UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ADRIANO QUILIÃO DE OLIVEIRA

Síntese de Vistas com Depth-Image-Based Rendering (DIBR)

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Marcelo Walter
Co-orientador: Prof. Dr. Cláudio Rosito Jung

Porto Alegre
2016

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

de Oliveira, Adriano Quilião

Síntese de Vistas com Depth-Image-Based Rendering (DIBR) / Adriano Quilião de Oliveira. – Porto Alegre: PPGC da UFRGS, 2016.

72 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2016. Orientador: Marcelo Walter; Co-orientador: Cláudio Rosito Jung.

1. DIBR. 2. Hole filling. 3. Síntese de vistas. 4. FTV. 5. TV3D. I. Walter, Marcelo. II. Jung, Cláudio Rosito. III. Título.

“Sometimes it’s the very people who no one imagines anything of who do the things that no one can imagine.”

— Sir Alan Mathison Turing

AGRADECIMENTOS

Primeiramente, agradeço aos meus orientadores, Profs. Marcelo Walter e Cláudio Jung. Obrigado por acreditar em mim, e possibilitar esta excelente oportunidade de aprendizado. Não tenho palavras para agradecer-lhes por toda paciência, disponibilidade e conhecimento compartilhado. Muito obrigado!

Agradeço aos meus pais, Carlos e Elke, por me apoiar incansavelmente em cada nova etapa da minha vida. Obrigado por toda amizade e pela participação ativa na luta pelos meus sonhos, vocês são o meu exemplo e também a razão de tanto esforço e dedicação. Ao meu irmão Rafael, por um dia me fazer acreditar que não existem problemas insolúveis, e que aquele computador poderia ser algo tão fascinante. Tratando-se da minha família, não poderia deixar de agradecer às minhas tias, tios e primos, os quais são parte fundamental da minha vida. À minha namorada, Drika, por me aguentar e apoiar durante o desenvolvimento de parte desta pesquisa.

Adicionalmente, gostaria de agradecer aos meus amigos de POA, principalmente ao Marcelo e o Arthur, pelas incansáveis correções, auxílio e aprendizado durante este período (muito obrigado mesmo!). Também, aos amigos malucos de Cachoeira, principalmente aos de maior convívio, Bohrdy, Mijado, Jozé, Troj, Roni, Puff, Atos, Panxo, Rafa, Razzera e Dedeh, meu muito obrigado por todos os churrascos, conversas e pelo apoio incondicional. Por fim, mas não menos importante, agradeço aos amigos de Alegrete, Rafa e Robson, a amizade e apoio de vocês foi muito importante.

Por fim, agradeço aos colegas da UCS, em particular ao André e a Patricia, por todos os conselhos, conversas e chopes. Agradeço também aos colegas da Moveideias, em especial ao Tiago, Renan e Everton por proporcionarem um excelente ambiente de trabalho e pelo incentivo durante o desenvolvimento desta dissertação.

RESUMO

Esta dissertação investiga soluções para o problema genérico de geração de vistas sintéticas a partir de um conjunto de imagens utilizando a abordagem *Depth-Image-Based Rendering*. Essa abordagem utiliza um formato compacto para a representação de imagens 3D, composto basicamente por duas imagens, uma colorida para a vista de referência e outra em tons de cinza com a correspondência de disparidade para cada pixel. Soluções para esse problema beneficiam aplicações como *Free Viewpoint Television*. O maior desafio é o preenchimento de regiões sem informação de projeção considerando o novo ponto de vista, genericamente denominados *holes*, além de outros artefatos como *cracks* e *ghosts* que ocorrem por oclusões e erros no mapa de disparidade. Nesta dissertação apresentamos técnicas para remoção e tratamento de cada uma das classes de potenciais artefatos. O conjunto de métodos propostos apresenta melhores resultados quando comparado com o atual estado da arte em geração de vistas sintéticas com o modelo DIBR para o conjunto de dados *Middlebury*, considerando-se as métricas SSIM e PSNR.

Palavras-chave: DIBR. hole filling. síntese de vistas. FTV. TV3D.

