# Representação Esparsa para Preenchimento de Buracos de Expansão em Sínteses de Vistas Virtuais baseada em Profundidade

Danilo Amaral Ribeiro, Bruno Macchiavello e Camilo Dorea

***Resumo*— Renderização de imagem baseada em profundidade (do inglês *Depth Image Based Rendering* - DIBR) é uma técnica para sintetizar uma vista virtual com imagens de textura e mapas de profundidade disponíveis. No final do processo desoclusões e buracos de expansão podem estar presentes. Neste trabalho, investigamos o problema de preenchimento de buracos de expansão. Utilizou-se um método previamente proposto baseado no histograma de profundidade para identificar esses pixels. Posteriormente, propomos uma técnica de representação esparsa, baseada na aprendizagem não-paramétrica bayesiana de dicionários usando técnica recursiva para preencher esses buracos. Nossa abordagem superou o método de inpainting usado no software referência VSRS+ em até $6.19dB$ e superou trabalhos anteriores em preenchimentos de buracos de expansão.**

***Palavras-Chave*— Renderização de Imagens Baseada em profundidade, Interpolação de imagens, Representação Esparsa, Aprendizado Bayesiano não-paramétrico.**

***Abstract*— Depth Image Based Rendering - DIBR is a technique for synthesizing a virtual view with available texture images and depth maps. At the end of the process desocclusions and expansion holes may be present. We investigate the problem of filling in expansion holes. A previously proposed depth-based histogram was used to identify these pixels. Subsequently,in this wok we propose a sparse representation technique, based on Bayesian non-parametric learning of dictionaries using half-recursive technique to fill these holes. Our approach outperformed the inpainting method used in reference software VSRS+ up to $6.19dB$ and outperformed previous work on expansion hole filling.**

***Keywords*— Depth-image-based rendering, image inpainting, sparse representation, non-parametric Bayesian learning.**

## I. Introdução

Aplicações como *Free-Viewpoint TV* (FTV), permitem vários pontos de observação de uma mesma cena tridimensional (3D). No entanto, o custo de transmitir todos os pontos de vista desejáveis é alto, devido a quantidade de imagens necessárias para cada vista diferente. Uma abordagem mais prática é sintetizar vistas virtuais a partir de um determinado número de posições de cena fixas. A técnica de renderização de imagem baseada em profundidade (DIBR - *Depth Image Based Rendering*) é uma deformação 3D baseada em imagens de textura (imagens a cores convencionais) e mapas de profundidade (onde cada pixel representa a distância relativa do objetos na cena 3D e a câmera de captura)[1], que permite a síntese de vista. Em resumo, DIBR é um mapeamento pixel a pixel da vista de referência para um local na vista virtual, este local é derivado do pixel de profundidade correspondente na visão de referência. Devido às oclusões (áreas obstruídas) na vista de referência, podem aparecer buracos chamados desoclusões na vista sintetizada. Isto pode ser atenuado usando mais de uma vista de referência e/ou preenchidos usando algoritmos de *inpainting*. Vários trabalhos anteriores focaram o preenchimento dessas desoclusões [2].

No entanto, em aplicações imersivas, como teleconferência, um espectador em posição sentada observa as imagens renderizadas em uma tela de exibição 2D, onde os pontos de vista da imagem são sintetizados de acordo com as localizações rastreadas pela cabeça do visualizador [3]. Neste cenário, há movimentos de cabeça nas direções $x$ (direita e esquerda), $y$ (para cima e para baixo) e $z$ (aproximação e distanciamento da câmera). Poucos trabalhos na literatura abordaram formalmente o problema de sintetizar imagens de pontos de vista correspondentes a movimentos $z$-dimensionais [4] [5] [6]. Este trabalho foca nesse problema.

