

Codificação de Imagem e Vídeo Baseada em Casamento Parcial de Padrões

Marcos Ricardo A. Morais, Francisco Madeiro e Elmar Uwe K. Melcher

Resumo—Este artigo apresenta uma técnica de codificação sem perda de coeficientes quantizados, oriundos de transformadas de bloco aplicadas à compressão de imagem e vídeo. A técnica de casamento parcial de padrões (CPP) é aplicada como pré-processamento da etapa de codificação entrópica de codificadores de imagem e vídeo baseados em transformada. Para imagem, utilizamos blocos já codificados como dicionário de busca. No caso de vídeo, analisamos a compressão nas imagens de resíduo obtidas pelos processos de estimação e compensação de movimento: as imagens anteriores de resíduo são utilizadas na formação do dicionário. O algoritmo de casamento parcial de padrões baseia-se no fato de haver correlação residual entre os mapas de significância dos blocos de coeficientes transformados. A técnica proposta neste trabalho é avaliada por meio de simulações realizadas com um codificador MPEG-4, modificado para “acomodar” o método CPP.

Palavras-Chave— Codificação de imagem, codificação de vídeo, casamento parcial de padrões.

Abstract—This paper presents a technique for lossless coding of quantized coefficients, obtained from block transforms applied to image and video compression. The partial pattern matching technique is applied as a preprocessing stage in entropy coding for transform based image and video coders. For image coding we use the already coded blocks as a dictionary. For video we analyze the compression of residue images after the motion estimation and compensation steps, using the previous residue transformed elements in the dictionary formation. The partial pattern matching algorithm is based on the fact that there is residual correlation among the blocks of transformed coefficients. The technique proposed in the present work is evaluated by simulations performed by using the MPEG-4 coder, modified to include CPP.

Keywords—Image coding, video coding, partial pattern matching.

I. INTRODUÇÃO

A técnica de casamento parcial de padrões (CPP) visa identificar seqüências de símbolos que possuam algum grau de similaridade. Na codificação usando essa técnica, substitui-se a seqüência atual por aquela encontrada como similar.

O casamento imperfeito, ou parcial, de padrões pode ser aplicado à codificação sem perda se um novo passo de codificação, normalmente denominado de passo de refinamento, for aplicado ao resíduo obtido. O resíduo é a diferença entre os símbolos reais e os símbolos codificados.

O casamento parcial de padrões tem sido utilizado recentemente como a base para diversos algoritmos de compressão. O trabalho de Bookstein e Klein [1] é direcionado à compressão

de vetores (mapas) de bits correlacionados, tais como imagens binárias e imagens textuais¹.

Um exemplo claro da utilização de CPP é o padrão de compressão de imagens binárias JBIG2 [2]–[5], baseado no formato DjVu da AT&T [6]. Esse sistema de codificação atinge uma taxa de compressão significativamente maior com relação ao padrão anterior JBIG [7], graças ao uso de casamento parcial de padrões.

Outros trabalhos de codificação com perda ou sem perda de imagens textuais [8]–[11] também são baseados no uso de elementos de referência, que formam um dicionário ou biblioteca de símbolos. Os elementos da imagem são codificados por meio da substituição dos símbolos. No caso sem perda, o resíduo, que é a diferença entre a imagem original e a formada pelos símbolos, também deve ser codificado.

O caso clássico da utilização de casamento de padrões na codificação entrópica é o codificador Ziv-Lempel [12]. Nesse codificador as seqüências são casadas apenas se forem idênticas. O Ziv-Lempel faz parte da classe de codificadores entrópicos de dicionário, ou seja, que realiza busca por cadeias de símbolos. A cada cadeia de símbolos encontrada é associado um novo símbolo. O conjunto de novos símbolos e suas cadeias associadas formam um dicionário de busca. O algoritmo de busca guloso (*greedy*), que tenta casar a maior sub-cadeia possível, é o mais utilizado na classe de codificadores de dicionário.

Há também o método Lempel-Ziv com perda [13], [14], em que o casamento pode ser imperfeito. A aplicação do Lempel-Ziv com perda diretamente a conjuntos de pixels de uma imagem e entre os quadros de uma seqüência de vídeo tem sido avaliada [15], [16].

