Universidade Federal Fluminense

Escola de Engenharia

Engenharia de Telecomunicações

Athayde Licério Vieira Frauche

Compressão de Sinais de Radares Meteorológicos Usando o Algoritmo MMP (Multidimensional Multiscale Parser)

Niterói-RJ

Março 2008

ATHAYDE LICÉRIO VIEIRA FRAUCHE

COMPRESSÃO DE SINAIS DE RADARES METEOROLÓGICOS USANDO O ALGORITMO MMP (MULTIDIMENSIONAL MULTISCALE PARSER)

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Telecomunicações.

Orientador: Prof. MURILO BRESCIANI DE CARVALHO, D.Sc.

Niterói-RJ

Março 2008

ATHAYDE LICÉRIO VIEIRA FRAUCHE

COMPRESSÃO DE SINAIS DE RADARES METEOROLÓGICOS USANDO O ALGORITMO MMP (MULTIDIMENSIONAL MULTISCALE PARSER)

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Sistemas de Telecomunicações.

Aprovada em MARÇO de 2008.

BANCA EXAMINADORA

MURILO BRESCIANI DE CARVALHO, D.Sc. - Orientador
Universidade Federal Fluminense - UFF

EDUARDO ANTÔNIO BARROS DA SILVA, Ph.D.
Universidade Federal do Rio de Janeiro - UFRJ

WEILER ALVES FINAMORE, Ph.D.
Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio

DÉBORA CHRISTINA MUCHALUAT SAADE, D.Sc.
Universidade Federal Fluminense - UFF

Niterói-RJ
2008

Dedico a presente obra à minha esposa Rosanne, à minha filhinha Maria Clara e aos meus pais Henrique e Mirthes cuja presença e incentivo nos momentos mais difíceis serviram de inspiração.

Agradecimentos

Ao Deus, Todo-Poderoso, por ter me proporcionado saúde e força de vontade ao longo da jornada para suplantar todos os obstáculos que pareciam, por vezes, intransponíveis.

À minha amada esposa Rosanne que esteve, em todos os momentos, ao meu lado sempre com um sorriso estampado no rosto e uma mão pronta a me acariciar e reconfortar. As suas xícaras de café, acompanhadas dos inesquecíveis biscoitos, foram de essencial importância nas muitas noites atravessadas em claro.

À minha sementinha linda, chamada Maria Clara, que recarregou as minhas energias todas as vezes em que apareceu à porta do escritório com o seu rostinho angelical. Minha filha, a sua simples existência me faz crer que tudo é possível.

Este trabalho somente foi possível devido ao apoio inestimável do meu orientador Prof. Dr. Murilo. Muitas foram as vezes em que o cansaço se sobrepôs à vontade, porém nunca me faltou, da sua parte, uma palavra de incentivo e compreensão. Como esquecer as muitas noites em que saímos da UFF porque os passos dos vigias nos lembravam que não havia mais ninguém na Escola de Engenharia.

Agradeço, também, ao Prof. Dr. Alexandre Santos de la Vega pelo tempo e paciência nas aulas de programação. A grande inquietação da notícia de que a minha dissertação não seria possível, sem a elaboração de programas, deu lugar, nos seis meses de convivência constante, à certeza de que programar não era um problema sem solução.

Ao Prof. Dr. Carlos Frederico de Angelis, do Centro de Previsão de Tempo e Estudos Climáticos (CPTEC), pelos valiosos esclarecimentos acerca dos produtos meteorológicos existentes e em desenvolvimento.

Ao Prof. Dr. Carlos Augusto Morales Rodrigues, do Instituto de Astronomia, Geofísica e Ciências Atmosféricas da Universidade de São Paulo (USP), pelas profícuas conversas sobre a obtenção dos dados brutos de radares meteorológicos.

Aos membros da Divisão de Meteorologia, do Departamento de Controle do Espaço Aéreo

(DECEA), em especial ao Major Martim e ao Sargento Nogueira, pelas informações sobre produtos meteorológicos e pelo tempo e paciência despendidos na revisão do texto do capítulo concernente a radares meteorológicos.

Aos companheiros do Subdepartamento Técnico (SDTE) do DECEA, principalmente aos Brigadeiro Castro, Coronel Duarte, Tenente Coronel Corbelli, Tenente Coronel Tuna, Major Waldir, Capitão Jansen e Suboficial Lima, os meus mais sinceros agradecimentos pelo apoio prestado.

Ao Engenheiro Marcus, companheiro e amigo do SDTE, pela revisão do texto e sugestões apresentadas.

Lista de Figuras

Lista de Tabelas

Sumário

Resumo

Neste trabalho, é proposta uma extensão do algoritmo de compressão de sinais, que é conhecido como MMP (Multidimensional Multiscale Parser). Esta nova versão, chamada de MMP 3D, permite que o MMP comprima eficientemente fontes de dados tridimensionais.

Também é apresentada uma nova regra de segmentação, aplicável a todas as versões do MMP, que incrementa significativamente o desempenho taxa-distorção.

Os algoritmos foram implementados em programas de computador, usando programação orientada a objetos com a linguagem C++, e foram usados em sinais tridimensionais de radares meteorológicos da Força Aérea Brasileira.

Os resultados dos desempenhos taxa-distorção obtidos foram muito bons, com o MMP-3D superando códigos atuais que utilizam as técnicas mais modernas, tais como o H.264 e o JPEG 2000. Versões bidimensionais do MMP, usando a nova técnica de segmentação melhorada, também foram implementadas e usadas para comprimir imagens bidimensionais com níveis de cinza. Os resultados mostram que o algoritmo MMP, com a nova segmentação proposta, supera o desempenho do MMP original.

Palavras-chave: MMP, Radar Meteorológico, Sinais Tridimensionais.

Abstract

In this work, it is proposed an extension of the lossy compression algorithm known as Multidimensional Multiscale Parser (MMP). This new version, called 3D-MMP, allows the MMP algorithm to efficiently compress three-dimensional sources of data.

Also, a new segmentation rule, which can be used with all versions of MMP, that enhances significantly the rate-distortion performance of the algorithm is presented.

The algorithms were implemented in computer programs, using the object-oriented C++ programing language, and used to compress three-dimensional data from weather radars of the Brasilian Air Force.

The rate distortion performance results obtained were very good, with 3D-MMP outperforming state-of-the-art encoders, such as H.264 and JPEG 2000. Two dimensional versions of MMP, using the new enhanced segmentation, were also implemented and used to compress two-dimensional gray scale image data. The results show that the MMP algorithm, with the new segmentation proposed outperforms, the original version of MMP.

Keywords: MMP, weather radar, three-dimensional signals.

Capítulo 1

Introdução

A evolução tecnológica experimentada pela sociedade moderna permitiu que houvesse uma transformação dos sinais do formato analógico para o digital. Praticamente todos os sinais de vídeo e áudio são, atualmente, digitalizados a fim de serem transmitidos ou armazenados. Os novos serviços disponibilizados, bem como o aparecimento de novas aplicações de áudio e vídeo, têm demandado o desenvolvimento de novos algoritmos e circuitos de compressão que tendem a ajustar-se à crescente demanda. Particularmente, em relação às imagens e vídeos digitais, ocorreu o aparecimento de uma infinidade de aplicações nas áreas militar, médica, meteorológica, de entretenimento, entre outras.

O presente trabalho utiliza o algoritmo de compressão *Multidimensional Multiscale Parser* (MMP), conforme proposto por (CARVALHO, 2001), em imagens que são utilizadas correntemente pelos estudiosos do assunto de compressão de sinais. Também é estudada a aplicação do algoritmo a imagens de radar meteorológico, utilizados pelo Comando da Aeronáutica (COMAER) na vastidão do território brasileiro, com a finalidade de prover informações para o Controle do Tráfego Aéreo e para a sociedade civil como um todo.

Ressalta-se que na literatura, praticamente, não existem trabalhos de compressão de imagens de radar meteorológico. A pesquisa realizada conduziu ao trabalho de Makkapati (2007). Contudo, não foi possível a comparação direta entre os métodos porque seria necessário implementar o método proposto por ele, uma vez que os resultados apresentados para aquele método somente consideram, para fins do cálculo do erro quadrático, alguns *pixels* selecionados considerados *pixels* de fronteira.

O MMP é baseado no conceito de casamento de padrões multiescalas, conforme proposto por Carvalho (2001, 2002). Ele utiliza versões contraídas e dilatadas de vetores existentes em um dicionário a fim de prover uma melhor representação, em termos de compressão, dos blocos extraídos do sinal de entrada.

Em Filho et all (2004, 2008) é apresentado o código SM-MMP (*Side-Match MMP*). A diferença básica entre o MMP e o SM-MMP é a de que o MMP analisa um bloco isoladamente sem levar em conta os seus vizinhos. Desta forma, mesmo que um bloco de entrada tenha variações suaves, a representação escolhida no dicionário para representá-lo pode apresentar descontinuidades. Já no SM-MMP blocos vizinhos são utilizados para alterar o modelo probabilístico do bloco que está sendo codificado.

Outra variante do MMP é intitulada de MMP-INTRA, conforme apresentado em Rodrigues et all (2005). Neste caso, é feita uma codificação preditiva e o preditor utiliza o conhecimento dos blocos anteriores para, efetivamente, alterar o bloco que está sendo codificado.

Conforme descreve Duarte et all (2005), o MMP-Estereo é uma variação do MMP onde há mudanças e deformações dos blocos codificados anteriormente com os vetores existentes no dicionário, para o par de imagens capturadas sob pontos de vista diferentes, de modo a explorar de forma mais eficiente a redundância existente entre o referido par de imagens.

Há, também, o código chamado de MMP Espaço de Códigos Segmentado, de acordo com Filho et all (2008), que possibilita um melhor desempenho do MMP quando usado para a codificação *Lossless*. Neste caso, há um aumento do número de contextos probabilísticos utilizados pelo codificador aritmético em relação ao MMP original.

De acordo com Filho et all (2005), o MMP foi aplicado a uma base de dados do Massuchussetes Institute of Technology (MIT) para uso em eletrocardiogramas. Neste caso, os vetores de entrada para comparação são unidimensionais.

1.1 Motivação

A principal motivação para o presente trabalho foi a possibilidade do desenvolvimento de um código que fosse passível de ser utilizado para a transmissão e o armazenamento de imagens de radares meteorológicos do Comando da Aeronáutica (COMAER).

Isto se deve ao fato de que os dados provenientes da reflexão das ondas eletromagnéticas, que retornam ao sistema irradiante após serem refletidas por formações meteorológicas, constituírem arquivos binários de grande tamanho de onde são extraídas todas as informações para a elaboração de muitos produtos meteorológicos.

Tendo em vista os escassos recursos das redes WAN (*Wide-Area Network*) para a transmissão dos referidos arquivos até o ponto de interesse, que são os centros de estudos meteorológicos e órgãos de tráfego aéreo, são geradas imagens comprimidas, utilizando o *Portable Networks Graphics* (PNG)(World Wide Web Consortium, 2008), que não representam todos os dados constantes de varreduras sucessivas do radar, que são chamadas de varreduras volumétricas.

Quando, por motivos de pesquisa, principalmente, os dados brutos são transmitidos para as instituições de interesse, foi escolhido pelos meteorologistas um código de compressão sem perdas, que no caso é o GNU ZIP (GZIP)(GNU Operating System, 2008). Cabe ressaltar que apesar da compressão alcançada o volume de dados ainda fica sendo muito grande, tendo em vista as atualizações constantes de varreduras sucessivas.

A despeito da capacidade de avaliar muitas possibilidades de intensidade das formações meteorológicas por meio das imagens comprimidas com o PNG, e que são geradas onde estão instalados os radares, os meteorologistas se ressentem do fato de não receberem, praticamente em tempo real, os dados brutos dos radares. Com isto, o trabalho se justifica na busca de um código que permita uma maior eficiência na transmissão e no armazenamento dos dados brutos em relação ao GZIP.

1.2 Objetivo

No presente trabalho, o objetivo precípuo foi a utilização do MMP estendido para 03 (três) dimensões nas imagens de radares meteorológicos. Assim, foi utilizado o MMP para o desenvolvimento de todos os códigos que levaram ao desenvolvimento do MMP 3D.

Várias fases intermediárias tiveram que ser empreendidas para a consecução do objetivo final:

- A adaptação do MMP original (CARVALHO, 2001) em 02 (duas) dimensões (MMP 2D) para um programa orientado a objetos onde, neste caso, foi utilizado o C++. Isto aconteceu com a implementação de vários códigos iniciais (MMP Sem e Com Multiescalas) até a agregação da otimização RD (MMP 2DRD);

- A extensão do MMP 2DRD para 03 (três) dimensões, que foi intitulado de MMP 3DRD.

- A adaptação dos códigos MMP 2DRD e MMP 3DRD para o MMP Espaço de Códigos Segmentado a fim de melhorar o desempenho para o caso da compressão sem perdas.

Durante as realizações das implementações, os resultados alcançados conduziram a adaptações nos códigos MMP 2DRD e MMP 3DRD no que tange às formas de segmentação dos blocos de entrada. As alterações deram origem aos MMP 2DRDFT e MMP 3DRDFT, tendo FT o significado de *Full Tree*.

Os resultados obtidos apontam para uma possível utilização do MMP FT para o armazenamento de imagens que tenham sido, previamente, transmitidas do local onde o radar esteja instalado a um centro de estudos, como o Centro de Previsão do Tempo e Estudos Climáticos (CPTEC) e a Universidade de São Paulo (USP).

Ressalta-se que nesta fase do trabalho não se buscou a otimização da implementação que conduzisse a tempos de processamento melhores, sendo priorizado o desempenho em termos do binômio taxa-distorção.

1.3 Organização da Dissertação

Para iniciar a discussão, apresentam-se, no Capítulo 2, os principais conceitos com respeito à compressão, com e sem perdas, de sinais ligados a imagens. A necessidade da compressão é abordada e são apresentados alguns tópicos concernentes à teoria da informação.

No Capítulo 3, exploram-se os princípios básicos associados a radares e expõe-se a possível aplicação do código nas imagens dos radares meteorológicos no âmbito do Comando da Aeronáutica. Tal necessidade se justifica, pois desde o início do trabalho procurou-se aplicar os conceitos desenvolvidos às imagens dos referidos radares, tendo em vista que, atualmente, os dados brutos resultantes das varreduras não são, na maioria das vezes, transmitidos por limitações das redes de longo alcance (WAN), devido às taxas de transmissão disponíveis, e ao grande volume de dados gerados.

O Capítulo 4 descreve as versões implementadas para o MMP em duas dimensões, chamadas de MMP 2D. Na exposição, apresentam-se os resultados das versões com e sem o uso de multiescalas e com a otimização $\mathcal{RD}$. Imagens conhecidas, como a *Lena* e *Toy Vehicles* foram utilizadas no desenvolvimento do capítulo além de 03 (três) conjuntos de 04 (quatro) imagens de radares meteorológicos. A fim de proporcionar uma comparação com os resultados alcançados pelo GZIP, os vários códigos MMP 2D foram testados sem distorção para as imagens de radar meteorológico. Os resultados alcançados apontam para o melhor desempenho do MMP, qualquer que seja a versão, em relação ao GZIP. Também foi utilizado o MMP Espaço de Códigos Segmentado que proporcionou resultados ainda melhores em relação ao GZIP.

O Capítulo 5 explora o objetivo principal deste trabalho, que foi a aplicação dos conceitos do MMP a sinais tridimensionais de radar meteorológico, ou seja, a visualização volumétrica de nuvens resultante de várias varreduras. Também foram utilizadas, no capítulo, imagens tradicionais, como uma seqüência intitulada de *Toy Vehicles*. Foram feitas comparações entre o desempenho do MMP 3DRD e do MMP 2DRD, sendo que este último levou a melhores resultados qualquer que fosse o conjunto de imagens utilizadas. Com isto, foi feita uma adaptação do MMP 3DRD, que foi chamada de MMP 3DRDII, que proporcionou uma melhoria de desempenho para as imagens de radar meteorológico em relação ao MMP 3DRD, porém sem chegar aos resultados obtidos pelo MMP 2DRD. Com respeito ao conjunto de imagens de *Toy Vehicles*, o MMP 3DRDII apresentou resultados piores do que aqueles obtidos pelo MMP 3DRD.

Alguns resultados interessantes, e julgados importantes, foram conseguidos com a análise do comportamento dos codificadores MMP 2D e MMP 3D nas provas realizadas nos capítulos 4 e 5, motivando o que foi apresentado no capítulo 6, que foi uma melhoria substancial no desempenho do algoritmo MMP 2D e MMP 3D em relação ao objetivo inicial proposto.

O codificador MMP 2D modificado, chamado de MMP 2DRDFT, foi aplicado a um conjunto diferente de imagens, ilustradas no Apêndice A, a fim de comprovar a sua eficácia. Com respeito ao codificador MMP 3D modificado, batizado de MMP 3DRDFT, foram feitos testes nas imagens de radares meteorológicos, chamadas de *PPI*, nas imagens da seqüência de *Toy Vehicles* e foi feita uma comparação com todos os resultados conseguidos para as versões dos MMP 2D e MMP 3D desenvolvidas ao longo do trabalho. O fechamento das simulações trouxe a comparação dos desempenhos das simulações feitas com o MMP, mediante uso das imagens *PPI* e *Toy Vehicles*, com os conseguidos com os algoritmos JPEG 2000 e H.264.

O Capítulo 7 apresenta um resumo de todas as conclusões advindas das análises empreendidas, citando, também, os principais obstáculos encontrados. Por fim, são apresentadas algumas sugestões que poderão servir de base para trabalhos futuros, sobretudo com respeito à possibilidade do processamento de imagens com a utilização de algoritmos mais eficazes, porém que demandarão mais recursos computacionais para a comprovação de resultados possíveis.

Capítulo 2

Conceitos de Compressão

2.1 A Necessidade da Compressão

Todas as aplicações de áudio, imagens e vídeo, hoje usadas, utilizam um mecanismo de compressão para o armazenamento ou a transmissão, tendo em vista a grande quantidade de informação presente. De fato, não existiria o conceito de multimídia, ou seja, a junção de mídias discretas e contínuas, sem os códigos de compressão.

Pelo exposto, a dificuldade de armazenamento dos dados produzidos, a deficiência de velocidade de apresentação das informações, principalmente vídeo, para a utilização em tempo real e as dificuldades de transmissão de dados, concernentes à largura de faixa necessária, impulsionam, cada vez mais, o mercado de desenvolvimento de algoritmos compressores.

Um exemplo do acima exposto pode ser dado pela demanda de recursos para a transmissão de uma única imagem SVGA 1024x768 (24 *bits* por cada *pixel*). No caso da utilização de uma rede de longo alcance ("Wide Area Network" - WAN) com a taxa de transmissão nominal de 64 kbit/s (HALSALL, 2001), o que ainda é muito comum sobretudo em redes que utilizam satélites, o tempo requerido seria de 295 segundos. Logicamente, para qualquer aplicação de áudio e vídeo concorrente os recursos disponíveis não seriam suficientes, independentemente da aplicação ou não de mecanismos de qualidade de serviço (QoS).

2.2 Teoria da Informação

A teoria da informação tem a idéia de estudar a fonte de um ponto de vista probabilístico e associar a ela o quanto de informação que está sendo gerada. A base de todo o estudo recai sobre a Teoria da Informação, trabalho publicado por Shannon em 1948, com o objetivo de otimizar os sistemas de comunicação.

A teoria da informação estabelece toda a modelagem matemática, buscando respostas a duas questões essenciais:

- Qual é a complexidade irredutível abaixo da qual um sinal não pode ser comprimido?; e
- Qual é a máxima taxa de transmissão para uma comunicação que seja confiável em um canal que apresente ruído?

A primeira resposta pode ser encontrada na Teoria da Codificação da Fonte, que é o objetivo precípuo do presente trabalho, e a segunda, na Teoria da Codificação de Canal.

Normalmente, os estudos apontam para o limite que pode ser conseguido, em termos de compressão, para um determinado tipo de sinal. O desafio reside em como conseguir desenvolver-se o código associado. O problema da compressão é aplicar uma transformação nos dados gerados por uma fonte para, por exemplo, efetuar-se o armazenamento ou a transmissão de dados (BLAHUT, 1988).

A fonte pode ser discreta ou contínua. No caso da fonte discreta X são emitidos símbolos de um alfabeto $\mathbf{S} = (s_0, s_1, \ldots, s_{k-1})$ que podem ser modelados como variáveis aleatórias discretas (s_k).

Assim, as probabilidades associadas aos eventos são dadas por: $P(\mathbf{S} = s_k) = p_k$, para $k = 0, 1, \ldots, k-1$, satisfazendo:

$$\sum_{0}^{\textit{k-1}} p_k = 1 \tag{2.1}$$

Assume-se que a fonte discreta seja sem memória, ou seja, o símbolo emitido a qualquer momento é independente de uma escolha prévia. Um código de compressão de dados é um mapa que associa a cada símbolo, ou grupo de símbolos, emitido pela fonte uma *string* de comprimento variável de letras em um alfabeto. Em um código binário, as letras pertencem a um alfabeto binário, o que é o caso neste trabalho. A quantidade de informação, obtida pela realização de um único evento, é dada por:

$$I(s_k) = log_2(1/p_k) \tag{2.2}$$

Assim, quanto maior for a probabilidade de ocorrência de um determinado evento, menor será a quantidade de informação associada a ele. Durante um intervalo de tempo em que uma fonte produz informações, vários símbolos vão sendo gerados e, no caso de eventos discretos, a média da quantidade de informação $I(s_k)$ de uma fonte com alfabeto (**S**) é dada por:

$$H(S) = E[I(s_k)]$$

Por fim, define-se o conceito de entropia, que determina a quantidade média de informação por símbolo, ou evento, de uma fonte.

$$H(S) = \sum_{k=0}^{k-1} p_k log_2(\frac{1}{p_k}) \tag{2.3}$$

A entropia define o Limite Fundamental da Compressão sem Perdas (BLAHUT, 1988) que será mais explorada na Seção seguinte.

2.3 Compressão sem Perdas

O problema principal da compressão sem perdas, codificação por entropia ou compactação, como também é conhecida, é encontrar um mapa $\mathcal{C}$ que codifica todos os símbolos emitidos de forma que o comprimento médio do código seja mínimo.

O diagrama 2.1 ilustra o processo simplificado da codificação por entropia.

Se x é uma VA, logo c é um vetor aleatório $c = \mathcal{C}\ (x)$. O comprimento da palavra código associada ao símbolo também é uma VA.

Assim, o comprimento médio do código será dado por:

$$\bar{\ell}\,(c) = E[\ell\,(c)] \tag{2.4}$$

O grande problema para os códigos de compressão sem perdas resume-se em encontrar o código de tal forma que o comprimento médio do mesmo seja mínimo (GONZALEZ, 2002). Duas restrições são aplicadas e que devem ser observadas:

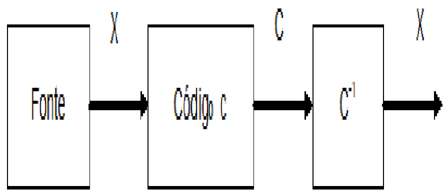


Figura 2.1: Representação da Codificação por Entropia

1. $c = \mathcal{C}$ deve possuir inversa, ou seja, para cada símbolo da fonte há uma palavra código diferente; e

2. A concatenação de blocos de símbolos subseqüentes deve ser unicamente decodificável.

Se for desejável decodificar símbolos que tenham sido emitidos por uma fonte exatamente tais quais eles foram gerados, o código é dito inversível e representa a compressão sem perdas. Caso contrário, ou seja, se vários símbolos puderem ser mapeados por uma única representação do dicionário, a compressão é dita com perdas.

Partindo-se do pressuposto de que se conhece a entropia, o principal desafio para o desenvolvimento de algoritmos é que se consiga aproximar o comprimento médio do código, tanto quanto seja possível, do limite teórico, tendo em vista que o comprimento médio é maior ou, na melhor as situações, igual à entropia. Com isso:

$$L \geq H(S) \tag{2.5}$$

Na utilização da codificação por entropia, um fator limitante é o estabelecimento de códigos prefixo, que são aqueles que apresentam decodificação inequívoca.

As palavras código de uma codificação que respeitam a condição do prefixo são determinadas por um meio prático, que é o estabelecimento das folhas de uma árvore binária, conforme pode ser visto na Figura 2.2.

É possível perceber-se que uma árvore que possua profundidade ℓ_{max} terá $2^{\ell_{max}}$ nodos na referida profundidade. Outra conclusão que se pode tirar é a de que se há a seleção de uma palavra código com comprimento $\boldsymbol{l}$, ocorre a impossibilidade de utilização

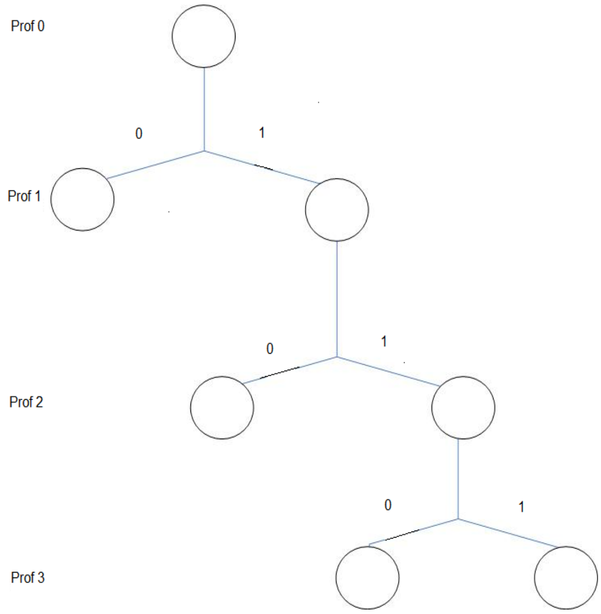


Figura 2.2: Árvore Binária

de $2^{\ell_{max}-\ell}$ palavras de comprimento ℓ_{max}.

Assim:

$$\sum_i 2^{\ell_{max}-\ell_i} \leq 2^{\ell_{max}} \tag{2.6}$$

Por fim, chega-se à desigualdade de Kraft-McMillan que é uma condição necessária, mas não suficiente, para a verificação se um código é ou não prefixo.

$$\sum_i 2^{-\ell i} \leq 1 \tag{2.7}$$

As subseções, a seguir, tratam das principais técnicas de compressão sem perdas.

2.3.1 Codificação Run-Length

A codificação *run-length* (por carreira) é uma codificação por entropia que pode ser utilizada para áudio, imagens e vídeo. O princípio baseia-se na supressão de seqüências de *bytes* e a inserção de um símbolo que caracterize a repetição do que foi suprimido. Logicamente, a compressão depende dos dados de entrada e pode ser que o uso do código não seja eficiente em relação à transmissão da seqüência de entrada original (HALSALL, 2001).

2.3.2 Codificação de Huffman

O código de Huffman é o mais popular entre aqueles desenvolvidos para a remoção de redundância, ou seja, compressão sem perdas. O código analisa a seqüência de caracteres e a freqüência relativa de cada símbolo é determinada. A codificação envolve a criação de uma árvore desbalanceada que apresenta alguns galhos mais curtos do que outros, dependendo da probabilidade dos símbolos (GONZALEZ,2002).

Normalmente, a árvore é construída de forma invertida, sendo o nó superior chamado de raiz e aqueles, em que há a terminação dos galhos, denominados de folhas. Há, também, os nós intermediários, que são aqueles que geram galhos em uma hierarquia inferior (GOMES, 2001). A seguinte seqüência é observada na criação da árvore:

1. Os símbolos da fonte são agrupados em ordem decrescente de probabilidade, sendo que os dois de menor já são rotulados com os *bits* 0 e 1 no início;

2. Os dois símbolos com menor probabilidade são reunidos em um único que representa a soma daqueles que lhe deram origem. As probabilidades são novamente listadas em ordem decrescente em uma segunda coluna que leva em conta o resultado da soma dos dois símbolos anteriores que tinham as menores probabilidades de ocorrência. É importante salientar-se que para cada agrupamento são atribuídos os *bits* 0 e 1;

3. O processo é repetido até que restem somente dois símbolos, aos quais são atribuídos novamente os *bits* 0 e 1; e

4. As palavras códigos binárias são obtidas fazendo o caminho inverso, a partir da última coluna em direção à primeira, anotando-se os *bits* encontrados em cada percurso, até cada símbolo listado na primeira coluna.

A Figura 2.3 ilustra a formação de uma árvore de Huffman.

O código de Huffman realiza operação de soma de *floats* no codificador. No lado do decodificador, as operações descritas não são necessárias e ele tem que fazer, simplesmente, uma consulta a uma tabela previamente conhecida. Normalmente, nem todos os símbolos têm uma representação em código na tabela de Huffman, apenas aqueles com alta probabilidade de ocorrência. Os demais, são codificados diretamente com a adição de um *flag* especial.

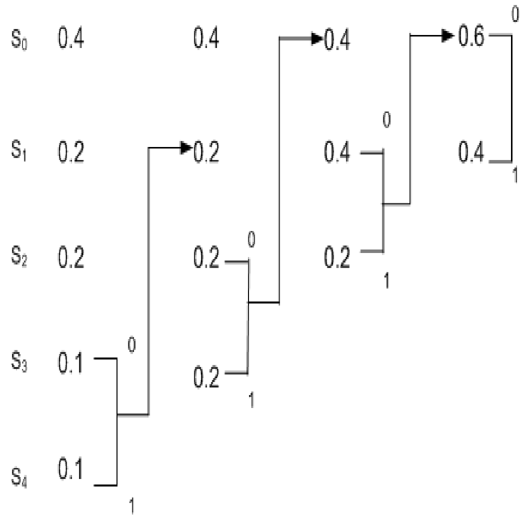


Figura 2.3: Exemplo de Codificação com Huffman

2.3.3 Codificação Lempel-Ziv Welsh

O código Lempel-Ziv Welsh (LZW) é uma variação de outro, chamado LZ78, sendo um dos códigos de compressão sem perdas mais utilizados atualmente (GONZALEZ, 2002). O princípio básico consiste em utilizarem-se palavras código de tamanho fixo para seqüências de símbolos de entrada de tamanho variável sem requerer um conhecimento prévio da estatística da fonte.

Apesar de ser patenteado nos Estados Unidos com o número 4.558.302 (GONZALEZ, 2002), o codificador foi incorporado em uma série de formatos, tais como: *Graphic Interchange Format* (GIF), *Tagged Image File Format* (TIFF), e *Portable Document Format* (PDF), que podem ser vistos com detalhes em Halsall (2001).

O funcionamento do LZW prevê que os dicionários, do codificador e decodificador, sejam construídos, dinamicamente, à medida que a mensagem vai sendo processada. No caso específico de um texto, os dicionários são iniciados com o código ASCII, por exemplo, e vão aumentando paulatinamente. Para imagens monocromáticas com 8 *bits*, a inicialização ocorrerá com os 256 níveis de cinza, ou seja, de 0 a 255 que poderão ser usados para representar cada *pixel*.

2.3.4 Codificador Aritmético

Neste tipo de codificador não existe a correspondência de um para um entre os símbolos da fonte e as palavras código, sendo que para uma seqüência de símbolos atribuise um código aritmético. Para tal, é utilizado um intervalo real de 0 a 1 com a partição inicial correspondente à probabilidade de ocorrência dos símbolos da fonte (GONZALEZ, 2002).

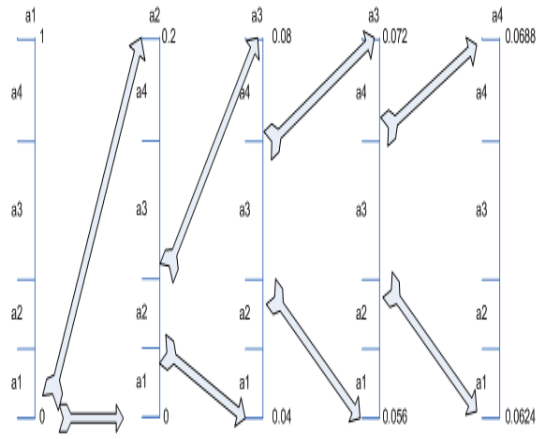


Figura 2.4: Exemplo de Codificação Aritmética

A Figura 2.4 ilustra a aplicação do codificador aritmético com uma seqüência de 5 (cinco) símbolos de uma fonte que contém 4 (quatro) representações possíveis.

A Tabela 2.1 representa os subintervalos admitidos inicialmente para a seqüência a ser codificada. Nota-se que são subintervalos fechados em seus limites inferiores e abertos nos superiores, pois as probabilidades são cumulativas.

Símbolo da Fonte	Probabilidade	Início Intervalo
a1	0.2	[0.0, 0.2)
a2	0.2	[0.2, 0.4)
a3	0.4	[0.4 0.8)
a4	0.2	[0.8 1.0)

Tabela 2.1: Tabela Exemplo do Codificador Aritmético

A cada símbolo transmitido é feito um rearranjo no subintervalo seguinte, adequando os símbolos com respeito às suas probabilidades iniciais. Assim, como exemplo, se o primeiro símbolo transmitido for o a_1, o segundo subintervalo terá os valores compreendidos entre [0.0,0.2) (GONZALEZ, 2002).

À medida que cresce o número de símbolos da mensagem, o intervalo utilizado para a representação fica menor e o número de *bits* necessários para representar o intervalo fica maior. Este código possui melhor desempenho quando comparado ao código de Huffman, pois não há a necessidade de representar cada símbolo por um número inteiro, chegando mais próximo do limite da compressão sem perdas.

Um aspecto que pode ocorrer, caso haja uma seqüência de mensagens, é que se coloque um símbolo que indique o término daquela mensagem, o que pode ser um ponto.

Do lado do decodificador, ocorre a análise do número decimal recebido e todos os símbolos da mensagem são decodificados seguindo os passos vistos para o codificador. Como exemplo, se o número recebido foi o 0.06759, o decodificador é capaz de descobrir que o primeiro símbolo foi o a_1 haja vista ser o único que se enquadra no subintervalo [0.0, 0.2) (GONZALEZ, 2002).

Após a breve análise dos princípios da compressão sem perdas, a próxima seção tratará dos principais conceitos atinentes à compressão com perdas.

2.4 Compressão com Perdas

No caso de se tolerar que haja distorção, ou seja, quando se admite que a representação de um símbolo decodificado não seja uma versão exata do que foi processado, a entropia da saída será menor do que a da entrada, ou seja, $H(y) < H(x)$. A Figura 2.5 ilustra o referido processamento.

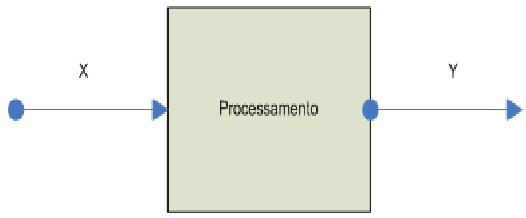


Figura 2.5: Codificação da Fonte

A premissa básica para toda a análise que se segue é a de que se há a transmissão da informação por um canal, e este está isento de qualquer erro, o processo de compressão que introduzirá uma distorção.

Analisando-se um bloco de símbolos transmitidos, pode-se chegar à conclusão de que o mapeamento seja determinístico, ou seja, a um bloco de símbolos na entrada será reproduzido um bloco de mesmo número de símbolos na saída. Contudo, se a análise for particularizada para cada símbolo, pode-se concluir que o mapeamento não será determinístico, partindo-se do pressuposto que não se saiba qual seja exatamente o símbolo de entrada dentre todos os possíveis.

Assim, para a análise em que se conheça a seqüência da mensagem, o comportamento poderá ser considerado determinístico. Considerando-se o processamento como um canal fictício, ocorrerá a minimização da informação mútua entre a entrada e a saída $I(X,Y)$ e, por conseguinte, ter-se-á a menor entropia de saída $H(Y)$, pois a entropia condicional $H(Y/X)$ será 0. A informação mútua representa a informação média recebida após a observação da saída (GOMEZ, 2001).

$$I(X,Y) = H(Y) - H(Y/X) \tag{2.8}$$

Com isso, define-se a função $R(D)$ como a mínima informação mútua $I(X/Y)$ sujeita à restrição de que o valor esperado da distorção entre X e Y seja menor ou igual a uma distorção selecionada $E[d(x;y) \leq D]$. A interpretação prática para a função $R(D)$ pode ser dada como a relação de compromisso entre a taxa de codificação e a distorção, especificando o número mínimo de *bits*, em média, necessários para representar uma saída de uma fonte com distorção média igual a D.

