbilt University Press*, , 1999.

[50] STARCK, J.-L., CANDÈS, E. J., DONOHO, D. L., "The Curvelet Transform for Image Denoising", *IEEE Transactions on Image Processing*, v. 11, n. 6, pp. 670–684, June 2002.

[51] CANDÈS, E. J., DONOHO, D. L., "Curvelets – A Suprisingly Effective Nonadaptive Representation for Objects with Edges", *Curve and Surface Ditting, A. Cohen, C. Rabut, and L. L. Schumaker, Eds., Saint-Malo, Vanderbilt University Press*, .

[52] DO, M. N., VETTERLI, M., "Contourlets: a directional multiresolution image representation", *IEEE International Conference on Image Processing*, v. 1, pp. 357 – 360, September 2002.

[53] WELLAND, G., *Beeyond Wavelets*. Academic Press, 2003.

[54] GOYAL, V. K., VETTERLI, M., THAO, N. T., "Quantized Overcomplete Expansions in $\mathcal{R}^N$: Analysis, Sinthesis, and Algorithms", *IEEE Transactions on Information Theory*, v. 44, n. 1, pp. 16–31, January 1998.

[55] MALLAT, S., *A Wavelet Tour of Signal Processing*. San Diego, California, Academic Press, 1998.

[56] RAO, K. R., YIP, P., *Discrete Cossine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990.

[57] ANTONINI, M., BARLAUD, M., MATHIEU, P., *et al.*, "Image Coding Using Wavelet Transform", *IEEE Transactions on Image Processing*, v. 1, n. 2, pp. 205–220, April 1992.

[58] DAUBECHIES, I., “Orthonormal Bases of Compactly Supported Wavelets”, *Communications on Pure and Applied Mathematics*, v. XLI, pp. 909–996, 1988.

[59] DAUBECHIES, I., “Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme”, *SIAM Journal on Mathematical Analysis*, v. 24, n. 2, pp. 499–519, March 1993.

[60] DAUBECHIES, I., “The Wavelet Transform, Time-Frequency Localization and Signal Analysis”, *IEEE Transactions on Information Theory*, v. 36, n. 5, pp. 961–1005, September 1990.

[61] MALLAT, S. G., “Multifrequency Channel Decompositions of Images and Wavelet Models”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 37, n. 12, pp. 2091–2110, December 1989.

[62] VETTERLI, M., HERLEY, C., “Wavelets and Filters Banks: Theory and Design”, *IEEE Transactions on Signal Processing*, v. 40, n. 9, pp. 2207–2232, September 1992.

[63] QUEIROZ, R. L. D., MALVAR, H. S., “On The Asymptotic Performance of Hierarchical Transforms”, *IEEE Transactions on Signal Processing*, v. 40, n. 10, pp. 2620–2622, October 1992.

[64] SILVA, E. A. B. D., SAMPSON, D. G., GHANBARI, M., “A Successive Approximation Vector Quantizer for Wavelet Transform Image Coding”, *IEEE Transactions on Image Processing, Special Issue on Vector Quantization*, v. 5, n. 2, pp. 299–310, February 1996.

[65] VETTERLI, M., KOVACEVIĆ, J., *Wavelets and Subband Coding*. Englewood Cliffs, New Jersey, Prentice Hall PTR, 1995.

[66] FIELD, D. J., “Scale-invariance and Self-similar Wavelet Transforms: An Analysis of Natural Scenes and Mammalian Visual Systems”. In: Farge, M., Hunt, J. C. R., Vassilicos, J. C. (eds.), *Wavelets, Fractals and Fourier Transforms*, Oxford, Clarendon Press, pp. 151–193, 1993.

[67] BEAUDOIN, N., BEAUCHEMIN, S. S., "A new numerical Fourier transform in d-dimensions", *IEEE Transactions on Signal Processing*, v. 51, n. 5, pp. 1422–1430, May 2003.

[68] MICHAEL, G., PORAT, M., "On signal reconstruction from Fourier magnitude", *The 8th IEEE International Conference on Electronics, Circuits and Systems*, v. 3, pp. 1403–1406, September 2001.