II. CASAMENTO DE PADRÕES NA CODIFICAÇÃO DE TRANSFORMADAS DE BLOCO

De uma forma geral, o contexto da codificação de sinais de imagem e vídeo, utilizando a técnica de casamento parcial de padrões, é muito vasto e deve ser melhor delimitado. Trabalharemos sob um modelo baseado em transformadas de bloco. Devido à existência de técnicas já bem desenvolvidas de otimização da quantização e da seleção dos modos de codificação [17]–[19], nos atemos à codificação entrópica dos coeficientes já quantizados. No caso de vídeo, podemos encontrar contribuições às técnicas de estimação e codificação de movimento [20], [21], como também de escolha dos diversos modos possíveis de codificação dos blocos [22], [23].

Foi apresentada em [24] uma análise do número de bits utilizados na codificação do mapa de significância, sinal e magnitude dos coeficientes de cada bloco transformado e

¹Denotamos uma imagem textual como uma imagem binária formada principalmente por texto impresso.

Marcos Ricardo A. Morais, Universidade Federal de Campina Grande, Campina Grande, PB, Brasil, e-mail: morais@dee.ufcg.edu.br. Francisco Madeiro, Escola Politécnica de Pernambuco – Universidade de Pernambuco, Recife, PE, Brasil, e-mail: franciscomadeiro@yahoo.com.br. Elmar Uwe K. Melcher, Universidade Federal de Campina Grande, Campina Grande, PB, Brasil, e-mail: elmar@dsc.ufcg.edu.br. Francisco Madeiro e Elmar Melcher também estão com o Instituto de Estudos Avançados em Comunicações (Iecom), Campina Grande, PB, Brasil.

quantizado de uma imagem. Observou-se que uma grande proporção dos bits é utilizada para codificar o mapa de significância. Trabalhos que tratam da codificação usando mapa de significância [25]–[30] mostram que o sinal e a magnitude são de difícil compressão e os resultados obtidos possuem ganho apenas marginal. O mapa de significância é uma imagem binária, de fácil tratamento.

Como podemos observar na Fig. 1, os elementos significativos estão na sua maioria agrupados no canto superior esquerdo de cada bloco. Este fato pode ser interpretado como uma estrutura intra-bloco, em que a significância dos coeficientes vizinhos é correlacionada, ou inter-bloco, em que a correlação ocorre entre coeficientes relativos à mesma sub-banda. Estas correlações podem ser aproveitadas de forma direta por meio da seleção dos contextos utilizados. A reutilização e a adaptação dos contextos entre blocos permite o aproveitamento indireto destas correlações.

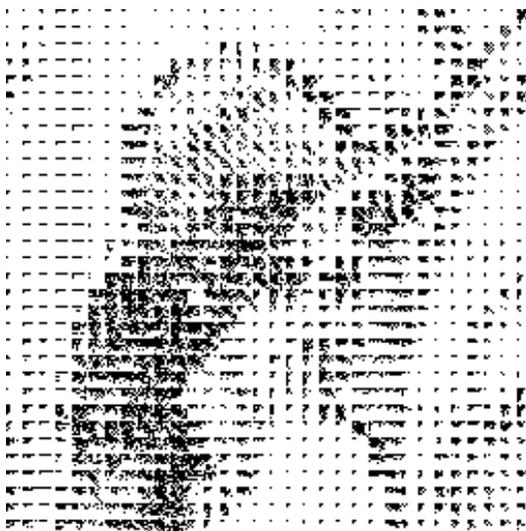


Fig. 1. Mapa de significância para a imagem Lena 256×256 [24].

Desta feita, o escopo deste trabalho fica delimitado a codificar imagens e seqüências de imagens por meio do casamento de padrões utilizando apenas os mapas de significância de cada bloco. Além da simplicidade de tratamento, pode-se justificar a busca de padrões similares no interior da mesma imagem, formando dicionários adaptativos ou semi-adaptativos, aproveitando-se a regularidade e localidade inerente às imagens naturais, nas quais as estruturas tendem a se agrupar. Observam-se regiões de **textura**, identificadas pela existência de muitas bandas espectrais de alta freqüência; **bordas**, mais facilmente identificadas no domínio espacial, e as **zonas homogêneas**, onde há a predominância de baixas freqüências. Desta forma, podemos encontrar características espectrais semelhantes em blocos vizinhos e próximos de uma imagem. Como o mapa de significância revela a existência de uma determinada freqüência espacial naquele bloco com amplitude superior a um determinado limiar, é possível encontrarmos mapas de significância com certo grau de similaridade entre blocos próximos de uma imagem. Esta característica pode ser aproveitada durante a codificação.