A função $R(D)$ é sempre contínua, decrescente em um intervalo $[0, D_{max}]$ para o caso discreto.

Por exemplo, a função $R(D)$ da fonte gaussiana, sem memória, com as variáveis aleatórias com média $\mu = 0$, variância σ^2 e distorção medida pelo critério da distorção quadrática $d = (x-y)^2$ é dada por:

$$R(D) = \frac{1}{2} log(\frac{\sigma^2}{D}) \tag{2.9}$$

A expressão 2.9 possibilita algumas conclusões a respeito da função gaussiana:

1. a maior distorção média que pode ser conseguida para uma fonte gaussiana é σ^2.

2. se a distorção é zero, a representação se dará com infinitos *bits*, haja vista tratar-se de uma fonte contínua.

A Figura 2.6 apresenta a representação gráfica para a função $R(D)$ gaussiana.

2.4.1 Solução de Compressão com Perdas em Dois Passos

Na solução em dois passos são usadas uma etapa de quantização seguida de uma etapa de codificação por entropia.

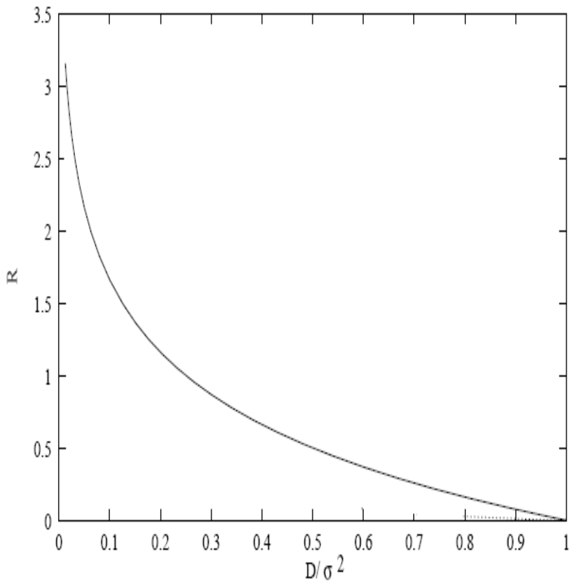


Figura 2.6: Função $R(D)$ da fonte Gaussiana sem memória

Fonte: Carvalho (2001)

A etapa de quantização é a responsável pela introdução das perdas. A idéia é construir uma representação aproximada da fonte de entrada, porém de entropia mais baixa. Desse modo, a etapa seguinte de codificação por entropia irá produzir uma representação mais compacta do que seria possível, caso fosse aplicada diretamente à fonte original.

A quantização pode ser feita por meio de um quantizador escalar (QE) ou vetorial (QV). No tipo escalar, cada amostra de entrada x, que é uma V.A, é comparada com uma tabela que representa os valores de reprodução possíveis de saída $Y \in (\gamma_1, \gamma_2, \cdots, \gamma_N)$. Pelo anteriormente exposto, o desafio é escolher os valores de γ_k que possibilitem o melhor desempenho, de um ponto de vista taxa-distorção.

No caso vetorial, comparam-se blocos de N amostras da entrada com um dicionário de vetores N-dimensionais e novamente o desafio é otimizar o dicionário. Nota-se, portanto, que a quantização escalar é um caso particular da quantização vetorial quando $N = 1$. Além disso, sabe-se que o desempenho tende a melhorar à medida que N aumenta. Por exemplo, um QV de dimensão N pode ter seu dicionário construído como o produto cartesiano de um dicionário de um QV de dimensão $N - 1$ e um QE. Contudo, esta solução particular pode não ser a ótima para o QV N-dimensional, mesmo que os dois quantizadores originais sejam ótimos em suas respectivas dimensões. Assim, o quantizador N-dimensional ótimo pode ter um desempenho superior ao de um quantizador

$(N-1)$-dimensional ótimo, mas não inferior. Apesar da melhoria que pode ser obtida no desempenho, a complexidade computacional aumenta exponencialmente com o aumento da dimensão N. Além disso, o problema de otimização se torna mais complicado à medida que N aumenta devido ao surgimento de múltiplos mínimos locais. Estes dois fatores limitam o uso prático da QV.

A figura 2.7 apresenta a representação gráfica para a compressão com perdas em dois passos.

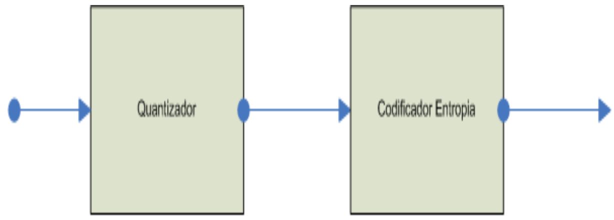


Figura 2.7: Codificação em Dois Passos

2.4.2 Solução de Compressão com Perdas em três Passos

Na solução composta por três passos há a adição de um terceiro bloco, na entrada do sistema, que faz uma transformação da informação da fonte. Um motivo para a popularização desta técnica é que ela permite implementar quantizadores vetoriais de dimensão relativamente elevada a um custo computacional baixo. Por exemplo, o algoritmo de compressão de imagens JPEG utiliza uma etapa de transformação DCT (Discrete Cosine Transform) 8×8 seguida de um banco de 64 quantizadores escalares (GONZALEZ, 2002). A combinação dos dois é equivalente a um QV de dimensão 64, porém a complexidade computacional é a mesma de 64 quantizadores escalares operando em paralelo, que é muito menor do que a de um QV-64. Além disso, a dificuldade de projetar o dicionário do QV-64 torna, na prática, o desempenho da solução sem transformada inferior ao da solução com transformada (GONZALEZ, 2002).

Percorrendo uma matriz de uma imagem monocromática digitalizada, percebe-se que de *pixel* para *pixel* há uma mudança do nível de cinza de um para outro elemento da matriz. A taxa de mudança de amplitude é chamada de **freqüência espacial** e é conseguida pela varredura da matriz nas linhas e colunas.

Como o olho humano é menos sensível às altas freqüências espaciais em relação às mais baixas, o fato é aproveitado na implementação da transformação da forma espacial inicial em outra que envolva freqüências.

Alguns exemplos de aplicação para a transformada podem ser vistos na implementação da codificação do JPEG, que utiliza DCT, conforme visto anteriormente, e no JPEG 2000, que faz a aplicação do conceito da DWT (Discret Wavelet Transform). A DCT é mais simples e rápida de ser implementada; a DWT apresenta melhor desempenho com o prejuízo da necessidade de maiores recursos computacionais (GONZALEZ, 2002).

O segundo bloco da transformação em três passos é o quantizador, que é aplicado, no caso do JPEG, à matriz inteira com a adição de um nível de corte abaixo do qual a representação espacial terá o valor zero, ou, no caso do JPEG 2000, as subbandas com os chamados níveis de decomposição, que estão presentes em número de quatro (GONZALEZ, 2002).

Por fim, aparece o terceiro bloco com a codificação por entropia onde aparecem, com mais freqüência, os codificadores de Huffman, aritmético e carreira.

A Figura 2.8 apresenta a representação gráfica para a solução em 03 (três) passos.

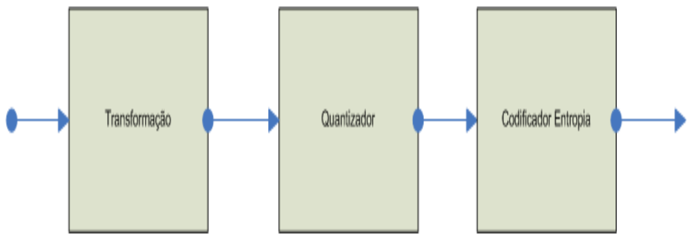


Figura 2.8: Codificação em Três Passos

Os conceitos da teoria da informação e de compressão são importantes na elaboração dos códigos MMP que serão utilizados no decorrer do trabalho. Tendo em vista o objetivo precípuo deste trabalho, que é a utilização do MMP para sinais de radares meteorológicos, serão apresentados, no capítulo seguinte, os princípios de funcionamento dos radares e uma particularização para os usados na meteorologia.

Capítulo 3

Conceitos de Radar Meteorológico

3.1 Introdução

O Radar Meteorológico é uma das melhores ferramentas para o acompanhamento e análise das formações meteorológicas adversas. Estes fenômenos podem afetar a operacionalidade dos aeródromos, colocando em risco, não somente, as vidas de tripulantes e passageiros que constantemente cruzam os céus do território brasileiro, mas, também, aquelas de pessoas, em terra, que podem sofrer as conseqüências de tempestades.

Em adição, os radares meteorológicos são de suma importância para o monitoramento da atmosfera, pois não somente detectam a presença de hidrometeoros como, também, os movimentos das massas de ar. Desta forma, o uso de radares meteorológicos apóia, decisivamente, a áreas como: gerenciamento e planejamento de recursos hídricos, agricultura, controle do tráfego aéreo, previsão do tempo de curto prazo, vigilância meteorológica, etc.

Assim, as informações colhidas dos radares instalados são processadas e utilizadas pelas instituições responsáveis por monitorar e auxiliar na previsão das condições meteorológicas em grande parte do território brasileiro. Tão logo são geradas, as mesmas são disponibilizadas para a utilização de previsões meteorológicas para os aeronavegantes e auxiliam os meteorologistas no rastreio e detecção de tempestades severas, vendavais, precipitações de granizo entre outros fenômenos.

A integração da rede de radares pertencentes ao Departamento de Controle do Espaço Aéreo (DECEA), ao Instituto de Pesquisas Meteorológicas (IPMET), Órgão do governo do Estado de São Paulo, e à Indústria Privada Tectelcom Aeroespacial Ltda

(TECSAT) está sendo realizada em convênio com o Centro de Previsão de Tempo e Estudos Climáticos (CPTEC), Órgão do Instituto Nacional de Pesquisas Espaciais (INPE), desde 2004. Esta é a primeira ação concreta no Brasil de se integrar, de forma operacional, uma rede de nove radares situados nas regiões sul, sudeste e centro-oeste do país, operando, de forma sincronizada, 24 horas por dia (BRASIL, 2001).

A rede produz dados que são enviados aos órgãos operacionais do controle do tráfego aéreo e, também, ao CPTEC. Esses dados são processados e disponibilizados em forma de mosaicos, que representam a síntese e a visualização, em uma tela, dos sinais de mais de um radar, englobando uma vasta área de cobertura de cada um deles.

O INPE, por meio do CPTEC, e o Instituto Nacional de Meteorologia (INMET) são exemplos de instituições governamentais brasileiras que utilizam os produtos disponibilizados pelos radares meteorológicos do COMAER na área da vigilância meteorológica e na previsão do tempo (BRASIL, 2001).

Paralelamente aos trabalhos desenvolvidos pelo CPTEC para o desenvolvimento de produtos meteorológicos, são feitos estudos na USP com respeito ao processamento dos dados de radar para o desenvolvimento de visualizadores dos produtos gerados.

A Figura 3.1 ilustra um radar meteorológico que está instalado no Destacamento de Controle do Espaço Aéreo do Gama (DTCEA-GA).

3.2 Abordagem Histórica sobre Radares

Muitos pensam que os sistemas de radar foram inventados durante a Segunda Grande Guerra Mundial. Na verdade, houve um alavancamento tecnológico muito grande em meio ao conflito, possibilitando o desenvolvimento daquilo que já havia sido inventado na década de 1920. É fato que os períodos em que a humanidade mais percebe os desenvolvimentos tecnológicos são durante aqueles em que há crises que convergem para as guerras entre povos.

A Inglaterra, entre 1935 e 1937, produziu e instalou radares denominados CHAIN HOM. Os radares tinham antenas de cem metros de altura, alcance de detectar uma aeronave a cem milhas da costa e a, aproximadamente, 10.000 pés de altitude. Foram instaladas cerca de 20 (vinte) estações-radar para orientar sistemas antiaéreos e as aeronaves inglesas de defesa aérea em toda a costa leste (Canal da Mancha), os quais muito

Figura 3.1: Radar Meteorológico do DTCEA-GA

Fonte: BRASIL (2007)

contribuíram para a vitória sobre os alemães (BRASIL, 2005).

Os alemães desenvolveram 02 (dois) tipos de radares bastante utilizados durante a guerra. O primeiro chamava-se FREYA, que achava os alvos a até 140 km de distância, varria 360 graus de azimute, mas não media a altitude. Já o segundo, o WÜRZBURG, funcionava como radar secundário e detectava aviões bombardeiros a até 40 km (BRASIL, 2005).

Também ficou evidenciado, durante toda a II Guerra Mundial, o desenvolvimento e o uso dos sistemas de Guerra Eletrônica, conhecimento bastante utilizado até os dias de hoje. A Alemanha nazista de Hitler usou, nos seus bombardeiros e em solo, durante as tentativas de invasão da Grã-Bretanha, equipamentos que faziam com que os aviões fossem guiados até determinados alvos pré-estabelecidos a fim de que suas bombas atingissem os objetivos. A Inglaterra, por sua vez, desenvolveu técnicas que permitiam interferir nos sistemas alemães.

Um fato bastante curioso foi o de que de ambos os lados todo o contingente de engenheiros eletrônicos esteve engajado no desenvolvimento de sistemas de rastreio para achar alvos em terra, do lado alemão, e para a detecção de aeronaves em vôo, no inglês. Assim, a todo novo artifício utilizado por um dos lados, correspondiam contramedidas eletrônicas ao seu emprego.

A diferença fundamental entre os alemães e ingleses, e que deu aos britânicos a vantagem tática na Batalha da Inglaterra, foi a utilização que estes últimos fizeram das informações obtidas do radar, centralizando-as e, logo após, divulgando-as, rapidamente, ao comando de caças e a outros órgãos de defesa envolvidos no conflito.

Outra contribuição decisiva ao radar foi a construção da válvula MAGNETRON, em 1940, pelos professores ingleses J.T. Randall e A.H. Bott, capaz de gerar dez mil watts na freqüência de 3000 MegaHertz (BRASIL, 2005), sendo que ela é utilizada até os dias de hoje em muitos radares e é uma válvula osciladora de alta potência para microondas.

No final de 1942, a companhia alemã GEMA desenhou e construiu o radar tridimensional WASSERMAN que informava altitude, rumo e distância de um avião até cerca de 250 km, sendo que utilizava uma enorme antena de área em torno de 240 m^2, retangular e montada sobre torre rotativa.

Quase ao mesmo tempo em que se desenvolviam os sistemas de radar de solo e defesa aérea pelos ingleses e alemães, outros radares eram aperfeiçoados, como, por exemplo, os destinados à navegação aérea e interceptações aéreas e marítimas, dentre eles, podem ser citados:

1. o H2S britânico (primeiro radar a ter uma antena giratória, instalada na barriga de um avião);

2. o sistema de sonar muito utilizado pela marinha alemã para detectar alvos (submarinos e navios), abaixo do espelho de água por meio do princípio da reflexão das ondas sonoras.

Como a maior parte dos equipamentos eletrônicos atuais que apresentam finalidades duais, ou seja, foram desenvolvidos para a utilização militar e se descobrem formas de uso no meio civil, a utilização dos radares militares para emprego na meteorologia teve seu início determinado a partir do fim da II Grande Guerra com a utilização daqueles que já se encontravam fora de uso, levando, com isso, os meteorologistas a investigar as possíveis aplicações destes equipamentos, como por exemplo, na localização de nuvens.

O acentuado uso e pesquisa dos Radares Meteorológicos Doppler durante as décadas de 50, 60 e, principalmente, na de 70 fez com que o sistema ganhasse em importância com o início de um ciclo de investimentos na busca da obtenção de dados meteorológicos operacionais mais significativos, tais como:

1. detecção e acompanhamento de ciclones e trovoadas;
2. verificação da morfologia das nuvens (correntes ascendentes e descendentes de cumulonimbos);
3. verificação dos campos de vento (fator de risco para a atividade aérea); e
4. intensidade, volume e duração de água precipitável, dentre outros.

3.3 Princípio de Funcionamento dos Radares

As primeiras experiências sobre a detecção de objetos não cooperativos foram baseadas no fato de que a presença de um objeto dentro do campo de uma antena perturbava seu padrão de irradiação. A medida era feita, por exemplo, utilizando uma antena de emissão e uma antena de recepção sobre a qual eram medidas as variações de impedância logo que um obstáculo exterior vinha modificar o acoplamento entre as duas antenas (BRASIL, 2005). Este procedimento que permite determinar apenas a existência é insuficiente, pois ele não permite determinar a localização deste objeto. É necessário, então, complementar esta técnica por uma medida, permitindo a localização, o que pode ser realizado utilizando impulsões segundo o princípio exemplificado na figura 3.2.

Seja um sinal muito breve ou uma impulsão de duração T_0, igual a alguns microsegundos, produzido num transmissor e dirigido para uma antena omnidirecional. A onda assim excitada se desloca na atmosfera com velocidade aproximadamente igual a da luz, ou seja, $c = 3 \times 10^8 m/s$.

Este sinal se propaga em todas as direções e se encontra a um instante T, repartido entre duas esferas de raios $C.T$ e $C.(T + T_0)$. Em outros termos, há a formação de uma onda esférica.

Após o intervalo de tempo T_1, o sinal atinge o alvo; este tempo é proporcional à distância antena-alvo, e se deduz pela equação $T_1 = D/C$.

Uma parte do sinal é refletida pelo alvo. Diz-se que o alvo é iluminado e reirradia uma parte da energia de maneira igualmente omnidirecional. No fim de um intervalo de tempo T_2 igual a T_1, a onda refletida atinge de novo a antena que capta uma parte

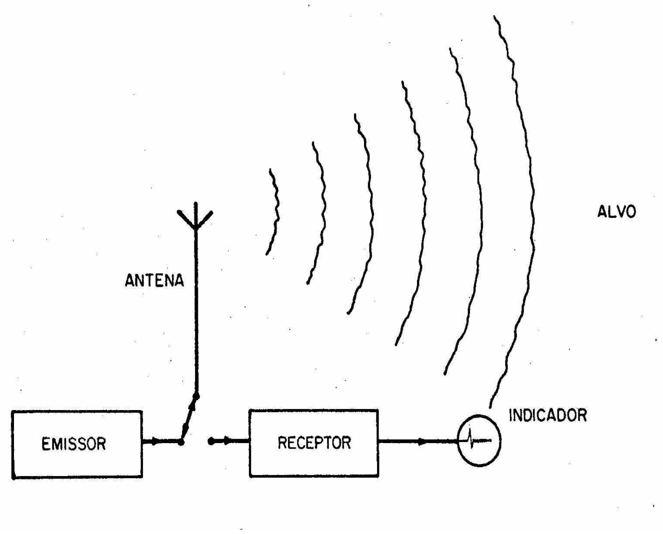


Figura 3.2: Princípio de Funcionamento dos Radares

Fonte: BRASIL 2005

da energia refletida pelo alvo. Esta antena foi religada, neste intervalo, a um receptor muito sensível que amplifica o sinal captado e fornece, então, uma impulsão análoga à transmitida, mas fortemente atenuada, pois somente uma pequena porção da energia transmitida retorna para o radar.

O intervalo de tempo que separa o instante de transmissão e o instante de recepção e permite que se saiba a distância até o alvo é, então, $T = T_1 + T_2$. Como $T_1 = T_2$, isto implica que $T = 2T_1$. Sendo $T_1 = D/C$, chega-se à relação:

$$D = \frac{T.C}{2} \tag{3.1}$$

sendo D a distância do alvo.

A Figura 3.3 ilustra o cálculo da distância para um eco que, no exemplo, é um avião.

O sinal de eco detectado e amplificado sai do receptor do radar e vai para o indicador, onde se fazem as medidas do tempo decorrido e a apresentação usual do sinal de eco.

Pode-se medir o tempo T com um tubo de raios catódicos análogo àqueles utilizados nos osciloscópios clássicos, utilizando-o da seguinte forma: sobre placas de desvio horizontal, aplica-se um sinal em "dente de serra", cujo objetivo é sincronizá-lo com a impulsão de emissão. Sobre as placas de desvio vertical, injeta-se o sinal captado pelo receptor, convenientemente amplificado.

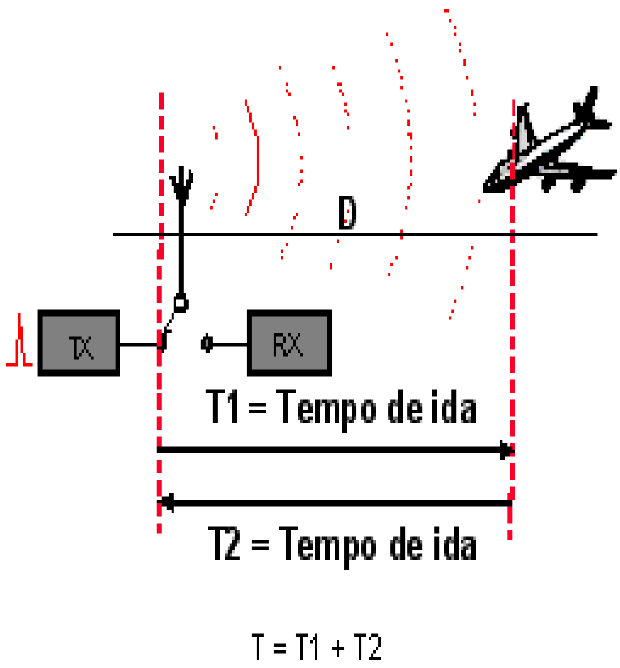


Figura 3.3: Exemplo do Cálculo da Distância

Fonte: BRASIL 2005

Assim a posição horizontal é proporcional ao tempo decorrido, após a emissão, e um desvio vertical, ao sinal da presença de um alvo.

Nas subseções, que se seguem, serão detalhados alguns dos parâmetros e características principais dos radares.

3.3.1 Distância Mínima de Detecção

Por definição, a distância mínima de detecção é a menor distância que um alvo pode estar de uma antena para que possa ser detectado e aparecer representado na tela do indicador. Para tal, é utilizado um elemento, chamado de **duplexador** que faz com que a transmissão do pulso e a recepção do eco possam ocorrer utilizando-se uma mesma antena e guias de ondas.

Quando ocorre a emissão de um pulso, a ação do duplexador também é iniciada nesse instante, de modo que o receptor fica efetivamente desconectado do circuito durante o tempo de transmissão do pulso tp. A largura do pulso transmitido estabelece um mínimo absoluto de distância, e se o alvo estiver a uma distância inferior a tp ele não será detectado (BRASIL, 2005).

Como exemplo, se um radar tem a $tp = 1\ \mu s$ e considerando-se que o tempo para percorrer uma milha náutica é de $12,36\ \mu s$, então a distância mínima para a deteção de um alvo, em milhas náuticas, será:

$$\frac{tp}{12,36\mu s} = 0,08NM \tag{3.2}$$

Na prática, contudo, a largura do pulso não é o único fator determinante da deteção mínima do alvo, tendo em vista que o dispositivo de chaveamento entre a emissão e recepção do pulso não possui uma resposta instantânea.

3.3.2 Relação entre Largura de Pulsos e Intervalos entre Pulsos

Como discutido anteriormente, um sistema de radar pulsado transmite um pulso de energia em RF e escuta os ecos por intervalo de tempo após a transmissão. Repete-se, então, o ciclo, o que poderia ser representado como na Figura 3.4.

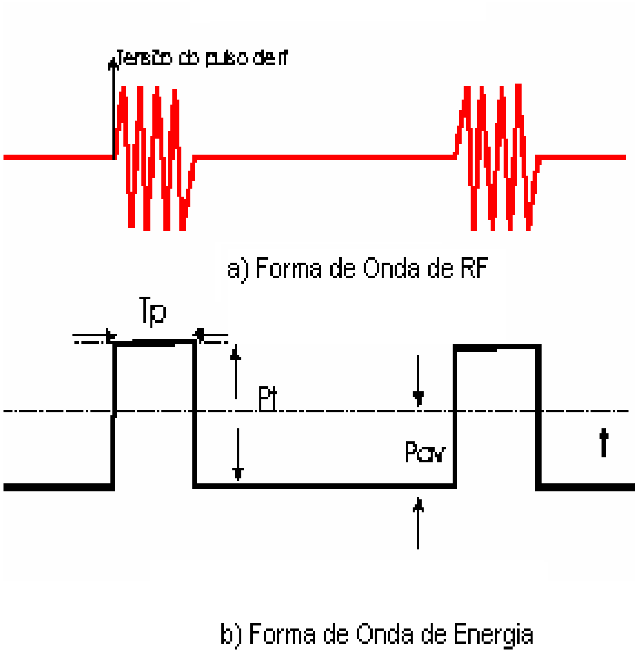


Figura 3.4: Formas de Onda de RF e Energia

Fonte: BRASIL 2005

A parte A da Figura 3.4 representa a forma de onda de RF e, a B, a de energia. Assim, tem-se a potência de pico P_t, a largura do pulso t_p e o período entre pulsos T.

A potência média de um dispositivo transmissor P_{av} é a potência que se poderia obter em uso contínuo. Em operação, a quantidade de energia contida no pulso não pode exceder a energia que se poderia obter se o dispositivo fosse operado continuamente durante todo o período $Pav.T$ (BRASIL, 2005), ou seja:

$$P_t \times t_p = P_{av} \times T \tag{3.3}$$

3.3.3 Separação Radial

Um dos parâmetros mais importantes com respeito ao desempenho dos radares é capacidade que ele tem em poder discriminar dois alvos radialmente.

Sejam dois alvos A e B que estão localizados a distâncias diferentes do radar. Se a distância do alvo A ao radar é D_1, então o tempo de recebimento do eco será $T_1 = 2D_1/c$ e a do alvo B, $T_2 = 2D_2/c$. O posicionamento dos dois alvos aparece na Figura 3.5.

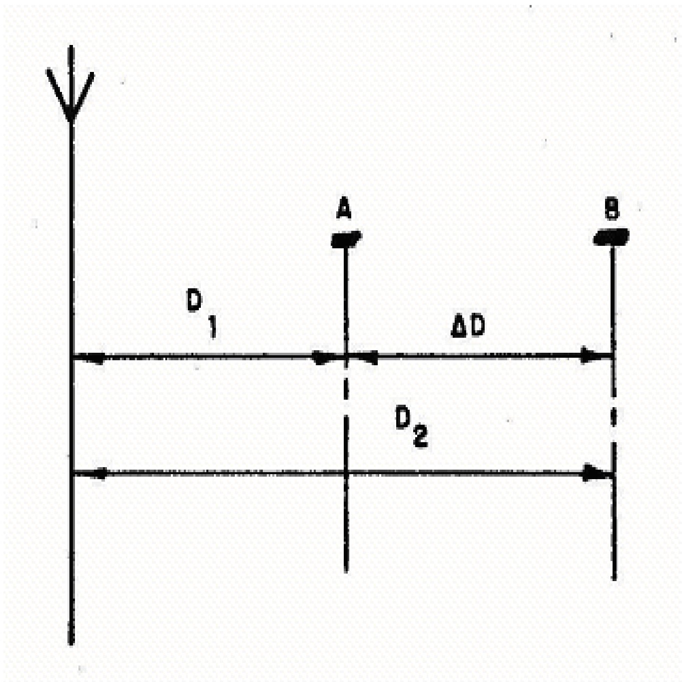


Figura 3.5: Resolução Radial

Fonte: BRASIL 2005

A resolução radial, ou seja, a capacidade que o radar tem de não confundir dois alvos que estejam em uma mesma radial, é dada pela largura de pulso t_p. Assim, se o $\Delta D > t_p$ os alvos A e B conseguirão ser "enxergados" pelo radar como dois alvos distintos.

3.3.4 Separação Azimutal

O poder de separação em azimute refere-se tanto ao sentido vertical quanto ao horizontal e estão, intimamente, ligados aos dados construtivos das antenas de radar.

Forma-se, assim, um ângulo sólido definido pelos pontos de meia potência (3 dB) em elevação (θs) e azimute (θg), conforme discriminado na Figura 3.6 (BRASIL, 2005) .

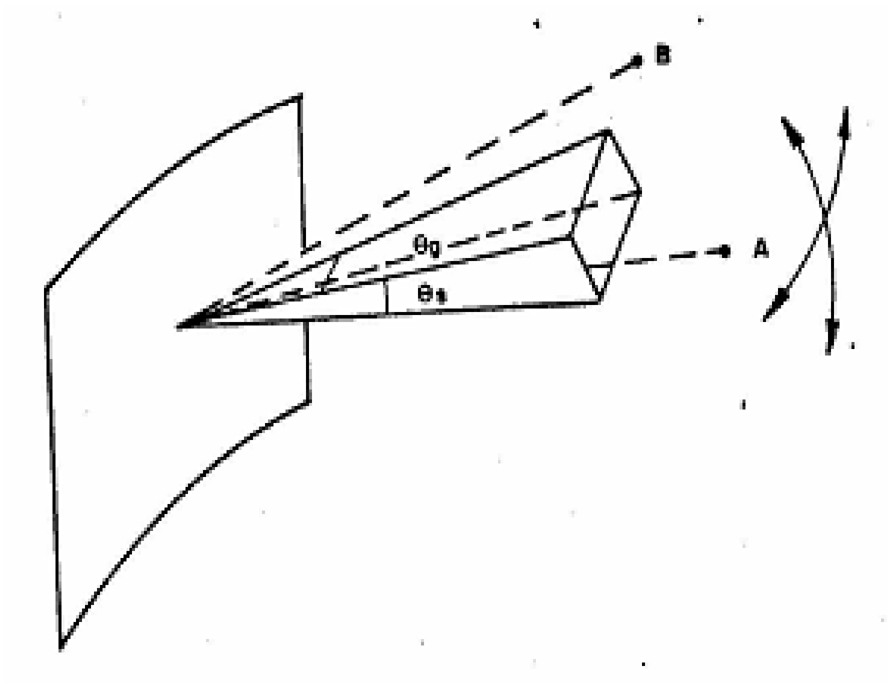


Figura 3.6: Resolução em Azimute

Fonte: BRASIL 2005

Se dois alvos A e B estão separados de um ângulo sólido tal que eles não possam, em nenhum caso, se encontrar simultaneamente dentro do ângulo sólido θg e θs, é admitido que eles estão separados angularmente. Assim, por uma varredura conveniente, a antena pode vir a iluminar o alvo A e, depois, o alvo B.

No caso de um radar panorâmico, onde somente há a rotação em torno do eixo vertical, ocorrerá que se podem ter alvos separados convenientemente, em termos do ângulo sólido, só que a detecção não ocorrerá, em função da própria constituição da antena que não varrerá os dois alvos. A Figura 3.7 exemplifica o descrito.

Com respeito à detecção dos alvos verifica-se o seguinte:

1. os alvos A e B1 estão separados desde que sua diferença angular seja superior a θg;

2. os alvos A e B2 não são jamais separados angularmente, sua separação é inferior a θs;

3. os alvos A e B3 são teoricamente separados, mas a antena não ilumina o alvo B3 que, por este fato, jamais é visto pelo radar.

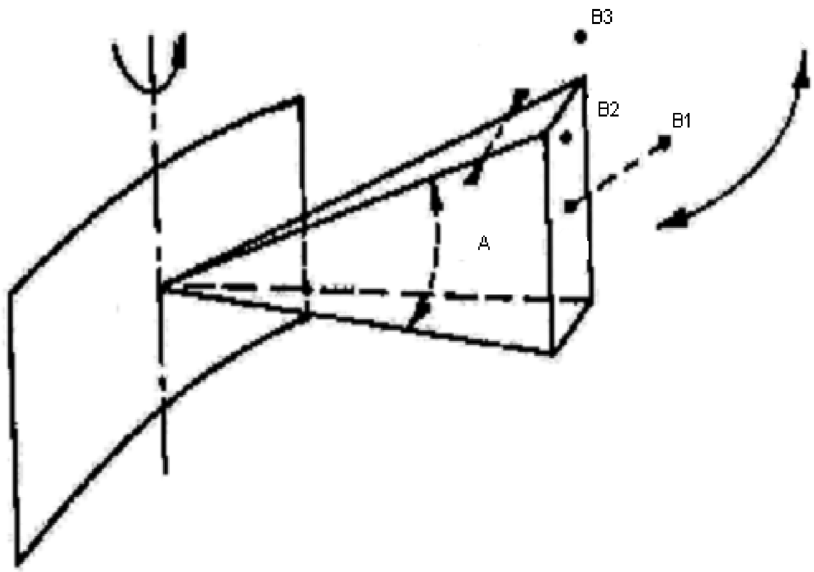


Figura 3.7: Resolução para o Radar Panarâmico

Fonte: BRASIL 2005

3.3.5 Horizonte de Propagação

A antena de radar irradia os pulsos, levando em conta a diretividade da antena. Acontece, porém, que ocorre a refração dos pulsos na atmosfera e um alvo, que não seria visto normalmente, aparece na tela do radar.

O efeito da refração acontece, basicamente, devido à parte inferior do feixe de irradiação propagar-se num meio mais denso que o da parte superior. Dessa forma, a velocidade de propagação é menor para a parte inferior do feixe (BRASIL, 2007).

A extensão do horizonte radar por este processo vai depender da variação da densidade atmosférica com a altitude. No entanto, em condições normais, o encurvamento aumenta o horizonte radar, tornando-o equivalente ao horizonte ótico correspondente à esfera cujo raio fosse 4/3 do raio atual da Terra. O resultado final, em termos práticos, é que o alcance do radar é aumentado de 1/3 em relação ao horizonte óptico. A Figura 3.8 detalha o fenômeno.

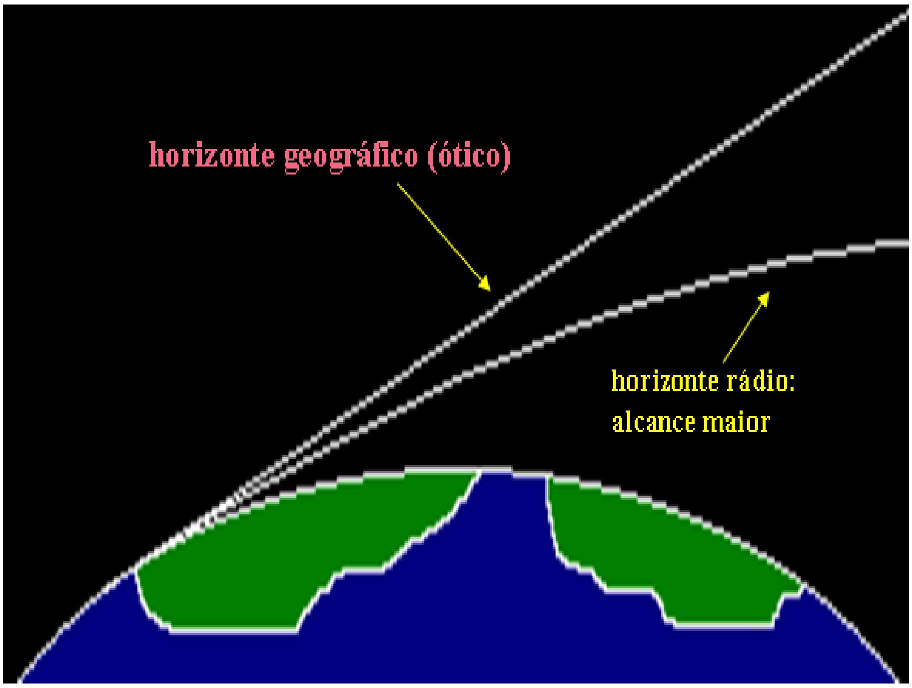


Figura 3.8: Horizonte Radar

Fonte: BRASIL 2007

3.4 O Radar Meteorológico

Como descrito anteriormente, o radar meteorológico é uma das melhores ferramentas para o acompanhamento e a análise das formações meteorológicas adversas que podem afetar a operacionalidade dos aeródromos, colocando, em risco, inúmeras vidas de tripulantes e passageiros que constantemente cruzam os céus do território brasileiro.

O seu funcionamento baseia-se na geração de pulsos de ondas eletromagnéticas com alta concentração de energia para alcançar grandes distâncias que, ao passar por uma nuvem, causam, em cada gota de água, uma ressonância que, por sua vez, irradia ondas eletromagnéticas em todas as direções. Parte dessa energia retorna ao refletor (antena/radar). Da mesma maneira que acontece em todos os outros radares, usados para a determinação da posição de aeronaves, sabendo-se o momento em que o feixe de onda foi emitido pelo refletor e o tempo gasto para retornar, determina-se a distância do alvo, que, neste caso, é a nuvem, ao refletor (BRASIL, 2005). A intensidade do sinal que retornou é diretamente proporcional ao tamanho e distribuição das gotas no volume iluminado pelo feixe do radar.