[69] DAUBECHIES, I., *Ten Lectures on Wavelets*. Philadelphia, Pensilvania, Society for Industrial and Applied Mathematics, 1991.

[70] GOYAL, V. K., "Quantized Overcomplete Expansions in $\mathcal{R}^N$: Analysis, Synthesis, and Algorithms", July 1995, In: Memorandum UCB/ERL M95/97, Eletronics Research Laboratory, College of Engineering, University of California,Berkley,Berkley.

[71] BURRUS, C. S., GOPINATH, R. A., GUO, H., *Introduction to Wavelets and Wavelets Transforms a Primer*. Upper Saddle River, New Jersey, USA, Prentice Hall, 1998.

[72] DAVIS, G., *Adaptive Nonlinear Approximations*. Computer Science Department, New York University, Ph.D. Thesis, Courant Institute of Mathematical Sciences, 1994.

[73] GOODWIN, M. M., VETTERLI, M., "Matching Pursuits and Atomic Signal Models Based on Recursive Filters Banks", *IEEE Transaction on Signal Processing*, v. 47, n. 7, pp. 1890–1902, July 1999.

[74] AL-SHAYKH, O. K., MILOSLAVSKY, E., NOMURA, T., *et al.*, "Video Compression using Matching Pursuits", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 9, pp. 123–143, February 1997.

[75] DONOHO, D. L., VETTERLI, M., DEVORE, R. A., *et al.*, "Data Compression and harmonic Analysis", *IEEE Transactions on Information Theory*, v. 44, n. 6, pp. 2435–2476, October 1998.

[76] KRIM, H., TUCKER, D., MALLAT, S., *et al.*, "On Denoising and Best Signal Representation", *IEEE Transactions on Information Theory*, v. 45, n. 7, pp. 2225–2238, November 1999.

[77] GOODWIN, M. M., *Adaptive Signal Models: Theory, Algorithm, and Audio Approximations*. New York, NY, USA, Kluwer, Kluwer Internationsl Series in Engineering and Computer Science, 1 ed., 1998.

[78] SUBOTIC, N. S., BURNS, J. W., "Adaptive Decomposition in Electromagnetics", *in Frontier in Electromagnetics, Douglas H. Werner (editor) and Raj. Mitra (editor)*, pp. 437–473, November 1999.

[79] CHEN, S. S., DONOHO, D. L., SAUNDERS, M. A., "Atomic Decomposition by Basis Pursuit", *SIAM Journal on Scientific Computing*, v. 20, n. 1, pp. 33–61, 1999. URL: `http://www-stat.stanford.edu/~donoho/Reports/1995/30401.pdf`.

[80] DAUBECHIES, I., "Time-frequency Localization Operators: A Geometric Phase Space Approch", *IEEE Transactions on Information Theory*, v. 34, pp. 605–612, April 1988.

[81] COIFMAN, R. R., WICKERHAUSER, M. V., "Entropy-Based Algorithms for Best Basis Selection", *IEEE Transactions on Information Theory*, v. 38, n. 2, pp. 713–718, March 1992.

[82] WICKERHAUSER, M. V., *Adapted Wavelet Analysis from Theory to Software*. A. K. Peters, Ltd, 1994.

[83] GOLUB, G., LOAN, C. V., *Matrix Computations*. Baltimore, John Hopkins University Press, 1989.

[84] ZIYAD, N. A., GILMORE, E. T., CHOUIKHA, M. F., "Dictionary Approaches to Image Compression and Reconstruction". In: *Proceeding of the IASTED International Conference on Signal and Image Processing*, pp. 12–16, Las Vegas, Nevada, USA, September 1998.

[85] DANTZIG, G. B., *Linear Programming and Extensions*. Princeton University Press, 1963.

[86] BLOOMFIELD, P., STEIGER, W., *Least Absolute Derivations: Theory, Applications, and Algorithms*. Boston, Birkhauser, 1983.