A. Descrição resumida do método

Podemos resumir a codificação por casamento de padrões nos seguintes passos:

- 1) Encontre no espaço de busca um bloco *similar* ao atual;
- 2) Codifique a posição relativa ao bloco atual;
- 3) Calcule e codifique o resíduo.

O espaço de busca, ou dicionário, é um conjunto de elementos pré-estabelecidos, indexados, que são varridos na busca pelo elemento mais adequado à codificação. Um exemplo de espaço de busca dependente da imagem é uma região bidimensional causal ao redor do bloco atual.

B. Considerações

A busca e a representação de padrões similares podem ser interpretadas como uma modelagem de mais alto nível das características estatísticas do conjunto de coeficientes quantizados. O nível de uma modelagem para condicionamento é relativo ao número de símbolos utilizados na formação de um contexto. Assim, como na busca de padrões diversos elementos são varridos na procura da maior similaridade, há uma junção de informações de diversas fontes na composição dos contextos.

Um bloco de coeficientes pode ser representado por um mapa de significância, que indica que elementos são não nulos, e por suas respectivas magnitudes e sinais. No presente trabalho a ênfase é o casamento de padrões de mapas de significância. Isto é justificado pelo fato de a maior parcela dos bits serem utilizados na codificação deste mapa, para diversas taxas de bits, e pela pouca estrutura apresentada pela magnitude e sinal dos coeficientes.

O CPP pode ser visto como um estágio de pré-processamento, no qual os mapas de bits formam aglomerados que são casados entre si. Por exemplo, o casamento pode ser realizado através da minimização da distância de Hamming. Nesse caso, a diferença entre os blocos é obtida pela operação XOR. O número de bits 1 resultante é o valor a ser minimizado. O resultado da operação XOR, ou seja, o resíduo, deve ser enviado juntamente com o índice do bloco no dicionário. Com essa métrica a diferença obtida possui um número menor de bits 1, o que é vantajoso. Para diversas técnicas de codificação de imagens binárias, ocorre uma relação linear entre o número de bits obtidos pela compressão e o número de bits 1 na imagem original [1]. Isto reforça a idéia de realizar um pré-processamento que reduza o número de bits 1 em um mapa binário. Para o caso específico de um mapa de significância de coeficientes DCT de imagens, é mostrado em [31] que a relação entre o número de coeficientes não nulos após a quantização e a taxa final também é aproximadamente linear.

III. IMPLEMENTAÇÃO DE CASAMENTO DE PADRÕES

Nesta seção apresentamos as técnicas utilizadas e os detalhes de implementação do método de casamento de padrões como codificador entrópico em um codificador MPEG-4.

A. Modificações no codificador MPEG-4

As informações de cabeçalho, de vetores de movimento e de textura (elementos DCT) são codificadas no MPEG-4 por meio

de um código de comprimento de sequência (*Run Length*). Na codificação dos elementos DCT utiliza-se também código de comprimento variável (VLC, *Variable Length Code*). Denominamos esse codificador de RL+VLC. Implementamos um codificador baseado em casamento de padrões em substituição ao RL+VLC. Dentre os vários elementos de sintaxe de um feixe de bits de um vídeo (tais como, cabeçalho, vetores de movimento, etc.), apenas os coeficientes DCT são codificados com CPP. Os demais seguem o procedimento convencional.

Para realizar essa substituição de codificador, foram realizados dois tipos de implementação: (a) inserção de um sistema de coleta dos elementos dos blocos DCT e gravação em arquivo para tratamento externo e (b) inclusão direta dos dados codificados com CPP e código aritmético no *bitstream* do MPEG-4.

A implementação (a) é mais simples porque não envolve mudanças no processamento do codificador MPEG-4. Uma vez retirados os elementos DCT, obtemos uma imagem digital com as mesmas dimensões dos quadros de vídeo com 16 bits de precisão por elemento. O arquivo gerado é processado por um programa de casamento de padrões que gera um *bitstream* (arquivo) de saída. O arquivo pode ser decodificado e deve levar ao mesmo resultado do arquivo original, uma vez que estamos trabalhando com codificação sem perdas.

