Codificação eficiente de mapas de profundidade com base em predição e aproximação linear

Luís F. R. Lucas, Nuno M. M. Rodrigues, Carla L. Pagliari, Eduardo A. B. da Silva, Sérgio M. M. de Faria

***Resumo*— Este artigo trata do problema da compressão de mapas de profundidade para aplicações de vídeo 3D, baseadas na síntese de vistas virtuais. Neste sentido, é proposto um algoritmo alternativo aos atuais padrões de codificação de imagem, que evita os problemas conhecidos na compressão de mapas de profundidade. O algoritmo proposto é baseado numa segmentação flexível e predição hierárquica, apropriados para a representação das bordas abruptas dos objetos. O sinal de resíduo é aproximado por uma função linear.**

Quando comparado com os algoritmos concorrentes, os experimentos mostram que os mapas de profundidade codificados pelo nosso algoritmo possuem desempenho estado-da-arte na síntese de vistas virtuais.

***Palavras-Chave*— Codificação de mapas de profundidade, aproximação linear, codificação preditiva, síntese de vistas.**

***Abstract*— This paper studies the problem of depth map compression for 3D video applications, based on virtual view synthesis. In this context, we propose an alternative algorithm to the current image coding standards, which avoids the known problems of depth map compression. The proposed algorithm is based on a flexible segmentation, combined with an hierarchical prediction step, that efficiently represent the objects' sharp edges. The residue signal is approximated by a linear function.**

When compared to other alternative algorithms, the experiments show that depth maps compressed with our algorithm achieve state-of-the-art performance on virtual view synthesis.

***Keywords*— Depth map coding, linear approximation, predictive coding, view synthesis.**

I. INTRODUÇÃO

O desenvolvimento dos últimos anos nos sistemas e tecnologias de vídeo 3D é uma consequência direta do esforço que as produtoras de conteúdos multimídia têm realizado para proporcionar uma melhor experiência visual aos usuários. A disponibilidade dos conteúdos 3D foi para além dos tradicionais mercados restritos, na medida em que este é agora acessível de uma forma inovadora num vasto leque de plataformas, tais como equipamento de entretenimento, dispositivos móveis, computadores pessoais, etc.

A adoção dos sistemas 3D é determinada pela capacidade que estes têm de criar a sensação de imersão na cena. O sistema 3D convencional, também denominado de sistema estéreo, permite que o usuário observe a cena com sensação de profundidade, mas apenas a partir de um ponto de vista. Este sistema é baseado na transmissão de dois sinais de vídeo, cada um associado a uma vista do sistema binocular humano. A compressão do conteúdo estéreo pode ser realizada codificando as duas vistas de forma independente com os atuais algoritmos padronizados para vídeo 2D (*simulcast*), ou, numa segunda abordagem, explorando a redundância entre vistas, através da compensação de disparidade.

O sistema multivistas é uma extensão do sistema estéreo que possui um número maior de vistas. Este sistema pode ser usado num leque mais vasto de aplicações. Ao permitir a observação de uma mesma cena a partir de diferentes pontos de vistas, ele melhora a experiência visual do usuário. A compressão do vídeo multivistas pode ser realizada com base no modelo *simulcast* ou explorando a redundância entre vistas. O aumento do número de vistas disponíveis é uma tendência nos sistemas futuros, por melhorar a experiência oferecida ao usuário. A desvantagem do aumento do número de vistas é a grande quantidade de informação associada, o que penaliza a transmissão ou armazenamento.

O codificador MVC (*Multiview Video Coding*) é o padrão estado-da-arte proposto para codificação de vídeo multivistas. Este algoritmo é uma extensão do codificador de vídeo monovista H.264/AVC [1]. Embora o MVC use algoritmos de compensação de disparidade baseados no algoritmo de *block-matching*, os ganhos obtidos explorando a redundância intervistas não são muito significativos.