View Synthesis with Depth-Image-Based Rendering (DIBR)

ABSTRACT

This dissertation investigates solutions to the general problem of generating synthetic views from a set of images using the Depth-Image-Based Rendering approach. This approach uses a compact format for the 3D image representation, composed basically of two images, one color image for the reference view and other grayscale image with the disparity information available for each pixel. Solutions to this problem benefit applications such as Free Viewpoint Television. The biggest challenge is filling in regions without projection information considering the new viewpoint, usually called holes, and other artifacts such as cracks and ghosts that occur due to occlusions and errors in the disparity map. In this dissertation we present techniques for removal and treatment of each of these classes of potential artifacts. The set of proposed methods shows improved results when compared to the current state of the art generation of synthetic views using the DIBR model applied to the Middlebury dataset, considering the SSIM and PSNR metrics.

Keywords: DIBR, hole filling, view synthesis, FTV, 3DTV.

LISTA DE ABREVIATURAS E SIGLAS

DIBR	Depth-Image-Based Rendering
FTV	Free-Viewpoint Television
TV3D	Televisão 3D
SSD	Sum of Squared Differences
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index
RGB	Red, Green, Blue
V+D	Video plus Depth
GPU	Graphics Processing Unit
CUDA	Compute Unified Device Architecture

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação

Com a recente popularização de *displays* estéreo em dispositivos como televisores, celulares e tablets, uma maior quantidade de conteúdo utilizando múltiplas câmeras e sensores de profundidade tem sido gerada. Em paralelo, aplicações para apreciação deste conteúdo, como televisão 3D (TV3D) e *Free-Viewpoint Television* (FTV), tem alavancado pesquisas, por parte da academia e da indústria, relacionadas com a sua viabilidade técnica. Estas aplicações tem como objetivo fornecer para o usuário um ambiente de visualização mais interativo e realista como é o caso, por exemplo, da FTV que visa proporcionar ao usuário a livre navegação e seleção do ponto de vista de uma cena. Contudo, utilizando as atuais tecnologias é necessário capturar e transmitir um grande número de vistas simultaneamente.

Porém, a transmissão simultânea de um grande volume de imagens facilmente extrapola os limites disponíveis de largura de banda da infraestrutura física existente. Neste contexto, Fehn (2004) propôs um modelo denominado *Depth-Image-Based Rendering* (DIBR), o qual possibilita a renderização de múltiplos pontos de vista utilizando apenas uma imagem de referência e seu respectivo mapa de disparidades. O processo consiste basicamente em utilizar informação de disparidade[1] para projetar a imagem colorida de referência para um ponto de vista virtual.

A qualidade do mapa de disparidades impacta diretamente no processo de geração de vistas sintéticas. Normalmente, estes mapas são gerados a partir de um par de imagens estéreo retificadas, utilizando técnicas de *stereo matching*. O objetivo de tais técnicas é detectar para cada linha a correspondência ponto a ponto entre as duas imagens (SCHARSTEIN; SZELISKI, 2002). Algoritmos de *stereo matching* tratam de problemas difíceis, como a não correspondência de pontos devido a oclusões e a detecção de similaridade em grandes regiões homogêneas e, portanto, costumam apresentar pequenas incoerências de estimativa de disparidade. Outra forma de obter o mapa de disparidades é através de sensores de profundidade (por exemplo, Kinect), contudo estes costumam apresentar menor precisão. Dessa maneira, é comum encontrar ruído e imprecisões em mapas gerados com ambas as técnicas, o que causa erros de projeção no processo de formação

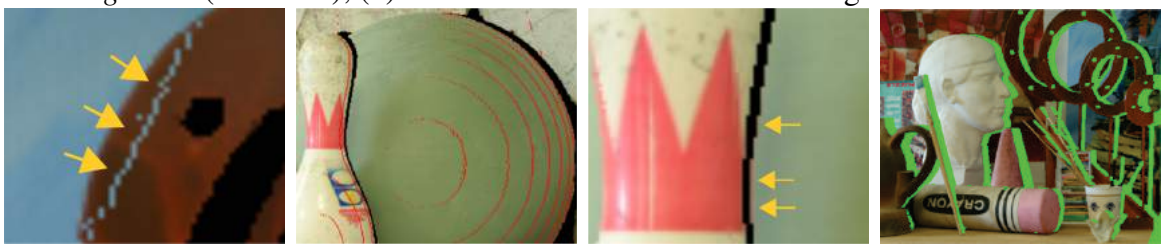
[1] Nesta dissertação a palavra disparidade refere-se ao deslocamento em número de *pixels* entre dois pontos correspondentes em duas imagens estéreo retificadas, obtidas por duas câmeras idênticas simultaneamente.

das imagens sintéticas.