Dependendo das dimensões do alvo, o eco é classificado em puntiforme ou volumétrico. No puntiforme, os alvos têm dimensões concentradas no espaço, tais como aeronaves, veículos terrestres, aves, etc. Nos ecos volumétricos, os alvos são distribuídos no espaço, isto é, preenchem um volume considerável dentro do feixe do radar, tais como chuvas e os ecos de solo (clutter). A Figura 3.9 dá uma idéia de como se faz o reconhecimento de um alvo volumétrico.

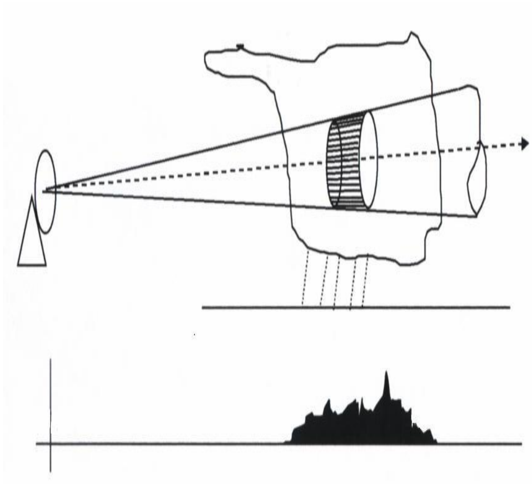


Figura 3.9: Representação de um Eco Volumétrico
Fonte: BRASIL 2007

Diferentemente dos radares convencionais, o radar meteorológico movimenta-se em azimute e em elevação. Desta forma, é possível a detecção de formações em suas reais dimensões, pois os alvos são tridimensionais.

Outra peculiaridade dos radares meteorológicos deve-se ao fato de que ele não mede apenas a refletividade do alvo. O radar usa o efeito Doppler e, além de medir a potência recebida a partir da amplitude do sinal de retorno, mede, também, a diferença entre a fase do sinal de retorno e a fase do sinal transmitido, proporcionando informações de refletividade, velocidade radial e largura espectral dos alvos.

Do mesmo modo que para os radares convencionais, os radares meteorológicos também sofrem os efeitos da propagação das ondas eletromagnéticas na atmosfera, sendo os principais:

1. Espalhamento da energia eletromagnética na superfície terrestre, podendo ocorrer, inclusive, interferências destrutivas por multi-percursos;

2. Refração causada pela atmosfera não-homogênea, o que aumenta a probabilidade de detecção de alvos além do horizonte geográfico;

3. Atenuação pelos gases, partículas e precipitações na atmosfera;

4. Ruídos externos; e

5. Reflexão pela superfície da Terra, por chuva, neve, pássaros e outros, sendo que todas estas reflexões indesejáveis são chamadas de *clutter* e podem ser eliminadas ou reduzidas por meio de processamento.

3.5 Parâmetros Estimados para o Radar Meteorológico

Os ecos úteis retornados, ou seja, que não são *clutter*, são tratados a fim de que seja gerada uma série de produtos para o uso de meteorologistas.

Um dos parâmetros processados é chamado de **Refletividade Equivalente - z**. É um parâmetro básico, pois, em função dele, são gerados outros. As unidades de refletividade são milímetros à sexta potência por metro cúbico ou dBZ (BRASIL, 2007). Está relacionada com a potência dos ecos recebidos, conforme a Equação do Radar Meteorológico, e depende da quantidade e das dimensões das gotas contidas em cada volume de resolução radar.

Outro parâmetro é a **Velocidade Radial - v**, que é derivado da refletividade, estando relacionada com a variação no tempo da fase dos sinais dos ecos recebidos em relação à fase dos sinais dos pulsos transmitidos. É calculada mediante a velocidade Doppler e baseia-se no fato de que quando a energia, a uma dada freqüência, é refletida pelos alvos movendo-se para fora do radar, retorna a uma freqüência menor que a originalmente emitida. Quanto mais rápido os alvos afastam-se, menor será a freqüência retornada.

A **Largura Espectral - w** está relacionada com a variância da distribuição do espectro de velocidades Doppler e é gerada pelas contribuições de cada gota no volume de resolução radar.

Também existe o parâmetro do **Potencial de Precipitação - R** que está relacionado, proporcionalmente, com a refletividade. Quanto maior for a intensidade da formação (Z), maior será o seu potencial de chuva (R).

As principais características dos radares meteorológicos instalados no Sistema de Controle do Espaço Aéreo Brasileiro (SISCEAB) são (BRASIL, 2001):

1. radar pulsado: significa que o radar transmite um pulso curto de alta freqüência.

Seguindo-se a esta transmissão inicial, um certo tempo é destinado a ouvir qualquer sinal refletido. Quando este período de audição termina, repete-se o ciclo de operação e assim por diante;

2. funcionam na banda S: $2,7$ a $2,9$ GHz;

3. comprimento de onda λ de 10 cm;

4. cobertura - até 400 km de raio;

Quanto à finalidade, os sistemas de radar meteorológico são classificados da seguinte forma:

1. tele-detecção radar de alvos meteorológicos;

2. mapeamento da distribuição espaço-temporal e dinâmica dos ecos detectados: nuvens; velocidade da formação e turbulência dentro da formação;

3. distribuição de imagens processadas aos usuários, por meio da rede de comunicações;

4. telecontrole e tele-supervisão do sistema, a partir dos centros de operação remota.

Dentre as muitas aplicações do sistema de radar meteorológico do COMAER, pode-se citar algumas, tais como:

1. detecção avançada dos fenômenos associados às frentes no cone sul do território brasileiro;

2. suporte de vigilância meteorológica, a fim de atender aeronavegantes via fonia (VOLMET) e o controle de tráfego aéreo;

3. exploração de ocorrência de precipitações e ventos em: formações estratiformes, convecções na camada limite da atmosfera e trovoadas.

3.6 Tipos de Varreduras

Como mencionado anteriormente, a antena do radar meteorológico movimenta-se em azimute e em elevação para determinar a estrutura tridimensional dos alvos.

Os sinais de retorno detectados e amplificados serão exibidos na forma de cores para facilitar a sua interpretação. Conforme explicitado anteriormente, as imagens que são geradas onde estão instalados os radares, a partir dos pulsos retornados, estão em formato PNG e são transmitidas para o uso dos meteorologistas.

Salienta-se que as informações do radar estão baseadas no Norte Verdadeiro (NV). Dois ou mais alvos que couberem dentro do volume iluminado pelo pulso *range bin*, serão tomados como um só alvo, resultando em um único eco (BRASIL, 2001). Fixado um certo ângulo da antena com relação à horizontal, chamado de ângulo de elevação, o radar efetua uma varredura da atmosfera girando 360 graus, colhendo um determinado número de amostras em cada radial. O tempo de transmissão/escuta é tão insignificante que a antena do radar fica girando continuamente, não sendo necessário parar o sistema irradiante até a recepção do eco.

Em essência, são 03 (três) os tipos de varreduras em um radar meteorológico, dependendo do interesse na observação dos fenômenos.

3.6.1 Varredura em Azimute

Também chamada de *Plan Position Indicator* (PPI), é resultante de uma única varredura do radar. É utilizada quando a nuvem está muito distante ou quando o radar está sendo utilizado no modo vigilância (BRASIL, 2007). É útil quando o tempo presente na área de cobertura do radar se apresenta estável e se deseja acompanhar a entrada de **frentes** ou o início de formações. A Figura 3.10 apresenta o resultado de uma varredura *PPI*.

Resultante da varredura em azimute aparece a imagem *Plan Position Indicator* que é a projeção ortogonal no plano cartesiano de uma varredura, sendo exemplificada na Figura 3.11.

3.6.2 Varredura em Elevação

É o tipo de varredura utilizada quando há a necessidade da observação de uma determinada formação. A varredura em elevação é tomada com a cobertura em vários ângulos do sistema irradiante e pode ser compreendida pela visualização da Figura 3.12.

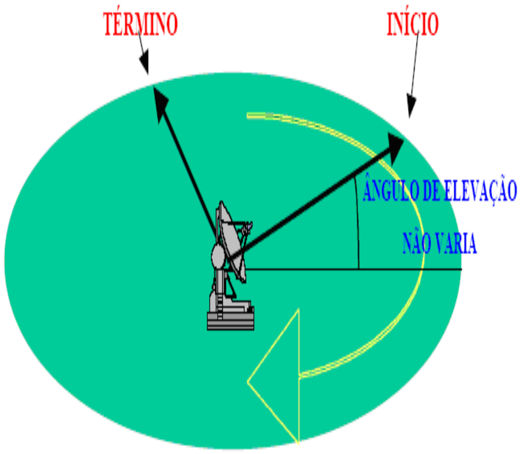


Figura 3.10: Varredura em Azimute

Fonte: BRASIL 2007

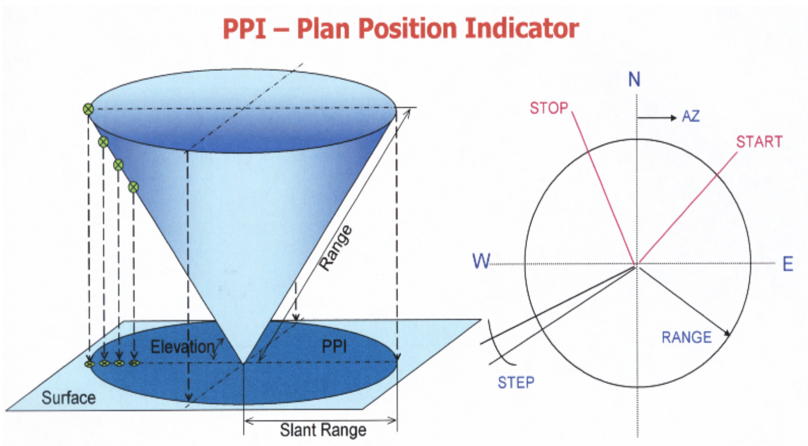


Figura 3.11: Visualização PPI

Fonte: BRASIL 2007

3.6.3 Varredura Volumétrica

É a varredura composta de várias em azimute, ou *PPI*, justapostas e com ângulos variáveis do sistema irradiante. É muito útil quando se deseja observar detalhes evolutivos de formações existentes dentro da região intermediária de cobertura do radar (BRASIL, 2007). A Figura 3.13 mostra o resultado de várias varreduras *PPI*, formando a volumétrica.

Por meio da varredura volumétrica, é possível analisar a atmosfera, tanto em um plano de elevação (vertical) como também, em um plano em distância (horizontal), o que permite extrair o maior número de dados (produtos meteorológicos) da atmosfera.

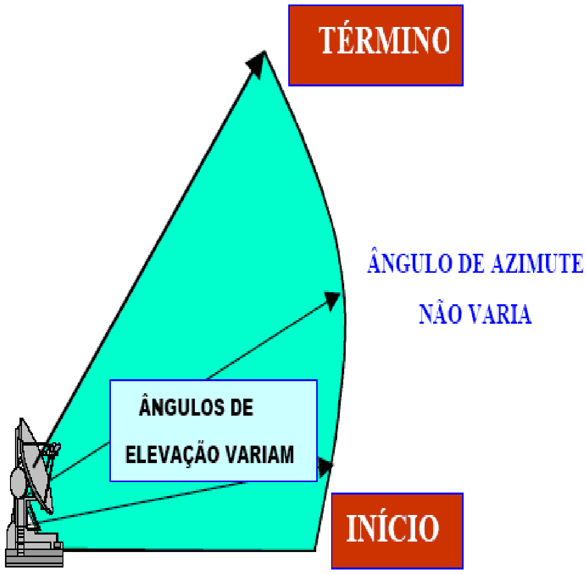


Figura 3.12: Varredura em Elevação

Fonte: BRASIL 2007

Um dos principais produtos, extraídos dos parâmetros do radar presentes nas varreduras volumétricas, é a visualização CAPPI (Constant Altitude PPI). O produto representa a projeção, no plano horizontal, dos dados contidos em uma camada na altura constante H sobre o radar, mostrados em uma janela retangular, com o uso de um código de cores. As imagens CAPPI são muito úteis para o exame e acompanhamento das condições do tempo onde existem rotas aéreas (BRASIL, 2007).

As imagens CAPPI, ilustradas pela Figura 3.14, são usadas, principalmente, nos Modos de Operação Evolução e Análise onde a quantidade de varreduras é aumentada em função da importância da formação a ser analisada.

Pelo exposto, nota-se que, a uma dada altitude, ocorrerá o retorno de pulsos pertencentes a ângulos de elevação subseqüentes. É necessário, pois, fazer o processamento de forma a não haver distorções na visualização do produto CAPPI.

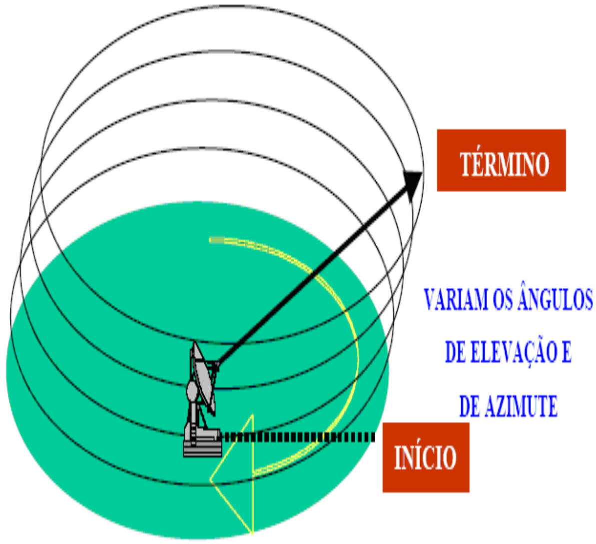


Figura 3.13: Varredura Volumétrica

Fonte: BRASIL 2005

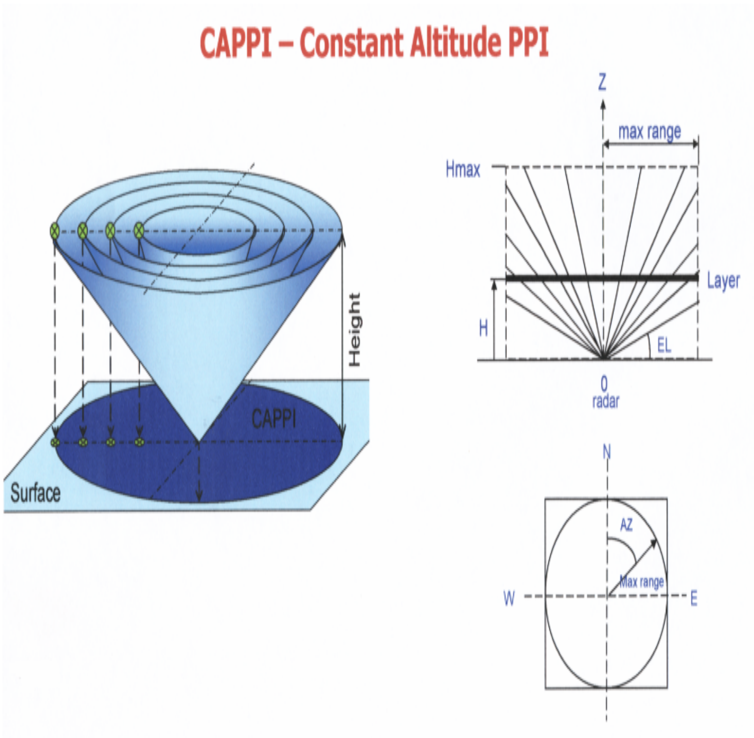


Figura 3.14: Imagem CAPPI

Fonte: BRASIL 2005

Capítulo 4

O Algoritmo MMP 2D

O algoritmo *Multidimesinal Multiscale Parser - MMP* (CARVALHO, 2001) é um método para compressão com perdas baseado no conceito de casamento aproximado de padrões recorrentes multiescalas.

No presente capítulo, são apresentados os conceitos fundamentais para o uso em duas dimensões. Na Seção 4.1, são apresentados todos os conceitos básicos do MMP e os resultados da primeira versão sem a utilização da atualização dos subdicionários com transformações de escala e da otimização $\mathcal{RD}$. Para tal, foi utilizada a imagem *Lena* com vários parâmetros de distorção selecionados.

Está presente, na Seção 4.2, a melhoria proporcionada por meio da utilização do conceito de casamento de padrões multiescalas com a inclusão nos subdicionários de todos os vetores decorrentes das transformações de escala. Neste caso, há a comparação gráfica entre a versão da Seção 4.2 com o desempenho alcançado por aquela da Seção 4.1.

O algoritmo final do Capítulo 4, que utiliza o conceito de otimização $\mathcal{RD}$, foi alcançado por meio do desenvolvimento, passo a passo, do MMP.

Na Seção 4.3, há a comparação do desempenho dos 3 (três) programas com uma seqüência de 4 (quatro) imagens de *Toy Vehicles* e, também, o comportamento do algoritmo frente a 4 (quatro) imagens de Radar Meteorológico *PPI*.

Na Subseção 4.3.2, são feitos alguns comentários a respeito dos tipos de imagens de radar meteorológico, como são obtidas e processadas nas estações instaladas no território nacional. Evidenciou-se, também, a utilização atual do GZIP para o armazenamento e, quando necessário, a transmissão dos sinais por meio da INTERNET ou das redes WAN do COMAER. Por fim, são mencionados alguns resultados do MMP para distorção D

igual a zero para uma comparação ao GZIP.

4.1 Conceitos Básicos

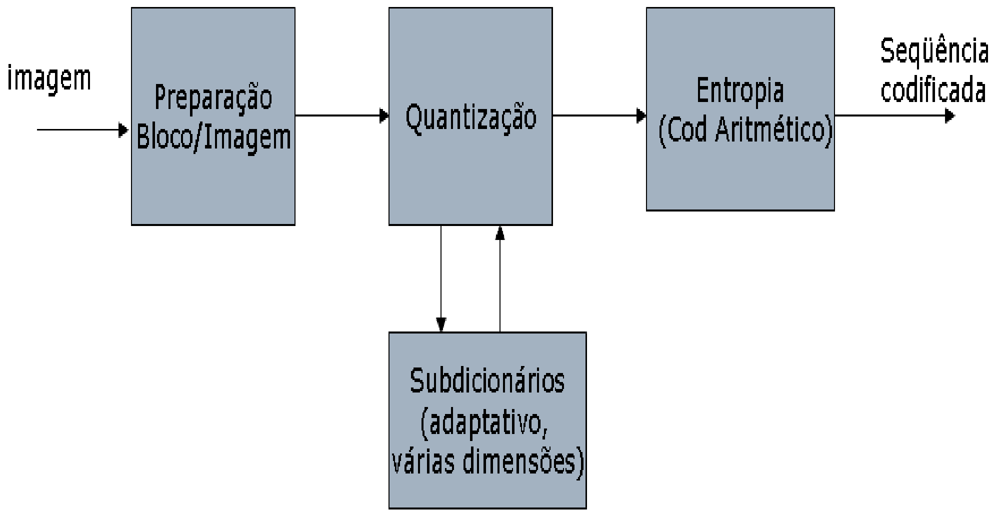


Figura 4.1: Diagrama Bloco em 2 Dimensões

No MMP as imagens são divididas em blocos, que são processados seqüencialmente. Em sua versão mais simples, o algoritmo compara os blocos de entrada com blocos pertencentes a um dicionário, previamente construído, com o intuito de se achar uma representação que seja aceitável em termos de distorção. A Figura 4.1 mostra o diagrama em bloco simplificado do algoritmo para 2 (duas) dimensões. É importante salientar-se que quanto maior for o bloco de entrada, maiores serão os recursos computacionais demandados.

A tônica do MMP é a aproximação do vetor de entrada $\mathbf{X}_i$ para um dos elementos do dicionário. O critério adotado é a diferença média quadrática entre cada *pixel* do bloco de entrada com cada um de um determinado elemento $\mathbf{V}_k$ do dicionário $\mathcal{D} = \{\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_{L-1}\}$.

$$(\mathbf{X}, \mathbf{V}_k) \leq D \tag{4.1}$$

$$(\mathbf{X}, \mathbf{V}_k) = \frac{1}{\ell(\mathbf{X})} \sum_i \sum_j (x_{i,j} - v_{i,j})^2 \tag{4.2}$$

Caso a distorção seja menor ou igual do que uma distorção D_{alvo} pré-selecionada, o bloco é codificado e é colocado um *flag* binário '1' seguido do índice correspondente ao elemento do dicionário. Caso contrário, é agregado um *flag*, também binário '0', e o bloco é partido na direção das linhas ou das colunas. Nos algoritmos MMP em duas dimensões utilizados no presente trabalho, a menos das melhorias implementadas pelo MMPFT descrito no Capítulo 6, adotou-se a partição inicial no sentido das colunas. A Figura 4.2 ilustra o descrito.

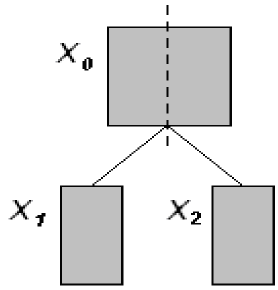


Figura 4.2: Critério de Partição de Blocos

Caso a condição seja satisfeita no nível abaixo, ou seja, na comparação dos blocos de entrada $\mathbf{X}_1$ e $\mathbf{X}_2$ com os elementos do dicionário $\mathcal{D}$, o valor médio quadrático da distorção for menor ou igual do que o *Dalvo*, então serão escolhidos os vetores $\hat{\mathbf{X}}_1$ e $\hat{\mathbf{X}}_2$, de cada comparação dos blocos partidos com o dicionário $\mathcal{D}$, para representar $\mathbf{X}_1$ e $\mathbf{X}_2$. Observa-se que a codificação de $\mathbf{X}_1$ ocorre antes da codificação de $\mathbf{X}_2$, e também que qualquer um deles pode ser subsequentemente subdividido, dependendo dos resultados das medidas de distorção comparadas à distorção alvo. Como os testes são independentes, $\mathbf{X}_2$ pode ser subdividido em dois subblocos $\mathbf{X}_5$ e $\mathbf{X}_6$ independentemente de $\mathbf{X}_1$ ter sido subdividido em dois subblocos $\mathbf{X}_3$ e $\mathbf{X}_4$ ou não.

O dicionário do MMP aumenta de tamanho à medida que a codificação dos blocos e subblocos ocorre. Assim que dois blocos $\mathbf{X}_{2n}$ e $\mathbf{X}_{2n+1}$ são codificados, suas representações $\hat{\mathbf{X}}_{2n+1}$ e $\hat{\mathbf{X}}_{2n+2}$ passam a ser conhecidas pelo codificador e pelo decodificador. Assim, estes subblocos podem ser concatenados de modo a formar uma representação aproximada $\hat{\mathbf{X}}_n$ para o subbloco $\mathbf{X}_n$, que deu origem aos subblocos $\mathbf{X}_{2n+1}$ e $\mathbf{X}_{2n+2}$. O resultado da concatenação $\hat{\mathbf{X}}_n$ pode ser, portanto, incluído no dicionário $\mathcal{D}$, tanto no codificador como no decodificador, sem a necessidade de transmitir-se nenhuma informação adicional.

O dicionário do MMP é composto de vetores de várias dimensões, daí derivando o nome multiescala. Na descrição anterior, um subbloco $\mathbf{X}_n$ deve ser comparado a todos os blocos $\mathbf{V}_k$ pertencentes ao dicionário $\mathcal{D}$, independentemente de seus tamanhos. Para que isto seja possível, deve-se ajustar o tamanho de cada $\mathbf{V}_k$ ao tamanho do bloco de entrada

$\mathbf{X}_n$ a cada tentativa de casamento. Uma forma de aumentar a velocidade do algoritmo, e que foi usada na presente dissertação, é pré-calcular todas as expansões e contrações possíveis para cada elemento $\mathbf{V}_k$, armazenando o resultado em subdicionários diferentes, cada um contendo elementos de mesmo tamanho.

No caso de 2 (duas) dimensões, os blocos são tratados como matrizes bidimensionais $\mathbf{X}^{m,n}$, sendo m o número de linhas e n o de colunas. Com isto, o número de subdicionários é dado por:

$$N = 1 + \log_2(mn) \tag{4.3}$$

Assim, admitindo-se que $m = n = 8$, ocorrerão as possibilidades de partição mostradas na Figura 4.3. Nota-se que os subdicionários possíveis são em número de $7(sete)$.

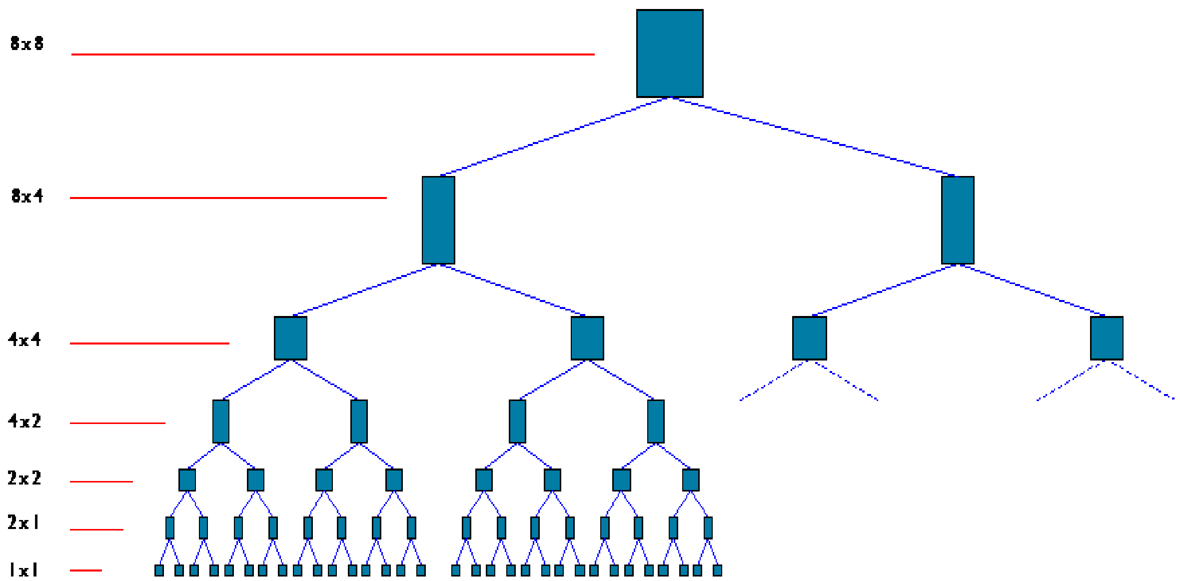


Figura 4.3: Critério de Partição de Blocos no MMP2D

O MMP possui subdicionários que são adaptativos e isto representa uma grande vantagem, pois à medida que os blocos de entrada vão sendo codificados, ocorre o aprendizado da estatística da fonte e os subdicionários vão crescendo em número de elementos.

É importante salientar, e isto estará exemplificado na Seção 4.2, que muitas vezes um subdicionário chega à dimensão de mais de 10.000 elementos, após a codificação de todos os blocos da imagem em teste.

Como comentado, na partição dos blocos constantes do MMP, convencionou-se, nesta dissertação, partir primeiro nas colunas e, caso não se consiga uma distorção $D \leq D_{alvo}$, parte-se nas linhas.

Do bloco inicial $\mathbf{X}_0$, que é chamado de nó raiz, há a derivação dos nós filhos $\mathbf{X}_1$ e $\mathbf{X}_2$. Considerando-se uma matriz de dimensões $m \times n$ e se $m = n$, a segmentação acarretará o aparecimento de duas matrizes de dimensões $m \times n/2$. Em termos de nós de uma árvore binária, admite-se a matriz original como o nó pai $\mathbf{X}_k$ e os nós filhos gerados, $\mathbf{X}_{2k+1}$ e $\mathbf{X}_{2k+2}$.

Enfatiza-se que se o número de linhas m é igual ao de colunas n, a segmentação ocorre no sentido das colunas. Caso contrário, ocorre a segmentação nas linhas. Isto é atingido de forma dinâmica por meio de chamadas recursivas que vão testando, sempre, a condição $D \leq D_{alvo}$.

Um exemplo prático, que pode ser dado na comparação de blocos de entrada com os subdicionários construídos, previamente, é com o segundo bloco 8×8 da imagem *Lena.256*.

165	161	162	160	155	163	160	155
165	161	162	160	155	163	160	155
165	161	162	160	155	163	160	155
165	161	162	160	155	163	160	155
165	161	162	160	155	163	160	155
160	160	155	160	154	154	156	154
160	162	157	162	155	155	157	152
159	161	161	159	154	154	153	151

Tabela 4.1: Codificação de um Bloco de Entrada

Para a codificação do bloco 2 da *Lena*, que está representada pela tabela 4.1, foi escolhido um $D_{alvo} \leq 12$. Com isto, houve a segmentação mostrada na Figura 4.4.

O codificador possui um método que retorna um vetor com o D calculado pela distância média quadrática e o índice K correspondente. Assim, considerando-se um dicionário $\mathcal{D} = \{0, 1, 2, \ldots, 255\}$, o bloco codificado possui a seguinte seqüência de saída: '0','1',161,'0','1',158,'1',155. A justificativa para a inclusão dos *flags* repousa no fato de que o decodificador (UNMMP) necessita interpretar o tamanho do vetor de entrada em

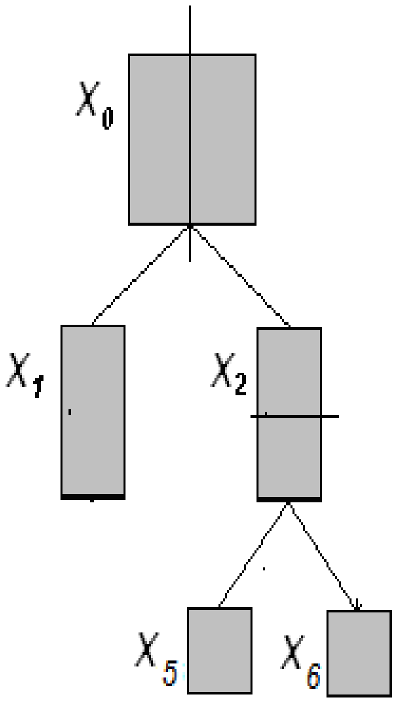


Figura 4.4: Árvore de Segmentação

função das partições feitas no codificador.

A conclusão que se pode tirar é a de que dos 64 *bytes* necessários, a princípio, para a representação do bloco, somente serão necessários 3 *bytes*, que representam os índices, mais 5 *bits* decorrentes dos *flags* '0' e '1'.

A representação da segmentação da árvore da Figura 4.4 aparece na Figura 4.5.

Tudo o que foi exposto, até agora, no presente capítulo possibilitou o desenvolvimento da versão inicial, que não contempla o conceito da mudança de escalas do vetor resultante da concatenação de dois blocos e a inclusão em todos os subdicionários e nem da otimização *RD*. Mesmo assim, o código foi utilizado para servir de padrão de comparação com as demais versões.

Como exemplo da aplicação acima descrita, para a imagem *Lena* consegue-se uma curva $PSNR(dB) \times R(bits/pixel)$, conforme a Figura 4.6. Para tal, basta aplicar a fórmula $PSNR(dB) = 10 \log(255 \times 255)/D$.

Na criação do gráfico, elegeram-se 5 (cinco) pontos distribuídos com espaçamentos iguais entre um $D_{alvo} \leq 20$ e um $D_{alvo} \leq 100$.

Uma outra forma de representação, e que pode ser vista na Figura 4.7, pode ser feita mediante utilização de gráficos $D \times R(bits/pixel)$, sendo D o resultado conseguido por

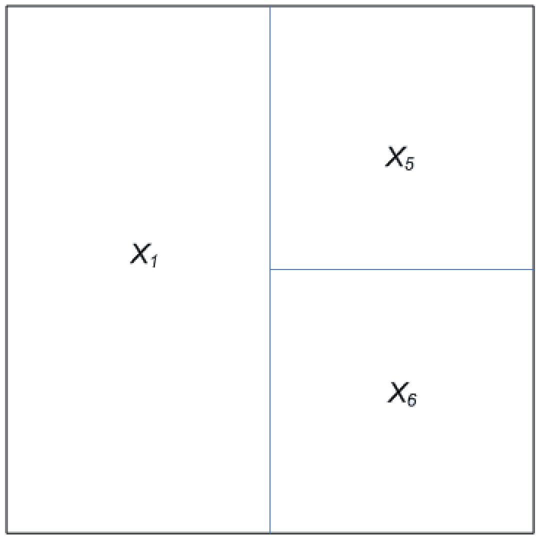


Figura 4.5: Bloco Segmentado

meio da aplicação da Fórmula (4.2), tendo, como limite, o D_{alvo} passado como parâmetro de entrada.

Observa-se que a compressão não é boa e para uma distorção escolhida de $D_{alvo} \leq 20$, consegue-se uma taxa de $2,17 bits/pixel$, ou seja, aproximadamente 1/4.

A imagem associada ao gráfico da Figura 4.6 aparece na Figura 4.8.

Logicamente, à medida que se faz a opção pela escolha de uma distorção D_{alvo} maior, como parâmetro de entrada, menor será a taxa conseguida com o prejuízo da PSNR. O efeito prático é percebido pelo aparecimento mais nítido de blocos ao longo da imagem, como pode ser percebido na Figura 4.9.

A Seção 4.2 apresenta a versão MMP Com Multiescalas para a definição da melhor forma de representação do bloco **X**. Sendo assim, o pseudo-código correspondente à seção atual agregado ao conceito de multiescalas será mostrado.

4.2 MMP Multiescala

A segunda versão, que é apresentada nesta Seção, representa uma evolução agregando o conceito de mudança de escalas do vetor que foi concatenado e a inclusão em todos os subdicionários.

Com isso, discute-se, na presente seção, o desenvolvimento do algoritmo chamado de MMP Com Multiescala onde se incorporam dilatações e contrações de vetores componentes dos subdicionários, o que permite fazer a comparação de um vetor de entrada **X**

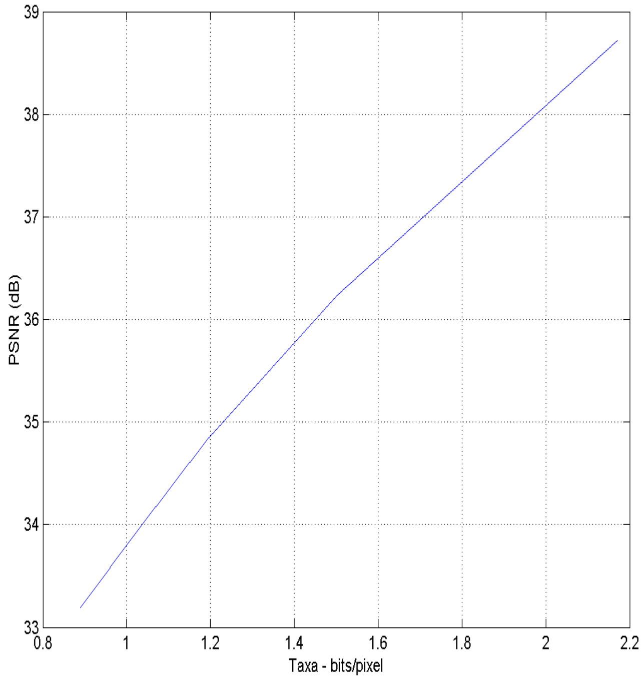


Figura 4.6: $PSNR \times Taxa(bit/s)$ para a Lena MMP sem Multiescala

com elementos de dimensões diferentes.

4.2.1 Mudança de Escala

Tanto a codificação sem perdas, utilizando *Lempel-Ziv*, quanto aquela com perdas, em que se use quantização escalar ou vetorial tradicionais, utilizam o conceito de que os vetores de entrada e os do dicionário de comparação têm as mesmas dimensões.

Com a utilização do casamento de padrões multiescalas, não há a necessidade de que as dimensões supramencionadas sejam iguais.

A idéia de transformação de escala é bastante simples. Supondo, por exemplo, um vetor de entrada unidimensional $\mathbf{X}_n$ com 04 (quatro) elementos e haja a necessidade de que seja representado por um vetor de saída $\mathbf{Y}_{n/2}$, ou seja, com 2 (dois) elementos, basta fazer a contração segundo uma regra pré-estabelecida, conforme notado na Figura 4.10.