Esse método possui como atrativo, além da simplicidade, o fato de podermos realizar vários testes de sintonias de parâmetros do codificador e avaliar de forma simples o resultado, o tamanho do arquivo de saída. Além disso, o arquivo de elementos pode ser usado para fazer uma verificação rápida da taxa de compressão obtida com o uso de codificadores universais, como Ziv-Lempel, PPM ou BWT [12], facilmente encontrados, em comparação com a codificação dedicada realizada pelo MPEG-4 e pelo CPP.

Nesse método, entretanto, perde-se a hierarquia entre macroblocos e blocos e, no caso de CPP, os quadros utilizados na construção do dicionário precisam ser aplicados juntamente com o quadro atual no codificador.

Na implementação (b) devemos observar cuidadosamente que elementos de sintaxe (cabeçalho geral, cabeçalhos dos macroblocos, vetor de movimento, textura, etc.) devem ser codificados. Uma vez que o *bitstream* utilizado é o mesmo do MPEG-4 e como não se pretende trocar a codificação de todos os elementos de sintaxe, faz-se necessário um processo de multiplexação de *bitstreams*. Por sorte, na codificação dos macroblocos, os únicos elementos de sintaxe codificados com CPP são sempre os últimos campos a serem codificados no MPEG-4. Assim, o cabeçalho global é codificado diretamente usando as tabelas originais do MPEG-4. Não há mudança no código desta parte. A codificação dos macroblocos foi repetida para a inserção do código de CPP e codificador aritmético. Para tanto, utilizou-se um modo especial do codificador aritmético CABAC (*context adaptive binary arithmetic coder*, codificador aritmético binário adaptativo com contextos). Nesse modo, chamado modo *skip*, codificam-se elementos em que se sabe que a entropia é perto de 0,5 e não é preciso realizar adaptação. Esse modo é utilizado para codificar todos os bits que não sejam elementos DCT. Entre eles encontramos os elementos de cabeçalho dos macroblocos. Todos esses elementos são codificados no modo *skip* e utilizam exatamente o mesmo número de bits que utilizariam se fossem codificados

diretamente. Isto ocorre porque as tabelas de codificação VLC ainda são utilizadas, mesmo que não para saída direta dos bits, mas para inserção no codificador aritmético.

Os dados dos elementos DCT são os últimos codificados em um macrobloco. Nos macroblocos *inter* é necessário também codificar os vetores de movimento. Utilizou-se, para tal, o mesmo método de outros elementos de sintaxe, ou seja, aproveitar a tabela VLC e enviar os bits obtidos diretamente para o codificador aritmético.

B. Estrutura de dados do dicionário

Como estrutura de dados para armazenar o dicionário foi implementado um *heap*. Um *heap* é uma estrutura de dados que armazena coleções de objetos (com chaves de busca) e tem as propriedades de ser uma árvore binária completa e ter uma ordenação de *heap*. É normalmente implementada como uma matriz em que cada nó da árvore corresponde a um elemento da matriz.

Uma árvore binária completa é uma árvore binária que está preenchida completamente em todos os seus níveis, com uma possível exceção do último nível, que é preenchido da esquerda para a direita.

A propriedade de ordenação de *heap* pode ser definida como para cada nó v , que não seja a raiz, a chave armazenada em v é maior ou igual (ou menor ou igual) à chave armazenada no nó pai de v . Como exemplo, temos uma *heap* de máximo, em que o maior valor é armazenado na raiz e temos nos descendentes valores sempre menores, conforme ilustra a Fig. 2.

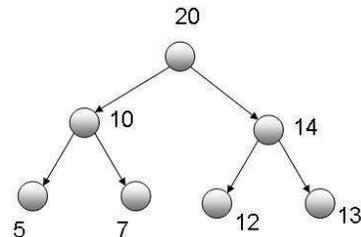


Fig. 2. *Heap* de máximo.

Na implementação do *heap* para o casamento de padrões, a chave utilizada foi o número de vezes em que determinado padrão foi usado na codificação de algum outro bloco. Cada vez que um bloco é identificado no dicionário como o mais adequado à codificação de outro bloco, incrementa-se o contador de utilização e realiza-se um rearranjo de todo o *heap*, para manter a propriedade de ordem de *heap*. Esse rearranjo é denominado *heapify*. Para manter a ordem de *heap*, os valores de todos os filhos de um nó devem ser menores que o valor armazenado nesse nó.