Artefatos decorrentes de problemas no mapa de disparidades são frequentes na geração de vistas sintéticas. Estes podem ser separados basicamente em duas classes *cracks* e *ghosts*. A primeira é decorrente em problemas de quantização das disparidades, e pode apresentar-se na forma translúcida, onde são preenchidos com informação do objeto de *background* (Figura 1.1(a)) ou vazia, quando não possuem nenhuma informação de projeção (Figura 1.1(b)). A outra classe, os *ghosts*, ocorrem quando a disparidade não está bem definida no domínio da imagem, mantendo a silhueta de objetos do *foreground* no *background* da imagem projetada, como pode ser visto na Figura 1.1(c).

Os *holes* são regiões sem informação de projeção na vista sintética, e ocorrem principalmente quando existe a sobreposição de objetos na cena, como pode ser observado na Figura 1.1(d). Isto ocorre porque a vista sintética é gerada em um ponto de vista diferente do utilizado para obter a imagem de referência, e algumas regiões oclusas neste ponto podem ser visualizadas na imagem projetada (PURICA et al., 2015). Desta forma, sempre que novos pontos de vista forem gerados, e houver a sobreposição de objetos em cena, estas áreas sem informação devem aparecer. O tamanho destas áreas varia de acordo com o deslocamento entre a vista de origem e o ponto no qual a imagem sintética é gerada. Portanto, o desafio nesta etapa consiste em estimar corretamente a informação dessas áreas e preenche-las de maneira coerente.

Figura 1.1: Artefatos e problemas encontrados na geração de imagens sintéticas com o modelo DIBR. (a) *Crack* translúcido em azul, indicado por setas amarelas; (b) *Cracks* vazios destacados em vermelho; (c) *Ghost* indicado por setas amarelas na borda do objeto do *background* (em verde); (d) *Holes* destacados em verde na imagem.



(a) (b) (c) (d)

Fonte: Adaptado de (MIDDLEBURY, 2016).

Uma das principais razões que justificam a investigação do referido problema consiste no fato de que, mesmo existindo inúmeras abordagens para a geração de imagens sintéticas com o modelo DIBR, estas geralmente desconsideram os limites de banda disponíveis atualmente para transmissão televisiva. Isso ocorre pois tais abordagens utilizam múltiplas imagens no processo de renderização como, por exemplo, (ZINGER; DO;

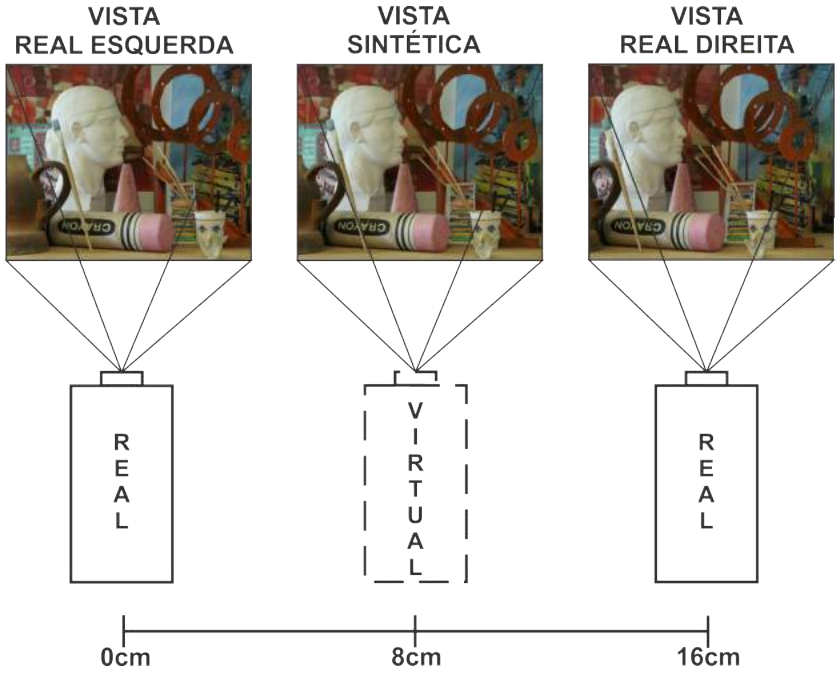
WITH, 2010) e (MORI et al., 2008). Dessa forma, estas podem se mostrar infactíveis para serem aplicadas na prática. Alternativamente, é possível gerar imagens sintéticas com boa qualidade para pontos de vista arbitrários tendo com base apenas uma imagem de referência e seu respectivo mapa de disparidades como, por exemplo, nos trabalhos de (SOLH; ALREGIB, 2012) e (GAUTIER; MEUR; GUILLEMOT, 2011). Este modelo adicionaria apenas uma pequena sobrecarga de transmissão (abaixo de 10-20% da taxa de bits do vídeo) em comparação com a TV digital 2D convencional (FEHN, 2004). Ainda, é importante destacar que por utilizar apenas uma imagem colorida, esta abordagem não recai em problemas como diferenças de contraste, brilho e cor das imagens de dois ou mais pontos de vista, que poderiam prejudicar a sensação estereoscópica do usuário. No entanto, para realizar o processo de renderização é preciso criar soluções para problemas que não estão resolvidos em sua completude, como o preenchimento de *holes* e a remoção e tratamento de artefatos.