Quando a câmera virtual está localizada mais perto da cena do que a câmera de vista de referência, os objetos próximos da câmera aumentarão de tamanho. Isto significa que o mapeamento pixel a pixel durante o DIBR não será suficiente para completar todas as superfícies dos objetos renderizados. Esses objetos terão os buracos de expansão [4]. Os buracos de expansão diferem das desoclussões pois os objetos são visíveis nas vistas de referência, mas as amostras de pixel são insuficientes para representá-los. Este tipo de buracos são caracterizados por serem menores em área do que as desoclusões, mas eles normalmente estão presentes por toda a imagem. Um exemplo pode ser visto na Figura 1.

Por outro lado, tem havido um interesse significativo em representações esparsas em diferentes contextos, tais como *denoising* [7] [8], *inpainting* [9] [10] e codificação [11]. Essas aplicações exploram o fato de que a maioria das imagens pode ser esparsamente representada em um dicionário apropriado.

Esse trabalho propõe um procedimento em duas etapas para preenchimento de buracos de expansão em imagens sintetizadas com DIBR usando representação esparsa devido às características dos buracos de expansão. Utilizou-se primeiro um método previamente proposto baseado em histogramas de profundidade [4] para identificar pixels como buracos de expansão. Em seguida, uma representação esparsa é usada para o preenchimento do buraco de expansão na imagem de vista virtual. Os resultados experimentais mostraram que a identificação e o preenchimento apropriados dos buracos de

Danilo Amaral Ribeiro, Bruno Macchiavello e Camilo Dorea Departamento de Ciência da Computação, Universidade de Brasília, Brasília-DF, Brasil, E-mails: daniloamaralr@gmail.com, bruno@image.unb.br, camilodorea@unb.br.

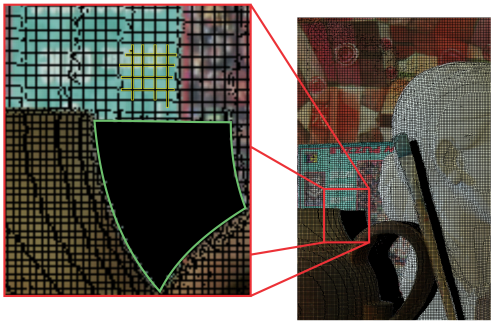

Fig. 1. Desoclusões (área interna da linha verde) e Buracos de Expansão (área interna da linha amarela)

expansão superaram o procedimento de *inpainting* empregado no VSRS 3.5 (até 6.19 dB). A representação esparsa utilizada aqui é uma modificação de uma representação não-paramétrica bayesiana para elaboração e atualização de dicionário [8].

Foi modificado o método de *inpainting* proposto para melhor ajustar o problema de preenchimento do buraco de expansão. O algoritmo é modificado para trabalhar recursivamente em resolução menores com menos buracos de expansão. Em cada iteração aplica-se o método de *inpainting*, a resolução é aumentada, os pixels preenchidos são replicados nas resoluções superiores usado, portanto em cada iteração os buracos são diminuídos.

## II. Definição do Problema

Conforme discutido anteriormente, após o DIBR, em aplicações onde exista movimento na dimensão $z$, vão existir buracos de desoclusão e expansão na imagem sintetizada. Sendo assim, definimos um buraco de expansão como: uma área espacial de superfície de um objeto na vista virtual, cuja área correspondente na vista de referência é visível, porém tem menos amostras de pixels para representar essa área. Diferente das desoclusões, os buracos de expansão são internos aos objetos. Logo, é importante identificar com precisão os buracos de expansão na vista virtual, de modo a preencher de forma diferente das desoclusões. Sendo assim, neste trabalho, foi utilizado apenas a síntese de vista virtual com movimentos de câmera $z$-dimensionais. Desta maneira é usado somente uma imagem com refêrencia, ao invés das duas comumente usadas. Para identificar os buracos de expansão, utilizamos um algoritmo já presente na literatura [4], o qual será descrito na seção seguinte.

### A. Identificação dos buracos de expansão

Foi realizado o mesmo procedimento para identificar buracos de expansão na imagem de vista virtual sintetizada com DIBR, conforme proposto em trabalhos anteriores [4], [5]. A técnica será resumida aqui, para mais detalhes são recomendados os trabalhos originais.