A manutenção dos elementos mais utilizados com índices menores do que os elementos menos utilizados permite a aplicação de métodos de codificação de inteiros, como os códigos de Golomb-Rice [1], com grande eficiência.

Os mapas binários dos blocos 8×8 foram codificados como inteiros de 64 bits. Isto permite uma comparação mais rápida dos valores, através de uma única operação XOR.

C. Mecanismo de busca

Para determinarmos o mecanismo de busca e armazenamento no dicionário é necessário definirmos uma métrica de distância entre os símbolos armazenados. Nesta implementação foi utilizada a distância de Hamming, obtida pela contagem do número de bits 1 residuais à utilização da operação XOR. O número de bits gastos na representação de uma mapa binário possui alta correlação com o número de posições significativas (com valor 1) no mapa. Como no CPP o número de bits necessários para representar um bloco é a soma do número de bits gastos na representação do índice no dicionário com o número de bits gastos para representar os resíduos, foi inserido no mecanismo de busca uma aproximação do número de bits devido à codificação do índice como sendo o número de níveis na *heap*.

De posse da aproximação do número de bits utilizados para codificar utilizando CPP, o codificador pode decidir se irá codificar cada bloco de forma direta ou utilizando algum elemento do dicionário como predição.

IV. RESULTADOS

O casamento parcial de padrões é aplicado à codificação sem perda ou entrópica dos coeficientes que resultam das transformadas de bloco em imagens e vídeo.

A técnica de CPP é avaliada em uma arcação de codificação de vídeo compatível com o padrão MPEG-4. Os quadros do tipo *intra* são codificados como imagens, sem referência a outros quadros. Assim, utilizando o mesmo arcação, podemos obter resultados para imagem (*intra*) e vídeo (*inter*).

Foram codificados o plano de luminância Y e os planos de crominância U e V , tanto nos quadros *intra* quanto *inter*.

Adiante veremos os detalhes da substituição do codificador entrópico para os coeficientes DCT do MPEG-4 pelo CPP.

O CPP utiliza como etapa final um codificador aritmético otimizado para dois símbolos com possibilidade de seleção adaptativa de contextos [12], [32], o CABAC, utilizado no recente padrão de codificação de vídeo H.264, também conhecido com MPEG-4 AVC ou parte 10 [33]. A adaptação de contextos, se bem explorada, pode proporcionar uma diminuição no número de bits utilizados. No CPP, a seleção dos contextos foi realizada de forma similar ao CABAC original. Algumas modificações foram necessárias devido ao fato de que o padrão H.264 trabalha com blocos de DCT 4×4 .

No CABAC, no lugar de contextos temos estados, que representam a probabilidade dos símbolos de forma aproximada. No H.264 são utilizados 276 estados para codificar todos os possíveis elementos de sintaxe [34]. Na implementação aqui realizada, os estados não são utilizados para codificar os elementos de sintaxe, a menos dos coeficientes DCT. Dessa forma, utilizamos um conjunto completamente diferente de estados. O conjunto de estados aqui utilizados pode ser visto na Tabela I.

O valor dos coeficientes é codificado em dois passos. Em primeiro lugar, é realizada uma binarização (separação em *bins*) do valor utilizando um código exponencial de Golomb truncado [34]. Em seguida, os *bins* obtidos são codificados com os estados selecionados. Nesta implementação o sinal dos coeficientes é codificado utilizando um modo especial

TABELA I

ESTADOS ASSOCIADOS COM ELEMENTOS DE SINTAXE.

Elementos de sintaxe	Estado
Valor dos coeficientes	0 - 19
Significância sem CPP	20-83
Bit indicativo de último sem CPP	100-163
Significância com CPP	180 - 313
Índice do CPP	320 - 336

do CABAC para símbolos não compressíveis. Isto acontece porque o sinal é considerado descorrelacionado.

Os índices encontrados durante a busca de casamento de padrões também são codificados utilizando código exponencial de Golomb truncado. Graças ao uso do *heap* como estrutura de dados de armazenamento do dicionário e da utilização da frequência de uso como chave de busca, observa-se uma distribuição em que os elementos de índices menores ocorrem com muito mais frequência.

