Preenchimento de Buracos em Síntese de Vista baseado em Mapa de Profundidade

Ennio W. L. Silva, Bruno Macchiavello e Camilo Dorea

***Resumo*— Um sinal de vídeo 3D digital é frequentemente composto de duas ou mais vistas, onde cada vista possui um sinal de textura e um sinal de profundidade. Dentro desta representação é possível criar de forma virtual outras vistas utilizando Renderização Baseada em Imagem de Profundidade. Um problema inerente que surge neste processo é o aparecimento de buracos de desoclusões. Os buracos de desoclusões são regiões que se encontram ocultas nas vistas de referência mas devem estar aparentes na vista virtual. Neste sentido, este artigo propõe um algoritmo de *inpainting* para preenchimento destes buracos. O algoritmo proposto baseado na técnica de exemplos (*exemplar-based*) utiliza-se das características do mapa de profundidade para auxiliar o preenchimento dos buracos. Quando comparados com os algoritmos concorrentes, os resultados objetivos e subjetivos mostram que o algoritmo proposto possui um bom desempenho, obtendo um ganho de até 0.45 dB na PSNR média do vídeo comparando com o estado-da-arte.**

***Palavras-Chave*— síntese de vista, *inpainting*, mapa de profundidade.**

***Abstract*— A 3D digital video signal is frequently composed of two or more views where each view has a texture and a depth signals. Within this representation it is possible to create a virtual views using Depth Image Based Rendering. An inherent problem which arises in this process is the appearance of disocclusion holes. The disocclusion holes are regions that are hidden in reference views but should be apparent in the virtual view. Thus, this paper proposes an inpainting algorithm to fill these holes. The proposed algorithm based on exemplar-based technique uses the depth map features to assist hole filling. When compared with competing algorithms, the objective and subjective results show that the proposed algorithm has a good performance. It can yield a gain up to 0.45 dB on average PSNR compared to the state-of-the-art.**

***Keywords*— view synthesis, inpainting, depth map.**

I. Introdução

A evolução dos sistemas e tecnologias de vídeo 3D nos últimos anos é uma consequência real do empenho das produtoras de conteúdos multimídia no intuito de proporcionar uma melhor experiência visual aos usuários. O sistema 3D tradicional, também conhecido como sistema estéreo, permite que o usuário veja a cena com sensação de profundidade, mas somente a partir de um único ponto de vista. O sistema multivistas, por outro lado, é uma extensão do sistema estéreo que possui um número maior de vistas, normalmente obtido a partir de um conjunto de câmeras sincronizadas que capturam a mesma cena. Este sistema pode ser usado em diversos tipos de aplicações, tais como a Televisão 3D (3DTV) e Televisão de Ponto de Vista Livre (FTV, do inglês *Free-viewpoint Television*).

Ennio W. L. Silva, Bruno Macchiavello e Camilo Dorea Departamento de Ciência da Computação, Universidade de Brasília, Brasília-DF, Brasil, E-mails: enniowillian@gmail.com, bruno@image.unb.br, camilodorea@unb.br.

Uma alternativa para transmitir o contéudo 3D requerido nas aplicações anteriormente citadas é a utilização dos formatos vídeo+profundidade (V+D, do inglês *video-plus-depth*) e o multivista+profundidade (MVD, do inglês *multiview-plus-depth*), já que os mesmos garantem a renderização de vistas virtuais tanto para displays estereoscópicos como para displays autoestereoscópicos [1].

Um dos processos chaves que otimizam a utilização dos sistemas V+D e MDV consiste na síntese de vista, que geralmente é obtida através da Renderização Baseada em Imagem de Profundidade (DIBR - do inglês *Depth-image-based Rendering*) também chamada de deformação 3D [2]. A síntese de vista permite a criação de pontos de vistas virtuais.