O sistema vídeo+profundidade (*video+depth*) [2] é uma alternativa ao sistema de vídeo estéreo, onde apenas a informação de uma das vistas e o respectivo mapa de profundidade são transmitidos. A ideia é que a vista não transmitida possa ser sintetizada com base num processo denominado DIBR (*depth-image-based rendering*), a partir das informações de textura e profundidade da outra vista. Uma vez que a profundidade usa geralmente menos taxa do que uma vista de textura, esta representação é mais eficiente que o sistema estéreo, sendo uma abordagem promissora. O fato de considerar apenas uma vista de textura também permite manter a compatibilidade com os sistemas 2D. Além disso, este sistema possibilita a síntese de um grande número de vistas, dentro de um intervalo angular limitado, o que aumenta a sua utilidade. Estas são algumas das características que justificam a definição desta solução no padrão MPEG-C Parte 3 [3].

Uma extensão óbvia do sistema vídeo+profundidade é o sistema multivista+profundidade. Segundo este, as vistas visualizadas são sintetizadas a partir da informação de um conjunto limitado de vistas e mapas de profundidade. Isto significa que os pontos de captura das vistas transmitidas não correspondem

Luís F. R. Lucas[†,*], Nuno M. M. Rodrigues[†,§], Carla L. Pagliari[‡], Eduardo A. B. da Silva[*], Sérgio M. M. de Faria[†,§], [†]Instituto de Telecomunicações, Portugal; [§]ESTG, Instituto Politécnico de Leiria, Portugal; [‡]DEE, Instituto Militar de Engenharia, Brasil; [*]PEE/COPPE/DEL/POLI, Universidade Federal do Rio de Janeiro, Brasil; E-mails: luis.lucas@lps.ufrj.br, nuno.rodrigues@co.it.pt, carla@ime.eb.br, eduardo@lps.ufrj.br, sergio.faria@co.it.pt. Este trabalho foi financiado pela FCT (Fundação para a Ciência e Tecnologia, Portugal), com recursos da bolsa SFRH/BD/79553/2011, e projeto COMUVI (PTDC/EEA-TEL/099387/2008).

necessariamente aos pontos de observação gerados pelas vistas virtuais sintetizadas. Desta forma, o número de vistas que é efetivamente transmitido pode ser muito inferior àquele que é requerido por um sistema multivistas que não usa informação de profundidade, permitindo um ganho significativo de desempenho.

No contexto dos sistemas assistidos por mapas de profundidade surge a necessidade de criar representações eficientes para esta informação. Dado que as intensidades destes mapas representam os valores das profundidades/disparidades que serão usadas na síntese de vistas, alterações nos valores absolutos, imputadas pelo processo de compressão, podem ser desastrosas. Os métodos existentes para compressão de imagens genéricas visam preservar a qualidade visual sem a preocupação de preservar os valores das intensidades e/ou bordas. Neste sentido, este trabalho investiga técnicas eficientes de codificação de mapas de profundidade, do qual resultou um algoritmo com desempenho superior ao das propostas existentes na literatura.

O restante artigo está organizado da seguinte forma. A próxima seção discute o problema da codificação dos mapas de profundidade referindo alguns dos algoritmos apresentados na literatura. A seção III descreve o algoritmo proposto neste trabalho, enquanto os resultados experimentais são mostrados na seção IV. Por fim, o artigo é concluído na seção V.

II. Codificação de mapas de profundidade

Os mapas de profundidade relacionam as distâncias das superfícies dos objetos de uma cena a partir de um ponto de vista. Esta representação é feita por intermédio de uma imagem em escala de cinza, na qual a intensidade dos *pixels* é mais clara quanto mais próximo da câmera estiver o objeto correspondente. Estes são constituídos majoritariamente por zonas suaves correspondentes a regiões com profundidade semelhante, e zonas com variações abruptas associadas às bordas dos objetos localizados a diferentes profundidades. As abordagens mais simples à codificação destes mapas são baseadas nos atuais padrões de codificação de imagens naturais, tais como o H.264/AVC, ou o emergente HEVC (*High Efficiency Video Coding*) [4]. Contudo, estes assumem que as imagens possuem uma natureza essencialmente suave, não sendo adequados para a codificação de algumas das características dos mapas. Ao utilizarem técnicas baseadas em transformadas, estes codificadores acabam descartando a informação de altas frequências, que no caso dos mapas de profundidade está associada às variações acentuadas de profundidade nas bordas de objetos. Por este motivo, os codificadores padronizados tendem a produzir artefatos indesejáveis junto às bordas mais abruptas (*ringing*), principalmente nas taxas mais baixas. Embora, os codificadores padronizados permitam manter a compatibilidade com as tecnologias existentes, os artefatos gerados por estes algoritmos prejudicam o desempenho dos algoritmos de síntese, que realizam uma reconstrução defeituosa junto das bordas que possuem os erros de codificação.