1.2 Objetivos

1.2.1 Objetivos Gerais

A presente dissertação propõe métodos para todos os passos que compõem a geração de imagens sintéticas com o modelo DIBR, criando diferentes pontos de vista de uma mesma cena, ao longo de um segmento de reta horizontal que conecta as duas câmeras reais. Cabe salientar que neste trabalho não são exploradas projeções verticais. Adicionalmente, observa-se que o limite do campo de visão horizontal permitido para aplicações reais é delimitado pela disponibilidade de vistas reais que auxiliem na geração das imagens sintéticas. Desta forma, utilizando os métodos desenvolvidos neste trabalho deve ser possível permitir a um usuário a variação de escolha do ponto de vista de uma cena, considerando apenas deslocamentos horizontais. A Figura 1.2 representa a escolha de um ponto de vista virtual (conforme pode ser determinado por um usuário), gerado entre dois pontos de vista reais. No modelo desta figura, aplicam-se técnicas de *stereo matching* para estimar os mapas de disparidades nas imagens reais retificadas, onde um destes é utilizado para projetar sua respectiva imagem para o ponto de vista virtual. Para a exibição final da vista, a imagem gerada é processada para remoção de artefatos e preenchimento de regiões sem informação.

Figura 1.2: Par de câmeras reais ilustrando o *setup* empregado na geração de vistas sintéticas utilizado no modelo DIBR. A vista central (virtual) pode ser obtida pela projeção e pós-processamento de qualquer uma das duas imagens reais vizinhas ao ponto de vista virtual.



Fonte: O Autor.

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho são descritos de forma resumida nos itens a seguir:

- Desenvolvimento de um método para a detecção de *cracks* vazios;
- Desenvolvimento de um método para a detecção de *cracks* translúcidos;
- Avaliação de diferentes algoritmos para o preenchimento dos *cracks*;
- Desenvolvimento de um método para a detecção e tratamento dos *ghosts*;
- Proposta de um novo método de *inpainting* baseado em profundidade para o preenchimento dos *holes*.

1.3 Organização do Trabalho

O restante desta dissertação está distribuído como segue. No Capítulo 2, apresenta-se a revisão teórica e o atual estado da arte para métodos de renderização de imagens sintéticas com o modelo DIBR. Posteriormente, no Capítulo 3, são apresentados os algo-

ritmos propostos neste trabalho. O Capítulo 4 exibe os resultados obtidos com as técnicas propostas, comparando-os com os principais métodos apresentados na literatura. Por fim, no Capítulo 5, são apresentadas as considerações finais sobre esta dissertação, com indicações de trabalhos futuros.

2 CONCEITOS BÁSICOS E REVISÃO BIBLIOGRÁFICA

Este capítulo tem por objetivo descrever alguns conceitos básicos para a compreensão do tema estudado, além de apresentar uma visão geral do atual estado da arte para a geração de vistas sintéticas com o modelo DIBR. Desta forma, a primeira seção fornece a fundamentação teórica necessária para a compreensão do trabalho desenvolvido. Após, na Seção 2.2, são discutidos os principais trabalhos propostos para a renderização de vistas sintéticas utilizando o modelo estudado.