Um problema inerente dos algoritmos de deformação 3D é o fato de que cada pixel não necessariamente existe em outros pontos de vista. Consequentemente, a síntese de vista pode expor as áreas da cena que estão obstruídas na vista de referência e tornar-se visível na vista virtual [3]. Estas áreas, também denominadas desoclusões ou buracos, devem ser preenchidas para melhor visualização da vista gerada. Devido a este fator, os algoritmos de deformação 3D normalmente utilizam-se de técnicas de *inpainting* para preenchimento dos buracos resultantes da síntese de vista. Este procedimento é critico quando a síntese de vista é realizada a partir de uma única imagem de referência, gerando grandes áreas de desoclução.

O conceito de *inpainting* digital foi introduzido mais formalmente por Bertalmio e Sapiro em [4], estabelecendo o termo na comunidade científica. Os autores apresentam um algoritmo baseado em equações diferenciais parciais para realizar preenchimento em imagens estáticas, onde o usuário seleciona a área a ser trabalhada, e o algoritmo recupera áreas com falhas, ou remove objetos da cena.

Uma técnica de *inpainting*, a qual permite reparar largas áreas através de reprodução das texturas, foi desenvolvida por Criminisi et al. [6]. O algoritmo proposto utiliza técnicas de síntese de texturas para reconstruir não apenas a estrutura das áreas danificadas, mas também para preencher seu conteúdo com a informação mais adequada.

Por outro lado, Daribo et al. em [3] baseado em trabalhos anteriores [7], [8] propôs uma alteração ao algoritmo de Criminisi para preenchimento de imagens resultantes do processo de síntese de vista. Em seu trabalho, o mesmo utiliza o mapa de profundidade em dois processos: cálculo de prioridade dos pixels e casamento de blocos. Gautier et al. em [9] utilizam os princípios contidos em [6], porém definindo uma nova expressão para o termo de dados. Gautier et al. realiza cálculo da prioridade dos pixels através da exploração da informação de profundidade, em primeiro lugar, definindo um tensor 3D,

e depois restringindo o lado por onde começar o *inpainting*. Ahn et al. em [10] também idealizaram uma nova forma de calcular a prioridade dos pixels pertencentes à borda baseando-se na propriedade de gradiente direcional da matriz Hessiana. Entretanto, não leva em consideração o mapa de profundidade no processo de casamento.

Neste trabalho propomos uma técnica que modifica o algoritmo de Criminisi levando em consideração a informação de profundidade de forma inovadora durante a priorização dos pixels a serem preenchidos, bem como durante o processo de casamento de blocos. As duas principais contribuições do algoritmo proposto são (i) o uso de um novo termo de relevância para modificar as prioridades dos pixels a serem preenchidos e (ii) o uso de múltiplas amostras, mediante uma combinação linear, durante o processo de preenchimento.

II. *Inpaiting* DE IMAGENS BASEADAS EM EXEMPLO

Como mencionado anteriormente, uma técnica do *inpainting* que permite reparar largas áreas através de reprodução das texturas, foi desenvolvida por Criminisi et al. em [6]. O algoritmo utiliza técnicas de síntese de texturas para reconstruir não apenas a estrutura das áreas danificadas, mas também como preencher seu conteúdo com a informação mais adequada, tendo como foco cenas do mundo real. O algoritmo considera a divisão da reconstrução da região de *inpainting* em duas fases: estrutura e textura, a diferença de trabalhos anteriores [5], a imagem original não é dividida em duas partes, mas divide a tarefa de reconstrução.

Uma das observações consideradas importante pelos autores é que uma síntese de textura que utiliza uma técnica simples de cópia baseada em exemplos é suficiente para o *inpainting* de largas áreas texturizadas, desde que a ordem de preenchimento seja correta.

O algoritmo utiliza o vetor gradiente na fronteira da região de *inpainting* (Ω) para a análise fundamental da informação nos pixels da imagem. A propagação dos pixels é uma questão de determinar qual pedaço da imagem original é mais adequado para a região falha, preservando a orientação dos gradientes calculados.