Os demais elementos de sintaxe são codificados utilizando as próprias tabelas VLC do MPEG-4. Os elementos de cabeçalho são codificados diretamente para o *bitstream* de saída. Os elementos de sintaxe que fazem parte dos macro-blocos, como a indicação de codificação, o campo CBP e os vetores de movimento, passam pelo codificador CABAC no modo *skip*, ou seja, com probabilidade 0,5, sem solicitar do codificador aritmético a atualização de probabilidades.

Para avaliar se os blocos transformados de uma imagem possuem mapas de significância similares dentro de uma certa vizinhança espacial, foi realizado um experimento de busca de padrões. Os testes foram realizados numa vizinhança causal de 8×8 blocos. A métrica utilizada foi a distância de Hamming. O total de bits 1 antes e após a operação XOR é relatado na Tabela II, para a codificação da imagem Bárbara (Fig. 3) para algumas taxas de bits. Observa-se, para todas as taxas de codificação consideradas, que, após a operação de busca de padrões, menos da metade dos coeficientes permaneceram significativos.

TABELA II

AVALIAÇÃO DO NÚMERO DE POSIÇÕES SIGNIFICANTES ANTES E APÓS A APLICAÇÃO DO CASAMENTO DE PADRÕES.

Taxa (bpp)	Razão compressão	Coefic. significantes	Após XOR
1,00	1:8	46.052	21.132
0,50	1:16	22.455	10.153
0,25	1:32	10.875	3.912

A seguir são apresentados resultados da técnica de CPP.

As primeiras imagens das seqüências utilizadas para a realização dos testes de compressão podem ser vistas nas Figs. 4 (*Flower Garden*) e 5 (*Foreman*). A seqüência *Flower Garden* está no formato SIF, que possui 352×240 pixels. A seqüência *Foreman* está no formato CIF, que corresponde a 352×288 pixels. As seqüências estão no formato YCbCr 4:2:0 com 8 bits de precisão. As figuras mostram apenas o canal de luminância.

Nas Figs. 6 e 7 observamos a codificação do primeiro quadro das seqüências de imagens *Flower Garden* e *Foreman*, respectivamente, utilizando o casamento parcial de padrões. Para o propósito de comparação, são apresentados resultados obtidos com o MPEG-4 convencional (denotado nas figuras simplesmente por MPEG-4) e com a substituição do RL-VLC



Fig. 3. Imagem Bárbara original.



Fig. 4. Primeiro quadro da seqüência *Flower Garden*. Cena com movimentação rápida de toda a imagem devido à movimentação da câmera com relação à imagem. Cena com muitas texturas. Formato SIF.

pele codificador aritmético de contextos (denotado nas figuras por ARIT). As figuras mostram que CPP e ARIT apresentam uma superioridade em relação ao MPEG-4, levando a um menor número de bytes para todos os valores de fator de qualidade (Q) considerados.

Os benefícios introduzidos pela técnica de CPP podem ser observados na Fig. 6. De fato, para a seqüência *Flower Garden*, para todos os valores de fator de qualidade considerados, o número de bits utilizados na codificação é menor que o obtido com o uso de MPEG-4 e ARIT.

Para a seqüência *Foreman*, cujos resultados estão apresentados na Fig. 7, observamos que há benefício da utilização de CPP em substituição a ARIT para altas taxas de codificação, ou seja, para parâmetros de quantização baixos. Para taxas



Fig. 5. Primeiro quadro da seqüência *Foreman*. Movimentação rápida do objeto principal com um fundo estático. Formato CIF.



Fig. 6. Codificação *intra* do primeiro quadro da seqüência *Flower Garden*: número de bytes utilizados na textura de acordo com o fator de qualidade Q .



Fig. 7. Codificação *intra* do primeiro quadro da seqüência *Foreman*: número de bytes utilizados na textura de acordo com o fator de qualidade Q .

de codificação mais baixas, o número de coeficientes não nulos é pequeno. Embora seja possível encontrar padrões de semelhança nestas condições, o custo, ou seja, o número de bits usados para representar o padrão normalmente é maior do que para a codificação do bloco original. Nas taxas de codificação mais altas, há um número elevado de coeficientes significativos. Através do casamento de padrões é possível diminuir o número de bits significativos do mapa de significância.