Na visão de referência, cada pixel $(x, y)$ é mapeado na vista virtual com a função de mapeamento $\mathcal{F}(x, y) = (x', y')$. É possível obter a função de mapeamento inverso para cada pixel na vista sintetizada para a vista de referência, isto é $\mathcal{F}'(x', y') = (x, y)$. Esses mapeamentos são realizados para a imagem de textura e mapa de profundidade de referência. A função, $\mathcal{F}'$ será usada para identificar os buracos de expansão.

O mapa de profundidade da vista sintetizada é dividido em blocos que não se sobrepõem e para cada bloco é aplicado o seguinte procedimento: O histograma com os valores de profundidade do bloco é gerado, então, todos os valores são agrupados em $k$ camadas, onde a primeira camada são os pixels que estão mais próximos da câmera. Cada camada $l_k$ é então processado onde todos os pixels do bloco das camadas mais altas são considerados como pixels vazios $p$. Figura 2 mostra um exemplo.

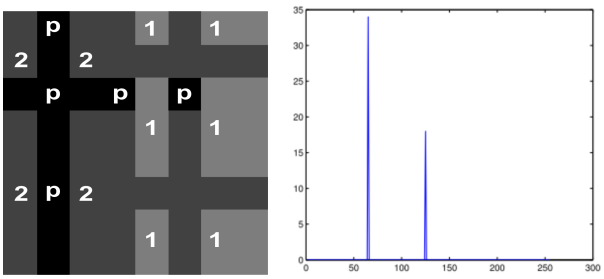


Fig. 2. Um bloco e o histograma do bloco. A camada rotulada como $l_1$ considera todos os pixels da camada $l_2$ e $p$ como pixels vazios e a camada $l_2$ considera somente os pixels marcados como $p$ como vazios

Cada pixel vazio $p$ tem quatro quadrantes dividindo sua vizinhança. Em cada quadrante, o pixel sintetizado $s$ mais próximo (não-vazio) é encontrado. A distância entre os pixels, usada para encontrar o pixel mais próximo, é medida por $H(x_p, y_p), (x_s, y_s)) = |x_p - x_s| + |y_p - y_s|$. O par do pixel vazio e seu pixel mais próximo em cada quadrante são referidos como pares vizinhos. Em seguida, é verificado se os pixels em cada quadrante são pixels próximos na visualização de referência. Especificamente, a função de mapeamento inversa $\mathcal{F}'$ é usada, e se $H(\mathcal{F}'(x'_p, y'_p), \mathcal{F}'(x'_s, y'_s)) < n$, onde $n$ é um limiar de distância predefinido, o par de pixels é considerado próximo ao pixel na vista de referência. Se dois ou mais pares vizinhos estiverem próximos de pixels na vista de referência, é declarado que este pixel vazio faz parte de um buracos de expansão. Isso é feito recursivamente para cada camada.

## III. Representação esparsa para preenchimento dos buracos

Dado um sinal $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}$ pode ter uma representação esparsa sobre um dicionário $\mathbf{D} \in \mathbb{R}^{n \times M}$, composto de $M$ colunas referidas como átomos. Este sinal pode ser encontrado com uma combinação linear de alguns átomos de $\mathbf{D}$ sendo representado aproximadamente por $\hat{\mathbf{x}} = \mathbf{D}\alpha$, onde $\alpha \in \mathbb{R}^M$ é esparso. Os buracos de expansão tendem a ter uma área menor e são geralmente espalhados por toda a imagem, fazendo um bom cenário para o uso de uma representação esparsa.

A extração de representação mais próxima e esparsa possível é procurada. Alguns métodos baseiam-se num pressuposto de que o dicionário é conhecido e fixo, enquanto em outros os dicionários **D** podem ser adquiridos com base em aprendizagem de máquina, onde o objetivo é encontrar o dicionário que produz uma representação esparsa para os dados de treinamento. Se **D** é dado, o vetor esparso $\alpha$ é estimado.