Os problemas dos codificadores padronizados têm motivado a investigação de métodos mais adequados para compressão de mapas de profundidade. Um deles é a codificação baseada em malha (*mesh*) [5]. Este divide o mapa de forma adaptativa numa malha irregular, segundo a estrutura de uma árvore triangular binária, conhecida por *tritree*. A partir dos nós da malha as outras amostras são interpoladas. A informação da árvore binária e valores de profundidade dos nós são codificados entropicamente e transmitidos, formando uma representação compacta do mapa. Um problema deste método é a necessidade de um grande número de remendos (*patches*) triangulares junto das bordas de objetos.

Uma proposta alternativa usa o padrão JPEG2000, e explora a possibilidade de definição de regiões de interesse (ROI) [6]. A ideia do algoritmo é atribuir cada objeto do mapa a uma ROI, e codificá-las com o algoritmo JPEG2000. Esta abordagem apresenta alguns problemas quando existem muitos objetos no mapa de profundidade.

O algoritmo *Multidimensional Multiscale Parser* (MMP) [7], originalmente apresentado como um codificador de imagens genéricas, foi proposto em [8] para codificação de mapas de profundidade. Sendo um algoritmo baseado no paradigma de casamento de padrões, o seu desempenho ultrapassa o dos outros métodos, contudo este apresenta uma complexidade computacional muito elevada, em ambos os lados do codificador e decodificador.

Um algoritmo de destaque para codificação de mapas de profundidade é conhecido por *Platelet* [9]. Neste algoritmo, o mapa é dividido segundo uma segmentação *quadtree* sendo que os blocos são aproximados por funções lineares. Esta abordagem baseia-se na suposição de que os mapas de profundidade são suaves e lineares por segmentos. Deste modo, os blocos suaves são aproximados usando uma função constante ou linear. Por outro lado, os blocos com descontinuidades são modelados pelas funções *wedgelet* ou *platelet*, definidas por duas funções constantes (*wedgelet*), ou duas funções lineares (*platelet*), separadas por uma linha reta. Todo o processo de decisão da divisão dos blocos e escolha das funções de aproximação é realizado de acordo com uma função de custo, que avalia a taxa e a distorção.

III. Algoritmo proposto

O trabalho proposto neste artigo consiste num algoritmo para codificação do mapa de profundidade de uma imagem 3D. Note que sequências de vídeo não serão consideradas neste trabalho. Isto acontece porque o nosso principal objetivo é investigar o desempenho de um algoritmo que apenas explora a redundância espacial, usando técnicas no domínio do espaço. Também não consideraremos a codificação conjunta de vários mapas de profundidade associados à mesma cena, i.e. capturados a partir pontos de vista distintos, pois neste caso, pelo mesmo motivo exposto acima, não exploramos a redundância inter-vistas. Em outras palavras, este trabalho trata apenas do problema principal da codificação de uma imagem, sendo que as outras abordagens (com exploração da redundância temporal e entre vistas) constituem uma interessante extensão deste trabalho.

A. Descrição do algoritmo

O algoritmo começa por dividir o mapa de profundidade em blocos 32×32. Durante o processo de codificação é usada uma