O CPP apresentou um bom desempenho principalmente para imagens (quadros *intra*) e resíduos (quadros *inter*) com muita textura e altas taxas de codificação. Em comparação com o codificador encontrado no padrão MPEG-4, observa-se que o CPP consegue realizar um melhor trabalho de codificação quando o número de coeficientes significativos é grande.

Para a seqüência *Flower Garden*, a Tabela III mostra que a substituição de MPEG-4 por CPP leva a uma economia em termos de número de bytes situada entre 10,65% e 22,46%. Uma leve superioridade de CPP sobre ARIT é observada para todos os valores de Q considerados – reduções em termos de número de bytes de 1,50% a 4,58% são obtidas ao se usar CPP em substituição a ARIT.

Em se tratando da seqüência *Foreman*, para a faixa de

TABELA III

REDUÇÃO DO NÚMERO DE BYTES OBTIDAS COM USO DE CPP EM
SUBSTITUIÇÃO A MPEG-4 E A ARIT, PARA DIVERSOS VALORES DE Q ,
PARA A SEQUÊNCIA *Flower Garden*.

Q	Redução sobre MPEG-4	Redução sobre ARIT
1	22,46%	3,00%
2	14,40%	4,38%
3	11,78%	4,37%
4	10,75%	4,04%
5	10,65%	4,20%
6	11,13%	4,20%
7	11,49%	4,45%
8	11,77%	4,58%
9	12,18%	4,48%
10	12,64%	4,43%
11	13,16%	4,47%
12	13,18%	3,96%
13	13,43%	3,76%
14	13,62%	3,62%
15	14,08%	3,42%
16	14,29%	2,96%
17	14,32%	2,63%
18	14,60%	2,44%
19	15,11%	2,12%
20	15,46%	1,50%

valores de Q em consideração, economias em termos de número de bytes situadas entre 7,67% e 18,57% são obtidas ao se substituir MPEG-4 por CPP. Uma superioridade de CPP sobre ARIT é observada para valores de Q entre 1 e 14.

V. CONCLUSÃO

Este trabalho apresentou uma técnica de codificação de imagem e vídeo baseada em casamento parcial de padrões (CPP). A técnica de codificação sem perdas apresentada neste trabalho é aplicada aos coeficientes quantizados, oriundos de transformadas de bloco aplicadas à compressão de imagem e vídeo. Os resultados obtidos mostram que a substituição do MPEG-4 convencional por sua correspondente versão modificada para acomodar a técnica de CPP ora apresentada leva a economias de número de bits no processo de codificação de imagem e vídeo. Como trabalho futuro, a técnica de CPP apresentada neste artigo será avaliada no padrão H.264.