Quando o dicionário **D** for aprendido, determinar com precisão o tamanho $M$ é muito importante. Para resolver este problema, análises de fatores tem sido usado recentemente em aplicações de aprendizagem de dicionário, como a formulação bayesiana não-paramétrica [8] [10]. Esses métodos, como o processo beta (*beta process*-BP) [12], permitem inferir os valores apropriados de $M$ e $\alpha$ para o dicionário **D** e $\alpha$ (neste caso uma vista sintetizada).

### A. Formulação do processo Beta-Bernoulli

Nós queremos o modelo $\hat{\mathbf{x}} = \mathbf{D}b$ e queremos aprender **D** e impor que $b$ seja esparso. Para isso, é considerado um dicionário $\mathbf{D} \in \mathbb{R}^{n\times K}$, com $K \to \infty$ e um vetor binário $b \in \{0,1\}^K$ para indicar qual das colunas $K$ de **D** são usadas para a representação $\hat{\mathbf{x}}$ de **x**. Se um componente específico de $b$ for igual a um, então a coluna correspondente de **D** é usada na representação de **x**. Também queremos que $b$ seja esparso, onde uma pequena fração das colunas de **D** são usadas para a representação de um dado **x**.

Especificamente, suponha que temos um conjunto de treinamento, como um dicionário inicial, $\mathbf{D} = [d_1, d_2, \cdots, d_K]$ para representar $\mathbf{x}_i = \mathbf{D}b_i$, e cada coluna deste dicionário está associado a vetores binários $\{b_i\}$ e o processo Beta-Bernoulli fornece uma forma conveniente de encontrar um dicionário apropriado usando esses vetores. Este processo foi desenvolvido em [12], e pode ser representado como:

$$\begin{aligned} H(d) &= \sum_{k=1}^{K} \pi_k \delta_{d_k}(d) \\ \pi_k &\sim B(a/K, b(K-1)/K) \\ d_k &\sim H_0 \end{aligned} \tag{1}$$

$H(d)$ representa um vetor de probabilidades $\pi_k$, com cada componente associado a um respectivo átomo $d_k$, no limite $K \to \infty$, $H(d)$ corresponde a um vetor infinito de probabilidades. A expressão $\delta_{d_k}(d)$ é igual a 1 se $d = d_k$ e é zero se o valor for diferente de $d \neq d_k$. $B()$ é a função distribuição de probabilidades Beta que associa a probabilidade a um evento de escolha de átomo do dicionário. Os parâmetros $a > 0$ e $b > 0$ determinam a forma da distribuição. $H_0$ é uma base inicial.

Usando $H(d)$, podemos agora determinar $N$ vetores binários $b_i$ e o componente $k$-ésimo de $b_i$ é dado por $b_{ik} \sim Be(\pi_k)$. Onde $Be()$ é a distribuição de probabilidade de Bernoulli. Estes $N$ vetores binários são usados para constituir uma matriz $\mathbf{Z} \in \{0,1\}^{K\times N}$, com $i$-ésima coluna correspondente a $b_i$ e a linha $k$-ésima associada ao átomo $d_k$. Neste problema os átomos corresponderão aos membros candidatos para constituir o dicionário **D** e o vetor binário $b_i$ define quais membros do dicionário são usados para representar a amostra $\mathbf{x}_i$.

O problema que existe é a imposição de que $b_i$ deve ser binário [8], porém isso é muito restritivo na hora de realizar a combinação dos átomos. Pode ser desejável que os átomos selecionados sejam ponderados por um determinado peso. Assim, o vetor esparso será agora $\alpha_i = b_i \circ w_i$, onde $\circ$ representa a multiplicação elemento por elemento de dois vetores e assim o sinal é $\mathbf{x}_i = \mathbf{D}\alpha_i$. Onde $w_i$ são pesos de 0 a 1.

A base $H_0$ é usada para projetar os elementos do dicionário, definidos pelos átomos $d_k$. Os elementos atuais e posteriores do modelo podem ser gerados via análise de amostragem de Gibbs. Depois de realizar essa inferência, mantemos aquelas colunas $d$ que são usadas na representação dos dados de treinamento, criando um dicionário **D** com tamanho $M < K$. Para garantir que $\alpha$ é esparso, ajuste nos parâmetros $a$ e $b$ pode ser feita.