Inicialmente, para cada pixel p da região de fronteira ($\delta\Omega$) é calculado um valor de prioridade. Para realizar este cálculo são levados em consideração os vizinhos do pixel, sua direção de propagação e quantidade de contribuição (informação de cor). Esta prioridade é calculada de acordo com a Equação (1), que define a prioridade $P(p)$ como a multiplicação do termo de confiança $C(p)$ com o termo de dados $D(p)$:

$$P(p) = C(p).D(p) \tag{1}$$

$C(p)$ representa a confiabilidade da informação no pixel p, que tende a diminuir à medida que se aproxima do centro da região de *inpainting*. $D(p)$ indica a consistência entre a orientação local da região válida na fronteira com a geometria da fronteira. Assim, quanto maior esta consistência, mais o pixel na fronteira penetra na região de *inpainting*, significando que é um ponto importante na estrutura da imagem, e deve ser o primeiro a ser preenchido, de forma a preservar esta estrutura.

Criminisi et al. define o termo de confiança e o termo de dados da seguinte forma:

$$C(p) = \frac{\sum C(q)_{q\in\Psi_p\cap(I-\Omega)}}{|\Psi_p|}, D(p) = \frac{|\nabla I_p^{\perp}.n_p|}{\alpha}, \tag{2}$$

onde $|\Psi_p|$ é a área de Ψ_p, α é um fator normalizador (por exemplo, $\alpha = 255$ para uma típica imagem em tons de cinza), n_p é um vetor unitário ortogonal à $\delta\Omega$ no ponto p e $\perp$ denota o operador ortogonal, conforme ilustra a Figura 1.

Para evitar possíveis problemas de borramento, a propagação da textura da imagem é feita por amostragem direta da região de origem. Para isso uma busca pelo padrão de textura mais próximo (casamento) é realizada. Um bloco (Ψ_p) é determinado em volta do pixel p, considerando unicamente os pixels preenchidos dentro de Ψ_p é feita uma busca exaustiva sobre a região da imagem já preenchida (Φ). Obtido o bloco mais próximo (Ψ_q), mediante SSD - *sum of squared differences*, esses valores são usados para preencher o buraco de Ψ_p.

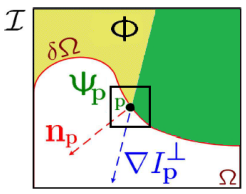


Fig. 1. Diagrama de notação de Criminisi sobre uma região de imagem já preenchida e a região de *inpainting*[6]

III. METODOLOGIA PROPOSTA

O algoritmo de *inpainting* proposto neste trabalho é dividido em quantro etapas: (A) Detecção de bordas, (B) cálculo da prioridade, (C) casamento de padrões e (D) preenchimento. O algoritmo possui como entradas a imagem virtual, seu respectivo mapa de profundidade, bem como a máscara que denota os buracos de desoclução.

A. *Detecção da borda*

No algoritmo base [6], a detecção da borda é feita ao longo de todo o buraco a ser preenchido. Porém, a imagen virtual gerada através da síntese de uma única imagem de referência possui a seguinte peculiaridade:

- Deformações 3D que utilizam somente a câmera de referência à esquerda geram buracos do lado esquerdo dos objetos que se encontram no primeiro plano da imagem ou *foreground*.
- Deformações 3D que utilizam somente a câmera de referência à direita geram buracos do lado direito dos objetos que se encontram no primeiro plano da imagem ou *foreground*.

Levando-se em consideração a peculiaridade dos buracos das imagens virtuais geradas e a necessidade de priorizar os elementos do *background* no preenchimento destes, percebe-se que há um sentido de preenchimento a ser adotado para a obtenção de um resultado mais adequado.

Tendo em vista esta singularidade relacionado ao sentido do preenchimento dos buracos, não há fundamento para o uso da borda inteira do buraco, já que a propagação dos pixels deve seguir somente um sentido. Portanto, para a detecção da borda neste trabalho foi utilizada a ideia proposta em [10], que detecta a borda somente do lado oposto aos objetos do primeiro plano. Logo, a borda ($\delta\Omega'$) é definida segundo a Equação (3);

$$\delta\Omega' = (Mascara \oplus E_1) - Mascara, \tag{3}$$

onde $\oplus$ representa a operação de dilatação e E_1 é um elemento estruturante definido como se segue:

$E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ se a deformação for para esquerda e $E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ se a deformação for para a direita.