REFERÊNCIAS

- [1] A. Bookstein and S. T. Klein. "Compression of Correlated Bit-vectors". *Information Systems*, vol. 16, no. 4, pp. 387–400, 1991.
- [2] I. JTC1/SC29/WG1. "JBIG2 Final Comittee Draft FCD-14492".
- [3] Y. Ye and P. Cosman. "Fast and Memory Efficient JBIG2 Encoder". In *Proceedings of the 2001 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001.
- [4] Y. Ye and P. Cosman. "Speedup Techniques for Text Image Compression with JBIG2". In *Proceedings of the the 35th IEEE Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, November 2001.
- [5] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, W. J. Rucklidge and F. Ono. "The Emerging JBIG2 Standard". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 838–848, November 1998.
- [6] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio and Y. LeCun. "High Quality Document Image Compression with DjVu". *Journal of Electronic Imaging*, 1998.
- [7] ITU-T. "Information Technology - Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression." ITU-T Recommendation T.82, 1993.
- [8] I. Witten, T. Bell, H. Emberson and S. Inglis. "Textual Image Compression: Two-stage Lossy/lossless Encoding of Textual Images", 1994.
- [9] P. G. Howard. "Text Image Compression using Soft Pattern Matching". *The Computer Journal*, vol. 40, no. 2/3, pp. 146–154, 1997.
- [10] S. Inglis and I. H. Witten. "Compression-Based Template Matching". In *Proceedings of IEEE Data Compression Conference (DCC)*, pp. 106–115, 1994.
- [11] S. Inglis. "Lossless Document Image Compression", 1999.
- [12] T. C. Bell, J. G. Cleary and I. H. Witten. *Text Compression*. Prentice Hall, Englewood Cliffs, NJ, USA, 1990.
- [13] W. Finamore, M. Carvalho and J. Kieffer. "Lossy Compression with the Lempel-Ziv Algorithm". In *Proceedings of 11th Brazilian Telecommunications Symposium*, pp. 141–146, 1993.
- [14] T. Luczak and W. Szpankowski. "A Suboptimal Lossy Data Compression Based on Approximate Pattern Matching". *IEEE Transactions on Information Theory*, vol. 43, pp. 1439–1451, 1997.
- [15] W. Szpankowski, M. Atallah and Y. Genin. "Pattern Matching Image Compression: Algorithmic and Empirical Results". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 618–627, 1999.
- [16] W. Szpankowski, M. Atallah and Y. Genin. "2D-Pattern Matching Image and Video Compression: Theory, Algorithms, and Experiments". *IEEE Transactions on Image Processing*, vol. 11, pp. 318–331, 2002.
- [17] V. Ratnakar and M. Livny. "RD-OPT: An Efficient Algorithm For Optimizing DCT Quantization Tables". In *Proceedings of IEEE Data Compression Conference (DCC)*, pp. 332–341, 1995.
- [18] V. Ratnakar and M. Livny. "Extending RD-OPT with Global Thresholding for JPEG Optimization". In *Proceedings of IEEE Data Compression Conference (DCC)*, pp. 379–386, 1996.
- [19] A. Ortega and K. Ramchandran. "Rate-Distortion Techniques in Image and Video Compression". *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.
- [20] M. Flierl, T. Wiegand and B. Girod. "Multihypothesis Pictures for H.26L". In *Proceedings of the IEEE International Conference on Image Processing, ICIP-2001*, Thessaloniki, Greece, October 2001.
- [21] R. Koenen. "Overview of the MPEG-4 Standard". Technical Report JTC1/SC29/WG11 N4030, ISO/IEC, March 2001.
- [22] G. J. Sullivan and T. Wiegand. "Rate-Distortion Optimization for Video Compression". *IEEE Signal Processing Magazine*, pp. 74–90, November 1998.
- [23] T. Wiegand, M. Lightstone, D. Mukherjee, T. George and S. K. Mitra. "Rate-distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 182–190, April 1996.
- [24] M. R. A. Morais. "Codificação de Imagem e Vídeo com Casamento Parcial de Padrões em Transformadas de Bloco". Ph.D. thesis, Universidade Federal de Campina Grande, COPELE, Campina Grande, PB, 2004.
- [25] A. Said and W. A. Pearlman. "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.
- [26] J. M. Shapiro. "Embedded Image Coding Using Zerotrees of Wavelet Coefficients". *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, December 1993.
- [27] S. Mallat. *A Wavelet Tour of Signal Processing - second edition*. Academic Press, 1999.
- [28] R. L. de Queiroz and T. D. Tran. *The Handbook on Transforms and Data Compression*, chapter Lapped transforms for image compression, pp. 197–265. CRC Press, October 2000.
- [29] Z. Xiong, K. Ramchandran and M. T. Orchard. "Space-Frequency Quantization for Wavelet Image Coding". *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 677–693, May 1997.
- [30] G. M. Davis and A. Nosratinia. "Wavelet-based Image Coding: An Overview". *Applied and Computational Control, Signals, and Circuits*, vol. 1, no. 1, Spring 1998.
- [31] Z. He and S. Mitra. "A Unified Rate-Distortion Analysis Framework for Transform Coding". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1221–1236, December 2001.
- [32] I. Witten, R. Neal and J. Cleary. "Arithmetic Coding for Data Compression". *Communications of the ACM*, vol. 30, pp. 520–540, 1987.
- [33] J. V. T. of ISO/IEC JTC1/SC29/WG11 and I. S. D. JVT-G050. *Draft ITU-T Recommendation H.264 and Draft ISO/IEC 14496-10 AVC*, Mar 2003.
- [34] D. Marpe, H. Schwarz and T. Wiegand. "Context-Based Adaptive Binary Arithmetic Coding in the H.264 / AVC Video Compression Standard (invited paper)". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, 2003.