No processo de criação das versões contraídas usou-se a seguinte lei: considerando o vetor de entrada exemplificado na Figura 4.10, aplicou-se a relação:

$$y(n) = \frac{x(2n) + x(2n+1)}{2} \tag{4.4}$$

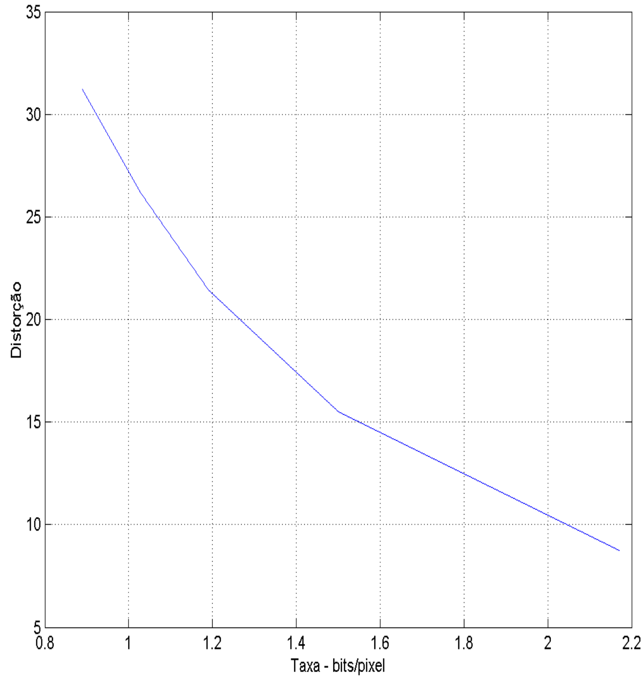


Figura 4.7: $D \times R(bit/s)$ para a Lena MMP sem Multiescala

O processo para a expansão é o mesmo, porém utilizam-se duas relações que são:

$$y(2n) = x(n) \tag{4.5}$$

No caso das amostras que serão criadas para o preenchimento das posições intermediárias entre as amostras múltiplas de 2 usa-se a seguinte expressão:

$$y(2n+1) = \frac{x(n) + x(n+1)}{2} \tag{4.6}$$

Depois da realização do processo de escalamento, o vetor de entrada $\mathbf{X}_i$ é comparado com a versão escalada de um dado elemento $\mathbf{V}_k^s$, pertencente ao dicionário $\mathcal{D}$. Há o processamento, após, para ver se a distância euclidiana $d(\mathbf{X}_i, \mathbf{V}_k^s) \leq \mathbf{D}_{alvo}$.

Considerando-se o uso de casamento de padrões multiescalas, pode-se construir os dicionários de duas formas básicas de implementação. Na primeira somente há uma dimensão e ao passo que os blocos vão sendo codificados acontecem contrações e dilatações dos elementos que são comparados com o bloco de entrada. Este método possui a vantagem de que não é necessário incluir em vários subdicionários versões escaladas de um vetor. Assim, a necessidade de recursos de memória diminui consideravelmente.

Figura 4.8: Imagem Recuperada Sem Multiescala com $D_{alvo} \leq 20$

Fonte: http://sipi.usc.edu/database/index.html

A segunda forma, e que foi a escolhida no presente trabalho, diz respeito à inclusão de versões escaladas em vários subdicionários, previamente, construídos. Neste caso, a complexidade computacional diminui sobremaneira, pois somente é necessário fazer a comparação de um bloco com uma versão escalada que já foi incluída em todos os subdicionários após a concatenação. Apresenta, contudo, a desvantagem de necessitar de mais recursos de memória.

4.2.2 Funcionamento do MMP

A fim de ser ilustrado o funcionamento do MMP, será feita a apresentação do algoritmo para uma dimensão e um vetor de entrada $\mathbf{X} = (X_0 \ldots X_{N-1})$. Utiliza-se, também, um dicionário $\mathcal{D} = \bigcup_{s=0}^{S-1} \mathcal{D}^s$, sendo S o numero de subdicionários, cada um possuindo L_s elementos. Toma-se por base, também, o fornecimento de um D_{alvo} para o processamento de cada bloco.

O procedimento padrão é mostrado, a seguir:

Passo 1: Codifica ($\mathbf{X}_i$, D_{alvo})

Passo 2: Encontra o menor D no subdicionário $\mathcal{D}^s$ cujos elementos têm a mesma

Figura 4.9: Imagem Recuperada Sem Multiescala com $D_{alvo} < 100$

Fonte: http://sipi.usc.edu/database/index.html

dimensão de $\mathbf{X}_i$, tal que $(\mathbf{X}_i, \mathbf{V}_k) = \frac{1}{\ell} \sum_i \sum_j (x_{i,j} - v_{i,j})^2$

Passo 3: Verifica se $d(\mathbf{X}_i, \mathbf{V}_k) \leq D_{alvo}$.

Passo 4: Se sim, coloca o *flag* '1' seguido do índice correspondente e retorna $\mathbf{V}_k$. Caso contrário, vá para o **Passo 5**.

Passo 5: Coloca o *flag* '0' e divide o vetor $\mathbf{X}_i$ em $\mathbf{X}_{2i+1}$ e $\mathbf{X}_{2i+2}$.

Passo 6: Chama recursivamente a rotina Codifica para os dois vetores $\mathbf{X}_{2i+1}$ e $\mathbf{X}_{2i+2}$.

Passo 7: Inclui ao subdicionário $\mathcal{D}^s$ na escala corrente o resultado da concatenação de $\hat{\mathbf{X}}_{2i}$ e $\hat{\mathbf{X}}_{2i+1}$, denotado por $\hat{\mathbf{X}}_i$.

Passo 8: Contrai e expande $\hat{\mathbf{X}}_i$ e inclui o resultado para todos os demais subdicionários $\mathcal{D}^s$. Caso um dado subdicionário já contenha $\hat{\mathbf{X}}_i$, o vetor não será incluído nele.

Passo 9: Retorna .

A diferença fundamental entre os procedimentos apresentados para o MMP Multiescalas para a versão Sem Multiescalas é que neste último caso não existe o **Passo 8** das contrações e expansões e, simplesmente, há ou não a inclusão no subdicionário do nível

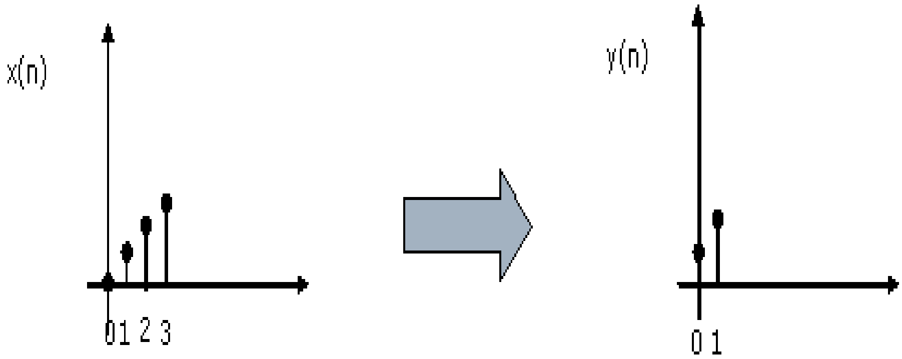


Figura 4.10: Exemplo de Transformação de Escala

acima ao da concatenação.

Como no caso do MMP Sem Multiescalas, foram feitos testes com a *Lena*, com 4 (quatro) imagens separadas dos *Toy Vehicles* e com 3 (três) conjuntos de 4 (quatro) imagens separadas de PPI de radar meteorológico.

A Figura 4.11 mostra os resultados comparativos com a *Lena* para os dois códigos. Fica patente a evolução com a utilização do padrão multiescalas que pode ser explicado pelo fato de que as inclusões, decorrentes de cada concatenação são expandidas, contraídas e incluídas em todos os subdicionários. Assim, há o aumento, considerável, da possibilidade do casamento do vetor de entrada $\mathbf{X}_i$ com um dos elementos $\mathbf{V}_{k,l}$ do subdicionário, escalados ou não, do subdicionário $\mathcal{D}^s$, sem a necessidade de partir o bloco.

Logicamente, que há um preço em termos de recursos computacionais a ser pago. Tendo em vista que na versão do MMP 2D Com Multiescalas há muito mais elementos nos subdicionários, resultantes dos processos de contração, expansão e inclusão nos subdicionários, muito mais comparações são efetuadas, necessitando de mais recursos de processamento em relação à versão do MMP 2D Sem Multiescalas.

Um exemplo dos tamanhos dos subdicionários pode ser dado pela codificação da *Lena* que apresenta 2813 elementos no subdicionário $\mathcal{D}^0$ para o MMP 2D Sem Multiescalas, e 16976 elementos para o mesmo subdicionário na versão MMP 2D Multiescalas.

4.3 MMP RD

A última versão do dicionário contempla os melhores resultados, quaisquer que sejam as imagens utilizadas, e é chamada de MMP RD, sendo RD o binômio $Taxa\times$

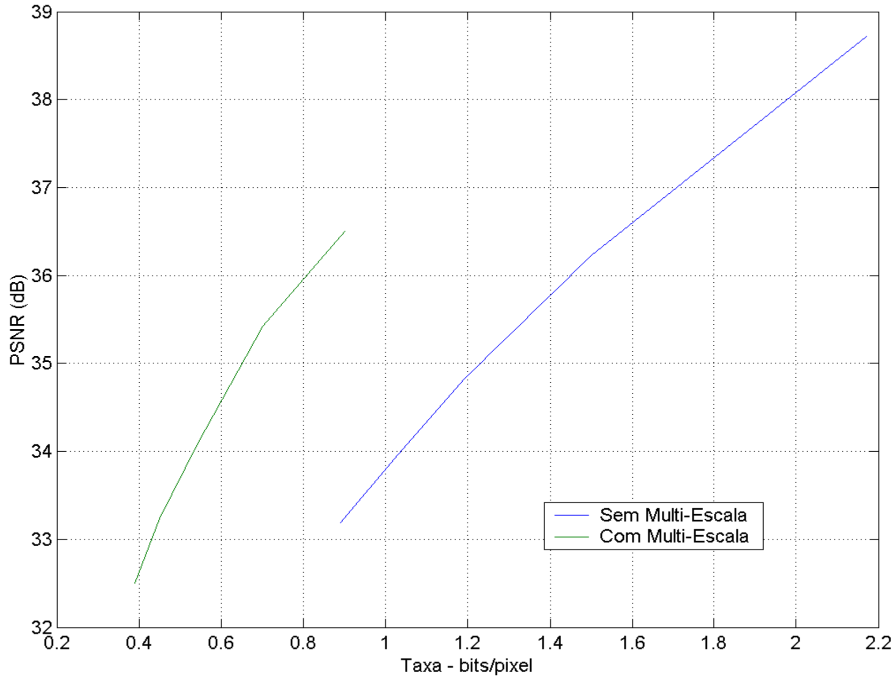


Figura 4.11: Comparação entre Códigos Com e Sem Multiescalas para a *Lena*

Distorção.

Até o presente momento, o critério para a divisão de um bloco de entrada $\mathbf{X}_i$ em outros dois, de um nível inferior $\mathbf{X}_{2i+1} e \mathbf{X}_{2i+2}$, foi que $D > D_{alvo}$.

A representação da segmentação dos blocos da Figura 4.4, em termos de árvore binária, é dada pela Figura 4.12.

Assim, a árvore mostrada na Figura 4.12 representa a segmentação do vetor de entrada $\mathbf{X}_0$ nos segmentos $(\mathbf{X}_1\mathbf{X}_5\mathbf{X}_6)$.

Imagine-se a situação em que o D_{alvo} estipulado seja de 30 e que na determinação da menor distância euclidiana, entre o vetor e o mais próximo elemento $\mathbf{V}_k$ do subdicionário $\mathcal{D}^0$, haja o retorno do valor 30.5. Chega-se à conclusão de que haveria a subdivisão em $\mathbf{X}_1$ e $\mathbf{X}_2$ simplesmente pelo fato de que o D retornado é um pouco maior do que o D_{alvo}. Gasta-se, com isso, mais *bits* para representar os *flags* e os índices resultantes.

Foi proposto um método de subdivisão (CARVALHO, 2001) em que não se leva em conta, somente, o D_{alvo} como parâmetro para a decisão de segmentação, mas, também, a incorporação da taxa R no critério.

Isto foi possível porque o codificador aritmético utilizado possui um método que retorna a taxa de cada símbolo em função da freqüência relativa do mesmo devido ao

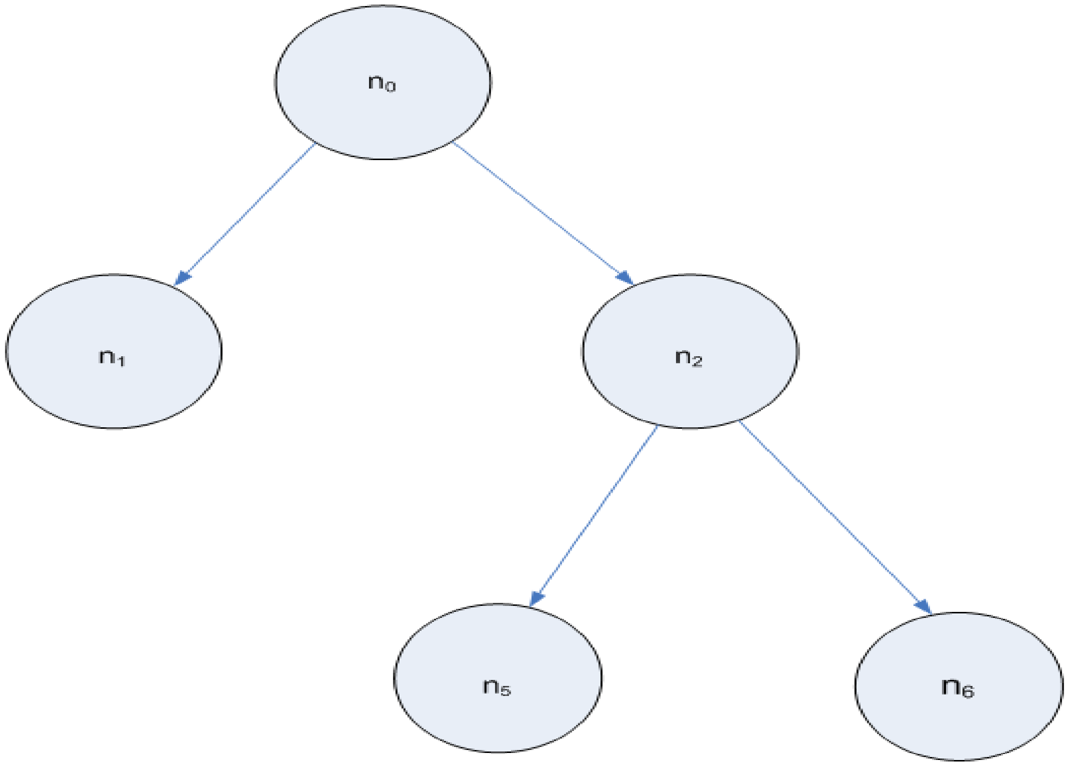


Figura 4.12: Árvore de Segmentação

número de ocorrências que vão acontecendo durante a codificação.

Pelo exposto, cada árvore possui uma distorção total e uma taxa total. No exemplo dado na Figura 4.12 a representação, usando somente os nós folhas fica:

$$\tau_f = (n_1, n_5, n_6) \tag{4.7}$$

Ao se fazer a codificação considerando as folhas e os critérios de taxa e distorção chega-se à relação de que a melhor árvore para representar o vetor de entrada $\mathbf{X}_i$ é dada por:

$$J\tau_f = (Dn_1 + Dn_5 + Dn_6) + \lambda[(Rn_1 + Rn_5 + Rn_6) + R_{flags}] \tag{4.8}$$

O problema se resume em encontrar a melhor árvore de segmentação que minimize a R_τ dado que $D_\tau \leq D_{alvo}$ ou o seu dual que é encontrar o mínimo D_τ dado que $R_\tau \leq R_{alvo}$.

O critério de otimização utilizado no desenvolvimento do MMP RD do presente trabalho foi:

$$J = D(\tau) + \lambda R(\tau) \tag{4.9}$$

O λ é o multiplicador de *Lagrange* e é o parâmetro passado em lugar do D das versões anteriores. O resultado é que há a escolha da melhor árvore entre todas as possíveis.

O critério de escolha da melhor árvore leva em conta que o D aumenta à medida que haja o aumento de λ e a taxa R diminui para a mesma situação. Isto é explicado pelo fato de que se o λ passado for muito grande, já há o *match* no primeiro subdicionário $\mathcal{D}$.

A D e a R relacionadas com a melhor árvore são calculadas a partir dos nós folhas, ou seja, debaixo para cima, até a chegada no nó raiz.

A idéia seguida é que se para a transmissão de dois nós filhos somados o custo for maior do que a transmissão do nó pai que os gerou, há a poda da árvore e decide-se pela transmissão do nó pai.

A fim de ilustrar o funcionamento do MMP RD, será feita a apresentação do algoritmo para um vetor de entrada bidimensional $\mathbf{X}^{m,n}$. Utiliza-se, também, um dicionário $\mathcal{D} = \bigcup_{s=0}^{S-1} \mathcal{D}^s$, sendo S o número de subdicionários, contendo L_s elementos cada um. Cada elemento $\mathbf{V}_k$ de cada subdicionário possui as dimensões que tornam compatíveis as comparações com o vetor $\mathbf{X}^{m,n}$ de entrada. Toma-se por base, também, o fornecimento de um λ_{alvo} para o processamento de cada bloco.

O procedimento padrão é mostrado, a seguir:

Passo 1: Codifique $(\mathbf{X}_i,\ \lambda_{alvo})$.

Passo 2: Inicialize $\mathcal{T}$ como a árvore completa de profundidade $log_2(mn) + 1$.

Passo 3: Faça $p = log_2(mn)$.

Passo 4: Para cada nó $n_l \in \mathcal{T}$ e profundidadede p, faça $Jmin = Jl$, sendo $l \in [2^{p-1} - 1, 2^p - 2]$ e calcule o seu custo associado $J_l = J(n_l) + \lambda R(n_l)$.

Passo 5: Caso $J_l < J_{2l+1} + J_{2l+2} + Rf_0$, sendo Rf_0 o custo do flag '0' para partir o nó pai n_l nos seus dois filhos n_{2l+1} e n_{2l+2}, que foram calculados na interação anterior com $p + 1$, então podam-se os nós filhos e o custo adotado é o do nó pai J_l. Caso contrário, o custo adotado é o dos nós filhos $J_{2l+1} + J_{2l+2} + Rf_0$.

Passo 6: Faça $p = p - 1$.

Passo 7: Repita os passos 4 a 6 até $p = 1$ e $\mathcal{T} = \mathcal{T}_0$.

4.3.1 Resultados Experimentais

Foram realizados uma série de experimentos a fim de verificar o desempenho do MMP 2DRD, sendo o 2D utilizado para indicar que o uso é para 02 (duas) dimensões, frente aos demais códigos MMP Sem e Com Multiescalas 2D. Utilizaram-se imagens 3D compostas de 04 (quatro) varreduras do radar meteorológico, cada uma correspondendo a uma elevação diferente. Cada varredura é composta de 360 raios com uma separação angular de 1 grau, sendo cada raio composto de 400 pontos. Para os testes realizados, foram compostas imagens bidimensionais retangulares, que são denominadas de *PPI*, onde cada linha corresponde a um raio.

Foram utilizadas, também, seqüências de vídeo digitalizadas. Os testes foram realizados nas imagens separadas e nas concatenadas de *PPI* (imagens de radar) e *Toy Vehicles* (imagens de vídeo). Também foram feitos os testes com a *Lena*.

Como exemplo, tomando uma única imagem *PPI*, com dimensões de 360×400 *pixels*, e aplicando os 3 (três) códigos acontece o resultado mostrado na 4.13 onde nota-se que o desempenho do MMP 2DRD foi muito superior aos demais. Enfatiza-se que em todas as demais imagens o resultado foi semelhante.

Assim, todos os resultados que aparecerão a partir deste ponto refletirão somente o uso do MMP 2DRD.

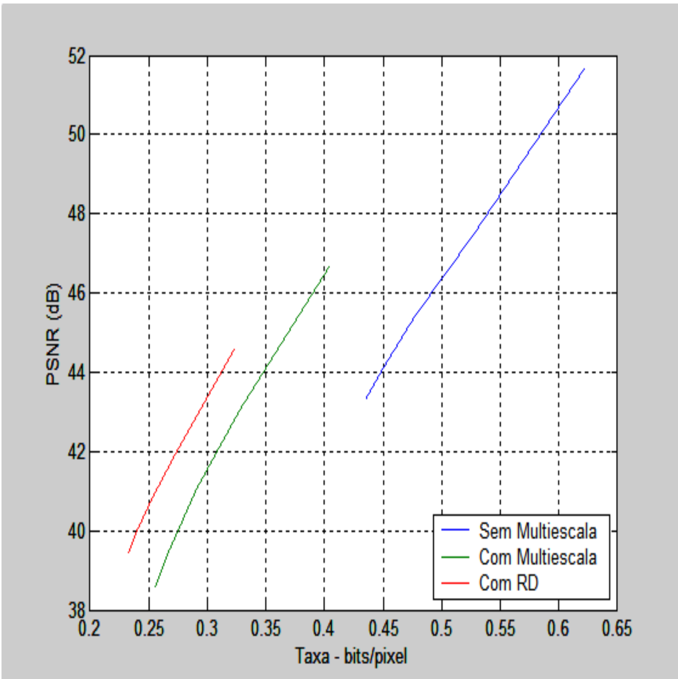


Figura 4.13: Resultados com a Primeira Elevação da Seqüência *PPI1*

As imagens de radar utilizadas (*PPI*) compreendem 3 (três) seqüências de 4 (quatro) imagens concatenadas, resultantes de 04 (quatro) elevações subseqüentes de uma varredura volumétrica, e uma das imagens separadas está exemplificada na Figura 4.14. Ressalta-se que aparecerão, como exemplo, no Apêndice A, as imagens concatenadas constantes de uma seqüência.

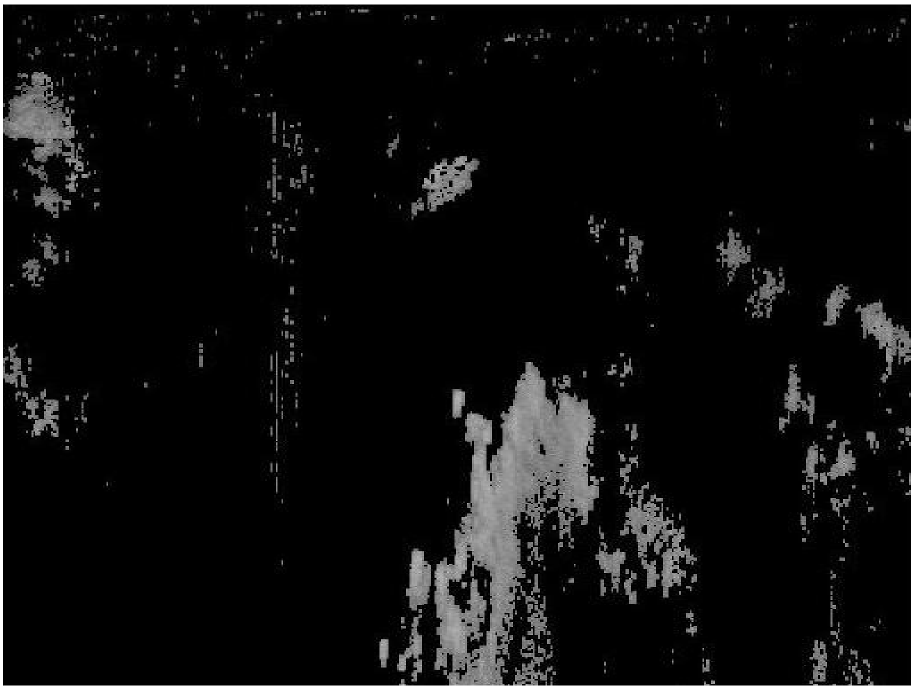


Figura 4.14: Exemplo de Imagem *PPI*

Fonte: Banco de Imagens DECEA

Com respeito às imagens *PPI* concatenadas, cujas dimensões são de 1440×400 *pixels*, foram feitos testes em blocos de matrizes 4×4, 8×8 e 16×16, por serem múltiplos das dimensões originais de cada *PPI*, a fim de verificar o comportamento do MMP 2DRD frente a blocos de dimensões diferentes para as imagens concatenadas e separadas. Nas imagens separadas foram considerados blocos de entrada de dimensões 8×8 e, para a realização de comparações com as imagens concatenadas, foram feitas médias aritméticas entre os parâmetros λ iguais de cada imagem processada.

A Figura 4.15 mostra os resultados alcançados. Nota-se que não há, praticamente, diferença entre as curvas 8×8 e 16×16. Verifica-se, também, que fazer a compressão das imagens separadamente conduz a um resultado prático pior do que executá-la com as imagens concatenadas. Uma explicação para o fato é que para as imagens concatenadas o dicionário que é usado para uma imagem subseqüente já cresceu em função das codificações das imagens anteriores, ou seja, proporciona mais opções de casamentos em função da estatística da fonte.

Com respeito às imagens dos *Toy Vehicles*, foram utilizados blocos de dimensões 8×8 e 16×16 por serem múltiplos das dimensões da imagem original de 512×512 (linhas por colunas) para uma imagem, e 2048×512, para as 04 (quatro) imagens concatenadas.

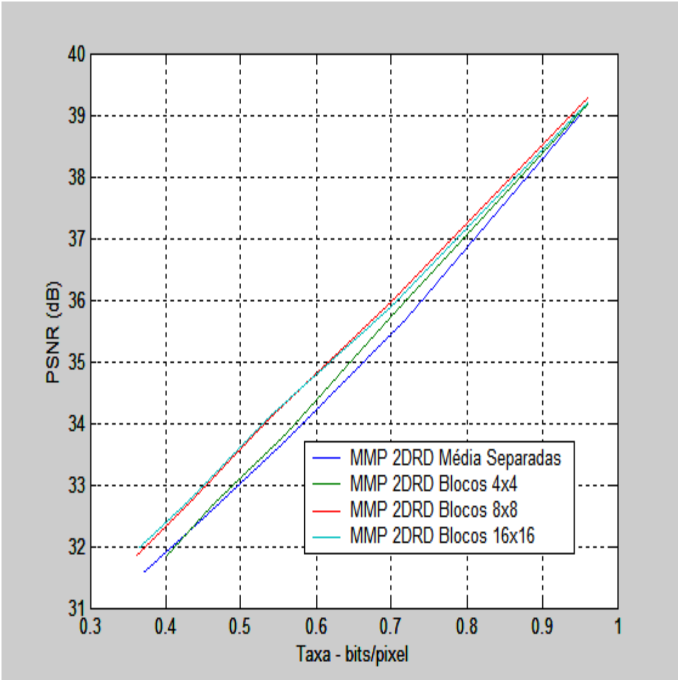


Figura 4.15: Resultados para a Seqüência PPI2

A Figura 4.16 apresenta os desempenhos do MMP 2DRD para tamanhos de bloco de entrada variados.

A diferença básica em relação ao teste com a *PPI* da Figura 4.15 está no fato de que conseguiu-se um desempenho superior com o uso dos blocos 16×16 em relação aos 8×8. A explicação encontrada foi a de que não há muita variação entre uma imagem e outra, possibilitando com que haja *match* para elementos que já se encontram nos subdicionários nos níveis mais altos sem a necessidade de muitas segmentações.

4.3.2 Particularidades na Aplicação das Imagens de Radar Meteorológico

Ao longo do trabalho foram utilizadas várias imagens separadas e concatenadas de radar meteorológico nos testes com os algoritmos MMP. Para tal, foram feitas visitas ao CPTEC, ao Instituto de Astronomia, Geofísica e Ciências Atmosféricas da Universidade de São Paulo (USP) e à Divisão de Meteorologia do DECEA a fim de se conhecer as aplicações das referidas imagens nos produtos meteorológicos.

Conforme explicado no Capítulo 3, há uma série de radares instalados no território brasileiro e, a cada 15 (quinze) minutos, aproximadamente, os dados referentes a uma

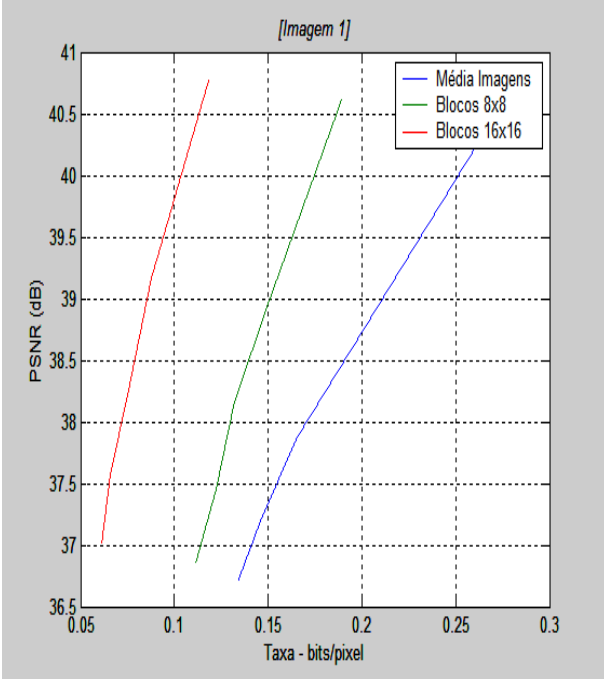


Figura 4.16: Resultados com as 4 Imagens *Toy Vehicles*

varredura volumétrica são enviados para locais de realização de pesquisas, como no caso do CPTEC, e para os Órgãos de Controle do Tráfego Aéreo.

Cada *PPI* é composto dos ecos provenientes do volume iluminado *BIN* e pelo número de radiais que é de 360, uma para cada grau de incremento durante a rotação da antena. Normalmente, o número de *BIN* varia de 400 a 512 e é configurado pelo meteorologista em metros, dependendo do interesse na formação. Assim, como exemplo, se o radar possui 400 *BIN* e o profissional seleciona a resolução de km em km haverá um retorno de eco a cada km e o alcance do radar será de 400 km, limitado por suas características técnicas.

Com isto, para uma imagem *PPI*, normalmente se tem 360×400 linhas por colunas. Tendo em vista que cada varredura volumétrica é constituída de vários *PPI*, sendo que o número de elevações depende da intensidade da formação meteorológica, e é um parâmetro escolhido pelos meteorologistas, há a formação de uma série de imagens, como a própria refletividade de um *PPI*, a imagem *RHI* (resultante do incremento do ângulo da antena em relação ao horizonte), a que representa a velocidade radial, dentre outras.

Sendo o volume de dados por varredura muito grande, é feita, para a transmissão, a transformação do volumétrico em imagens que são as projeções ortogonais em duas dimensões dos dados de interesse.

Assim, para efeitos práticos os dados volumétricos não são transmitidos e, quando são necessários, como no caso do CPTEC, é feita uma compressão sem perdas, com a utilização de *GZIP*. O mesmo ocorre no destino para o armazenamento da varredura volumétrica, ou seja, o arquivo é, novamente, compactado antes de ser carregado em um disco rígido ou outro meio.

Como o objetivo da dissertação é o estudo do MMP e, tendo em vista as próprias limitações do ser humano na percepção de uma distorção conveniente que seja passada como parâmetro, ele tem as suas aplicações voltadas à compressão com perdas.

Com isto, procurou-se dar a mesma atenção às imagens *PPI* das outras imagens utilizadas, ou seja, foram aplicados vários pontos equivalentes aos custos lagrangeanos a fim de serem levantados os pontos para plotar os gráficos.

Contudo, a fim de comparar o algoritmo MMP ao GZIP e ver se havia algum ganho, foram feitos testes com a distorção D igual a zero, no caso dos algoritmos MMP Com e Sem Multiescala, bem como o custo lagrangeano λ também igual a zero, para o MMP 2DRD.

O resultado encontrado foi bastante interessante e curioso, tendo em vista que as 3 (três) versões do MMP, Sem Multiescalas, Com Multiescalas e Com RD superaram o GZIP. Como exemplo, para uma imagem *PPI* de dimensões 360×400 o desempenho do MMP Sem Multiescalas foi melhor, aproximadamente, 10.5 %, enquanto os códigos MMP Com Multiescalas e MMP 2DRD superaram o GZIP em 7.3 %.

O tempo de processamento dos algoritmos os torna, ainda, de difícil aplicação prática para a transmissão dos dados. Ocorre, contudo, que para o armazenamento de dados nos destinos, após a transmissão, podem ser de utilidade. Conforme já mencionado, nesta fase de desenvolvimento do método não foram feitos esforços no sentido de otimizar-se o tempo de execução do código. Toda a ênfase do estudo foi dada à maximização do desempenho de compressão em termos dos curvas $R(D)$.

Outro ponto de destaque foi que dentre os 03 (três) algoritmos MMP o de melhor desempenho, para uma distorção D igual a zero, foi o MMP Sem Multiescalas, justamente a versão inicial e que apresentou o pior desempenho comparado aos demais códigos quando se admite qualquer distorção. A explicação para o fato é a de que para distorções D perto da entropia da fonte, ou seja, próximas a zero, a simples concatenação e inclusão nos subdicionários proporciona um desempenho melhor, pois o codificador aritmético usa, para a concatenação dos blocos, um número maior de *bits* para as versões MMP Com Multiescalas e MMP Com RD.

Uma variação dos códigos MMP foi desenvolvida com o intuito de melhorar o desempenho do MMP 2DRD para compressão sem perdas. O código foi batizado de MMP 2DRD Espaço de Códigos Segmentado. Nesta versão, são feitos subdicionários dentro dos subdicionários padrão que são inicializados em qualquer versão do MMP. No MMP 2DRD Espaço de Códigos Segmentado , o codificador aritmético usa um modelo estatístico separado para codificar a origem de um bloco que foi incluído em um subdicionário, resultante de expansão ou contração, atribuindo-o um índice, mais o modelo estatístico original do subdicionário. Com a mesma imagem testada para os demais códigos, a versão MMP 2DRD Espaço de Códigos Segmentado apresentou um resultado 12.5 % melhor em relação ao GZIP.

A Tabela 4.2 apresenta as vantagens percentuais de cada código MMP em relação ao GZIP para uma única imagem *PPI* de uma seqüência.

MMP Sem Multi	MMP Com Multi	MMP 2DRD	MMP Esp. Cod. Seg.
10.5	7.3	7.3	12.5

Tabela 4.2: Vantagens Percentuais dos Códigos MMP em Relação ao GZIP para uma *PPI*

Em todos os locais visitados, os profissionais demonstraram preocupação no possível uso de algoritmos de compressão com perdas para as imagens radar. As aplicações de compressores com perdas para a visualização já acontecem, correntemente, nos radares do SISCEAB e, cientificamente, são defendidas, conforme Makkapati (2007).

A impressão, passada pelos pesquisadores meteorologistas, é a de que não é possível o uso de sinais comprimidos com perdas nos ecos recebidos antes de eles gerarem outros produtos meteorológicos. Contudo, tendo em vista que na utilização dos radares meteorológicos há uma série de ruídos e interferências no sinal recebido, conforme descrito na Seção 3.4, é bem possível que a compressão com perdas possa ser utilizada sem qualquer prejuízo para as análises posteriores dos dados que levam à formulação dos diversos produtos meteorológicos. O valor admissível da distorção permanece em aberto, a ser determinado em investigações futuras.

Capítulo 5

O Algoritmo MMP 3DRD

Neste capítulo, são apresentados os conceitos e experimentos atinentes ao MMP 3DRD. Com isto, atinge-se o objetivo precípuo da dissertação, que era o de adaptar o código em duas dimensões para o processamento simultâneo de imagens de radar meteorológico como um volume.

A proposta baseou-se no fato de que, sendo o algoritmo multidimensional, seria possível fazer testes para seqüências de imagens, tratadas como um bloco, a fim de prover subsídios à comparação do desempenho com o algoritmo MMP 2DRD.

Na Seção 5.1 são discutidos os aspectos básicos do funcionamento do MMP 3DRD e a apresentação da primeira versão do algoritmo com uma partição do bloco de entrada que seguiu o padrão original de segmentação já utilizado para o MMP 2DRD. Para a exemplificação do desempenho, utilizam-se um conjunto de imagens dos *Toy Vehicles* e 03 (três) conjuntos de 4 (quatro) imagens de radar meteorológico com algumas dimensões diferentes definidas para o bloco de entrada. Ao longo da referida seção, são apresentados os principais resultados alcançados.