B. Cálculo da prioridade

Uma vez detectada a borda de onde se propagará o preenchimento do buraco, faz-se necessário o cálculo da prioridade de cada um os pixels desta borda para verificar qual bloco tem a preferência na propagação.

No algoritmo de Criminisi o termo de confiança ($C(p)$) tem seus valores inicializados de tal forma que todos os componentes da região de busca assumem o valor 1. Todavia essa forma de inicialização pode causar distorções no resultado do preenchimento. É importante ressaltar que o termo de confiança dos pixels vizinhos deve ser maior do que os pixels que estão mais distantes do buraco para evitar distorções no preenchimento. Dado este fato o algoritmo proposto utiliza a ideia de inicialização empregada em [10] na qual os pixels mais próximos a borda detectada possuem maior confiança:

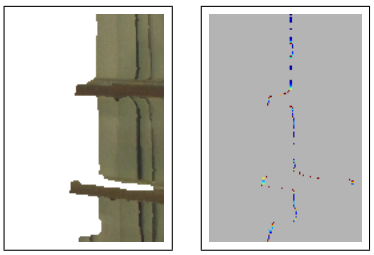
1) Inicialização: $h_0 = \delta\Omega'$
2) Repetir: $h_{k+1} = (h_k \oplus E_1) \cap (\neg Mascara)$ até k = floor(T_1/2),

onde $T_1 \in \mathbb{Z}$ limita o número de iterações e $\neg$ é a operação de negação.

Analisando o termo de dados ($D(p)$) propostos em Criminisi et al. [6], verifica-se que o mesmo está exposto à ruídos na predição da direção da estrutura da imagem, uma vez que o operador gradiente é aplicado na imagem sem nenhum pré-processamento afetando o preenchimento do buraco.

Uma vez que o termo de dados é utilizado como meio de refletir a estrutura da imagem, e sabendo que esta última é evidenciada pelas bordas e descontinuidades da imagem em si, neste trabalho é proposto a substituição do termo de dados por um termo de relevância ($R(p)$). Esse termo de relevância é baseado no termo utilizado em trabalho anterior de *inpaiting* de imagens não especifico à síntese de vista [13].

Diferentemente das abordagens tradicionais, o algoritmo proposto computa a direção dos pixels com base em uma imagem auxiliar, contendo informações de borda, e o mapa de profundidade realçado. A imagem auxiliar é obtida através da difusão anisotrópica proposta por Perona e Malik em [12]. O mapa de profundidade tem as bordas realçadas através do filtro *Unsharp Masking* para enfatizá-las no cálculo do termo de relevância.



(a) (b)

Fig. 2. Efeito do Termo de Relevância em regiões de borda (a) Parte da imagem contendo região preenchida e região de *inpainting* (b)Termo de relevância calculado indicando o realce das bordas

O novo cálculo de prioridade baseado no termo de relevância proposto é dado por:

$$P(p) = C(p).R(p)\ . \tag{4}$$

$R(p)$ é definido como:

$$R(p) = \left|\nabla(\Delta u_p).\nabla(m_p).\overrightarrow{d_p}\right| \tag{5}$$

onde ∇ é operador gradiente, Δ é o operador laplaciano, u é a imagem auxiliar obtida com a difusão anisotrópica, m é o mapa de profundidade realçado, e $\overrightarrow{d_p}$ é o vetor ortogonal ao gradiente definido como $\overrightarrow{d_p} = \frac{\nabla^{\perp} u_p}{|\nabla^{\perp} u_p|}$. A Figura 2 mostra o efeito do termo de relevância proposto neste trabalho.

C. Casamento de Padrões

O casamento consiste na procura por blocos similares ao bloco centrado no pixel p. Este processo em suas abordagens clássicas como em [6] tem sido realizado através de funções de custo tais como SAD (*Sum of Absolute Differences*) ou SSD (*Sum od Square Differences*) nos três canais de cores das imagens (R,G,B). Entretando, as cores em si não estão relacionadas diretamente com a estrutura da imagem, o que é uma das principais preocupações no processo de *inpainting*. Por outro lado, os mapas de profundidades trazem consigo uma distinção relevante entre diferentes partes da imagem a ser preenchida. Aproveitando essa informação adicional uma proposta anterior [3] obtém o bloco Ψ_q a partir da seguinte expressão:

$$\Psi_q = \arg\min_{\Psi \in \Phi} (d(\Psi_p, \Psi) + \beta.d^m(\Psi_p, \Psi)) \tag{6}$$

onde d é a distância SSD calculada para a imagem RGB, d^m é distância SSD calculada para o mapa de profundidade e β é um número inteiro maior que 1.