### B. Inpainting de Imagem

Suponha que seja dada uma imagem $\mathbf{I} \in \mathbb{R}^{N_x\times N_y}$ com pixels ausentes (*buracos de expansão*). Como é feito em [11], partimos a imagem em $N_B = (N_x - B + 1) \times (N_y - B + 1)$ blocos $\{\mathbf{x}_i\}_{i=1,N_B}$ para cada um dos quais $\mathbf{x}_i \in \mathbb{R}^{B^2}$ (neste trabalho usamos $B = 8$). Para aplicar a técnica de interpolação, em vez de observar diretamente $\mathbf{x}_i$ para montar o dicionário, observa-se um subconjunto dos pixels que não pertecem a nenhum buraco. Observe que aqui **D** e $\{\alpha_i\}_{i=1,N_B}$, que são usados para preencher os buracos de expansão são inferidos diretamente da vista sintetizada.

O nível de esparsidade é influenciado pelos parâmetros $a$ e $b$ mostrado em (1). Examinando o posterior $p(\pi_k|-) \sim Beta(a/K + \sum_{i=1}^{N} z_{ik}, b(K-1)/K + N - \sum_{i=1}^{N} z_{ik})$, condicionada a todos os outros parâmetros, foi encontrado que a maioria dos ajustes de $a$ e $b$ tendem a não ser informativos. Portanto, o nível médio de esparsidade da representação é inferido pelos próprios dados e cada amostra $\mathbf{x}_i$ tem sua própria representação esparsa, o que torna muito mais flexível do que aplicar o mesmo nível de esparsidade para cada amostra [8].

Este algoritmo de representação esparsa já foi utilizado para preenchimento de imagens, durante processo de restauração de imagens. Neste trabalho, modificados esta técnica para aplicar em buracos de expansão. Logo, não somente a aplicação é nova, mas também como a técnica é aplicada. O algoritmo descrito é executado recursivamente em diferentes resoluções. Isso não só reduzirá o tempo de execução, mas principalmente melhorará o processo de preenchimento. A imagem com a resolução original é reduzida pela metade recursivamente, realizando um mapeamento das posições de pixels. No primeiro nível $L_1$ (com a resolução mais baixa), o aprendizado de dicionário bayesiano não paramétrico anteriormente descrito é usado para interpolar os buracos de expansão presentes nesta imagem. Em seguida, um mapeamento é realizado para replicar os pixels de preenchimento na imagem do nível acima $L_2$. Os buracos que ainda existem neste nível são então

preenchidos usando o mesmo algoritmo. Isso é executado até que o último nível $L_k$ (resolução original) esteja com todos os buracos de expansão preenchidos. Desta forma, a área dos buracos a serem preenchida é sempre pequena, reduzindo o erro entre $\hat{\mathbf{x}}_i$ e $\mathbf{x}_i$.

## IV. Experimentos

Foi utilizado as imagens *Art* e *Laundry* nos conjuntos de dados disponível no *Middlebury's 2005 dataset* [13] como nossas sequências de teste de imagens multivistas. Foi usado a primeira vista, que contém imagem de textura e mapa de profundidade com uma resolução $1110 \times 1390$. Essas imagens são redimensionadas para $1104 \times 1384$ para que as linhas e colunas de pixels sejam múltiplos de $8$, o que é necessário para o algoritmo de detecção de expansões.