segmentação flexível, que possibilita um número de tamanhos de bloco superior à segmentação *quadtree*. Cada bloco 32×32 pode ser dividido na metade no sentido vertical ou horizontal, dando origem a blocos 32×16 ou 16×32, respectivamente. Por sua vez, estes dois tamanhos de bloco apenas podem ser divididos no sentido perpendicular à direção mais longa do bloco, resultando sempre em blocos 16×16. Estes blocos de tamanho maior são apropriados para codificação das zonas suaves de grandes dimensões, cuja ocorrência é frequente nos mapas de profundidade. Já a segmentação dos blocos 16×16 é realizada de forma mais flexível, com o intuito de melhorar a codificação das descontinuidades. Cada bloco (ou sub-bloco) pode ser segmentado vertical ou horizontalmente. Esta segmentação permite que um bloco 16×16 possa ser dividido em sub-blocos com tamanhos $2^m \times 2^n$, onde $m, n = 0, \ldots, 4$, o que corresponde a 25 escalas possíveis. Através da utilização de símbolos para sinalizar a ocorrência de segmentação e o sentido da mesma (vertical ou horizontal), é possível construir uma árvore de segmentação binária para cada bloco 32×32.

Para cada bloco da imagem, a primeira etapa do algoritmo é a predição hierárquica. Esta usa os mesmos 8 modos direcionais e o modo DC adotados no padrão H.264/AVC. A predição intra é útil ao nível da codificação das zonas suaves e descontinuidades. Ela permite não só reduzir a energia do sinal, e facilitar a sua codificação entrópica, como também codificar as descontinuidades existentes nos mapas através dos modos direcionais. A predição é testada nos vários sub-blocos resultantes da segmentação, excluindo os blocos menores, nomeadamente aqueles cuja maior dimensão é inferior a 8.

De modo a gerar uma predição mais precisa nas descontinuidades presentes nos mapas de profundidade, a filtragem passa-baixo proposta no H.264/AVC para a vizinhança do bloco a predizer não é considerada. Esta filtragem altera a estrutura das descontinuidades presentes na vizinhança do bloco, removendo as altas frequências da mesma. Como consequência, o desempenho da predição baseada nessa vizinhança filtrada tende a piorar.

A segmentação flexível e a predição hierárquica são as principais ferramentas capazes de codificar as descontinuidades dos mapas de profundidade. Para representar as zonas suaves propomos uma técnica de aproximação linear do resíduo produzido pela predição. O uso desta aproximação é razoável tendo em conta a característica suave dos mapas de profundidade, repletos de regiões aproximadamente constantes (ex. interior dos objetos) ou de regiões que variam suavemente (ex: o plano do solo ou paredes em perspectiva). Para um bloco de dimensão $2^m \times 2^n$, a aproximação linear é definida por:

$$\hat{f}(\tilde{x}, \tilde{y}) = \alpha_0 + \alpha_1 \tilde{x} + \alpha_2 \tilde{y} \quad , \tag{1}$$

onde $\tilde{x} = (x - 2^{m-1} + 1)$, $\tilde{y} = (y - 2^{n-1} + 1)$, (x, y) são as coordenadas dos *pixels* do bloco e α_i as constantes do modelo.

Dado que os valores das coordenadas (x, y) correspondem apenas a valores não-nulos, as transformações $\tilde{x}$ e $\tilde{y}$ deslocam seus valores para que estes possuam uma média aproximadamente nula. De fato, as médias de $\tilde{x}$ e $\tilde{y}$ não são exatamente zero, a menos que estes tivessem uma precisão numérica não-inteira. No entanto, a precisão não-inteira não é benéfica tendo em conta que a intensidade dos *pixels* só assume valores inteiros. A principal vantagem desta mudança de variável é a correspondência direta entre o coeficiente α_0 e a média do bloco de resíduo. Admitindo-se que o sinal de resíduo possui média aproximadamente nula, pode-se concluir através da relação anterior que a média do coeficiente α_0 também será nula. A utilização de coeficientes com média nula é importante porque permite que quantizadores simétricos sejam ótimos, o que facilita a codificação entrópica dos mesmos.

O método para estimação dos coeficientes lineares é baseado na minimização do erro médio quadrático entre os valores originais $f(x, y)$ e a aproximação linear $\hat{f}(\tilde{x}, \tilde{y})$. A minimização ℓ_2 foi escolhida por possuir uma solução fechada e bem conhecida, com um desempenho aceitável.