No método proposto neste trabalho, a informação adicional dos mapas de profundidade é também utilizada porém de forma distinta: k candidatos são obtidos Ψ_{q1},Ψ_{q2},...,Ψ_{qk} e posteriormente combinados linearmente. Os candidatos são os k blocos com menor distância segundo a Equação (6). Em nossa implementação $k = 3$ e $\beta = 3$.

D. Preenchimento

Dado os k candidatos definidos na Seção III-C são combinados para gerar a informação de preenchimento da seguinte forma:

$$\Psi'_p = (w_1\Psi_{q1}) + (w_2\Psi_{q2}) + ... + (w_k\Psi_{qk}); \quad (7)$$

onde Ψ'_p é a região a ser preenchidad de Ψ_p e w_k são pesos proporcionais à distância obtidos da seguinte forma:

$$w_i = \frac{(\frac{1}{d_i})}{(\frac{1}{d_1}) + (\frac{1}{d_2}) + ... + (\frac{1}{d_k})} \quad (8)$$

onde d_i é as distâncias do candidato Ψ_{qi} em relação o bloco referência Ψ_p.

IV. Resultados Experimentais

O algoritmo proposto foi implementado na plataforma Matlab®. Para os testes foram utilizadas as sequências de vídeo multivistas+profundidade denominadas *Ballet* (100 *frames*) e *Breakdancers* (100 *frames*) disponibilizadas pela Microsoft [11]. Estas sequências possuem 1024×768 pixels de resolução, sendo que as mesmas foram capturadas a partir de 8 câmeras. Para as duas sequências de vídeo foram geradas vistas virtuais da câmera 2 e 4 a partir da câmera 5 (BA54:Ballet/v5-v4, BA52:Ballet/v5-v2, BR54:Breakdancing/v5-v4, BR52:Breakdancing/v5-v2). As Figuras 3(a) e 3(b) ilustram um exemplo do resultado da deformação das vistas virtuais geradas.

Para avaliar a performance do algoritmo proposto, foram utilizadas duas métricas objetivas: PSNR e SSIM [14]. Os valores médios destas duas métricas são expressos na Tabela 1 e Tabela 2, onde são comparados o valores obtidos no algoritmo proposto e outros quatro métodos definidos na literatura [6][3][9][10]. Os valores em negrito denotam o melhor resultado obtido. É possível observar que em 2 dos 4 testes o algoritmo proposto superou objetivamente tanto em PSNR como em SSIM os algoritmos anteriores que representam o estado-da-arte. Na sequência de teste BA54 o algoritmo proposto tem o maior valor de SSIM, e o segundo maior valor de PSNR. Já a sequência BR52 foi onde o algoritmo proposto proporcionou os piores resultados. O motivo pelo qual o algoritmo não teve um bom desempenho para o sequência BR52 é porque os objetos na sequência *Breakdancers* possuem todos profundidades muito próximas, e quando a distância entre a câmera virtual e a câmera de referência aumentam, a geração do termo de relevância deixa de ser muito confiável. A Figura 3 apresenta alguns resultados do preenchimento dos buracos do algoritmo proposto e dos demais métodos para uma análise subjetiva. São mostrados um quadro de duas sequências, uma das quais inclusive onde nosso algoritmo não é superior na media de PSNR do vídeo (BA54). Neste caso, apesar de pior PSNR existe uma melhora subjetiva significativa que nosso algoritmo pode alcançar.