### A. Síntese na dimensão z

A síntese de DIBR na dimensão $z$ é realizada utilizando apenas uma imagem de referência. O parâmetro que determina a distância entre a câmera e a cena virtual é chamado $\Delta z$. O valor máximo absoluto de $\Delta z$ para cada cena é calculado por $\Delta z_{max} = 1/max(di(:))$, onde $max(di(:))$ é a maior profundidade da imagem, ou seja a distância do objeto mais próximo da câmera. Logo, pode ser escolhido um valor $\Delta z$, no intervalo $[-\Delta z_{max}, \Delta z_{max}]$. Valores negativos de $\Delta z$ representam um *zoom in* ou aproximação da cena da câmera e os valores positivos representam um *zoom out* ou distanciamento da cena da câmera. Assim, dado um pixel $p_r$ da imagem de referência com posição $(x, y)$, ele será mapeado para um novo pixel $p_v$ na posição $(x', y')$ na vista virtual. O processo de mapeamento é dado por $c(p_v) = round([x, y] - [w/2, h/2] * Z_r/Z_v + [w/2, h/2]$, onde $c(p_v)$ são as coordenadas $(x', y')$, $Z_{p_r} = 1/depth(p_r)$ e $Z_{p_v} = Z_{p_r} + \Delta z$, $w$ e $h$ são largura e a altura da imagem, respectivamente.

Nos experimentos foi considerado uma redução da distância entre a câmera e o objeto mais próximo pela metade, o que significa que a resolução espacial do objeto mais próximo pode ser aumentada em 2x. Utiliza-se essa razão para podermos comparar os resultados deste trabalho com os métodos propostos anteriormente [4] [5] [6].

Usamos a mesma metodologia de um trabalho anterior [4], usando como *ground truth*, a vista de referência fornecida em [13] chamada de $v_0$. Em seguida, usa-se o DIBR para gerar a visualização $v_r$ que é mais distante da câmera do que $v_0$. Nesta vista não haverá buracos de expansão. Usando mapas de textura e profundidade de $v_r$, aplica-se DIBR novamente para gerar a vista virtual na mesma posição que $v_0$, este processo é semelhante a um processo de *zoom-out / zoom-in* em câmeras digitais.

Quando o ponto de vista virtual está localizado mais perto da cena do que a referência de vista, os objetos mais próximos da câmera aumentarão em tamanho na visualização virtual mais do que objetos mais distantes. Um grande aumento no tamanho do objeto significa que uma parte dos pixels amostrados de uma superfície de objeto na vista de referência será espalhada para uma área espacial maior, o que resulta em buracos de expansão. Nesses pixels dispersos e seu consequente aumento de área podem bloquear outros objetos, gerando também as desoclusões. Portanto, é necessário identificar os pixels da síntese como desoclusão ou buracos de expansão, como explicado na Seção II-A. O procedimento para identificação foi conduzido com um tamanho de bloco de $B = 8$.

### B. Preenchimento dos buracos de expansão

Em seguida são preenchidos os *buracos de expansão* com um método de representação esparsa explicada na Seção III, feito de forma recursiva. Com os buracos de expansão identificados, utiliza-se o seguinte procedimento. Considere qualquer pixel $(p, j)$ onde $p, j \in [1, B]$, e deixe esse pixel constituir o pixel esquerdo-inferior em um novo bloco $B \times B$ . Então considera-se todos os blocos $B \times B$ com pixels esquerdo-inferior nas posições $p, j + B \cup p + B, j$ respeitando as dimensões da imagem. Este conjunto de blocos é denotado conjunto de dados $D_{pj}$, e considerando $1 \leqslant p \leqslant B$ e $1 \leqslant j \leqslant B$. Na primeira iteração de aprendizado de $\mathbf{D}$, é empregado os blocos em $D_{11}$, e para esta primeira rodada inicializa-se $\mathbf{D}$ e $\alpha_i$ com base em uma decomposição de valor singular (*Singular value decomposition - SVD*) dos blocos em $D_{11}$.

O dicionário $\mathbf{D}$ foi inicializado sem a presença de desoclusão e buraco de expansão. Fazemos várias iterações de Gibbs com $D_{11}$ retendo a última amostra de $\mathbf{D}$ e $\alpha_i$ da etapa anterior. Esses $\mathbf{D}$ e $\alpha_i$ são então usados a amostragem na segunda rodada, agora aplicado aos blocos $B \times B$ em $D_{11} \cup D_{21}$. A amostragem de Gibbs agora é executado nestes dados expandidos. Após várias iterações, a última amostra é retida, e o conjunto de dados é aumentado novamente. Isto é feito $B^2 = 64$ vezes até que todos os blocos são processados. Assim cada bloco da imagem tem sua representação atráves dos blocos(átomos) selecionados do dicionário e o buraco de expansão é preenchido com o valor do pixel respectivo em todos os blocos dividido pelo seu peso (probabilidade).