Na Seção 5.2, há uma comparação dos desempenhos dos códigos MMP 3DRD e MMP 2DRD onde chega-se a algumas conclusões que possibilitaram a extensão do trabalho. Alguns experimentos reproduzem resultados para blocos de entrada com o mesmo número de *bytes*, quer seja em duas ou três dimensões. Tenta-se, com isto, estabelecer um aspecto de paridade entre os códigos. Nada impede, contudo, que sejam utilizados blocos de entrada de dimensões maiores e algumas simulações também têm suas conclusões apresentadas e discutidas.

Uma análise dos resultados obtidos na Seção 5.2 motivou a criação de uma nova versão do MMP 3D intitulada de MMP 3DRDII. Com isto, na Seção 5.3, explicita-se uma nova forma de segmentação dos blocos e apresentam-se alguns resultados práticos para as seqüências de imagens *PPI*.

5.1 Princípios Básicos do MMP 3D

Ao longo de todo o Capítulo 4, utilizaram-se os conceitos de MMP Com e Sem Multiescalas e com a Otimização $\mathcal{RD}$. Tendo em vista que ficou comprovado que a otimização RD proporciona a melhor opção, não mais serão usados os códigos que não tenham a referida otimização. Logicamente, dada à facilidade de adaptação do MMP ao número de dimensões requerido, os códigos 3D Sem Multiescalas e Com Multiescalas foram implementados para a corroboração de resultados.

No caso do MMP 2D, usa-se uma matriz que representa uma imagem $\mathbf{X}^{m,n}$, com m significando o número de linhas e n, o de colunas. Nada impede que a matriz seja unidimensional $1 \times n$, sendo uma utilização prática o uso em eletrocardiogramas ou eletroencefalogramas.

O MMP 3D utiliza os mesmos conceitos aplicados ao MMP 2D. A Figura 5.1 ilustra o diagrama em blocos do algoritmo.

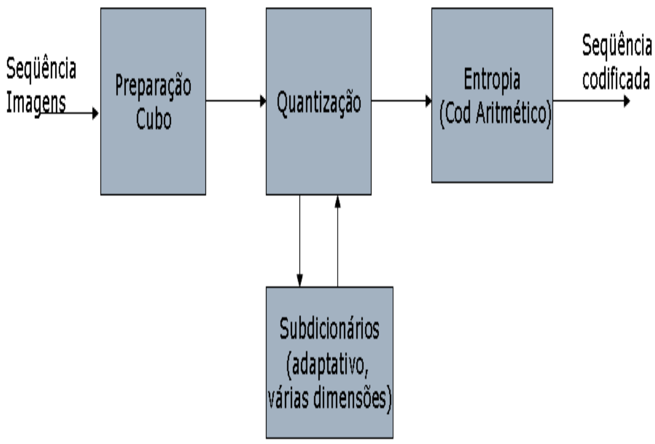


Figura 5.1: Diagrama em Bloco do MMP 3D

Comparando a Figura 5.1 com a apresentada em 4.1, vê-se que a diferença básica é que agora há a presença de um **cubo** em vez de uma matriz no bloco de entrada.

O processo de codificação do MMP 3DRD assemelha-se muito com o MMP 2DRD. Evidentemente há a adequação do código à nova situação. Assim, no MMP 2D havia uma matriz de entrada, após a preparação do bloco, e todo o processo levava em conta os valores correspondentes às dimensões das linhas e colunas.

No MMP 3DRD a matriz de entrada é tridimensional $X^{m.n,p}$ e trocam-se, somente, os nomes dos componentes do bloco preparado, que, agora, é chamado de cubo. Com isto, aparecem as dimensões chamadas altura, largura e profundidade. O nome cubo foi escolhido somente porque é um volume e o processamento é tridimensional. Contudo, não necessariamente as 03 (três) dimensões são iguais e isto foi explorado ao longo dos experimentos efetuados.

O objetivo básico, no processamento dos blocos pelo MMP 3DRD, é o mesmo daquele para o MMP 2DRD, ou seja, é o de descobrir a melhor aproximação, em um dicionário $\mathcal{D}$, para um vetor de entrada $\mathbf{X}_i$. A regra para a decisão é o custo lagrangeano para a codificação dos nós filhos $\mathbf{X}_{2i+1}$ e X_{2i+2} em relação ao nó pai.

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

Quanto à distorção D, é feita a diferença média quadrática entre os *pixels* de entrada e os dos elementos do dicionário, calculada por:

$$d(\mathbf{X}_i, \mathbf{V}_k) \leq D \tag{5.2}$$

$$d(\mathbf{X}_i, \mathbf{V}_k) \leq D = \sum_i \sum_j \sum_k (x_{i,j,k} - v_{i,j,k})^2 \tag{5.3}$$

A taxa R é composta pela contribuição dos índices dos nós folhas somado à taxa dos *flags* correspondentes, conforme descrito na Seção 4.3.

Mais uma vez enfatiza-se a finalidade dos *flags*, pois são eles que possibilitam que o decodificador, neste caso o UNMMP 3DRD, entenda a seqüência passada pelo codificador.

Conforme evidenciado no Capítulo 4 há, sempre, um critério de como partir o bloco de entrada. Nos algoritmos iniciais em 03 (três) dimensões desenvolvidos para esta dissertação, optou-se pela divisão, nesta ordem, da profundidade, largura e altura. Entende-se, como profundidade, o número de imagens carregadas para o processamento simultâneo. Para esta situação, nomeou-se os códigos de MMP 3DRD e UNMMP 3DRD, respectivamente, para o codificador e decodificador. A Figura 5.2 ilustra o processo inicial.

Todos os critérios de partição, discutidos na Seção 4.3 continuam válidos e, no caso do nível abaixo, se $\mathbf{X}_1$ e/ou $\mathbf{X}_2$ não atingirem a condição imposta de que o custo

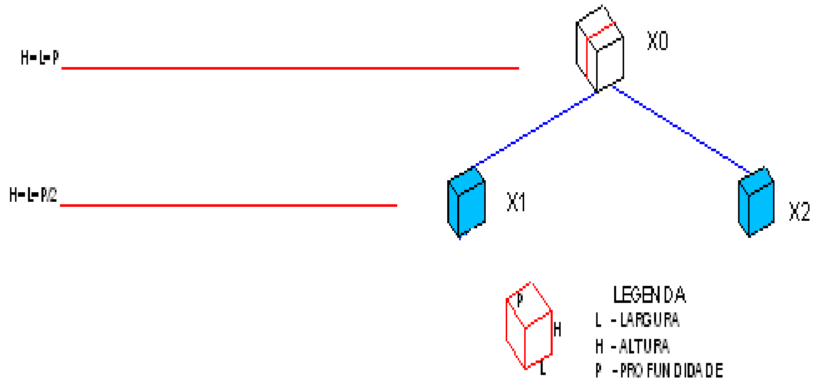


Figura 5.2: Critério de Partição 3D

lagrangeano seja menor do que o de seu pai, haverá a poda dos nós filhos, sendo que a regra é que um bloco pai $\mathbf{X}_j$ gera, se segmentado, os filhos $\mathbf{X}_{2j+1}$ e $\mathbf{X}_{2j+2}$.

O número de subdicionários, para o caso 3D, é dado por:

$$N = 1 + \log_2(mnp) \tag{5.4}$$

Como visto para o MMP 2D, são 07 (sete) o número de subdicionários para as dimensões dos blocos de entrada 8×8. No caso do MMP 3D, o mesmo número de subdicionários é atingido para um cubo de entrada com dimensões $4 \times 4 \times 4$ e as possibilidades de segmentação são mostradas na Figura 5.3.

Um fato que ficou demonstrado, ao longo do trabalho, foi que para imagens 2D com o mesmo número de *pixels* de uma imagem 3D, o processamento usando as versões 3D conduziu a tamanhos de subdicionários bem diferentes, sendo maior para os códigos 3D.

A fim de ilustrar os números de elementos de cada subdicionário após a codificação, a Tabela 5.1 traz os tamanhos com os parâmetros $\lambda = 60$ e $\lambda = 80$ para a seqüência de imagens *Toy Vehicles* com MMP 2DRD e MMP 3DRD. Nas linhas horizontais da referida tabela aparecem as escalas possíveis com os códigos 2D e 3D. Assim, como exemplo, a escala 1, para o processamento em duas dimensões, representa a situação em que o bloco de entrada, de dimensões 16×16 (escala 0), foi segmentado em 02 (dois) outros, de dimensões 16×8. A mesma análise pode ser aplicada aos códigos 3D. Assim, um bloco de entrada inicial de dimensões $8 \times 8 \times 4$, que representa a escala 0, originará, se segmentado, outros 2 (dois) blocos de dimensões $8 \times 4 \times 4$, que estão associados à escala 1.

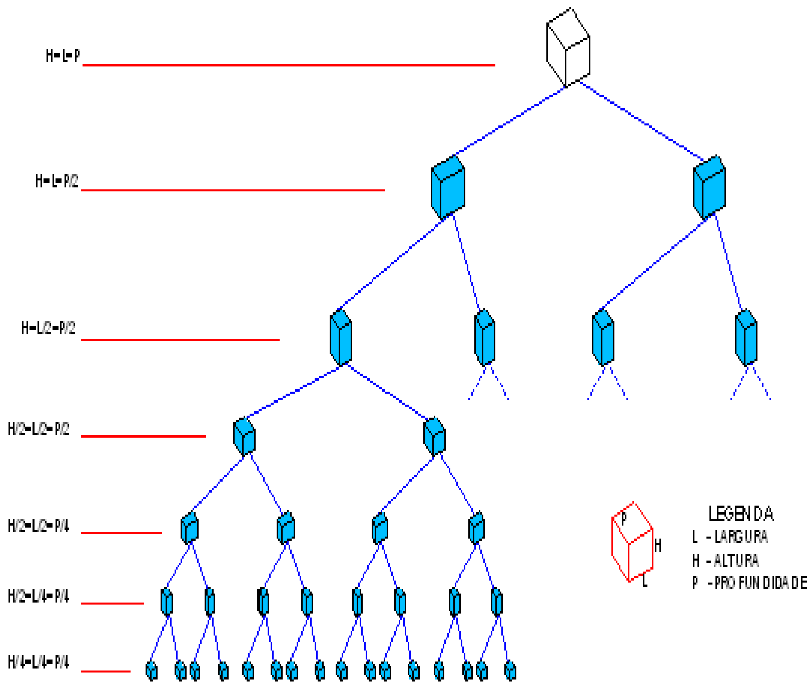


Figura 5.3: Critério de Partição dos Blocos no MMP 3D

Logicamente, os únicos subdicionários que não sofrem mudança no número de elementos são os relacionados às segmentações que conduzem aos menores blocos, ou seja, 1×1, para o MMP 2D, e $1 \times 1 \times 1$, para o MMP 3D, que são construídos com 256 elementos e terminam com o mesmo número após a codificação.

Outro fato que merece destaque, e isto acontece com qualquer simulação que envolva MMP, é que o tamanho dos subdicionários é proporcional às taxas conseguidas e aumentam com a diminuição do valor de λ empregado.

Nos experimentos para o MMP 3DRD, as 04 (quatro) imagens de *Toy Vehicles* foram codificadas para as dimensões de $4 \times 4 \times 4$ e $8 \times 8 \times 4$. A fim de ilustrar a seqüência são mostradas, por meio da Figura 5.4, as 2 (duas) primeiras imagens. Com o uso de um *player* configurado para uma determinada taxa de reprodução, tem-se a noção de movimento e, com isto, chega-se à conclusão de que o código pode ser usado para vídeos.

Nota-se, pela Figura 5.5, que a codificação com o bloco $8 \times 8 \times 4$ superou, com facilidade, aquela com o bloco $4 \times 4 \times 4$. Como exemplo, para uma taxa de 0.12 *bits/pixel*

MMP 3DRD			MMP 2DRD		
Escala	lambda=60	lambda=80	Escala	lambda=60	lambda = 80
0	5364	4526	0	3624	3096
1	5365	4525	1	3625	3096
2	5362	4530	2	3628	3092
3	5295	4510	3	3624	3091
4	5240	4460	4	3614	3075
5	5005	4253	5	3596	3062
6	4005	3424	6	3477	2964
7	2332	2097	7	2334	2054
8	256	256	8	256	256

Tabela 5.1: Tabela Exemplo de Tamanhos de Subdicionários

a diferença foi de cerca de 2 *dB*.

Com respeito às imagens de radares meteorológicos, os mesmos tamanhos de blocos foram aplicados às seqüências *PPI* de 04 (quatro) imagens com os resultados, de uma delas, reproduzidos pela Figura 5.6 onde há uma ligeira vantagem para os testes com blocos $4 \times 4 \times 4$. O desempenho do código foi praticamente idêntico para os 03 (três) casos, sendo que na *PPI2* houve sobreposição das curvas e para a *PPI3* os resultados com bloco $8 \times 8 \times 4$ foram, ligeiramente, melhores.

Os blocos $8 \times 8 \times 4$ possuem 4 (quatro) vezes mais *pixels*, do que no caso em que os blocos de entrada são de $4 \times 4 \times 4$. Com isto, os tempos de processamento, tanto do MMP 3DRD quanto do UNMMP 3DRD são maiores, quanto maiores forem os blocos de entrada $\mathbf{X}_i$. Outro fato que provoca o aumento do tempo de processamento é que o número de subdicionários $\mathcal{D}^s$ aumenta com o aumento do tamanho do bloco a ser comparado, conforme visto pela equação 5.4.

Pode-se concluir que não vale à pena aumentar indefinidamente o tamanho dos blocos entrada, porque o conseqüente aumento da complexidade computacional nem sempre será acompanhado de um ganho proporcional de desempenho. Uma outra conclusão alcançada é a de que no caso específico de imagens *PPI* pode ser notado, pela Figura 5.6, que nem sempre o aumento das dimensões do bloco do entrada é acompanhado da melhoria de desempenho.

Figura 5.4: Imagens Recuperadas com $\lambda = 20$

5.2 Comparação entre o MMP 2D e MMP 3D

Após a realização dos experimentos para o MMP 2DRD e para o MMP 3DRD foram feitas comparações dos desempenhos entre os códigos.

Para *Toy Vehicles* a simulação $2D$ com as dimensões 16×16, para a matriz de entrada $\mathbf{X}_i$, superou o resultado alcançado para as dimensões 8×8. De forma análoga, a versão $3D$ para as dimensões do cubo de entrada $8 \times 8 \times 4$ teve melhores resultados do que os conseguidos com as dimensões $4 \times 4 \times 4$.

Assim, a Figura 5.7 ilustra a comparação em duas e três dimensões para *Toy Vehicles*. Nota-se que à medida que a taxa diminui, ou seja, quanto maior for a distorção D, o desempenho do MMP 3DRD (Blocos $8 \times 8 \times 4$) se aproxima do MMP 2DRD (Blocos 16×16), porém para taxa maiores o MMP 2DRD apresentou desempenho melhor. Outro ponto de destaque foi que para o MMP 3DRD com blocos 3D ($4 \times 4 \times 4$) o desempenho foi melhor do que o seu correlato em duas dimensões, MMP 2DRD, com blocos 8×8.

Uma característica das imagens utilizadas, e que facilitou o processo de compara-

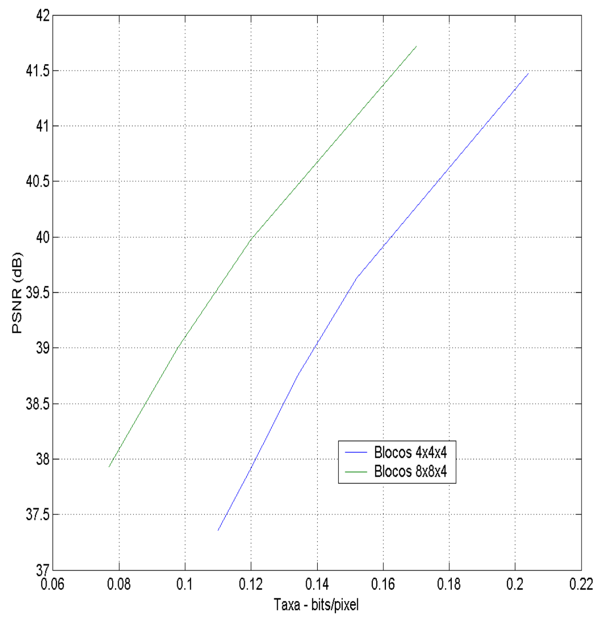


Figura 5.5: Comparação do MMP 3DRD *Toy Vehicles* com Blocos 8x8x4 e 4x4x4

ção, é a de que uma seqüência de 04 (quatro) imagens *Toy Vehicles* possui 2048×512 linhas por colunas e as 03 (três) seqüências com *PPI* possuem 1440×400. Com isto, há a possibilidade de comparação de processamento de blocos $4 \times 4 \times 4$ (3D) com 8×8 (2D) e $8 \times 8 \times 4$ (3D) com 16×16 (2D), pois há a manutenção do mesmo número de *bytes* e de mesmo número de subdicionários para o MMP 2DRD e para o MMP 3DRD.

A Figura 5.8 reproduz os resultados para um conjunto de imagens, chamado de *PPI1* equivalentes, em número de *bytes*, sendo que para o $2D$ foram utilizados blocos 8×8 e 16×16 e, para $3D$, blocos $4 \times 4 \times 4$ e $8 \times 8 \times 4$.

A fim de ilustrar o comportamento com mais um conjunto de imagens *PPI*, a Figura 5.9 traz o gráfico dos resultados para outra seqüência de 04 (quatro) imagens de radar meteorológico, chamado de *PPI2*.

Para *PPI1* e *PPI2* o resultado comprovou que com o MMP 2DRD o desempenho foi melhor em relação ao MMP 3DRD. Diferentemente da seqüência de imagens *Toy Vehicles*, não há, para as imagens *PPI*, a tendência de concorrência das curvas à medida que a distorção D aumenta. Outro dado comprovado foi que há, praticamente, a concordância de curvas das simulações com os tamanhos de blocos de entrada $8 \times 8 \times 4$ e $4 \times 4 \times 4$, o mesmo acontecendo com as simulações feitas para 02 (duas) dimensões, ou seja, o aumento das dimensões não proporcionou melhoria de desempenho.

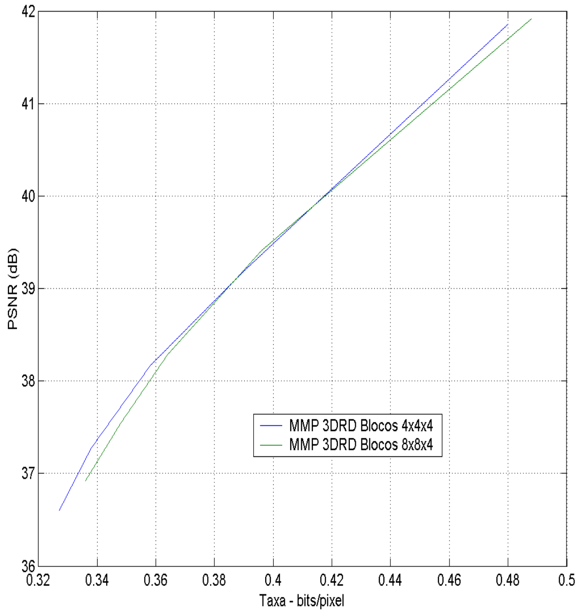


Figura 5.6: Comparação do MMP 3DRD para a Seqüência *PPI1*

Para as imagens *Toy Vehicles*, o aumento nas dimensões dos blocos de entrada para $8 \times 8 \times 4$ e 16×16 possibilitou melhores desempenhos *PSNR* $\times$ *Taxa* em relação às simulações feitas com blocos $4 \times 4 \times 4$ e 8×8, respectivamente.

Acrescenta-se que o tempo de processamento de cada seqüência aumenta sobremaneira com o aumento das dimensões dos blocos. Apenas como exemplo, uma seqüência de *PPI* com o $\lambda = 20$ levou $1:03:00$ h para ser codificada com as dimensões $4 \times 4 \times 4$. Para as dimensões $8 \times 8 \times 4$, a mesma seqüência consumiu $1:43:00$ h em um computador com processador *Intel Centrino Duo*, com 1.83 *GHz* de *clock* e memória *RAM* de 1 *GBytes*. Ressalta-se, mais uma vez, que todo o desenvolvimento, na atual fase, foi dedicado para a otimização do desempenho em termos da taxa-distorção.

5.3 Algoritmo MMP 3DRDII

Ao longo da Seção 5.2 foram feitas comparações entre as versões em duas e três dimensões para o MMP RD e ficou evidenciado que a versão $2D$ do código conduziu a melhores resultados.

No caso do código MMP 2DRD, as imagens são tratadas de forma contínua e justapostas com a varredura de toda a sua dimensão, obedecendo ao incremento das

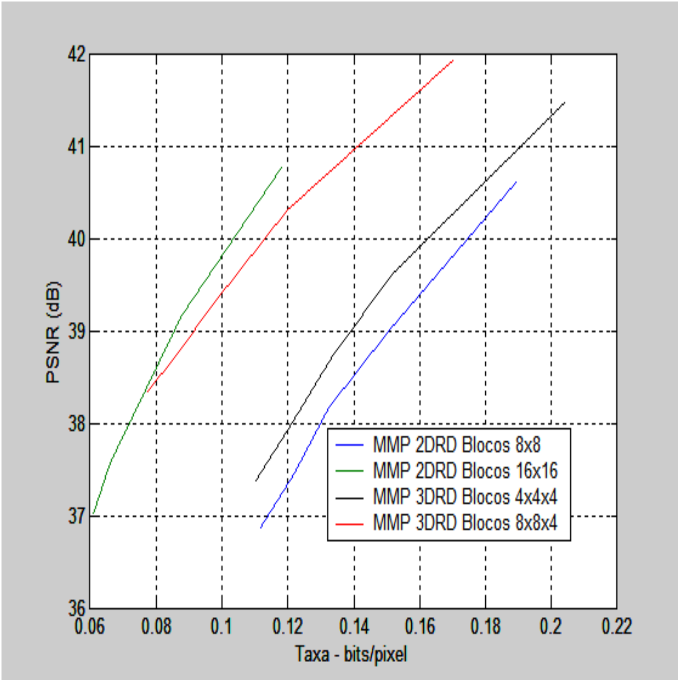


Figura 5.7: Comparação do MMP 3DRD com o MMP2DRD para *Toy Vehicles*

linhas e colunas, de acordo com as dimensões do bloco de entrada $\mathbf{X}_i$ passado.

Para a situação do MMP 3DRD, as imagens são tratadas sobrepostas umas às outras, seguindo os parâmetros de altura, largura e profundidade estipulados.

Esperava-se, antes da realização das simulações, que o MMP 3DRD apresentasse melhores desempenhos em relação ao MMP 2DRD, sobretudo com respeito seqüência de imagens de *Toy Vehicles*. A espectativa sustentava-se nas características da seqüência das imagens que apresentam pouca variação estatística de uma imagem para outra.

Ao contrário, para as imagens *PPI*, em que de uma rotação do sistema irradiante para a seguinte decorre o tempo aproximado de 06 (seis) segundos e, devido ao deslocamento das nuvens, a espectativa era de que o MMP 3DRD fosse superado pelo MMP 2DRD.

Contudo, a menos da situação para distorções maiores, para o caso dos blocos $8 \times 8 \times 4$, ou com o uso de blocos menores, para blocos de entrada $4 \times 4 \times 4$, em que o código MMP 3DRD começa a apresentar melhor desempenho, conforme demonstrado na Figura 5.7, a seqüência de imagens dos *Toy Vehicles* foi melhor codificada com a versão $2D$. Para as imagens *PPI*, o que foi pensado foi corroborado e o desempenho do MMP 2DRD foi bem melhor do que aquele do MMP 3DRD em todas as 03 (três) seqüências de imagens ensaiadas.

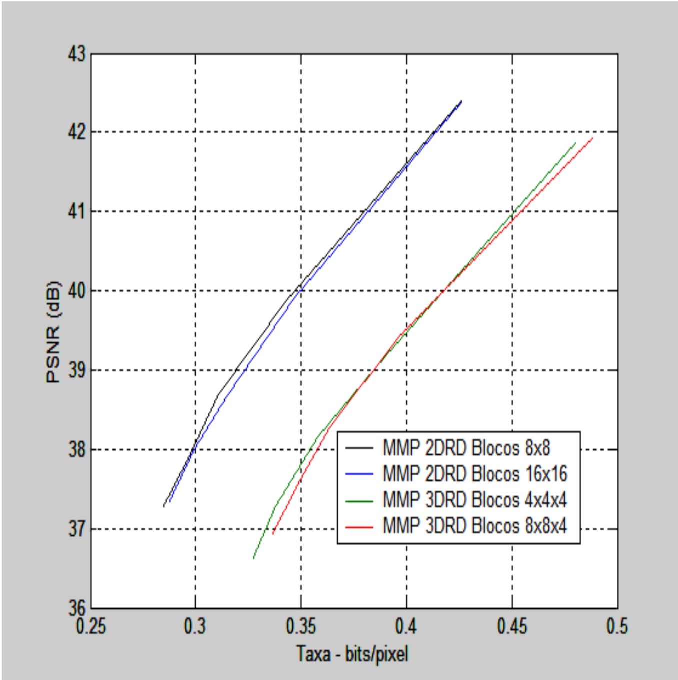


Figura 5.8: Comparação do MMP 3DRD com MMP2DRD para PPI1

Um fato que pode ser apontado para justificar os desempenhos é que, a despeito da segmentação do MMP 2DRD ser bidimensional, o processamento é tridimensional, pois a codificação das imagens posteriores leva em conta a estatística já incorporada aos dicionários das imagens anteriores. No caso do MMP 3DRD, o processamento é tridimensional, pois há o carregamento das imagens antes do processamento, e a segmentação também é tridimensional.

Face ao exposto, pensou-se em melhorar o código para 03 dimensões com a adição de um novo conceito de partição de blocos.

5.3.1 Conceitos Básicos do MMP 3DRDII

No MMP 3DRD, convencionou-se partir na seguinte seqüência: profundidade, largura e altura, conforme mostrado na Figura 5.3. Seja um novo critétio de segmentação dos blocos, onde a seguinte seqüência seja observada para chamadas recursivas: profundidade, dividida até a unidade, largura e altura.

A conclusão é que chega-se a um ponto em que o cubo original vira uma matriz de dimensões $altura \times largura \times 1$. Assim, no limite, o MMP 3DRD acaba funcionando como o MMP 2DRD.

A Figura 5.10 exemplifica os critério de partição de um cubo de dimensões $4 \times 4 \times 4$

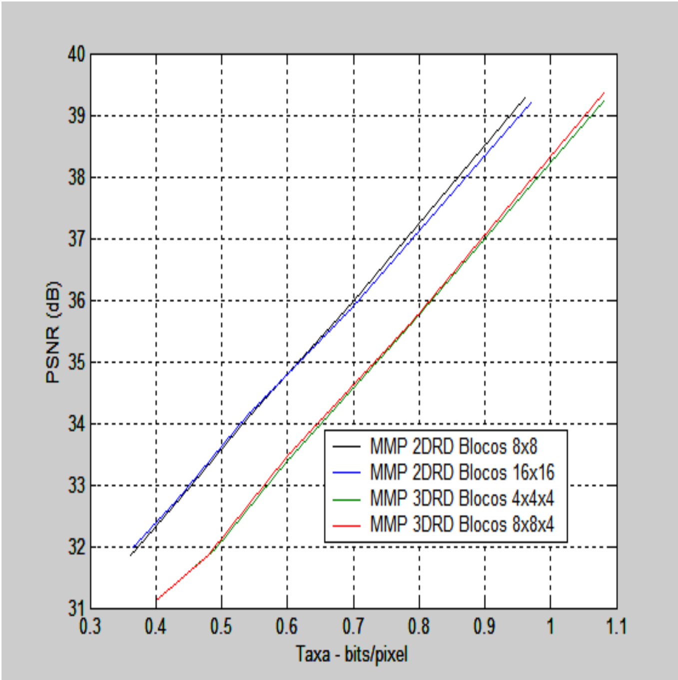


Figura 5.9: Comparação do MMP 3DRD com o MMP2DRD para PPI2

onde, a partir do terceiro nível, já se tem uma matriz de dimensões 4×4.

5.3.2 Resultados Experimentais com o MMP 3DRDII

Com a adoção da partição do MMP 3DRDII, alguns resultados interessantes foram atingidos e puderam comprovar, em parte, aquilo o que se pensava antes da execução das simulações.

A fim de proporcionar uma visão global das principais simulações levadas a cabo para as imagens dos *Toy Vehicles*, apresentam-se, na Figura 5.11 os produtos, não somente, das implementações com o MMP 2DRD, mas, também, os resultantes do MMP 3DRD e a incorporação dos conseguidos com a aplicação do MMP 3DRDII.

Com a utilização das imagens dos *Toy Vehicles* houve a comprovação de que não se obtém melhorias, em relação ao MMP 3DRD com dimensões $8 \times 8 \times 4$, utilizando-se o MMP 3DRDII com as mesmas dimensões. Isto foi uma surpresa porque esperava-se um desempenho semelhante ao obtido pelo MMP 2DRD. Contudo, para taxas menores, há uma tendência de concorrência entre o MMP 3DRDII (Blocos $8 \times 8 \times 4$), MMP 3DRD (Blocos $8 \times 8 \times 4$) e o MMP 2DRD Blocos (16×16).

Destaca-se, também, o desempenho dos MMP 3DRD e MMP 3DRDII, com blocos $4 \times 4 \times 4$ que, ao contrário do que aconteceu com os que utilizaram blocos $8 \times 8 \times 4$ em

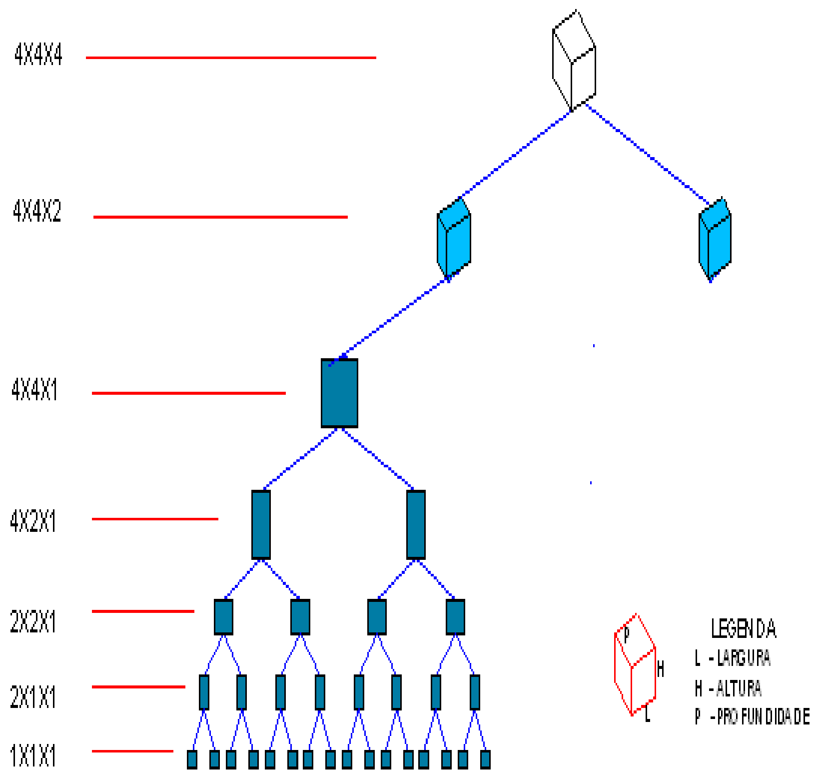


Figura 5.10: Critério de Partição dos Blocos no MMP 3DRDII

relação ao código com blocos 16×16, conseguiram superar o MMP 2DRD com blocos 8×8. Com isto, pode-se concluir que para volumes menores (Blocos $4 \times 4 \times 4$) conseguem-se mais casamentos nos níveis mais altos da árvore binária.

A Figura 5.12 finaliza o capítulo com os resultados obtidos para uma seqüência de imagens *PPI*, sendo que as outras 2 (duas) apresentaram resultados semelhantes aos apresentados.

Os resultados alcançados pelos MMP 2DRD, quer para dimensões 8×8 ou para 16×16, foram melhores do que qualquer implementação feita com os algoritmos MMP 3DRD e MMP 3DRDII. Uma observação pertinente é que para o caso das imagens de radar meteorológico *PPI*, que apresentam maior variação estatística do que para as *Toy Vehicles*, as simulações com o MMP 3DRDII levaram a melhores resultados do que os conseguidos com o MMP3DRD. Com isto, chega-se à conclusão de que há uma tendência à aproximação do desempenho do MMP 2DRD com a regra de segmentação levando a profundidade até a dimensão 1 antes de começar a segmentar a largura e altura.

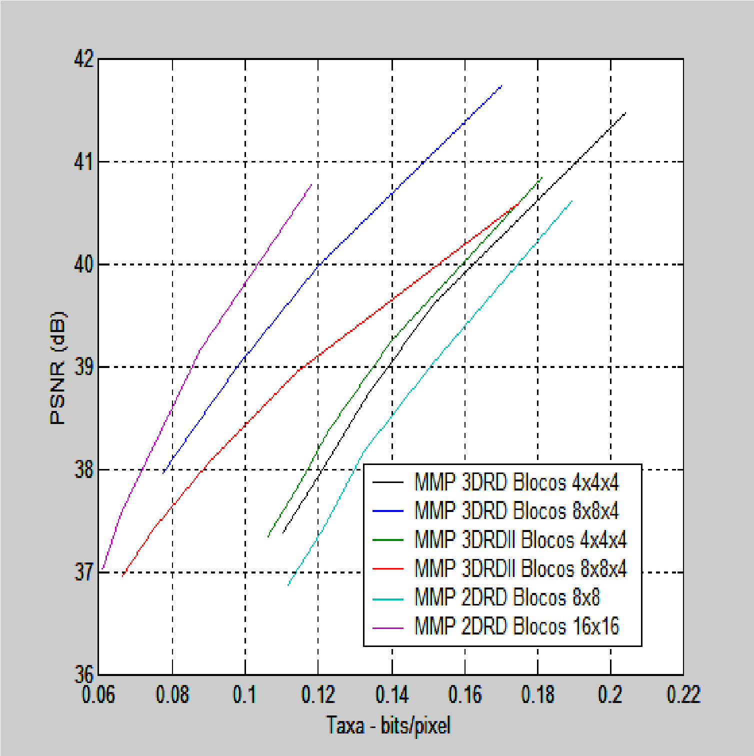


Figura 5.11: Comparação Global para *Toy Vehicles*

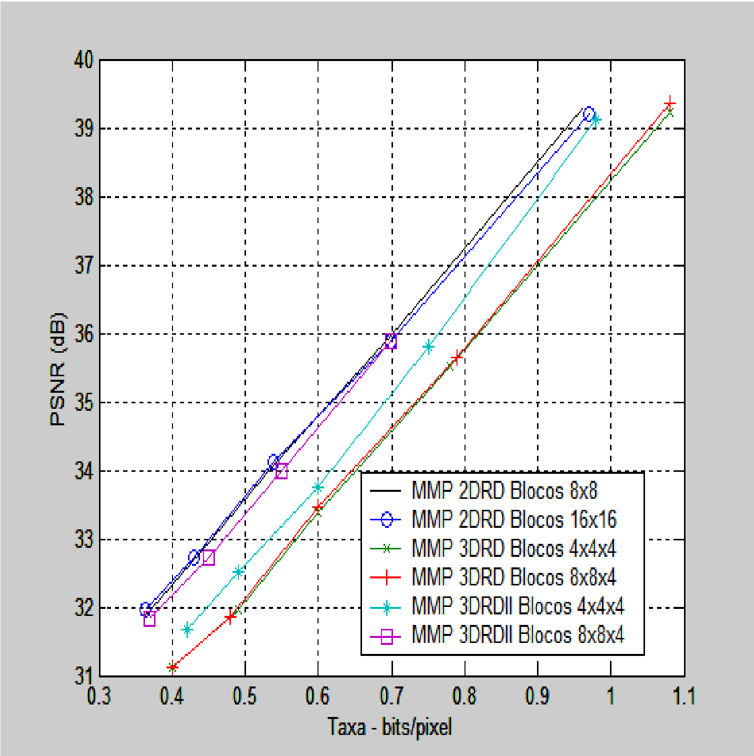


Figura 5.12: Comparação Global para *PPI2*

Capítulo 6

Algoritmos MMP RDFT

Durante o desenvolvimento do trabalho, algumas conclusões foram sendo tiradas e conduziram a novas implementações e adaptações, tanto do MMP 2DRD quanto do MMP 3DRD.