TABELA I
RESULTADOS UTILIZANDO PSNR [DB]

Seq	[6]	[3]	[9]	[10]	Proposto
BA54	27.09	28.91	28.70	**32.43**	32.12
BA52	25.29	24.96	24.51	25.78	**26.23**
BR54	29.46	24.71	28.59	29.52	**29.98**
BR52	26.82	26.82	26.26	**26.96**	25.42

TABELA II
RESULTADOS UTILIZANDO SSIM

Seq	[6]	[3]	[9]	[10]	Proposto
BA54	0.9024	0.9040	0.9130	0.9369	**0.9439**
BA52	0.7759	0.7686	0.7654	0.7858	**0.8197**
BR54	0.9022	0.8430	0.8891	0.9136	**0.9304**
BR52	0.8455	0.8333	0.8411	**0.8583**	0.8329

V. Conclusões

Neste artigo apresentamos um algoritmo para preenchimento de buracos em imagens resultantes do processo de síntese de vista. O algoritmo é baseado na técnica de exemplos. Através da utilização do mapa de profundidade no cálculo de prioridade e casamento de blocos, com a combinação linear de blocos, os buracos são preenchidos de tal forma que a estrutura da imagem como um todo é preservada, trazendo consigo um maior conforto visual. O algoritmo apresenta duas novidades com relação aos algoritmos já presentes na literatura, o uso do termo de relevância e o uso da combinação linear de blocos. Os resultados experimentais mostram que a nossa proposta pode alcançar um desempenho superior à diversos algoritmos concorrentes da literatura objetivamente e subjetivamente.

Referências

[1] Suryanarayana M. Muddala, Roger Olsson, and Marten Sjostrom, *Disocclusion handling using depth-based inpainting*. In The Fifth International Conferences on Advances in Multimedia, pages 136-141. IARIA, 2013.

[2] C. Fehn, *Depth-image-based rendering (dibr), compression, and trasmission for a new approach on 3d-tv*. In Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XI, pages 93-104, 2004.

[3] I. Daribo and B. Pesquet-Popescu, *Depth-aided image inpaiting for novel view synthesis*. In Multimedia Signal Processing (MMSP), pages 167-170. IEEE, 2010.

[4] M. Bertalmio et al, *Image inpainting*. In ACM Computer Graphics Proceedings (SIGGRAPH), pages 417-424, 2000.

[5] M. Bertalmio et al, *Simultaneous structure and texture image inpainting*. In IEEE Transactions on Image Processing, pages 882-889, 2003.

[6] P. Perez Crimisi and K. Toyama. Region filling and object removal by exemplarbased image inpainting. In IEEE Transactions on Image Processing. IEEE, 2004.

[7] S. et al C.M. Cheng,*Improved novel view synthesis from depth image with large baseline*.In International Conference on Pattern Recognition (ICPR), pages 1-4, 2008.

[8] S. K.J. Oh and Y.S. Ho, *Hole filling method using depth based inpainting for view synthesis in free viewpoint television and 3-d video*. In Proc. of the Picture Coding Symposium (PCS), pages 1-4, 2009.

[9] Josselin Gautier et al, Depth-based image completion for view synthesis. In The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), pages 1-4,2011.

[10] Ilkoo Ahn and Changick Kim, *Depth-based Disocclusion Filling for Virtual View Synthesis*

[11] Sequence microsoft ballet and breakdancers,2004,[Online] Available: http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload/.

[12] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*. In IEEE Transactions on Image Processing, pages 629-639, 1990.

[13] Wallace Casaca et al,*Combining anisotropic diffusion, transport equation and texture synthesis for inpainting textured images*. Pattern Recognition Letters, pages 36-45, 2014.

[14] Wang Zhou, Bovik, Alan C., Sheikh, Hamid R., and Simoncelli, Eero P. *Image Qualifty Assessment: From Error Visibility to Structural Similarity*. In IEEE Transactions on Image Processing, Volume 13, Issue 4, pp. 600-612, 2004.