De acordo com a descrição anterior, para cada bloco de resíduo, três coeficientes são quantizados e transmitidos. Contudo, esta representação do resíduo pode tornar-se pouco eficiente ao longo da codificação da imagem. Um modo de minimizar este problema consiste em melhorar a eficiência de codificação das aproximações transmitidas previamente. Esta ideia é levada em consideração no algoritmo através da definição de um dicionário com as aproximações transmitidas. Durante o processo de codificação existe a possibilidade da transmissão explícita dos coeficientes estimados, ou da transmissão de um índice do dicionário ao qual corresponde uma aproximação que foi anteriormente usada. A decisão entre a utilização de um elemento de aproximação do dicionário ou uma nova aproximação é detalhada na próxima sub-seção.

No início da codificação o dicionário contém apenas a aproximação correspondente aos 3 coeficientes nulos. Esta será provavelmente a aproximação mais frequente ao longo do processo de codificação, assumindo que a energia do resíduo é modelada por uma distribuição gaussiana de média nula. Para permitir a reutilização das aproximações lineares, o algoritmo adiciona um novo elemento ao dicionário, cada vez que uma nova aproximação é explicitamente transmitida para o codificador por meio dos 3 coeficientes do modelo linear.

B. Otimização taxa-distorção

O processo de codificação é otimizado de acordo com a minimização de uma função de custo baseada na taxa e distorção. O valor deste custo depende de vários parâmetros do algoritmo, nomeadamente das decisões de segmentação, modos de predição escolhidos, e aproximações lineares usadas. A escolha da melhor combinação destes parâmetros depende do resultado de uma otimização exaustiva realizada ao nível de cada bloco 32×32. A primeira etapa dessa otimização consiste em gerar uma árvore de segmentação totalmente expandida para cada bloco 32×32. A cada nó dessa árvore está associado um bloco de escala diferente, e um custo de codificação. Posteriormente a árvore expandida é podada a partir dos nós inferiores, de baixo para cima, com base nos custos de cada nó. O objetivo é achar uma árvore de segmentação otimizada, $\mathcal{T}$, que represente o bloco com o menor custo dado pela seguinte função Lagrangiana:

$$J(\mathcal{T}) = D(\mathcal{T}) + \lambda R(\mathcal{T}) \tag{2}$$

onde $D(\mathcal{T})$ é a distorção do bloco e $R(\mathcal{T})$ é a taxa necessária para codificar a informação associada à árvore ótima $\mathcal{T}$. A

parcela da taxa usada no cálculo do custo diz respeito aos bits usados pelo modo de predição, *flags* de segmentação e codificação do resíduo. Note que a codificação do resíduo envolve a otimização de uma sub-árvore de resíduo associada a cada nó da árvore principal. A otimização do resíduo é necessária porque o mesmo pode ser sub-particionado e codificado de duas formas: calculando um novo modelo linear ou usando uma aproximação existente no dicionário. A melhor aproximação linear para o resíduo é escolhida com base na avaliação de uma função de custo. No caso da estimação dos coeficientes do modelo linear (cuja aproximação gera a menor distorção), o custo do bloco da escala l_k é dado pela função:

$$J_{model}(\mathrm{l}_k) = D_{model}(\mathrm{l}_k) + \lambda\Big(R(\text{flag0}) + \sum_{j=0}^{2} R(\alpha_j)\Big), \quad (3)$$

Quando um bloco da escala l_k é aproximado por um elemento i do dicionário, o custo é calculado pela seguinte função:

$$J_{dic}(\mathrm{l}_k, i) = D_{dic}(\mathrm{l}_k, i) + \lambda\Big(R(\text{flag1}) + R(i)\Big), \quad (4)$$

Os parâmetros D_{model} e D_{dic} são as distorções associadas com os dois modos de aproximação, enquanto $R(\text{flag0})$ e $R(\text{flag1})$ correspondem à taxa dos símbolos $flag0$ e $flag1$ usados para informar o decodificador se o resíduo é aproximado por uma nova função linear ou por um índice do dicionário. Na fase de avaliação dos custos, se não existir nenhum elemento i no dicionário com custo $J_{dic}(\mathrm{l}_k, i)$, inferior ao custo $J_{model}(\mathrm{l}_k)$, é realizada a transmissão dos três coeficientes do modelo linear. Caso contrário, o índice i do dicionário que gera o menor custo é transmitido.