É fato, contudo, que uma das espectativas era a de que o desempenho em três dimensões suplantaria aquele em duas dimensões. Contudo, os resultados não confirmaram o esperado e foram feitas modificações no MMP 2DRD e MMP 3DRD a fim de que o desempenho melhorasse.

Com isto, o presente capítulo apresenta as últimas versões desenvolvidas, que são os algoritmos MMP 2DRDFT e MMP 3DRDFT, e os seus resultados em relação às versões anteriores.

Um outro objetivo é o de comparar o desempenho com o H.264 (Joint Video Team, 2005) que, correntemente, é o algoritmo de melhor desempenho para vídeos, e também frente ao JPEG 2000 (Taubman and Marcelin, 2001), que é usado, com freqüência, para imagens.

A Seção 6.1 apresenta os algoritmos MMP 2DRDFT e MMP 3DRDFT, sendo as letras *FT* a abreviação de *Full Tree*, e faz uma comparação entre a complexidade computacional entre estes e os outros principais códigos discutidos ao longo deste trabalho.

Na Seção 6.2, são vistos uma série de resultados conseguidos para as imagens mostradas no Apêndice A para o MMP 2DRD e o MMP 2DRDFT. Também são apresentados os resultados comparativos para as imagens de *Toy Vehicles* e os 03 (três) conjuntos de imagens *PPI* com os códigos MMP 3DRD, MMP 3DRDII e MMP 3DRDFT. Com o fito de proporcionar uma visão panorâmica acerca dos resultados de todos os códigos, estão presentes as comparações entre todos os programas MMP em 02 (duas) e (03) dimensões. Aparece, por fim, a comparação entre o desempenho dos códigos MMP, JPEG 2000 e H.264 para as seqüências de imagens de *Toy Vehicles* e de *PPI*.

6.1 O Código MMP RDFT

Nos Capítulos 4 e 5 foram ilustradas as formações dos códigos para o MMP 2DRD e o MMP 3DRD, respectivamente. O critério de divisão dos blocos de entrada foi o de achar o menor custo Lagrangeano $J(n_l) = D(n_l) + \lambda R(n_l)$. Definiu-se que a $R(n_l) = -\log_2(P(i_l))$ e o D, a diferença média quadrática entre os *pixels* do bloco de entrada e dos elementos do dicionário.

Se $Jn_l < Jn_{2l+1} + Jn_{2l+2} + R('0')$, então ocorre a poda da árvore e o nó folha é definido como Jn_l. Contudo, a regra de segmentação sempre seguiu uma sistemática fixa.

Nos códigos dos capítulos supracitados, do bloco inicial $\mathbf{X}_0$, designado nó raiz, há a derivação para os nós filhos $\mathbf{X}_1$ e $\mathbf{X}_2$, seguindo o critério de que os nós filhos são $\mathbf{X}_{2i+1}$ e $\mathbf{X}_{2i+2}$ até a construção da árvore completa. Para o caso do MMP 2D, em que o bloco de entrada é uma matriz $X^{m,n}$, sendo m o número de linhas e n o de colunas, optou-se pela primeira segmentação no sentido das colunas. Assim, as matrizes derivadas têm as dimensões $X^{m,n/2}$. Para todos os nós folhas foi tomada a condição $Jn_l < Jn_{2l+1} + Jn_{2l+2} + R('0')$ e caso não seja atingida, é feita a poda dos nós filhos $\mathbf{X}_{2i+1}$ e $\mathbf{X}_{2i+2}$ e codifica-se o nó pai $\mathbf{X}_i$.

No caso do MMP 3D, os mesmos princípios básicos se aplicam com o cubo de entrada de dimensões $X^{m,n,p}$, sendo m a altura, n a largura e p a profundidade. Como regra de segmentação, optou-se por partir na seqüência fixa: profundidade, largura e altura.

Tendo em vista os resultados alcançados no Capítulo 5, onde mesmo com o MMP 3DRDII o desempenho não igualou ao obtido pelo MMP 2DRD, desenvolveu-se um código que possibilitou a obtenção de resultados melhores.

A teoria básica para os códigos MMP 2DRDFT e MMP 3DRDFT baseia-se no fato de que ao invés de decidir-se pela partição segundo uma seqüência pré-estabelecida, aproveitam-se todas as possibilidades de partição para o nível abaixo, ou seja, testam-se as possibilidades de partição nas linhas e nas colunas, para o caso do MMP 2D, e na altura, largura e profundidade, para o caso do MMP 3D.

A Figura 6.1 apresenta a regra de segmentação para o MMP 2DRDFT.

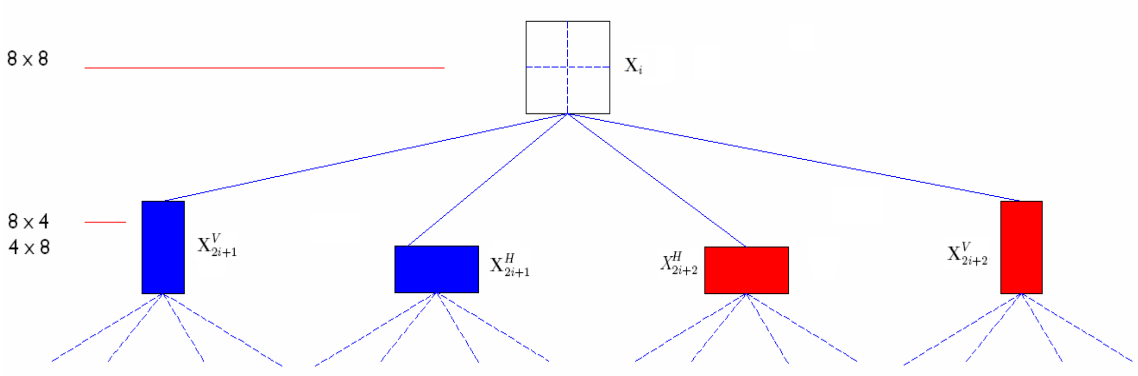


Figura 6.1: Árvore de Segmentação para o MMP 2DRDFT

Conforme pode se notado na Figura 6.1, o nó pai $\mathbf{X}_i$ deu origem a 04 (quatro) nós filhos que são dados por: $\mathbf{X}^V_{2i+1}$, $\mathbf{X}^V_{2i+2}$, com segmentação *vertical*, e $\mathbf{X}^H_{2i+1}$, X^H_{2i+2}, com segmentação *horizontal*. As duas opções, segmentação vertical e segmentação horizontal, são testadas, e a melhor opção no sentido RD é selecionada.

Com respeito ao MMP 3DRDFT, a mesma lógica do MMP 2DRDFT é utilizada com a diferença de que cada cubo segmentado no primeiro nível gera 06 (seis) no nível abaixo. A Figura 6.2 mostra o critério de segmentação para o MMP 3DRDFT.

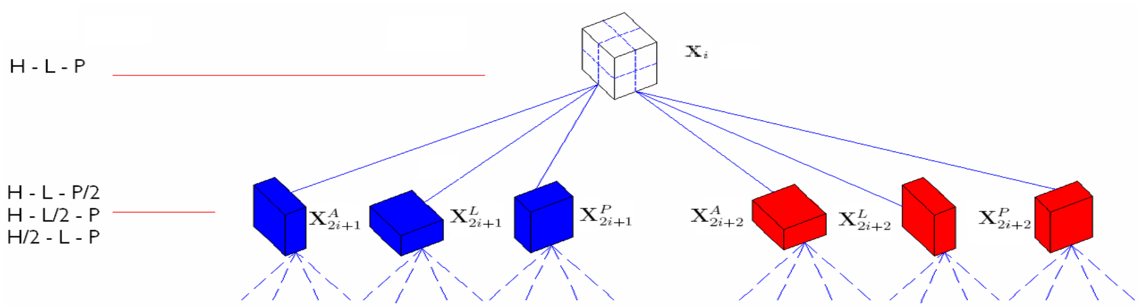


Figura 6.2: Árvore de Segmentação para o MMP 3DRDFT

Assim, o MMP 3DRDFT parte do nó raiz $\mathbf{X}_i$ e há a criação de seis novos nós: $\mathbf{X}^A_{2i+1}$, $\mathbf{X}^A_{2i+2}$, $\mathbf{X}^L_{2i+1}$, $\mathbf{X}^L_{2i+2}$, $\mathbf{X}^P_{2i+1}$, $\mathbf{X}^P_{2i+2}$.

Para o melhor entendimento acerca do MMP FT, será feita a apresentação do algoritmo para um vetor de entrada bidimensional $\mathbf{X}^{m,n}$. O mesmo processo pode ser empregado para o MMP 3DRDFT.

Utiliza-se um dicionário $\mathcal{D} = \bigcup_{s=0}^{S-1} \mathcal{D}^s$, sendo S o numero de subdicionários e L_s o número de elementos de cada subdicionário. Cada elemento $\mathbf{V}_k$ de cada subdicionário possui as dimensões que tornam compatíveis as comparações com o vetor $\mathbf{X}^{m,n}$ de entrada. Toma-se por base, também, o fornecimento de um λ_{alvo} para o processamento de cada bloco.

O procedimento usado para encontrar a árvore de segmentação ótima é mostrado, a seguir:

Passo 1: Inicialize $\mathcal{T}$ como a árvore *quaternária* completa de profundidade $P_{max} = log_2(NM) + 1$, considerando a partição nas linhas (H) e nas colunas (V).

Passo 2: Calcule os custos Lagrangeanos J_j^V e J_j^H ótimos para todos os nós, correspondentes a uma segmentação vertical n_j^V, bem como para todos os nós correspondentes a uma segmentação horizontal n_j^H, de $\mathcal{T}$. Observa-se que existirão $4^{P_{max}} - 1$ nodos distintos.

Passo 3: Faça $p = P_{max} - 2$.

Passo 4: Para todos os nós n_j^V na profundidade p, comparar o custo $JM = J_j^V + J_{FM}$ com a soma os custos dos descendentes verticais $JV = J_{2j+1}^V + J_{2j+2}^V + J_{FV}$ e com a soma os custos dos descendentes horizontais $JH = J_{2j+1}^H + J_{2j+2}^H + J_{FH}$ (JM, JV e JH são os custos para codificar os *flags* de match, segmentação vertical e segmentação horizontal respectivamente). Se JM for o menor custo dentre todos, as quatro subárvores descendentes de n_j^V devem ser podadas. Caso o menor custo seja JV, deve-se podar as duas subárvores descendentes horizontais de n_j^V e faz-se o custo do nodo $n_j^V = JV$. Caso o menor custo seja JH, deve-se podar as duas subárvores descendentes verticais de n_j^V e faz-se o custo do nodo $n_j^V = JH$.

Passo 5: Fazer o mesmo descrito no passo 4, porém agora para os nós n_j^H.

Passo 6: Fazer $p = p - 1$. Se $p \geq 0$, retorna ao passo 4. Senão, termina.

É importante ressaltar que, embora durante a otimização seja utilizada uma árvore quaternária, a árvore final resultante será binária porque nos passos 4 e 5 do procedimento de otimização pelo menos duas das subárvores dentre as quatro descendentes de cada nó serão podadas. Assim, para *codificação*, pode-se utilizar o mesmo procedimento anteriormente utilizado no MMP, apenas substituindo-se os *flags* binários ('0' representando segmentação e '1' representando casamento) por *flags* ternários ('V' representando segmentação vertical, 'H' representando segmentação horizontal e 'M' representando casamento).

6.1.1 Complexidade Computacional para o MMP 2DRD

A principal dificuldade para a implementação do MMP RDFT, quer seja em 02 (duas) ou 03 (três) dimensões está relacionada ao fato de que a complexidade computacional aumenta bastante em relação ao MMP RD.

Ao longo da presente subseção, serão discutidas as complexidades computacionais atinentes aos códigos desenvolvidos para o MMP em 02 (duas) dimensões.

Os recursos computacionais demandados estão relacionados ao número de operações feitas para o Quantizador Vetorial (Q_v) utilizado. Isto se deve ao fato de que a comparação entre os blocos de entrada e do quantizador vetorial ocorrem *pixel* a *pixel*.

A fração de programa, as seguir, é um método usado no MMP 2D, qualquer que seja a versão, e que ilustra a quantidade de cálculos matemáticos envolvidos.

```
float Matriz::DistanciaEntreMatrizes (Matriz Lida)
 {
long int n,m, MatDif;
float D;


D = 0;

for (n=0; n<linha; n++) {
  for (m=0; m<coluna; m++) {
    MatDif = Lida.pixel[n][m] - pixel[n][m];
    D = D + MatDif*MatDif;
  }
}
return(D/(linha*coluna));

}
```

Figura 6.3: Método Distância entre Matrizes

Como pode ser observado por meio da Figura 6.3, para um bloco de Dimensões $N \times N$ ocorrerão N^2 multiplicações, N^2 subtrações e $N^2 - 1$ somas. Considerando que o subdicionário tem dimensão L, a complexidade computacional é dada por:

$$Cm = LN^2$$

Sendo o número de níveis coincidente com o número de escalas e dado por:

$$K = 1 + 2\log_2(N)$$

Chega-se à expressão final para o cálculo da complexidade computacional para o

MMP 2DRD original e que será chamado C_m^1:

$$Cm^1 = LN^2(1 + 2\log_2(N)) \tag{6.1}$$

Para o caso do MMP 2DRDFT, aparecerão, até a profundidade $p = \log_2(N)$, o dobro do número de Q_v, ou seja,:

$$Q_v = 1 + 2 + \ldots + 2^{\log_2(N)}$$

$$\sum_{k=0}^{\log_2 N} 2^k = 2^{\log_2(N)+1} - 1 = 2N - 1$$

A partir deste ponto, alguns nodos gerarão 02 (dois) descendentes, enquanto outros, gerarão 04 (quatro). Por exemplo, para $N = 8$, na profundidade $p = 3$, aparecerão alguns blocos de dimensão 8×1 e outros, de dimensão 1×8. Cada um dos referidos blocos somente podem ser partidos em outros 02 (dois) de dimensões 4×1 e 1×4, respectivamente.

Assim, é possível calcular 02 (dois) limitantes para a complexidade computacional do MMP 2DRDFT: um inferior, onde supõe-se que se tem todos os blocos, a partir da profundidade $p = \log_2(N) + 1$, gerando 02 (dois) descendentes; o outro limitante, que é o superior, onde supõe-se que cada nodo, a partir da mesma profundidade $p = \log_2(N) + 1$, gera 04 (quatro) descendentes.

Para o limitante inferior:

$$Q_v = 1 + 2 + \cdots + 2^{\log_2 N} + 2^{\log_2 N} + \cdots + 2^{\log_2 N}$$

Que conduz a

$$Q_v = \sum_{k=0}^{\log_2 N} 2^k + \log_2(N)2^{\log_2(N)} = 2N - 1 + N\log_2 N$$

Para o caso do limitante superior, o total de Q_v de dimensões $N \times N$ utilizados será $Qv = 1 + 2 + 4 + \cdots + 2^{2\log_2(N)}$, que conduz a:

$$Qv = \sum_{k=0}^{2\log_2(N)} 2^k = 2N^2 - 1$$

Com isto, a complexidade computacional do limitante inferior do MMP 2DRDFT é dada por:

$$Cm^2 = LN^2(2N - 1 + N\log_2 N) \tag{6.2}$$

E para o limitante superior:

$$Cm^2 = LN^2(2N^2 - 1) \tag{6.3}$$

Observando-se cuidadosamente a segmentação gerada, nota-se que vários nodos n_j^V e n_k^V na árvore completa usada no passo 2 do algoritmo de otimização correspondem, na realidade, ao mesmo subbloco de entrada. Por exemplo, se partirmos $\mathbf{X}_0$ com uma árvore de três níveis obteremos em $l = 1$: $\mathbf{X}_1^V$, $\mathbf{X}_2^V$, $\mathbf{X}_1^H$, $\mathbf{X}_2^H$. Em $l = 2$: $\mathbf{X}_3^{VV}$, $\mathbf{X}_4^{VV}$, $\mathbf{X}_3^{HV}$, $\mathbf{X}_4^{HV}$, $\mathbf{X}_3^{VH}$, $\mathbf{X}_4^{VH}$, $\mathbf{X}_3^{HH}$, $\mathbf{X}_4^{HH}$, $\mathbf{X}_5^{VV}$, $\mathbf{X}_6^{VV}$, $\mathbf{X}_5^{HV}$, $\mathbf{X}_6^{HV}$, $\mathbf{X}_5^{VH}$, $\mathbf{X}_6^{VH}$, $\mathbf{X}_5^{HH}$, $\mathbf{X}_6^{HH}$. São ao todo 21 subblocos, contudo alguns são idênticos. Por exemplo, $\mathbf{X}_3^{VH} = \mathbf{X}_3^{HV}$, $\mathbf{X}_6^{VH} = \mathbf{X}_6^{HV}$, $\mathbf{X}_4^{VH} = \mathbf{X}_5^{HV}$ e $\mathbf{X}_5^{VH} = \mathbf{X}_4^{HV}$. Durante a execução do passo 2, podemos identificar os subblocos iguais e calcular os custos para apenas um deles, repetindo o resultado para os demais. Assim, o número de operações de quantização vetorial requeridas passa a depender do número de escalas distintas, e não do número de nodos da árvore completa.

O número de escalas distintas pode ser avaliado considerando-se que há $\log_2(N)+1$ dimensões diferentes na direção vertical e $\log_2(N) + 1$ na direção horizontal.

Assim, o número de escalas para este processo, que é chamado de MMP 2DRDFT acelerado, é dado por:

$$N_{esc} = (\log_2(N) + 1)^2$$

A complexidade computacional, com isto, é proporcionada por:

$$Cm^3 = LN^2(\log_2(N) + 1)^2 \tag{6.4}$$

Para saber o quanto um código demanda mais recursos do que outro basta fazer a relação das expressões obtidas em cada caso.

No caso da relação entre o MMP 2DRDFT e o MMP 2DRD, tem-se que o limitante inferior proporciona:

$$\frac{Cm^2}{Cm^1} = \frac{(2N-1) + N\log_2 N}{(1 + 2\log_2(N))}$$

Para o limitante superior, aparece a expressão:

$$\frac{Cm^2}{Cm^1} = \frac{2N^2 - 1}{(1 + 2\log_2(N))}$$

Admitindo-se um $N = 8$, chega-se à conclusão de que $Cm^2/Cm^1 = 39/7$, para o limitante inferior, e $Cm^2/Cm^1 = 127/7$, considerando-se o limitante superior.

Para a situação em que se queira a relação entre o MMP 2DRDFT acelerado e o MMP 2DRD, tem-se:

$$\frac{Cm^3}{Cm^1} = \frac{(\log_2(N) + 1)^2}{(1 + 2\log_2(N))}$$

Para o mesmo $N = 8$, conclui-se que a relação conduz a $Cm^3/Cm^1 = 16/7$, que é bem menor do que a obtida anteriormente para o MMP 2DRDFT, independentemente se é considerado o limitante inferior ou superior.

6.1.2 Complexidade Computacional para o MMP 3DRD

Na Subseção 6.1.1, apresentaram-se as expressões que conduziram à análise de quanto um código é mais complexo que outro para 2 (duas) dimensões. A mesma análise será feita, nesta seção, com respeito aos códigos em 3 (três) dimensões.

Para o MMP 3D, qualquer que seja a versão, aparece o método correlato ao apresentado na 6.3, que é mostrado a seguir.

```
float Cubo :: DistanciaEntreCubos(Cubo A) {
//Calcula a distancia ate o bloco A (erro quadratico)
int n, m, p;
float D=0;
float Dp;
        for(n = 0; n < altura; n++) {
                for(m = 0; m < largura; m++) {
                        for(p = 0; p < profundidade; p++) {
                                Dp = pixel[n][m][p] - A.pixel[n][m][p];
                                D = D + Dp*Dp;
                        }
                }
        }
        return(D);
}
```

Figura 6.4: Método Distância entre Cubos

A diferença básica entre os métodos em 2 (duas)e 3 (três) dimensões é que neste último tem-se a diferença média quadrática entre *pixels* em um volume em vez de em um plano.

No caso do MMP 3DRD normal, para um bloco de dimensões $N \times N \times N$ ocorrerão N^3 multiplicações, N^3 subtrações e $N^3 - 1$ somas. Assim, a complexidade computacional para o bloco será dada por:

$$Cm = LN^3$$

Do mesmo modo como no caso do MMP 2DRD, o número de escalas é coincidente com o número de níveis, ou seja:

$$K = 1 + 3\log_2(N)$$

Assim, a expressão final para a complexidade computacional do MMP 3DRD é:

$$Cm^1 = LN^3(1 + 3\log_2(N)) \tag{6.5}$$

No caso específico do MMP 3DRDFT, até a profundidade $p = \log_2(N)$ serão possíveis 3^p operações com Qv de dimensões $N \times N \times N$, ou seja, $Op = 3^p Qv$.

Desta forma, ocorre a triplicação de Q_v a cada nível da árvore até a profundidade $p = \log_2(N)$, ou seja:

$$Q_v = 1 + 3 + \cdots + 3^{\log_2(N)} = \frac{3^{\log_2 N+1} - 1}{2}$$

A partir da profundidade $p = \log_2(N)$, alguns nodos gerarão 04 (quatro) descendentes, porque uma das dimensões da matriz tridimensional chegou a 1, e outros, continuam dando origem a 06 (seis). Por exemplo, para $N = 4$, ocorrerão, na profundidade 2, alguns blocos $4 \times 4 \times 1$ e $1 \times 4 \times 4$. O primeiro bloco gera 02(dois) $4 \times 2 \times 1$ e 02 (dois) $2 \times 4 \times 1$. O segundo bloco dará origem a 02 (dois) $(1 \times 4 \times 2)$ e a mais 02 (dois) $1 \times 2 \times 4$. Continuando a construção da árvore, quando 02 (duas) das dimensões da matriz tridimensional se igualam a 1, haverá a geração, a partir deste nível, de somente dois (dois) descendentes por nó.

Da mesma forma que para o MMP 2DRDFT, pode-se calcular dois limitantes para o estabelecimento do número mínimo e máximo de Q_v que aparecem na construção da árvore.

Para o limitante inferior tem-se:

$$Q_v = 1 + 3 + \cdots + 3^{\log_2 N} + 3^{\log_2 N} + \cdots + 3^{\log_2 N} = \sum_{k=0}^{\log_2 N} 3^k + 2\log_2 N 3^{\log_2 N}$$

Que leva à expressão final:

$$Q_v = \frac{3^{\log_2 N+1} - 1}{2} + 3^{\log_2 N} 2\log_2 N$$

Para o caso do limitante superior, tem-se que o total de Q_v de dimensão $N \times N \times N$ é $Q_v = 1 + 3 + 9 + 27 + \cdots + 3^{P_{max}} = \frac{3N^3-1}{2}$.

A complexidade computacional, considerando-se o limitante inferior é dada por:

$$Cm^2 = LN^3(\frac{3^{\log_2 N+1} - 1}{2} + 3^{\log_2 N} 2\log_2 N)$$

E para o limite superior:

$$Cm^2 = LN^3(\frac{3N^3 - 1}{2})$$

Para o MMP 3DRDFT acelerado, o número de escalas é dado por:

$$N_{esc} = (\log_2(N) + 1)^3$$

A complexidade computacional, com isto, é dada por:

$$Cm^3 = LN^3(\log_2(N) + 1)^3 \tag{6.6}$$

No caso da relação entre o MMP 3DRDFT e o MMP 3DRD, tem-se para o limite inferior:

$$\frac{Cm^2}{Cm^1} = \frac{\frac{3^{\log_2 N+1}-1}{2} + 3^{\log_2 N} 2\log_2 N}{(1 + 3\log_2(N))}$$

E para o limite superior:

$$\frac{Cm^2}{Cm^1} = \frac{3N^3 - 1}{2(1 + 3\log_2(N))}$$

Admitindo-se um $N = 8$, chega-se à conclusão de que $Cm^2/Cm^1 = 202/10$, para o limite inferior, e $Cm^2/Cm^1 = 1535/20$, para o limite superior.

Para a situação em que se queira a relação entre o MMP 3DRDFT acelerado e o MMP 3DRD, tem-se:

$$\frac{Cm^3}{Cm^1} = \frac{(\log_2(N) + 1)^3}{(1 + 3\log_2(N))}$$

O mesmo exemplo com $N = 8$ conduz a $Cm^3/Cm^1 = 64/10$.

6.2 Resultados Experimentais

Todas as imagens presentes nesta seção foram comprimidas com perdas com o uso dos programas MMP 2DRD, MMP 2DRDFT. Nas imagens de *Toy Vehicles* e nos conjuntos de imagens *PPI* foram agregados os resultados com o MMP 3DRD, MMP 3DRDII e MMP 3DRDFT. Além disto, alguns experimentos apresentam resultados para imagens com o JPEG 2000 e H.264.

As imagens *Aerial*, *Baboon*, *Barbara*, *Bridge*, *F16*, *Gold*, *Lena*, têm as dimensões 512×512, estão ilustradas, sem compressão, no Apêndice A e foram baixadas de *http://sipi.usc.edu/services/database.* As imagens *PP1205* e *PP1209* foram digitalizadas do IEEE *Transactions on Image Processing*, volume 9, número 7, de julho de 2000, páginas 1.205 e 1.209, respectivamente, e também possuem as dimensões de 512×512. As imagens de *Toy Vehicles* foram conseguidas de *http://sipi.usc.edu/services/database* e têm as dimensões de 512×512 cada uma, sendo a seqüência de 04 (quatro) imagens composta de 2048×512 *bytes*. Quanto às imagens *PPI*, as dimensões são de 360×400, cada uma, ou seja, a seqüência de 04 (quatro) imagens possui 1440×400 *bytes* e foram conseguidas no banco de dados do Instituto de Astronomia, Geofísica e Ciências Atmosféricas da Universidade de São Paulo.

As imagens *Aerial*, *Baboon*, *Barbara*, *Bridge*, *F16*, *Gold*, *Lena*, *PP1205* e *PP1209* foram divididas, inicialmente, em blocos de 16×16 que foram processados seqüencialmente, segundo as regras de partição consideradas para o MMP 2DRD e MMP 2DRDFT. Para as seqüências de imagens *PPI* e *Toy Vehicles* também foram utilizados os programas MMP 2DRD e MMP 2DRDFT, tendo em vista que para a aplicação em 02 dimensões basta fazer as dimensões das matrizes 2048×512, para *Toy Vehicles*, e 1440×400 para *PPI*, sendo que as dimensões dos blocos de entrada foram consideradas com 8×8 e 16×16.

Os resultados para as imagens do parágrafo anterior estão mostrados da Figura 6.5 até 6.17. Pode ser notado o melhor desempenho do MMP 2DRDFT, aproximadamente de 0.5 a 1 dB, em relação ao MMP 2DRD para todas as imagens apresentadas entre as Figuras 6.5 e 6.13, exceção feita à imagem *PP1205* em que houve uma aproximação dos resultados. Isto pode ser explicado pelo fato de que a maior parte das imagens pode ser consideradas suaves, ou seja, não há a presença de muitas componentes de alta freqüência. Quanto à imagem *PP1205*, há a presença de componentes de alta freqüência por apresentar somente texto. A imagem *PP1209* é considerada uma mesclagem por possuir texto e figuras. Para os testes feitos com *PPI* e *Toy Vehicles* notam-se melhores resultados para o MMP 2DRDFT.

As mesmas seqüências de imagens também propiciam a utilização dos códigos MMP 3DRD, MMP 3DRDII e MMP 3DRDFT, considerando-se o volume correspondente a cada seqüência consistido das dimensões de $512 \times 512 \times 4$, para *Toy Vehicles*, e $360 \times 400 \times 4$, para as 03 (três) seqüências de imagens *PPI*.

Com respeito às seqüências usadas nas variadas versões do MMP 3D, as imagens *PPI* e *Toy Vehicles* tiveram os blocos de entrada com dimensões $4 \times 4 \times 4$ e $8 \times 8 \times 4$.

Analisando-se as Figuras 6.18, 6.19, 6.20, que representam as simulações com as seqüências de imagens *PPI* com todas as versões de MMP 3D, chega-se às seguintes conclusões:

1. À medida que se aumentam as dimensões do bloco de entrada, há a melhoria do desempenho, qualquer que seja o código MMP 3D utilizado;

2. O MMP 3DRDFT superou os demais códigos em todas as simulações, sendo que o MMP 3DRDII teve melhor desempenho do que o MMP 3DRD.

De acordo com a Figura 6.21, que representa os resultados com*toy Vehicles*, verifica-se um comportamento um pouco diverso ao apresentado pelas imagens *PPI* com as seguintes características básicas:

1. O desempenho de qualquer das versões do MMP 3D melhora com o aumento das dimensões dos blocos de entrada;

2. O MMP 3DRD teve desempenho melhor do que o apresentado pelo MMP 3DRDII para blocos de entrada de dimensões $8 \times 8 \times 4$;

3. Os resultados obtidos com o MMP 3DRDFT superam os conseguidos com os outros 02 (dois) códigos.

Comparando os resultados de *Toy Vehicles* com *PPI*, chega-se à conclusão de que, ao contrário do que acontece com as imagens de radar meteorológico, em que as nuvens apresentam movimento e mudança no formato de uma varredura para outra, as imagens de *Toy Vehicles* são mais suaves e não há vantagem no uso do MMP 3DRDII em relação ao MMP 3DRD.

Após a apresentação dos resultados com os códigos em 02 (duas) e 03 dimensões separadamente, é oportuna a apresentação de gráficos que mesclam as mais variadas situações de modo a obterem-se parâmetros de comparação. Para os códigos MMP 2DRD e MMP 2DRDFT foram usados blocos de dimensões 8×8 e 16×16; para os códigos MMP 3DRD, MMP 3DRDII e MMP 3DRDFT foram usados blocos de dimensões $4 \times 4 \times 4$ e $8 \times 8 \times 4$.

A Figura 6.22 reproduz as curvas comparativas para as imagens *Toy Vehicles*. Nota-se, claramente, que com a aplicação do MMP 3DRDFT os resultados ficam superiores aos obtidos com os códigos em 02 dimensões.

Da Figura 6.23 a 6.25 mostram-se os produtos conseguidos com as simulações das 03 (três) seqüências *PPI*. Diferentemente do que aconteceu com os carrinhos, os melhores desempenhos para as imagens de radar meteorológico foram obtidos por meio dos códigos MMP 2DRD.

Por fim, apresentam-se as comparações dos códigos MMP 2D e MMP 3D com o H.264 e o JPEG 2000. Os experimentos foram realizados com as imagens *Toy Vehicles* e *PPI*. Para tal, utilizaram-se as dimensões $8 \times 8 \times 4$ e 16×16 para o MMP 3D e MMP 2D, respectivamente.

Para o JPEG 2000 foi utilizado o programa KAKADU 2.2.3, sendo que os parâmetros passados foram:

1. As dimensões da imagem a ser comprimida, ou seja, 2048×512, para *Toy Vehicles*, e 1440×400, para as *PPI*;

2. As amostras não foram sinalizadas, ou seja, os valores estão entre 0 e 255;

3. Foi estipulada a precisão em 08 (oito) *bits*;

4. As taxas em função das qualidades desejadas.

Com respeito ao H.264, o *software* utilizado foi o JM.96 e os parâmetros adotados foram os seguintes:

1. GOP (*Group of Pictures*) igual a 04 (quatro) - IPPP;

2. A otimização RD foi ligada;

3. Foi utilizada a Transformada de Hadamard;

4. A janela de busca foi configurada em 400 (quatrocentos) *pixels*;

5. Foi selecionado o controle de *buffer* com taxa variável;

6. A codificação aritmética foi a CABAC.

Na comparação dos resultados entre os programas MMP, JPEG 2000 e H264, chega-se às seguintes conclusões:

1. Para a seqüência de *Toy vehicles*, cujo gráfico aparece na Figura 6.26, o H.264 teve melhor resultado, seguido do MMP 3DRDFT, MMP 2DRDFT e JPEG 2000, nesta ordem;

2. Nas simulações com *PPI*, mostradas nas Figuras 6.27, 6.28 e 6.29, o MMP 2DRDFT teve melhor resultado. Depois veio o MMP 3DRDFT, seguido do H.264 e JPEG 2000.