2.1 Fundamentação Teórica

Para a devida compreensão dos assuntos abordados nesta dissertação, é necessário, inicialmente, a definição de alguns conceitos chave. Desta forma, na Subseção 2.1.1, apresenta-se a metodologia utilizada para a aquisição e representação da profundidade das imagens. Na Subseção 2.1.2, descrevem-se as etapas e problemas encontrados na geração de vistas sintéticas com o modelo DIBR. Por fim, na Subseção 2.1.3, apresenta-se uma visão geral sobre algoritmos de *inpainting* e uma descrição detalhada da técnica desenvolvida por (CRIMINISI; PEREZ; TOYAMA, 2004), considerando-se que esta serve como base para os métodos desenvolvidos nesta dissertação.

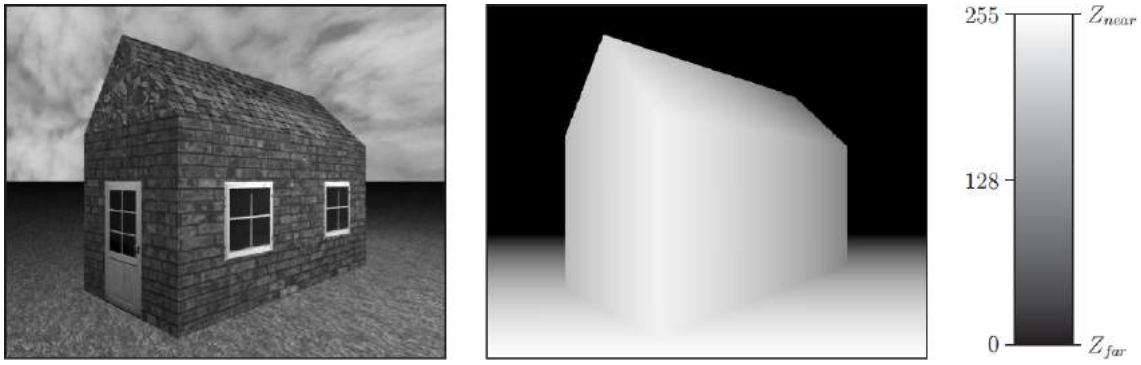
2.1.1 Aquisição da Profundidade

A informação de profundidade de uma cena possui diversas aplicações, tanto em visão computacional quanto em computação gráfica. Esta informação pode ser representada de uma maneira bastante simples, utilizando uma imagem em escala de cinza (mapa de profundidades), na qual cada *pixel* da imagem colorida é representado por um valor de intensidade, correspondente à profundidade. Contudo, devido a limitação para representação dos valores reais de profundidade em 8 bits, estes são normalizados em um intervalo entre 0 e 255. Desta forma, para obter os valores reais de profundidade do mapa, é necessário aplicar em cada ponto da imagem a seguinte equação:

$$Z = Z_{far} + v\frac{Z_{near} - Z_{far}}{255}, \tag{2.1}$$

onde Z_{near} e Z_{far} são, respectivamente, o maior e o menor valor de profundidade real na cena, e v indica o valor em escala de cinza correspondente a profundidade (conforme ilustrado na Figura 2.1).

Figura 2.1: Representação de uma imagem em tons de cinza acompanhada do seu mapa de profundidades.



Fonte: Retirado de (FEHN, 2004).