Relativamente à codificação entrópica dos símbolos o algoritmo usa um codificador aritmético. Durante a otimização da função de custo é possível ter uma estimativa da taxa que seria usada pelo codificador aritmético para representar um determinado símbolo. Para sinalizar a segmentação são necessários 5 símbolos: dois símbolos identificam a direção (vertical ou horizontal) da partição dos blocos de predição na árvore principal; outros dois símbolos indicam se a segmentação é realizada ao nível da predição ou do resíduo; por último um símbolo é utilizado para indicar que não existem segmentações num (sub-)bloco. Os 9 modos de predição escolhidos são explicitamente transmitidos para o decodificador usando 9 possíveis símbolos. Em relação aos coeficientes do modelo linear, a sua codificação envolve uma etapa de quantização não uniforme. O coeficiente α_0 associado ao valor médio do bloco de resíduo é quantizado numa faixa entre -255 e 255, possuindo um passo de quantização menor em torno do valor 0. Os outros dois coeficientes associados aos declives da aproximação linear são quantizados num intervalo entre -127 e 127, possuindo uma quantização não-uniforme semelhante. O uso de quantização reduz o número de valores possíveis dos coeficientes e permite que o codificador aritmético se adapte mais rapidamente às estatísticas de utilização dos mesmos.

A codificação usa geralmente a soma dos erros quadráticos (SSE - *Sum of Squared Error*) como medida de distorção na função de otimização. A principal motivação para esta medida é a utilização do PSNR (*Peak Signal-to-Noise Ratio*), baseado no erro quadrático, para avaliação da qualidade das imagens codificadas. No caso dos mapas de profundidade, uma vez que estes não são sinais diretamente observados pelo utilizador, uma medida mais importante de desempenho do codificador é a qualidade PSNR das vistas sintetizadas a partir dos mesmos.