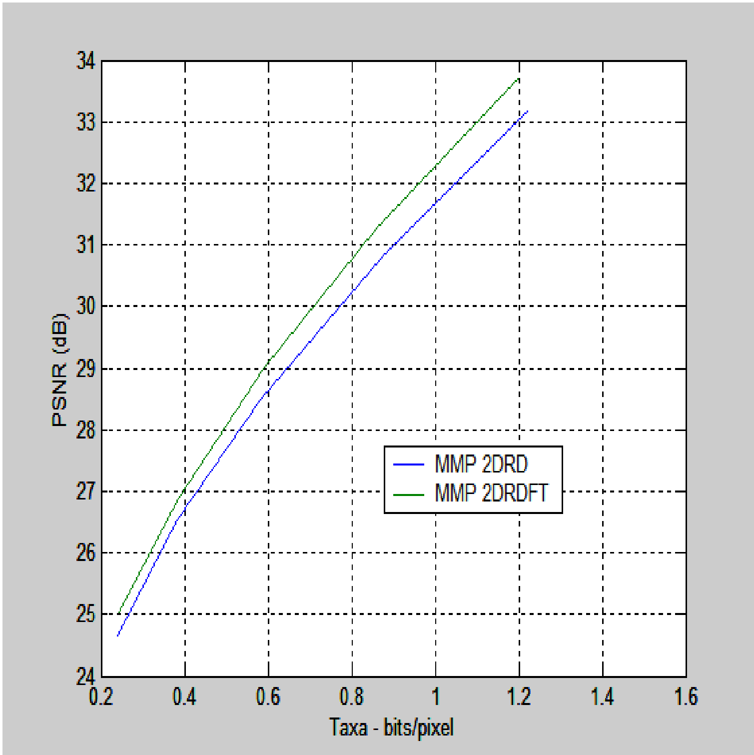


Figura 6.5: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *Aerial*

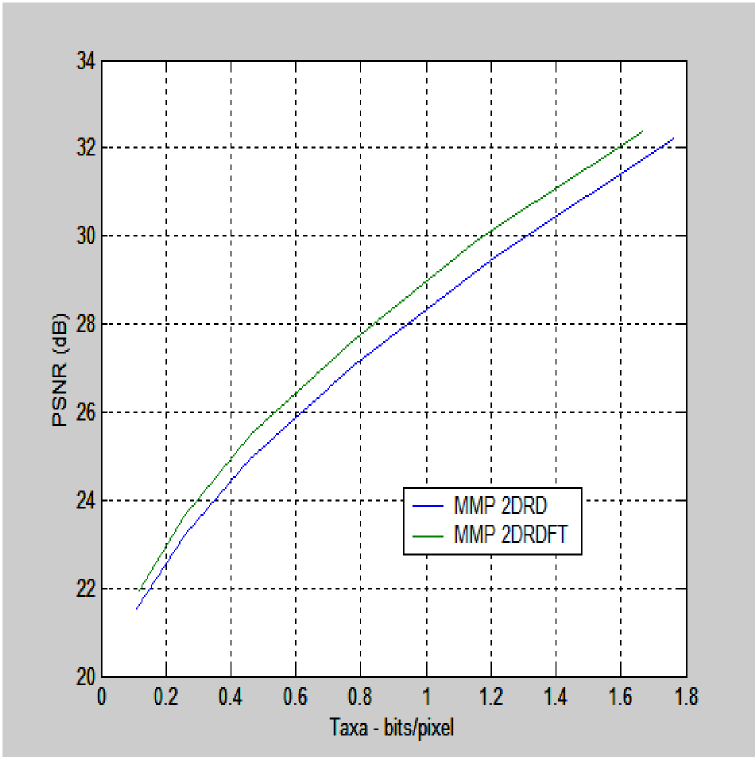


Figura 6.6: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *Baboon*

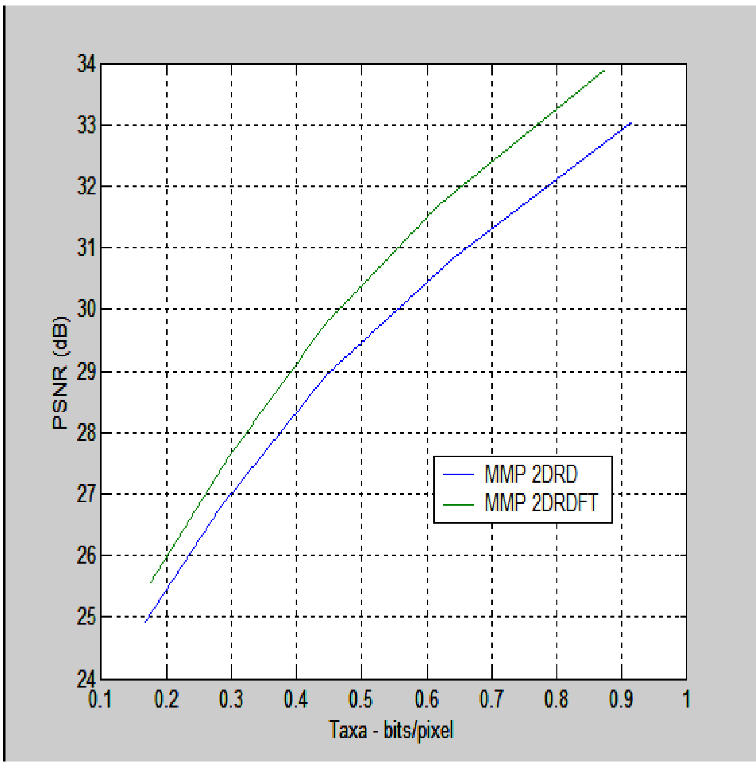


Figura 6.7: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *Barbara*

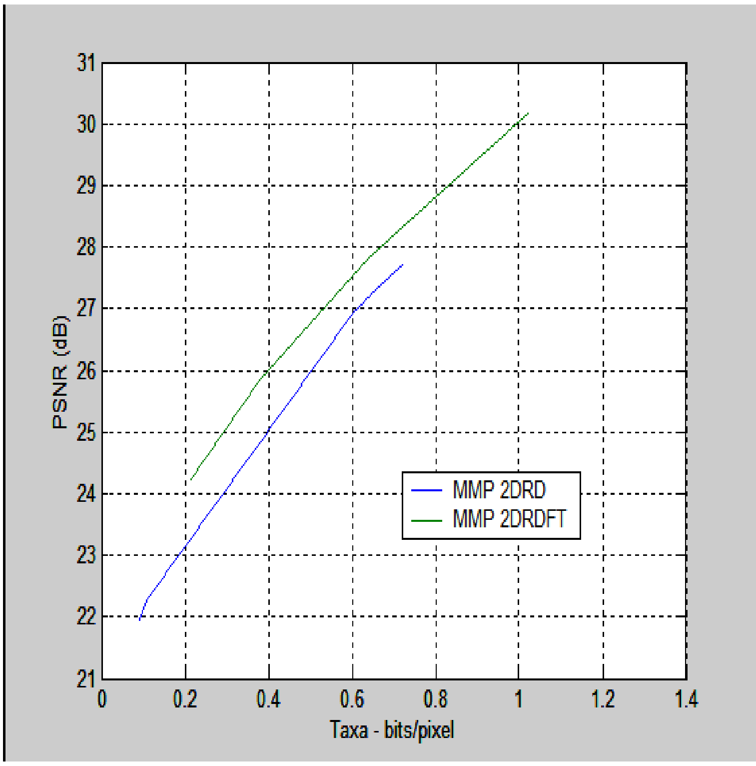


Figura 6.8: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *Bridge*

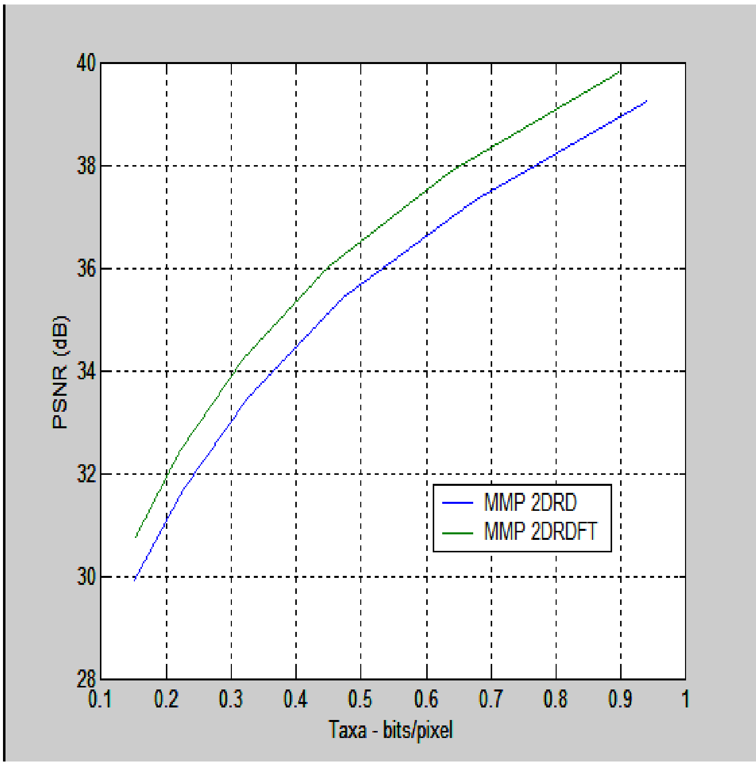


Figura 6.9: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *F16*

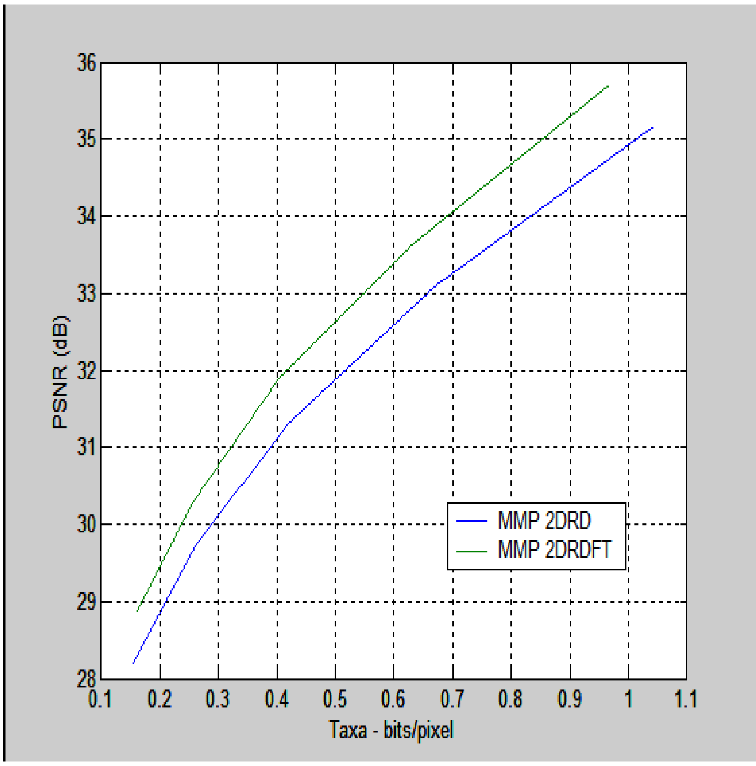


Figura 6.10: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *Gold*

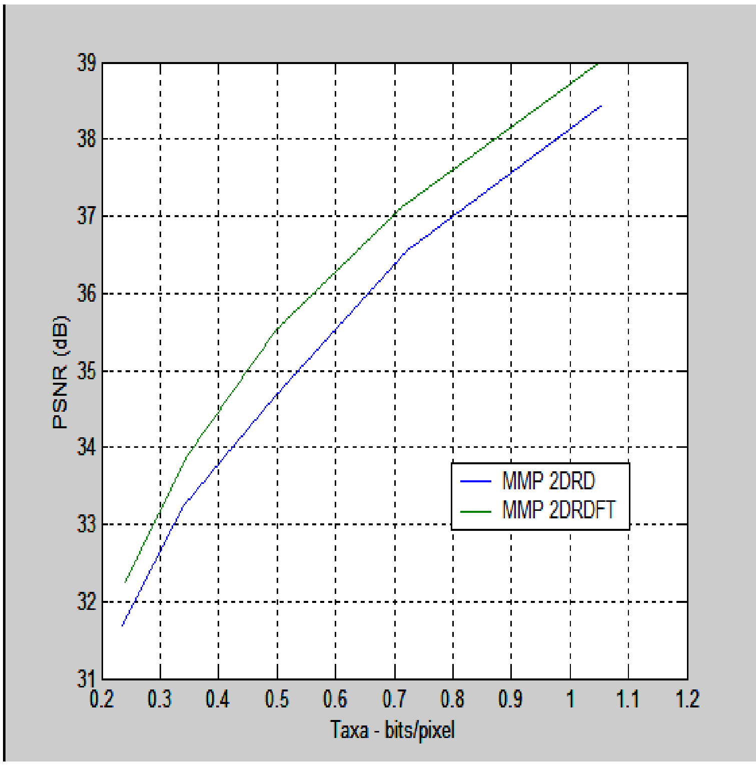


Figura 6.11: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *Lena*

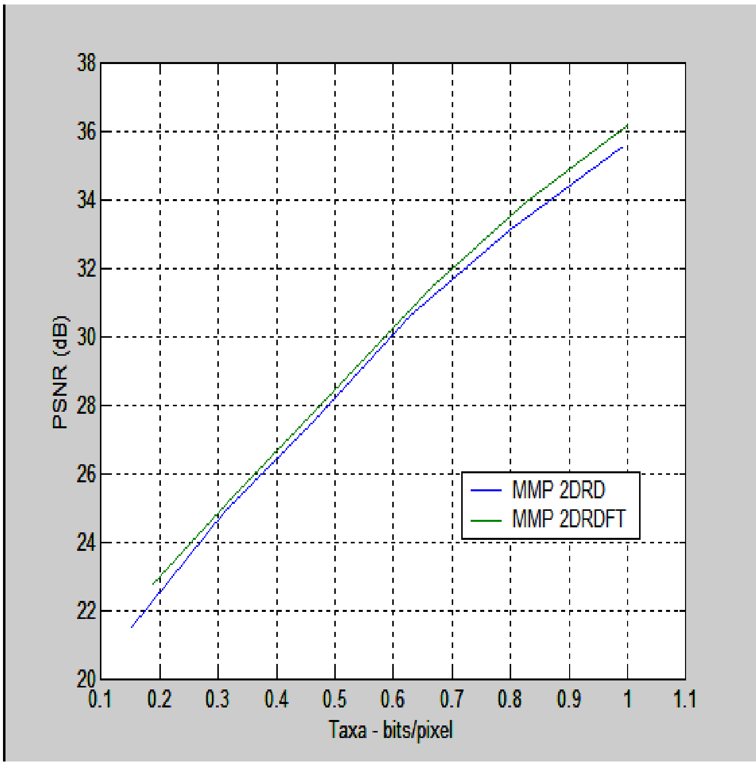


Figura 6.12: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *PP1205*

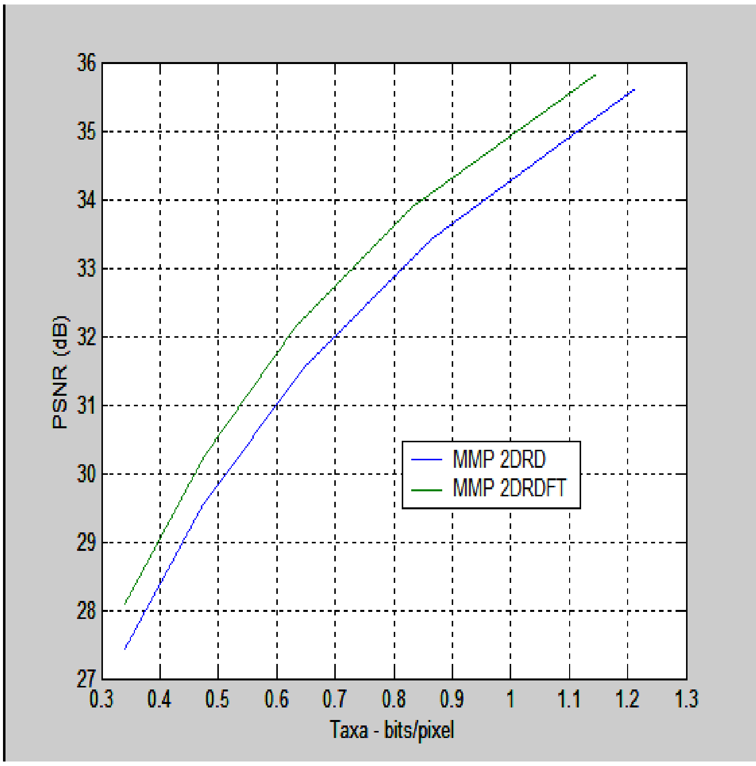


Figura 6.13: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *PP1209*

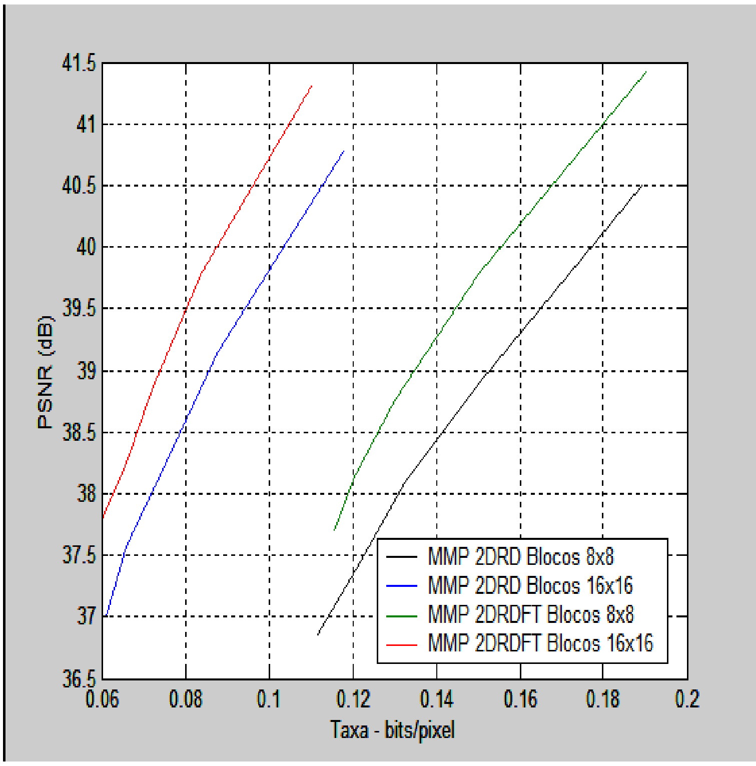


Figura 6.14: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *Toy Vehicles*

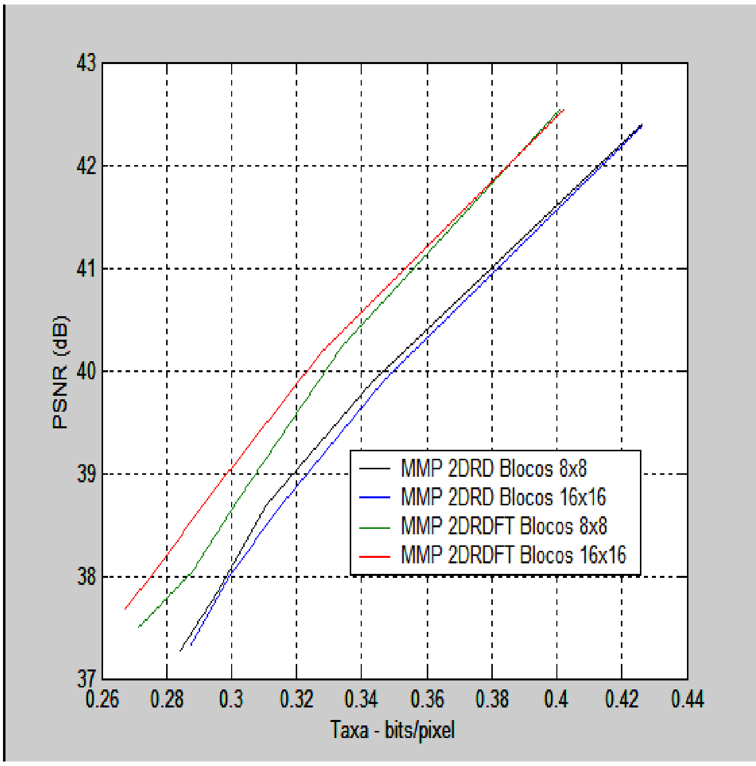


Figura 6.15: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *PPI1*

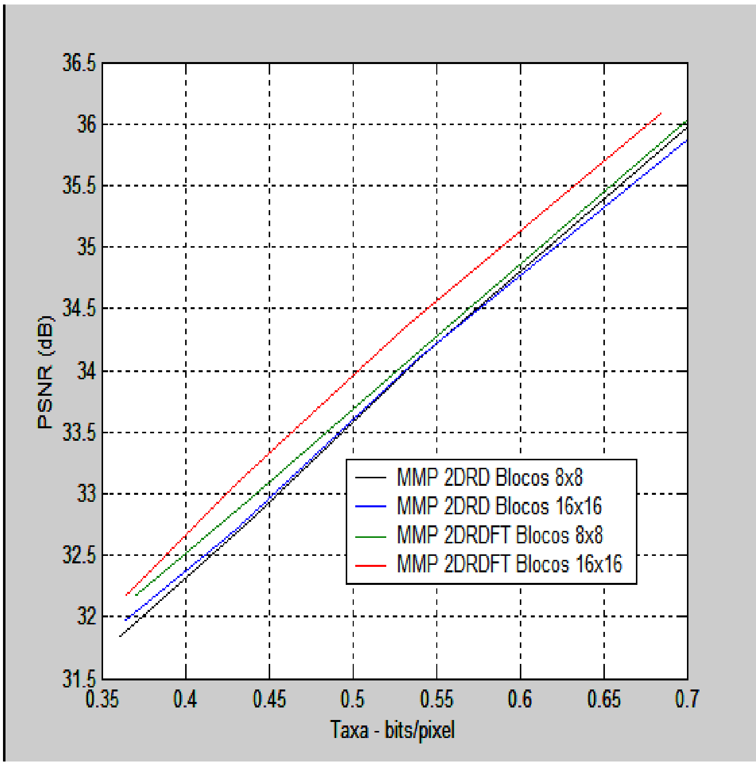


Figura 6.16: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *PPI2*

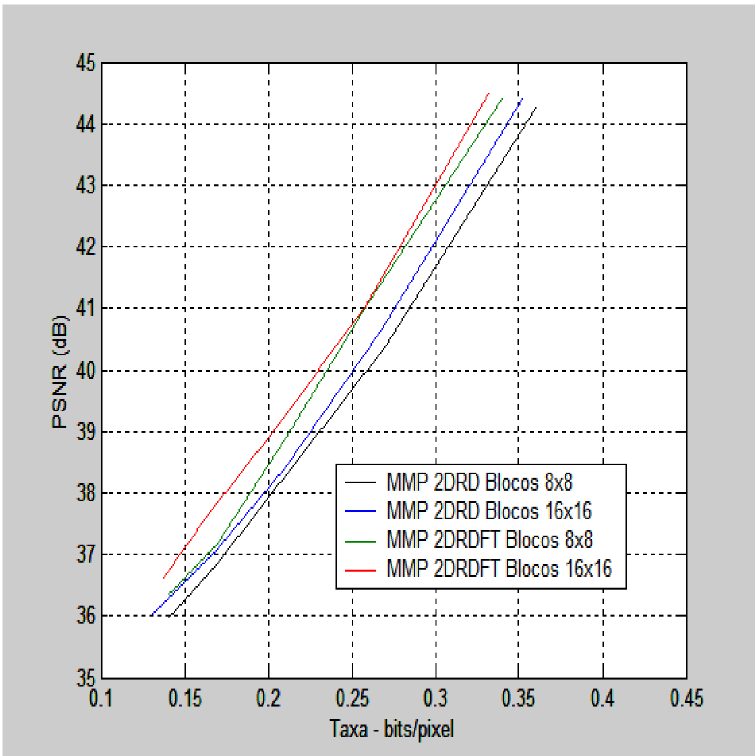


Figura 6.17: Taxa-Distorção para o MMP 2DRD e MMP 2DRDFT com *PPI3*

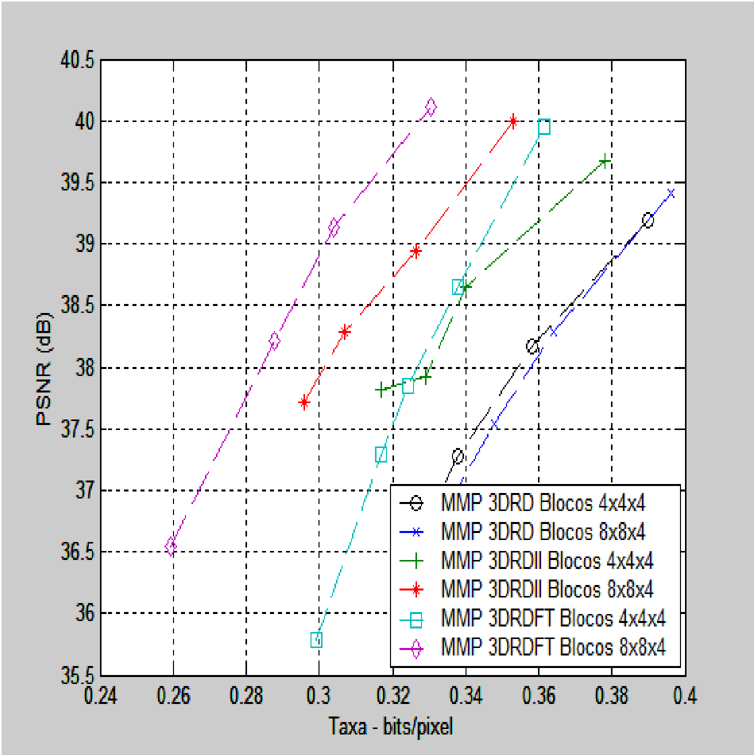


Figura 6.18: Taxa-Distorção para o MMP 3DRD, MMP 3DRDII e MMP 3DRDFT com *PPI1*

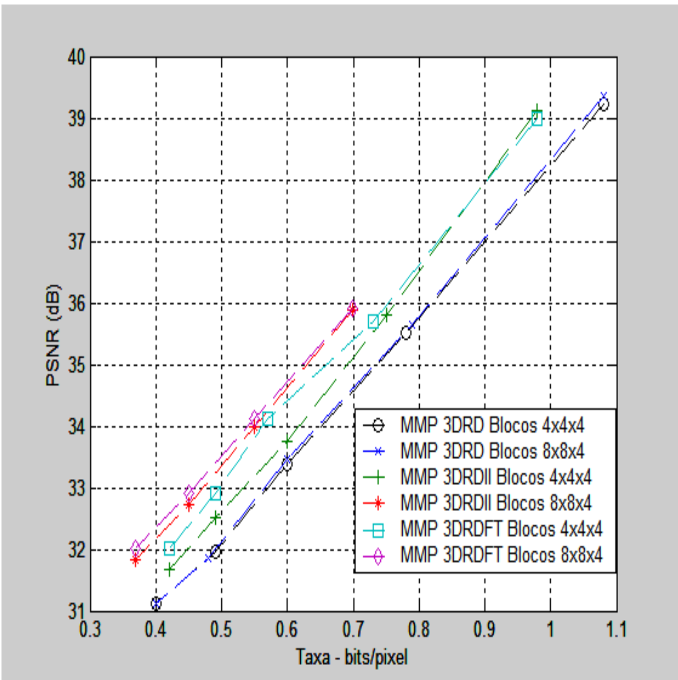


Figura 6.19: Taxa-Distorção para o MMP 3DRD, MMP 3DRDII e MMP 3DRDFT com *PPI2*

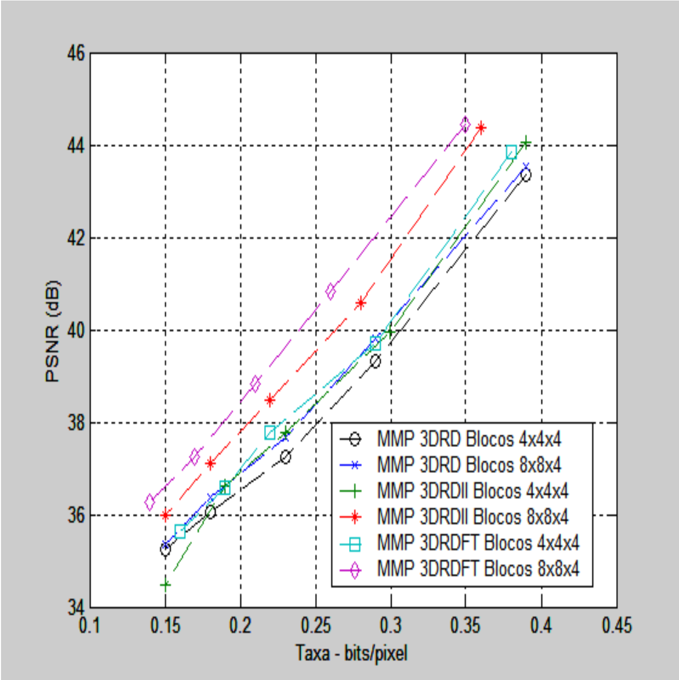


Figura 6.20: Taxa-Distorção para o MMP 3DRD, MMP 3DRDII e MMP 3DRDFT com *PPI3*

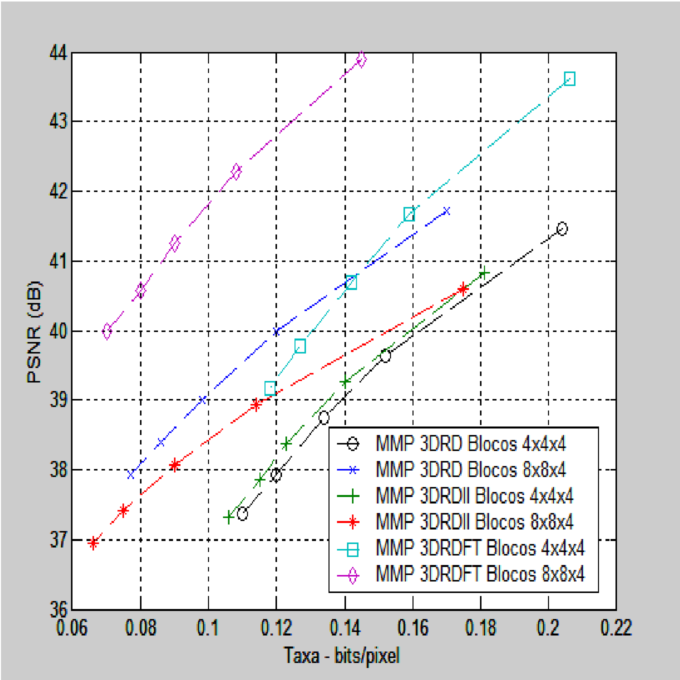


Figura 6.21: Taxa-Distorção para o MMP 3DRD, MMP 3DRDII e MMP 3DRDFT com *Toy Vehicles*

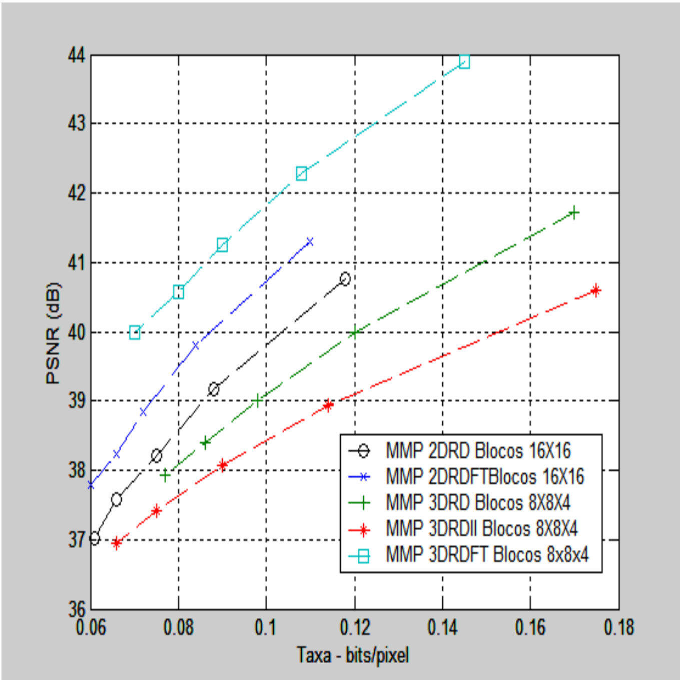


Figura 6.22: Taxa-Distorção para a Comparação dos MMP 2D e MMP 3D para *Toy Vehicles*

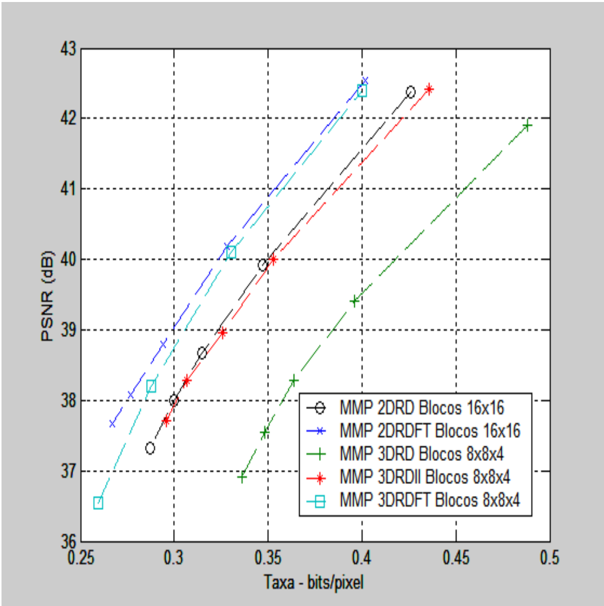


Figura 6.23: Taxa-Distorção para a Comparação dos MMP 2D e MMP 3D para *PPI1*

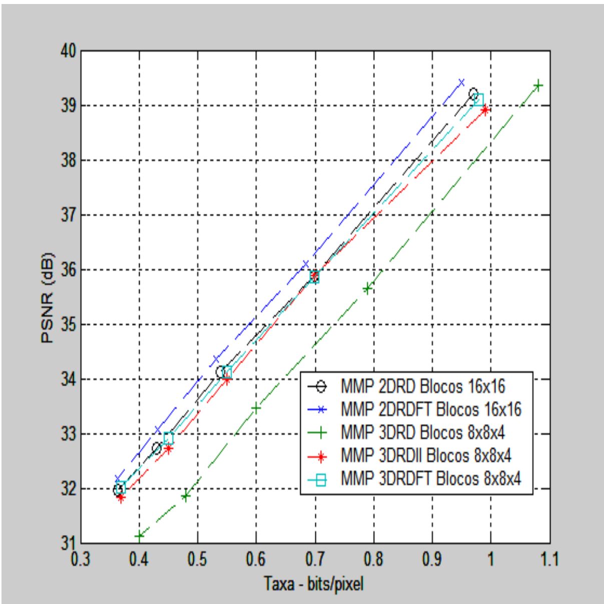


Figura 6.24: Taxa-Distorção para a Comparação dos MMP 2D e MMP 3D para *PPI2*

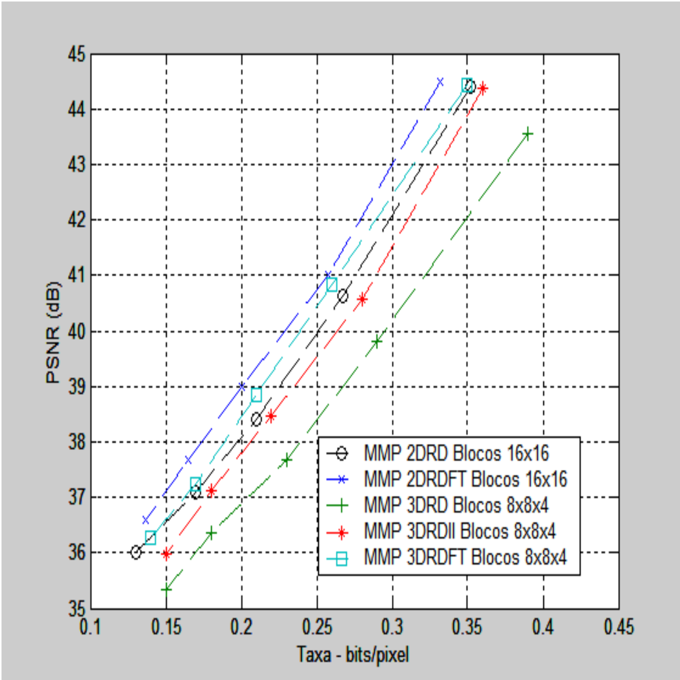


Figura 6.25: Taxa-Distorção para a Comparação dos MMP 2D e MMP 3D para *PPI3*

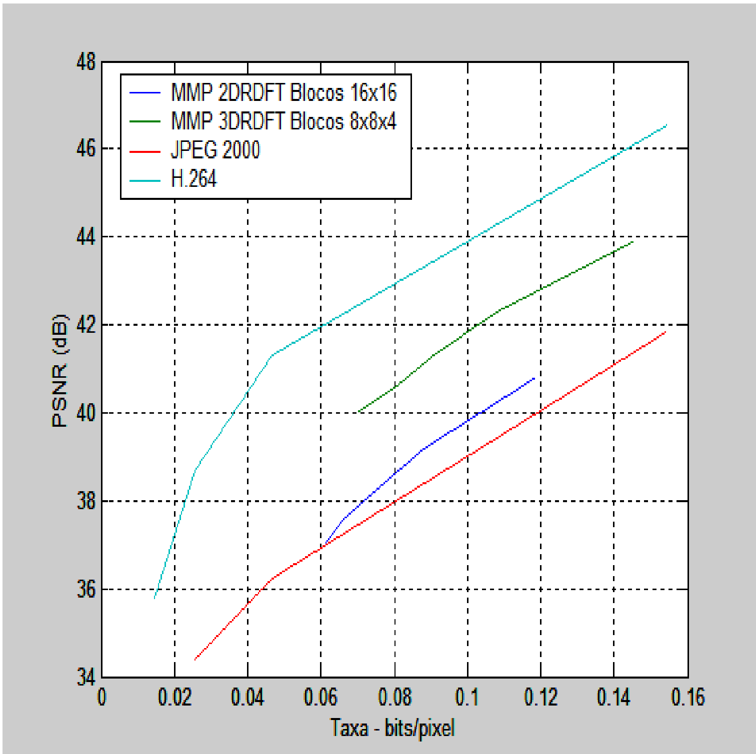


Figura 6.26: Taxa-Distorção para a Comparação do MMP. H.264 e JPEG 2000 para *Toy Vehicles*

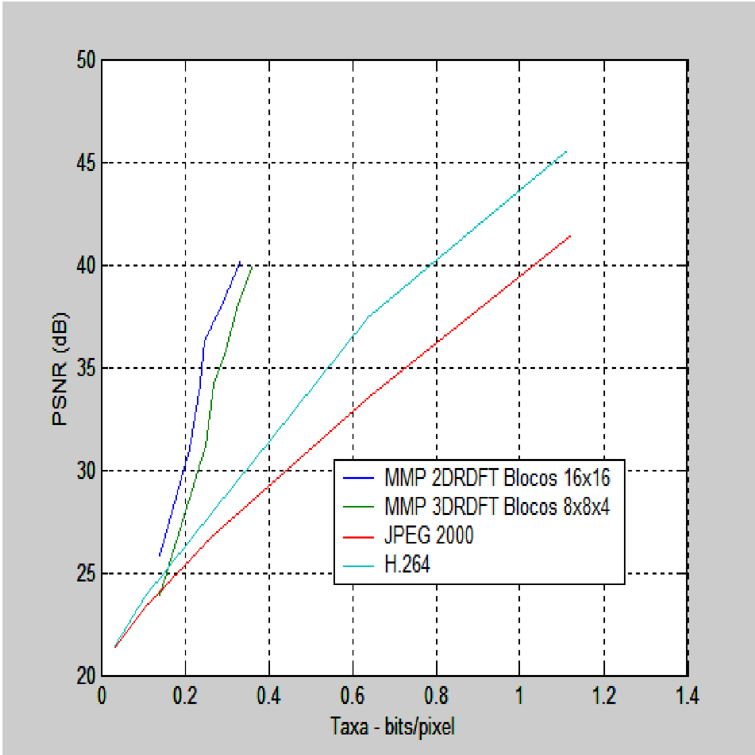


Figura 6.27: Taxa-Distorção para a Comparação do MMP. H.264 e JPEG 2000 para *PPI1*

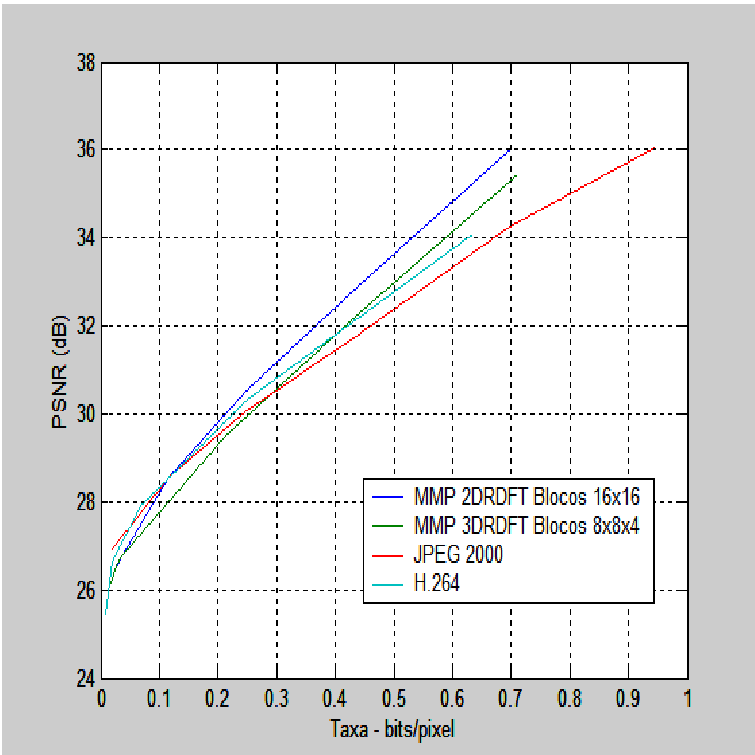


Figura 6.28: Taxa-Distorção para a Comparação do MMP. H.264 e JPEG 2000 para *PPI2*

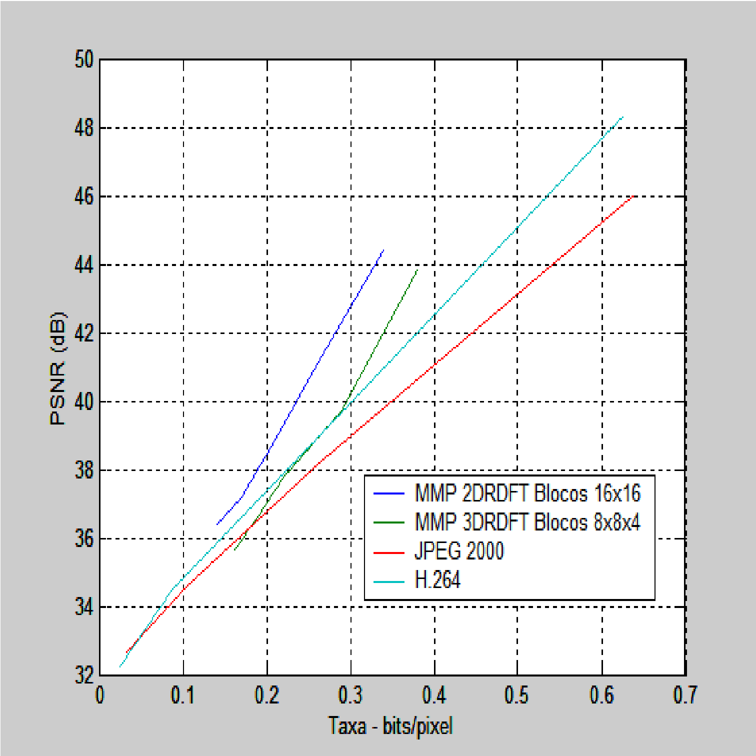


Figura 6.29: Taxa-Distorção para a Comparação do MMP. H.264 e JPEG 2000 para *PPI3*

Capítulo 7

Conclusão

A Força Aérea Brasileira é uma instituição que busca a otimização dos recursos econômicos sem se descuidar da segurança e da excelência dos serviços prestados à sociedade.