### C. Resultados

As imagens interpoladas resultantes são comparadas com trabalhos anteriores de [4], [5], [6]. Essas técnicas implementam respectivamente uma interpolação baseada em grafos com restrição de esparsidade (GBT), interpolação baseada em grafos com Médias Não-Locais (NLGBT) e Transformada de Fourier em grafos (AGFT).

Todas esses trabalhos relataram um valor de PSNR para um método que modificou o software VSRS versão 3.5 (chamada de VSRS+) e usam uma técnica de *inpainting* padrão para preencher todos os buracos na visualização virtual. Os valores em PSNR foram de 19.11 dB em imagem *Art* e 19.17 dB na imagem *Laundry*. Estes métodos propuseram a abordagem da camada de profundidade para identificar os buracos de expansão. Em todos os casos, o PSNR é calculado considerando somente as áreas de buracos de expansão.

Todas as imagens geradas e usadas são mostradas na Figura 3 e na Figura 4 para imagem *Art* e *Laundry* respectivamente. Em primeiro lugar é mostrada uma imagem de *ground-truth* (e

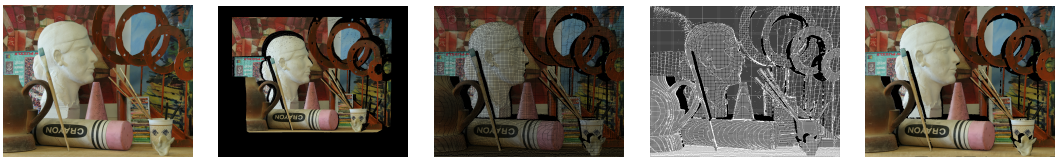

Fig. 3. Imagens para *Art*, Esquerda para direita: *Ground truth*, *zoom-out*, *zoom-in*, máscara dos buracos de expansão, vista preenchida.

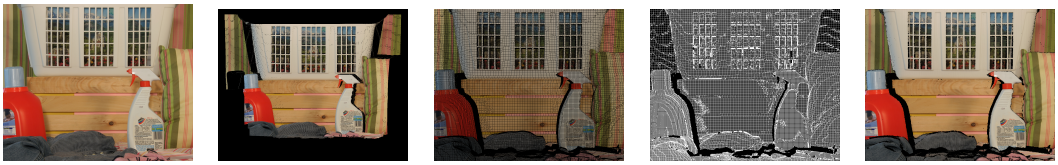

Fig. 4. Imagens para *Laundry*, Esquerda para direita: *Ground truth*, *zoom-out*, *zoom-in*, máscara dos buracos de expansão, vista preenchida.

vista de referência), as duas imagens seguintes são o processo de *zoom-out / zoom-in*, note no caso de *zoom-out* não há buracos de expansão. A última imagem mostra a imagem preenchida.

Na Figura 5 vemos uma imagem recortada de uma visão sintetizada da imagem *Art*, (a) é uma vista virtual com buracos de expansão e de desoclusão, (b) mostra a máscara onde os pixels brancos são identificados como buraco de expansão e (c) os pixels preenchidos com a representação esparsa explicada. O algoritmo recursivo foi definido para funcionar com 4 níveis.