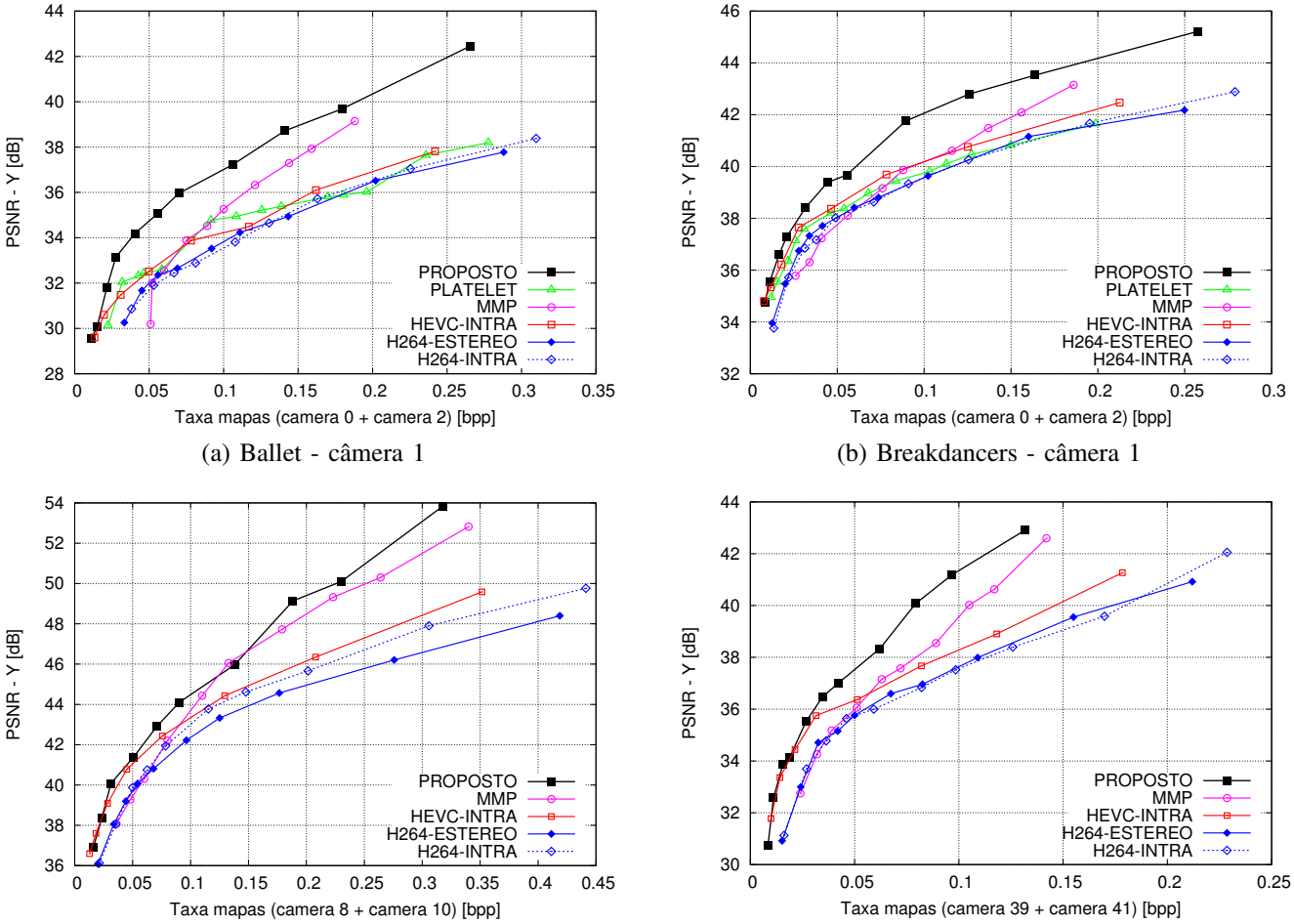
Dado que os mapas não são diretamente observados e avaliados, este trabalho propõe uma modificação na componente de distorção da função de custo, nomeadamente no uso de uma medida alternativa ao tradicional SSE. De fato, observou-se que o uso do erro absoluto (SAE - *Sum of Absolute Error*) na otimização taxa-distorção melhora o desempenho em termos do PSNR das vistas sintetizadas, embora o PSNR dos mapas de profundidade diminua. Esta observação mostra como o PSNR dos mapas de profundidade não é uma medida adequada para avaliação da qualidade dos mesmos. Sendo o processo de síntese o objetivo principal dos mapas de profundidade, escolhemos o SAE como medida de distorção para a função de otimização do custo no nosso algoritmo.

IV. Resultados experimentais

Para avaliar o desempenho do algoritmo desenvolvido, os mapas de profundidade codificados foram usados na síntese de vistas virtuais da *frame 0* para 4 sequências, especificamente a câmera 1 da *Ballet*, a câmera 1 da *Breakdancers*, a câmera 9 da *Book Arrival* e a câmera 40 da *Champagne Tower* [1]. No processo de síntese da vista associada à câmera n, foram considerados os mapas codificados associados às câmeras $n-1$ e $n+1$, e correspondentes imagens de textura originais. O algoritmo de DIBR utilizado foi o VSRS-3.5 [10], sendo que apenas o sinal de luminância foi usado na síntese. O código fonte do algoritmo proposto pode ser encontrado em [11].

A Figura 1 apresenta as curvas taxa-distorção das sequências referidas codificando dos mapas de profundidade com o algoritmo proposto, alguns algoritmos da literatura concorrentes, nomeadamente a *Platelet* [9] e o MMP [8], e os padrões estado-de-arte de codificação de imagem/vídeo, o H.264/AVC [1] (versão 18.0 do software JM, nos modos Intra e Estéreo com *High-profile*), assim como o futuro padrão HEVC [4] (*High Efficiency Video Coding*) usando a versão 5.2 do software HM no modo Intra. Relativamente, ao algoritmo *Platelet* apenas são apresentadas as curvas taxa-distorção para as sequências *Ballet* e *Breakdancers*, as únicas sequências para as quais os autores disponibilizaram os resultados [12]. Os resultados PSNR são calculados com base no erro entre as imagens sintetizadas com os mapas de profundidade originais e a versão codificada desses mapas. A componente da taxa apresentada nos gráficos é dada pela soma dos bits usados na codificação dos dois mapas de profundidade usados na síntese. Note, que as taxas das vistas de texturas não estão sendo computadas, porque estas não foram codificadas. Dado que as vistas de textura são comuns a todos os métodos, e o foco deste trabalho é nos mapas de profundidade, estas não foram codificadas para não mascarar os resultados da síntese. Desta forma, apenas eventuais artefatos introduzidos pelos diferentes métodos de compressão estão sendo comparados.