No setor de tráfego aéreo, o aspecto da segurança é de grande vulto por mexer com vidas humanas. Contudo, os recursos disponíveis são finitos para manter o complexo sistema de telecomunicações operando com qualidade. Por isso, o uso de códigos de compressão aos mais variados tipos de aplicação, incluindo-se a transmissão de sinais de radares meteorológicos, pode conduzir a economias na contratação de circuitos de telecomunicações, principalmente no tocante a redes WAN.

Assim, no desenvolvimento da dissertação, utilizou-se uma variada gama de imagens consagradas pelos profissionais que trabalham com compressão de dados, porém o foco sempre foi a possibilidade de utilização do algoritmo para as imagens de radares meteorológicos instalados em um grande número de sítios do COMAER e outras entidades civis.

O Capítulo 2 teve o objetivo de fazer uma breve introdução dos conceitos e necessidades de se comprimir dados, tendo em vista, principalmente, a demanda crescente de recursos de armazenamento e transmissão de sinais digitais.

No Capítulo 3, apareceram as principais definições sobre radares, e em especial os meteorológicos, com a finalidade de familiarizar os leitores com as particularidades de um equipamento muito importante para o homem, quer seja em momentos de crise, como as guerras, mas, também, no dia-a-dia do cidadão comum que se utiliza dos serviços meteorológicos ou de tráfego aéreo.

A partir do Capítulo 4, com a apresentação do MMP 2D (CARVALHO, 2001), e suas versões que foram sendo atingidas ao longo do tempo, o trabalho começou a tomar corpo e os primeiros resultados apareceram. Ficou patente que a utilização do MMP 2DRD sobrepujou os códigos iniciais, Com e Sem Multiescalas, com as imagens da *Lena* e algumas *PPI*.

Também no Capítulo 4, foi feita uma abordagem sobre as aplicações mais comuns das imagens *PPI* na elaboração de produtos meteorológicos. Ilustrou-se, também, como as imagens volumétricas são transformadas na projeção ortogonal em um plano para a transmissão aos centros que estudam os fenômenos, como o CPTEC (INPE), ou aqueles que são responsáveis pelas previsões meteorológicas para o uso em tráfego aéreo ou para a minoração dos problemas advindos por tempestades. No tocante à transmissão, salientou-se o uso do GZIP que teve o seu desempenho suplantado pelo MMP 2D, sendo que a utilização do MMP 2D Espaço de Códigos Segmentado proporcionou uma melhoria considerável. Verificou-se, por meio de conversas com profissionais da área de meteorologia, a preocupação em não se usar mecanismos de compressão com perdas que poderiam, segundo os especialistas, ocasionar erros e degradação na formação dos produtos meteorológicos.

O Capitulo 5 trouxe o desenvolvimento do MMP 3DRD, que é a simples utilização do conceito impresso na sigla MMP para o uso em sinais multidimensionais. Assim, as matrizes bidimensionais associadas a imagens separadas foram extrapoladas para um conceito tridimensional, sendo a terceira dimensão dada pelo carregamento de um número de imagens subseqüentes que foram processadas concomitantemente.

Nos experimentos em que foram comparados os desempenhos entre o MMP 2DRD e o MMP 3DRD ficou patente que, na maior parte das vezes, quanto maior for o bloco $\mathbf{X}_i$ de entrada melhores os desempenhos para a relação *Taxa-Distorção*, porém os recursos computacionais necessários são maiores.

Também ficou comprovado o melhor desempenho do MMP 2DRD em relação ao MMP 3DRD. Salientou-se, contudo, que quando são utilizadas seqüências de imagens concatenadas, as versões do MMP 2D podem ser consideradas de processamento tridimensional, tendo em vista que, no processamento de uma imagem da seqüência, leva-se em conta a estatística incorporada aos subdicionários por meio da codificação das imagens anteriores. Com isto, o conceito bidimensional fica restrito à segmentação dos blocos de entrada, que são matrizes bidimensionais.

Uma forma de melhorar o desempenho do MMP 3DRD foi apresentada com o nome de MMP 3DRDII. Nela, optou-se por um modo de partição que leva, em um dado momento, à condição de se ter uma matriz bidimensional com partições recursivas da dimensão da *profundidade* até a unidade, caso não haja a satisfação das condições pré-estabelecidas para a não segmentação. Os resultados alcançados pelo MMP 3DRDII superaram os conseguidos pelo MMP 3DRD nas imagens *PPI* e tenderam a se aproximar daqueles obtidos pelo MMP 2DRD. Para as imagens de *Toy Vehicles*, contudo, o desempenho do MMP 3DRD foi melhor do que o obtido pelo MMP 3DRDII.

A análise pormenorizada dos códigos MMP 2DRD, MMP 3DRD e MMP 3DRDII, bem como dos resultados associados, levaram ao desenvolvimento de um aperfeiçoamento do MMP original tanto para 02 (duas) quanto para 03 (três) dimensões, e que foi apresentado no Capítulo 6. Tais códigos foram batizados de MMP 2DRDFT e MMP 3DRDFT que conduziram a resultados que melhoram os desempenhos dos códigos anteriores, chegando ao ponto do MMP 3DRDFT superar o código MMP 2DRDFT para as imagens de *Toy Vehicles*.

Também foram feitas comparações com padrões consagrados, H.264 e JPEG 2000, e verificou-se um melhor desempenho do MMP em relação aos códigos citados quando aplicados a imagens de radar. Quando simulados com as imagens *Toy Vehicles*, o H.264 superou o MMP 2DRDFT e MMP 3DRDFT, sendo que estes tiveram melhor desempenho do que o JPEG 2000 para a mesma seqüência. Pelo exposto, os resultados foram encorajadores à continuação dos estudos, tendo em vista que há a possibilidade de serem aperfeiçoados os códigos MMP.

Com respeito aos códigos MMP 2DRDFT e MMP 3DRDFT as regras de partição mudaram sensivelmente em relação aos programas MMP apresentados nos Capítulos anteriores. Cabe o comentário de que a melhoria dos resultados foi acompanhada de um aumento considerável da complexidade computacional, o que foi abreviado pela adoção dos dois códigos MMP RDFT em 02 (duas) e 03 (três) dimensões nas suas formas *aceleradas*.

Alguns temas para trabalhos futuros incluem:

1. A implementação do MMP com espaço segmentado nas versões do MMP 2DRDFT e MMP 3DRDFT e comparar os resultados ao desempenho do GZIP para saber se

há um ganho maior em relação aos obtidos pelo MMP 2DRD Espaço de Códigos Segmentado.

2. Com o intuito de validar o uso dos códigos MMP, sugere-se que sejam feitos estudos quanto ao comportamento do código com perdas nos vários produtos do radar meteorológico e a interação com profissionais da área de meteorologia para saber se é possível a admissão de alguma distorção.

3. Tendo em vista a melhoria do uso dos MMP FT em relação aos demais, acredita-se ser viável o aprofundamento dos estudos a fim de encontrar uma forma de diminuir a complexidade computacional, tornando-o mais rápido.

4. Por fim, um outro segmento que pode ser explorado é o desenvolvimento do MMP com codificação preditiva que pode conduzir a resultados ainda melhores dos que os apresentados pelos códigos MMP 2DRDFT e MMP 3DRDFT e compará-los, por exemplo, com o JPEG 2000 e H.264.

Apêndice A

Figura 7.1: Imagem *Lena* Original, 512x512 *pixels* e 8bpp

Figura 7.2: Imagem *Barbara* Original, 512x512 *pixels* e 8bpp

Figura 7.3: Imagem *Aerial* Original, 512x512 *pixels* e 8bpp

Figura 7.4: Imagem *Bridge* Original, 512x512 *pixels* e 8bpp

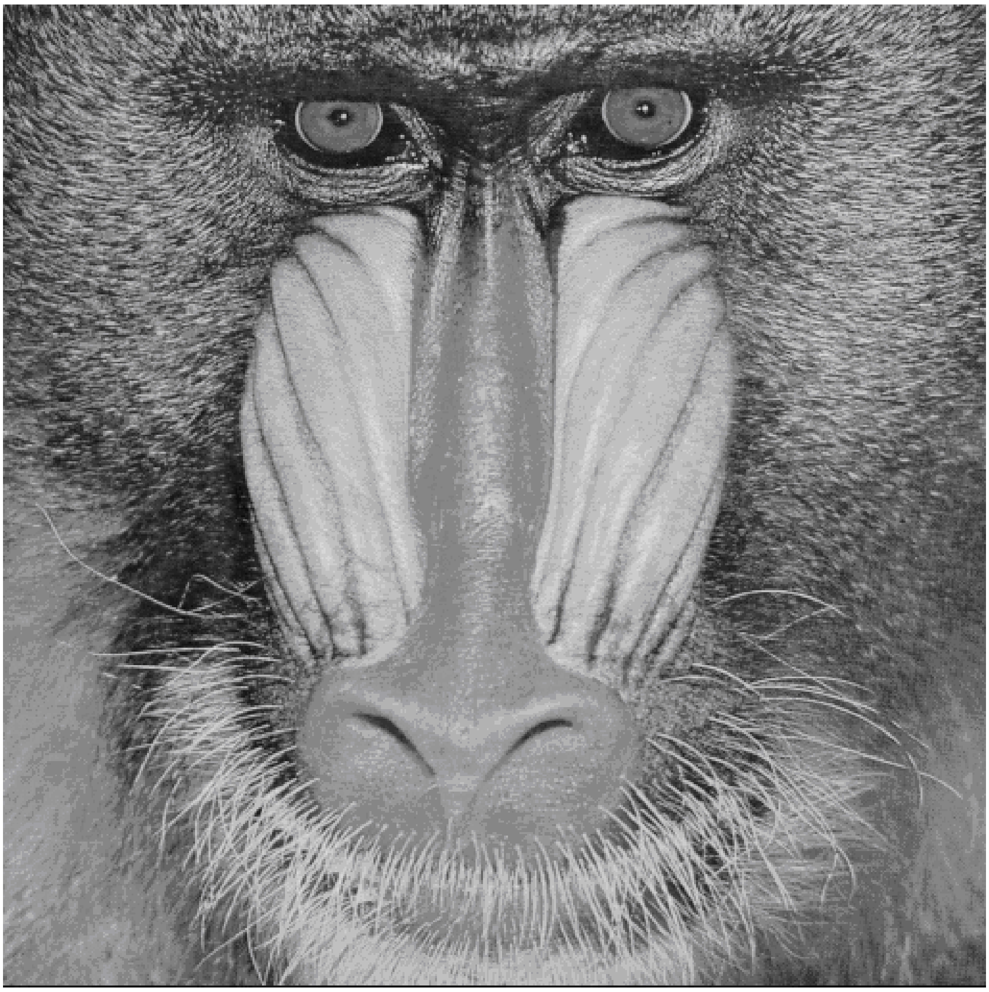
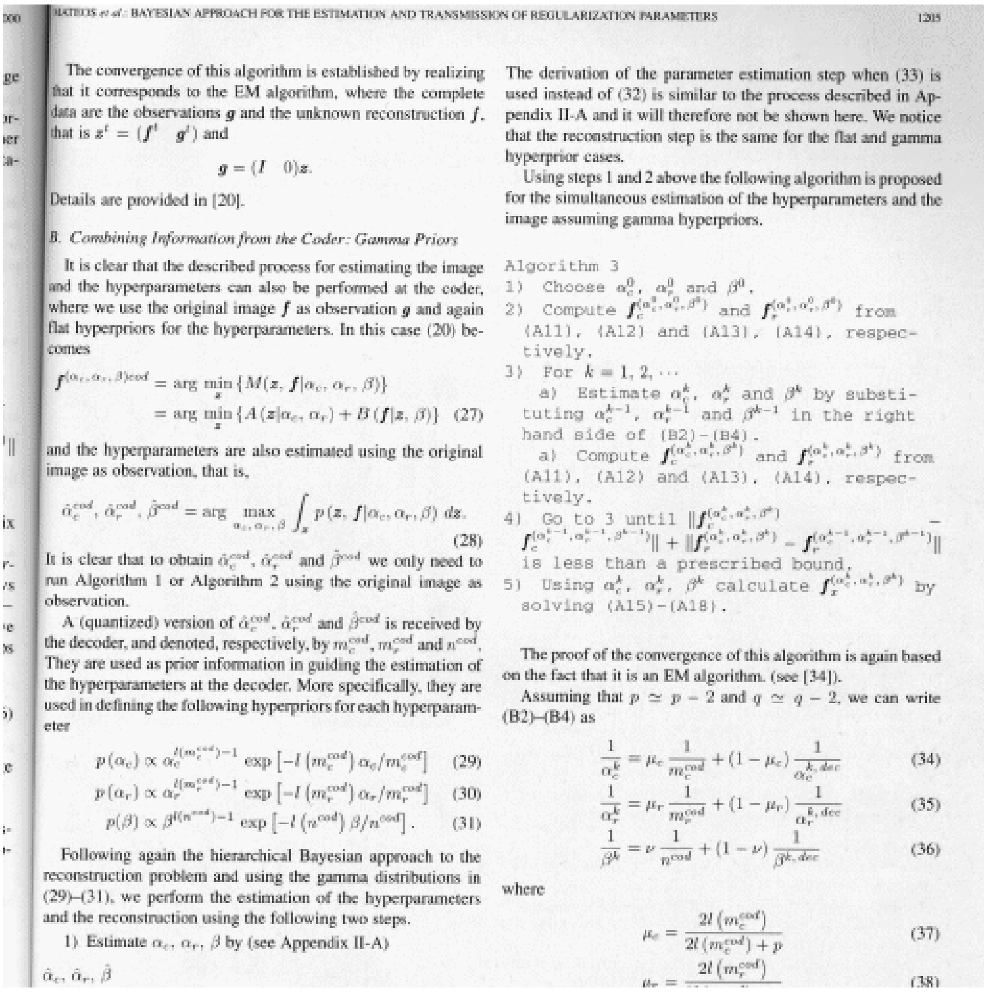


Figura 7.5: Imagem *Baboon* Original, 512x512 *pixels* e 8bpp

Figura 7.6: Imagem *Gold* Original, 512x512 *pixels* e 8bpp


MATEOS *et al.*: BAYESIAN APPROACH FOR THE ESTIMATION AND TRANSMISSION OF REGULARIZATION PARAMETERS 1205

The convergence of this algorithm is established by realizing that it corresponds to the EM algorithm, where the complete data are the observations $\boldsymbol{g}$ and the unknown reconstruction $\boldsymbol{f}$, that is $\boldsymbol{z}^t = (\boldsymbol{f}^t \quad \boldsymbol{g}^t)$ and

$$\boldsymbol{g} = (I \quad 0)\boldsymbol{z}.$$

Details are provided in [20].

B. Combining Information from the Coder: Gamma Priors

It is clear that the described process for estimating the image and the hyperparameters can also be performed at the coder, where we use the original image $\boldsymbol{f}$ as observation $\boldsymbol{g}$ and again flat hyperpriors for the hyperparameters. In this case (20) becomes

$$\begin{aligned}\boldsymbol{f}^{(\alpha_c, \alpha_r, \beta)cod} &= \arg\min_{\boldsymbol{z}} \{M(\boldsymbol{z}, \boldsymbol{f}|\alpha_c, \alpha_r, \beta)\} \\ &= \arg\min_{\boldsymbol{z}} \{A(\boldsymbol{z}|\alpha_c, \alpha_r) + B(\boldsymbol{f}|\boldsymbol{z}, \beta)\} \quad (27)\end{aligned}$$

and the hyperparameters are also estimated using the original image as observation, that is,

$$\hat{\alpha}_c^{cod}, \hat{\alpha}_r^{cod}, \hat{\beta}^{cod} = \arg \max_{\alpha_c, \alpha_r, \beta} \int_{\boldsymbol{z}} p(\boldsymbol{z}, \boldsymbol{f}|\alpha_c, \alpha_r, \beta)\, d\boldsymbol{z}. \quad (28)$$

It is clear that to obtain $\hat{\alpha}_c^{cod}$, $\hat{\alpha}_r^{cod}$ and $\hat{\beta}^{cod}$ we only need to run Algorithm 1 or Algorithm 2 using the original image as observation.

A (quantized) version of $\hat{\alpha}_c^{cod}$, $\hat{\alpha}_r^{cod}$ and $\hat{\beta}^{cod}$ is received by the decoder, and denoted, respectively, by m_c^{cod}, m_r^{cod} and n^{cod}. They are used as prior information in guiding the estimation of the hyperparameters at the decoder. More specifically, they are used in defining the following hyperpriors for each hyperparameter

$$p(\alpha_c) \propto \alpha_c^{l(m_c^{cod})-1} \exp\left[-l\left(m_c^{cod}\right)\alpha_c/m_c^{cod}\right] \quad (29)$$

$$p(\alpha_r) \propto \alpha_r^{l(m_r^{cod})-1} \exp\left[-l\left(m_r^{cod}\right)\alpha_r/m_r^{cod}\right] \quad (30)$$

$$p(\beta) \propto \beta^{l(n^{cod})-1} \exp\left[-l\left(n^{cod}\right)\beta/n^{cod}\right]. \quad (31)$$

Following again the hierarchical Bayesian approach to the reconstruction problem and using the gamma distributions in (29)–(31), we perform the estimation of the hyperparameters and the reconstruction using the following two steps.

1) Estimate α_c, α_r, β by (see Appendix II-A)

$\hat{\alpha}_c, \hat{\alpha}_r, \hat{\beta}$

The derivation of the parameter estimation step when (33) is used instead of (32) is similar to the process described in Appendix II-A and it will therefore not be shown here. We notice that the reconstruction step is the same for the flat and gamma hyperprior cases.

Using steps 1 and 2 above the following algorithm is proposed for the simultaneous estimation of the hyperparameters and the image assuming gamma hyperpriors.

Algorithm 3
1) Choose α_c^0, α_r^0 and β^0.
2) Compute $\boldsymbol{f}_c^{(\alpha_c^0, \alpha_r^0, \beta^0)}$ and $\boldsymbol{f}_r^{(\alpha_c^0, \alpha_r^0, \beta^0)}$ from (A11), (A12) and (A13), (A14), respectively.
3) For $k = 1, 2, \cdots$
a) Estimate α_c^k, α_r^k and β^k by substituting α_c^{k-1}, α_r^{k-1} and β^{k-1} in the right hand side of (B2)–(B4).
a) Compute $\boldsymbol{f}_c^{(\alpha_c^k, \alpha_r^k, \beta^k)}$ and $\boldsymbol{f}_r^{(\alpha_c^k, \alpha_r^k, \beta^k)}$ from (A11), (A12) and (A13), (A14), respectively.
4) Go to 3 until $\|\boldsymbol{f}_c^{(\alpha_c^k, \alpha_r^k, \beta^k)} - \boldsymbol{f}_c^{(\alpha_c^{k-1}, \alpha_r^{k-1}, \beta^{k-1})}\| + \|\boldsymbol{f}_r^{(\alpha_c^k, \alpha_r^k, \beta^k)} - \boldsymbol{f}_r^{(\alpha_c^{k-1}, \alpha_r^{k-1}, \beta^{k-1})}\|$ is less than a prescribed bound.
5) Using α_c^k, α_r^k, β^k calculate $\boldsymbol{f}_x^{(\alpha_c^k, \alpha_r^k, \beta^k)}$ by solving (A15)–(A18).

The proof of the convergence of this algorithm is again based on the fact that it is an EM algorithm. (see [34]).

Assuming that $p \simeq p-2$ and $q \simeq q-2$, we can write (B2)–(B4) as

$$\frac{1}{\alpha_c^k} = \mu_c \frac{1}{m_c^{cod}} + (1-\mu_c)\frac{1}{\alpha_c^{k,dec}} \quad (34)$$

$$\frac{1}{\alpha_r^k} = \mu_r \frac{1}{m_r^{cod}} + (1-\mu_r)\frac{1}{\alpha_r^{k,dec}} \quad (35)$$

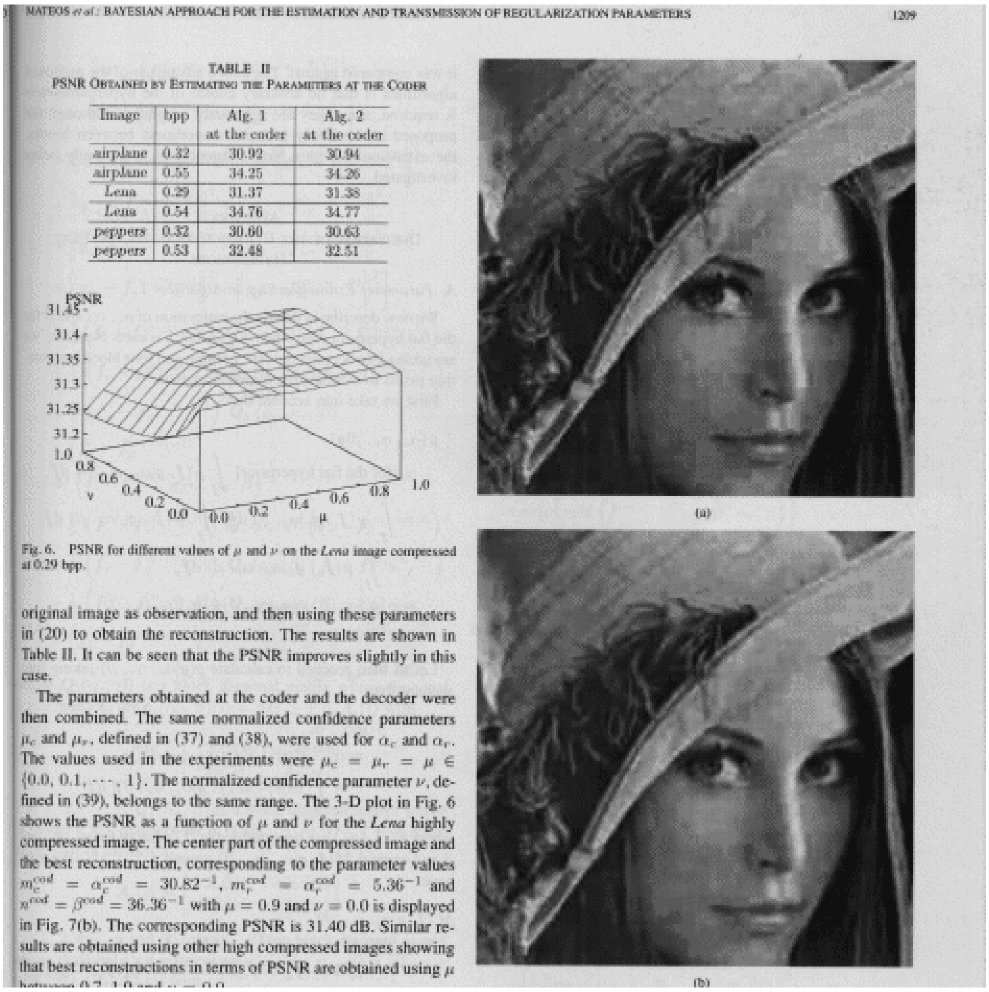
$$\frac{1}{\beta^k} = \nu \frac{1}{n^{cod}} + (1-\nu)\frac{1}{\beta^{k,dec}} \quad (36)$$

where

$$\mu_c = \frac{2l\left(m_c^{cod}\right)}{2l\left(m_c^{cod}\right)+p} \quad (37)$$

$$\mu_r = \frac{2l\left(m_r^{cod}\right)}{\ldots} \quad (38)$$

Figura 7.7: Imagem *pp1205* Original, 512x512 *pixels* e 8bpp



MATEOS *et al.*: BAYESIAN APPROACH FOR THE ESTIMATION AND TRANSMISSION OF REGULARIZATION PARAMETERS 1209

TABLE II
PSNR OBTAINED BY ESTIMATING THE PARAMETERS AT THE CODER

Image	bpp	Alg. 1 at the coder	Alg. 2 at the coder
airplane	0.32	30.92	30.94
airplane	0.55	34.25	34.26
Lena	0.29	31.37	31.38
Lena	0.54	34.76	34.77
peppers	0.32	30.60	30.63
peppers	0.53	32.48	32.51

Fig. 6. PSNR for different values of μ and ν on the *Lena* image compressed at 0.29 bpp.

original image as observation, and then using these parameters in (20) to obtain the reconstruction. The results are shown in Table II. It can be seen that the PSNR improves slightly in this case.

The parameters obtained at the coder and the decoder were then combined. The same normalized confidence parameters μ_c and μ_r, defined in (37) and (38), were used for α_c and α_r. The values used in the experiments were $\mu_c = \mu_r = \mu \in \{0.0, 0.1, \cdots, 1\}$. The normalized confidence parameter ν, defined in (39), belongs to the same range. The 3-D plot in Fig. 6 shows the PSNR as a function of μ and ν for the *Lena* highly compressed image. The center part of the compressed image and the best reconstruction, corresponding to the parameter values $m_c^{cod} = \alpha_c^{cod} = 30.82^{-1}$, $m_r^{cod} = \alpha_r^{cod} = 5.36^{-1}$ and $n^{cod} = \beta^{cod} = 36.36^{-1}$ with $\mu = 0.9$ and $\nu = 0.0$ is displayed in Fig. 7(b). The corresponding PSNR is 31.40 dB. Similar results are obtained using other high compressed images showing that best reconstructions in terms of PSNR are obtained using μ between 0.7–1.0 and $\nu = 0.0$.

(a)

(b)

Figura 7.8: Imagem *pp1209* Original, 512x512 *pixels* e 8bpp

Figura 7.9: Imagem *F16* Original, 512x512 *pixels* e 8bpp

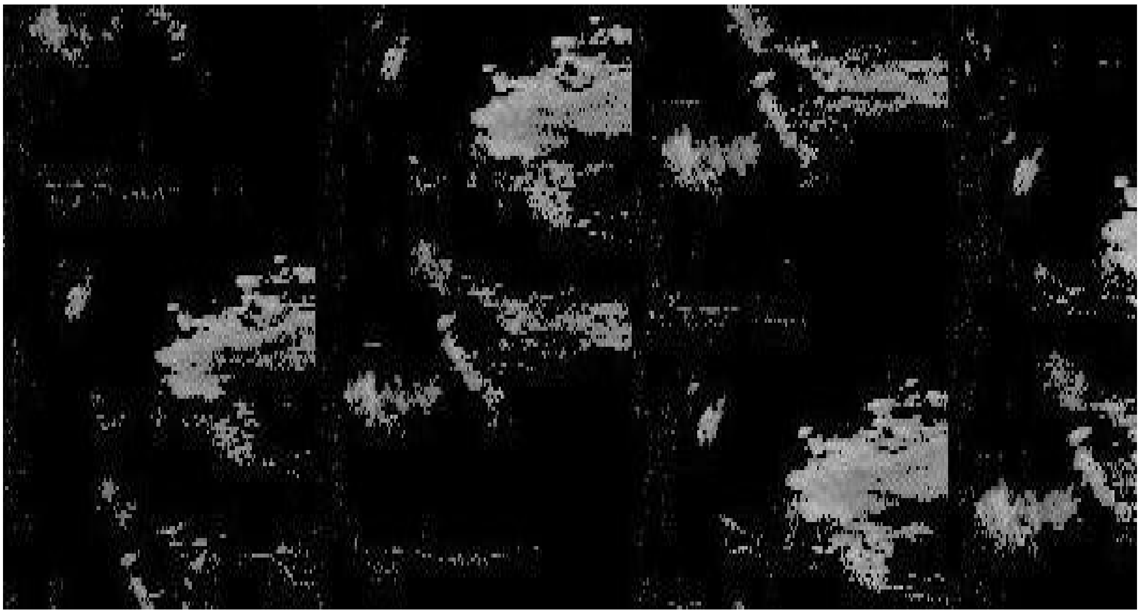


Figura 7.10: Imagem *PPI* Original, 1440x400 *pixels* e 8bpp

Figura 7.11: Imagem *Toy Vehicles* Original, 2048x512 *pixels* e 8bpp

Referências Bibliográficas

[1] BLAHUT, R.E., *Principles and Practice of Information Theory*, Edson-Wesley Publishing Company, 1988.

[2] BRASIL.Comando da Aeronáutica. Departamento de Controle do Espaço Aéreo. *Apostila CFOE QOEMET. Fundamentos de Radiocomunicação e Detecção* CIAAR, 2001.

[3] BRASIL.Comando da Aeronáutica. Departamento de Controle do Espaço Aéreo. *Apostila. Operação do Posto de Visualização Remota* ICEA, 2007.

[4] BRASIL.Comando da Aeronáutica. Departamento de Controle do Espaço Aéreo. *Apostila. Teoria Radar* ICEA, 2005.

[5] CARVALHO, Murilo B.,*Compressão de Sinais Multidimensionais Usando Recorrência de Padrões Multiescalas*,tese de doutorado, Programa de Engenharia Elétrica, COPPE/UFRJ, Rio de Janeiro, RJ, Março de 2001.

[6] CARVALHO, Murilo B., DA SILVA Eduardo A.B. and FINAMORE Weiler A., *Multidimensional Signal Processing Using Recurrent Patterns*, Signal Processing: Image and Video Coding beyond Standards, Elsevier Science,2002.

[7] DUARTE, M. H. V., CARVALHO, M. B., DA SILVA, E. A. B., PAGLIARI, C. L., MENDONÇA, G. V., *Multiscale Recurrent Patterns Applied to Stereo Image Coding*, IEEE Transactions on Circuits and Systems for Video Technology, 15(11):1434-1447, November 2005.

[8] FILHO, E.B.L, CARVALHO, M.B. and DA SILVA E.A.B. and FINAMORE Weiler A., *Multidimensional Signal Processing Using Recurrent Patterns with Smooth Side*

Match Criterion, IEEE Internacional Conference on Image Processing, Singapore, October, 2004.

[9] FILHO, E.B.L, DA SILVA E.A.B., CARVALHO, M.B., JUNIOR W.S.S., KOILLER, J., *Eletrocardiographic Signal Compression using Multiscale Recurrent Patterns*, IEE transactions on Circuits and Systems- 1: Regular Papers, Vol., 52 NO. 12, December 2005.

[10] FILHO, E.B.L., CARVALHO M.B., DA SILVA E.A.B., PINAGÉT, F.S., *Universal Image Compression using Multiscale Recurrent Patterns with Adaptative Probability Model*, IEEE Transactions on Image Processing, Vol. 17, P. 512-527, April 2008.

[11] GNU Operating System, *gzip (GNU zip) compression utility*, disponpivel em <http://www.gnu.org/software/gzip/manual/gzip.pdf>. Acesso em: 25 mar. 2008.

[12] GOMES, Geraldo G.,*Introdução à Teoria da Informação e da Codificação*,INATEL, 2001.

[13] GONZALEZ, Rafael C. and WOODS, Richard E.,*Digital Image Processing*,Prentice Hall, 2a edição, 2002.

[14] HALSALL, Fred.,*Multimedia Communications*,Addison-Wesley, 2001.

[15] (Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)), *Draft of Version 4 of (H.264/AVC)(ITU-T Recommendation (H.264) and ISO/IEC 14496-10 ((MPEG)-4 part 10) Advanced Video Coding)*,2005.

[16] MAKKAPATI, Vishnu V., *Extreme Compression of Wheather Radar Data*, IEEE Transactions on Geoscience and Remote Sensing, Vol. 45, NO. 11, November, 2007.

[17] World Wide Web Consortium, *Portable Network Graphics (PNG) Specification (Second Edition)*, disponpivel em <http://www.w3.org/TR/2003/REC-PNG-20031110/>. Acesso em: 20 mar. 2008.

[18] RODRIGUES, N.M.M., DA SILVA E.A.B., CARVALHO M.B., FARIA S.M.M., DA SILVA V.M.M., *Universal Image Coding using Multisacale Recurrent Patterns and*

Prediction, IEEE International Conference on Image Processing, ICIP, 2005., Vol 2, P. 11-14, September 2005.

[19] Taubman, D.S. and Marcelin, M.W., *JPEG2000: Image Compression Fundamentals and Practice*, Kluwer Academic Publishers, 2001.