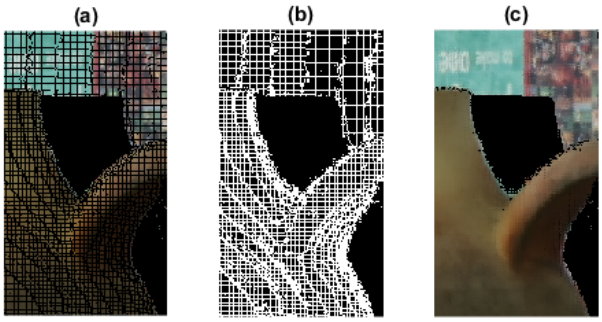


Fig. 5. (a) vista virtual,(b) máscara com buracos de expansão e (c) pixels preenchidos com representação esparsa proposta

Na Tabela I são mostrados os valores de PSNR para as imagens testadas. Nosso método superou os trabalhos anteriores para essas imagens. Para a imagem *Art* a diferença com VSRS + é muito significativa, em torno de 6.19 dB. E nosso método superou o GBT em 1.94 dB, NLGBT em 1.72 dB e AGBT em 1.61 dB

TABELA I

COMPARAÇÃO EM PSNR DOS MÉTODOS DE *inpainting*

| | GBT | NLGBT | AGBT | Proposto |
|---|---|---|---|---|
| *Art* | 23.36 | 23.58 | 23.69 | 25.30 |
| *Laundry* | 22.53 | - | 23.04 | 23.70 |

A imagem *Laundry* tem resultados semelhantes. Nossa abordagem tem um melhor desempenho e superou o VSRS + em 4.53 dB, GBT em 1.17 dB e AGBT em 0.66 dB. O algoritmo NLGBT não forneceu um valor PSNR para a imagem *Laundry*

## V. CONCLUSÕES

Neste trabalho, propusemos um método de *inpaiting* para buracos de expansão em síntese de vista DIBR com movimentos no eixo $z$. O método utiliza uma identificação consistente deste tipo de buracos que foi previamente proposto. A representação esparsa é uma modificação de também um trabalho anterior para preenchimento de buracos de expansão. Na literatura atual não há nenhum trabalho anterior que usa a representação esparsa para o preenchimento de buracos de expansão em DIBR. Além disso, a técnica de *inpainting* é realizada de forma recursiva, a fim de melhor ajustar o modelo de representação esparsa e reduzir o tempo de execução. O resultado superou o método inpainting usado no VSRS 3.5 em até 6.19 dB e também trabalhos anteriores na área.

## REFERÊNCIAS

[1] D. Tian, P. Lai, P. Lopez e C. Gomila, *View synthesis techniques for 3D video*. Proc. SPIE, 2009.

[2] M. Tanimoto e M. P. Tehrani e T. Fujii e T. Yendo, *Free-viewpoint TV*, *IEEE Signal Processing Magazine*, v. 28, 2011.

[3] C. Zhang, Z. Yin, D. Florencio, *Improving Depth Perception with Motion Parallax and Its Application in Teleconferencing*, *MMSP'09*, 2009.

[4] Y. Mao, G. Cheung, A. Ortega e Y. Ji, *Expansion hole filling in depth-image-based rendering using graph-based interpolation.*, *ICASSP*, 2013.

[5] Y. Mao, G. Cheung e Y. Ji, *Graph-based interpolation for DIBR-synthesized images with nonlocal means*, *Global Conference on Signal and Information Processing (GlobalSIP)*, 2013.

[6] Y. Mao, G. Cheung e Y. Ji, *Image interpolation for DIBR viewsynthesis using graph fourier transform*, *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2014.

[7] W. Dong, L. Zhang, R. Lukac e G. Shi, *Sparse Representation Based Image Interpolation With Nonlocal Autoregressive Modeling*, *IEEE Transactions on Image Processing*, 2013.

[8] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro e L. Carin, *Non-Parametric Bayesian Dictionary Learning for Sparse Image Representations*, *NIPS*, 2009.

[9] J. Mairal, M. Elad e G. Sapiro, *Sparse Representation for Color Image Restoration*, *Trans. Img. Proc.*, 2008.

[10] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro e L. Carin, *Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images*, *NIPS*, 2009.

[11] J. Mairal, F. Bach, J. Ponce e G. Sapiro, *Online Dictionary Learning for Sparse Coding*, *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[12] J. Paisley e L. Carin, *Nonparametric Factor Analysis with Beta Process Priors*, *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[13] 2005 Stereo datasets with ground truth ,2005,[Online] Dispnível em: http://http://vision.middlebury.edu/stereo/data/scenes2005/