[1] As sequencias *Ballet* e *Breakdancers* foram produzidas pela Interactive Visual Media (Microsoft Research), e as sequências *Champagne Tower* e *Book Arrival* foram geradas por Tanimoto Lab (Nagoya University) e FHG-HHI, respectivamente.



(a) Ballet - câmera 1

(b) Breakdancers - câmera 1

(c) Book Arrival - câmera 9

(d) Champagne Tower - câmera 40

Fig. 1: Resultados taxa-distorção para as vistas sintetizadas da câmera n de algumas sequências, usando os mapas codificados por diferentes algoritmos e as vistas de textura originais associados às câmeras $n-1$ e $n+1$.

Em qualquer um dos gráficos apresentados, observa-se que as imagens sintetizadas com o nosso algoritmo apresentam o desempenho mais elevado, gerando a menor distorção, para a mesma taxa. O padrão HEVC apresenta resultados semelhantes para as taxas mais baixas, contudo o nosso algoritmo destaca-se para a maioria dos pontos taxa-distorção. Relativamente ao algoritmo concorrente *Platelet*, verifica-se também que o seu desempenho fica aquém do método proposto.

V. Conclusões

Neste artigo apresentamos um algoritmo alternativo para codificação de mapas de profundidade. O algoritmo é baseado num esquema de segmentação flexível dos blocos da imagem, combinado com predição hierárquica e uma codificação de resíduo usando um modelo de aproximação linear. Através de um método baseado em dicionário, e uma medida de distorção dada pela soma do erro absoluto, os mapas de profundidade são codificados de forma mais eficiente, tendo em conta o desempenho observado nas imagens sintetizadas. Os resultados experimentais mostram que a nossa proposta possui um desempenho superior aos algoritmos concorrentes da literatura, e padrões estado-de-arte de codificação de imagem.

Referências

[1] ITU-T and ISO/IEC JTC1, "Advanced video coding for generic audiovisual services," *ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, 2010.

[2] K. Muller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proceedings of the IEEE*, vol. 99, pp. 643–656, 2011.

[3] Philips Applied Technologies, "MPEG-C part 3: Enabling the introduction of video plus depth contents," 2008, Suresnes, France.

[4] http://hevc.info.

[5] M. Sarkis, W. Zia, and K. Diepold, "Fast depth map compression and meshing with compressed tritree," in *ACCV*, 2010, vol. 5995, pp. 44–55.

[6] R. Krishnamurthy, B. Chai, H. Tao, and S. Sethuraman, "Compression and transmission of depth maps for image-based rendering," in *ICIP*, 2001, vol. 3, pp. 828–831.

[7] N. Rodrigues, E. da Silva, M. de Carvalho, S. de Faria, and V. Silva, "On dictionary adaptation for recurrent pattern image coding," *IEEE TIP*, vol. 17, no. 9, pp. 1640–1653, September 2008.

[8] D. Graziosi, N. Rodrigues, C. Pagliari, E. da Silva, S. de Faria, M. Perez, and M. de Carvalho, "Multiscale recurrent pattern matching approach for depth map coding," in *PCS*, Dec. 2010, pp. 294–297.

[9] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Image Commun.*, vol. 24, pp. 73–88, Jan. 2009.

[10] M. Tanimoto, T. Fujii, and K. Suzuki, "View synthesis algorithm in view synthesis reference software 3.5 (VSRS3.5) Document M16090, ISO/IEC JTC1/SC29/WG11 (MPEG)," May 2009.

[11] http://www.lps.ufrj.br/profs/eduardo/linear_approx.

[12] http://vca.ele.tue.nl/demos/mvc/PlateletDepthCoding.tgz.