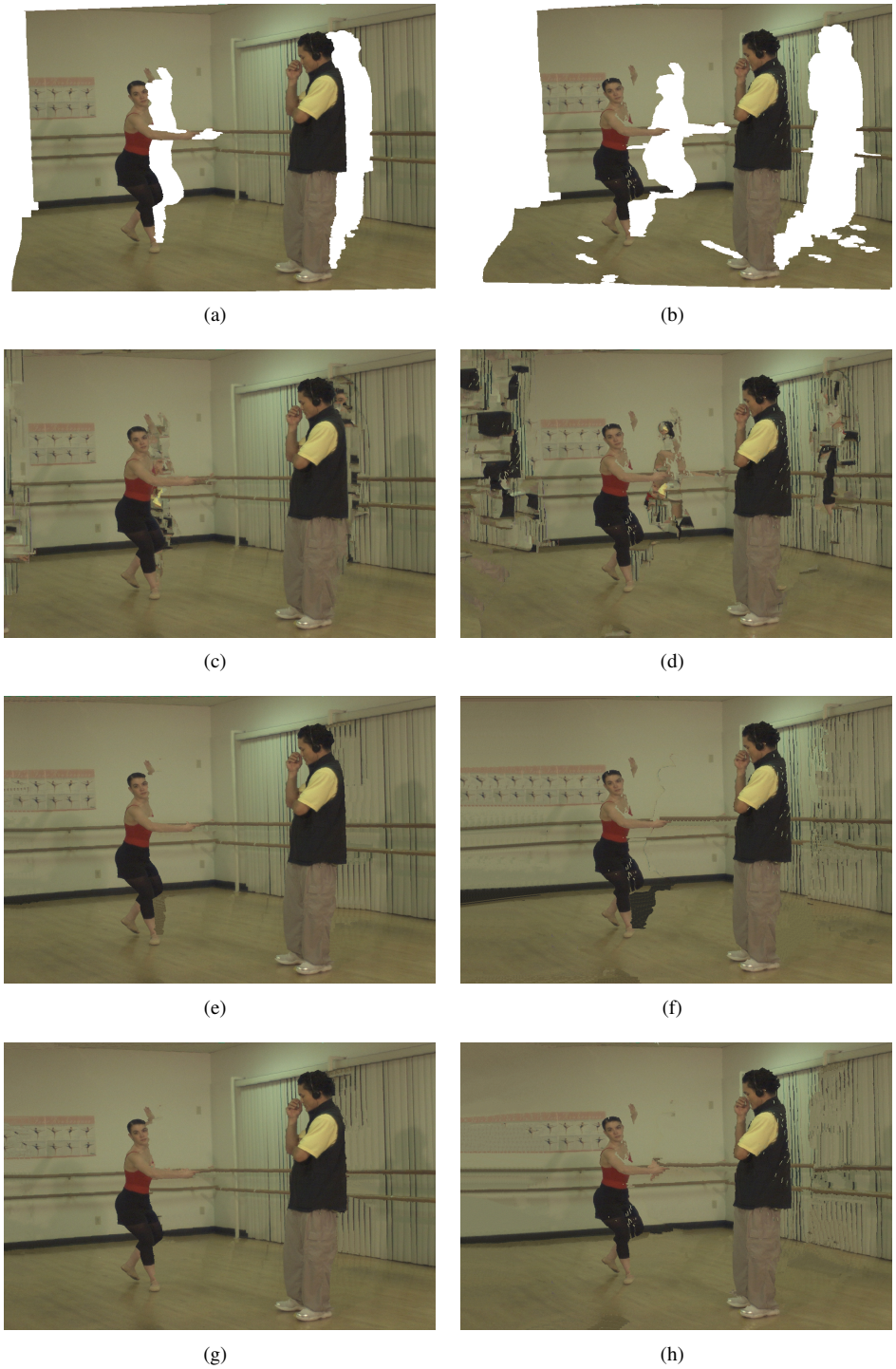


Fig. 3. Resultados subjetivos do algoritmo proposto e demais algoritmos da literatura. (a) BA54 : Síntese de vista (b)BA52 : Síntese de vista (c) BA54: *Inpainting* com método de Criminisi [6] (d) BA52: *Inpainting* com método de Criminisi [6] (e) BA54: *Inpainting* com método de Ahn Ilkoo [10] (f) BA52: *Inpainting* com método de Ahn Ilkoo [10] (g) BA54: *Inpainting* com método proposto (h) BA52: *Inpainting* com método proposto