[87] GILL, P. E., MURRAY, W., WRIGHT, M. H., *Numerical Linear Algebra and Optimization*. Redwood City, CA, Addison Wesley, 1991.

[88] DAVIS, G., MALLAT, S., AVELLANEDA, M., *Adaptive Greedy Approximations*. Springer-Verlag, New York, Constructive Approximations, 1997.

[89] HUSOY, J. H., EILEVSTJONN, J., EFTESTOL, T., *et al.*, "Removal of cardiopulmonary resuscitation artifacts from human ECG using an efficient matching pursuit-like algorithm", *IEEE Transactions on Biomedical Engineering*, v. 49, n. 11, pp. 1287–1298, November 2002.

[90] VERA-CANDEAS, P., RUIZ-REYES, N., ROSA-ZURERA, M., *et al.*, "Transient modeling by matching pursuits with a wavelet dictionary for parametric audio coding", *IEEE Signal Processing Letters*, v. 11, n. 3, pp. 349–353, March 204.

[91] GRIBONVAL, R., BACRY, E., "Harmonic decomposition of audio signals with matching pursuit", *EEE Transactions on Signal Processing*, v. 51, n. 1, pp. 101–111, January 2003.

[92] ETEMOGLU, C. O., CUPERMAN, V., "Matching pursuits sinusoidal speech coding", *IEEE Transactions on Speech and Audio Processing*, v. 11, n. 5, pp. 413–424, September 2003.

[93] VANDERGHEYNST, P., FROSSARD, P., "Efficient Image Representation by Anisotropic Refinement in Matching Pursuits", *IEEE International Conference on Acoustics and Signal Processing*, pp. 1757–1760, 2001.

[94] FROSSARD, P., VANDERGHEYNST, P., VENTURA, R. M. F., *et al.*, "A posteriori quantization of progressive matching pursuit streams", *IEEE Transactions on Signal Processing*, v. 52, n. 2, pp. 525–535, February 2004.

[95] BOURBAKI, M., *Groupes et Algebres de Lie*. Paris, France, Hermann, Chapitres 4, 5 et 6.

[96] COSTELLO, P. J., *Extreme Forms*. CJM, 3, 1951.

[97] COSTELLO, P. J., *Twelve Geometric Essays*. Carbondale, IL, Southern Illinois Press, 1968.

[98] COSTELLO, P. J., *Regular Polytopes*. Dover, NY, Southern Illinois Press, 3rd ed., 1973.

[99] COSTELLO, P. J., MOSER, W. O. J., *Generators and Relations for Discrete Groups*. Springer-Verlag, 4th ed., 1980.

[100] HAZENWINKEL, M., HESSELINK, W., SIERSMA, D., *et al.*, *The Ubiquity of Coxeter-Dynkin Diagrams (an introduction to the A-D-E problem)*. Nieuw Arch. Wisk., 25, 1977.

[101] BARNES, E. S., WALL, G. E., *Some Exteme Forms defined in terms of Abelian Groups*. JAMS, 1, 1959.

[102] LEECH, J., *Notes on Sphere Packings*. CJM, 19, 1967.

[103] WICKER, S. B., *Error Control Systems for Digital Communication and Storage.* New Jersey, USA, Prentice-Hall, 1995.

[104] CONWAY, J. H., SLOANE, N. J. A., "Voronoi Regions of Lattices, Second Moments of Polytopes, and Quantization", *IEEE Transactions on Information Theory*, v. IT-28, pp. 211–226, 1982.

[105] BARLAUD, M., SOLE, P., GAIDON, T., *et al.*, "Pyramidal Lattice Vector Quantization for Multiscale Image Coding", *IEEE Transactions on Image Processing*, v. 3, n. 4, pp. 367–381, July 1994.

[106] BELL, T. C., CLEARY, J. G., WITTEN, I. H., *Text Compression*. Englewood Cliffs, NJ, Prentice Hall, 1990.

[107] GERSHO, A., GRAY, R. M., *Vector Quantization and Signal Compression*. New York, Kluwer Academic Publishers, 1991.