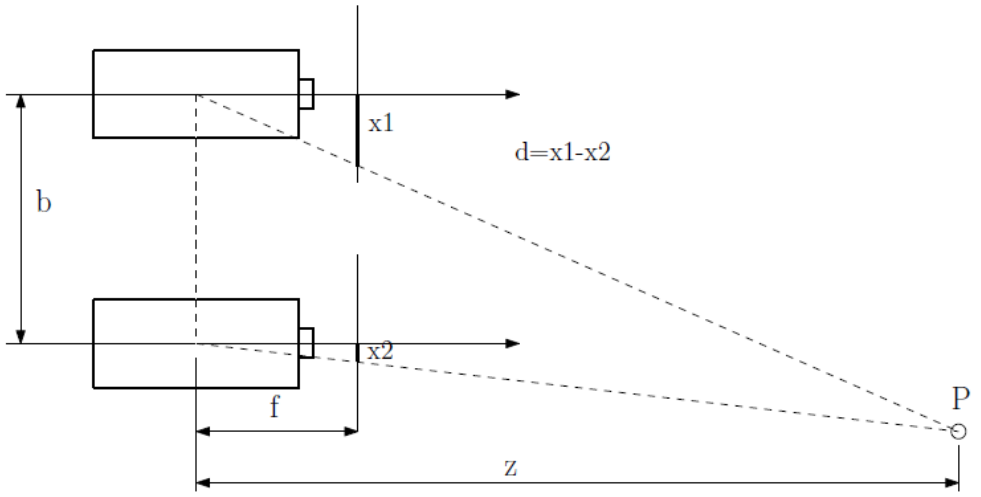
Um dos meios mais comuns para a aquisição da informação de profundidade é através do uso de técnicas de *stereo matching*. A partir de um par de câmeras estéreo retificadas, tal procedimento consiste em detectar a correspondência ponto a ponto (na mesma linha horizontal) nas imagens obtidas. A relação de correspondências entre cada ponto das imagens forma o mapa de disparidades, o qual denota o deslocamento d de cada ponto $x1$ até o seu similar $x2$ na outra imagem. Esta informação é de suma importância, uma vez que o valor de profundidade para cada ponto da cena pode ser obtido com base na disparidade d. Mais precisamente, para determinar o valor de profundidade de um ponto P, por meio de um valor de disparidade d, aplica-se a seguinte equação:

$$Z = \frac{fb}{d}, \tag{2.2}$$

onde f é a distância focal e b o *baseline* (distância horizontal entre os dois pontos de vista) no *setup* utilizado para a aquisição das imagens, através de um par de câmeras idênticas. A Figura 2.2 representa a relação dada por esta equação, onde um exemplo de *setup* para geração de imagens estéreo é ilustrado. Ressalta-se que Z é expresso em valores reais de profundidade, através do sistema de coordenadas de mundo, assim como b. Enquanto f e d são representados em *pixels* (FICKEL, 2015).

Por fim, observa-se que existem outros meios para a aquisição do mapa de profundidades de uma cena, como é o caso do uso de câmeras com sensores de profundidade embutidos. No entanto, estas ainda não apresentam a mesma precisão fornecida pelas

Figura 2.2: Relacionamento entre o baseline b, a disparidade d, a distância focal f e a profundidade Z.



Fonte: Retirado de (FICKEL, 2015).

técnicas de *stereo matching*.

2.1.2 *Depth-Image-Based Rendering*

O modelo DIBR apresenta como principal característica a renderização de múltiplos pontos de vista "virtuais" de uma cena, através da projeção de uma imagem colorida, de acordo com a informação de profundidade associada a cada *pixel*, utilizando a técnica de *warping* definida em (MCMILLAN, 1997; MARK, 1999). Fehn (2004), descreve o modelo DIBR, o qual utiliza um formato compacto para a representação de imagens 3D. Este formato é composto basicamente por duas imagens, uma colorida para a vista de referência e outra em tons de cinza com a correspondência de disparidade para cada ponto.

Utilizando o mapa de disparidades, é possível gerar vistas sintéticas apenas fazendo o *warping* da imagem colorida de referência para pontos de vista escolhidos arbitrariamente. Contudo, este processo de geração de vistas sintéticas apresenta alguns desafios, dentre os quais pode-se citar a presença de artefatos (*cracks* e *ghosts*) e a falta de informação de projeção devido a oclusões na imagem de referência (*holes*). Desta forma, para a renderização de uma imagem em um ponto de vista virtual, faz-se necessário primeiramente projetá-la para o ponto de vista desejado, e posteriormente remover/tratar os artefatos e preencher as regiões sem informação na imagem gerada.

2.1.2.1 Síntese de Vistas

O primeiro passo para a geração de vistas sintéticas com o modelo DIBR trata da projeção da imagem obtida pelo ponto de vista real para o ponto virtual. Neste trabalho, considera-se apenas o modelo de configuração paralela (SCHARSTEIN; SZELISKI, 2003) para a geração de novas vistas, no qual as câmeras utilizadas para a estimativa de disparidade encontram-se alinhadas verticalmente, separadas por uma pequena distância horizontal. As imagens utilizadas para a geração dos mapas de disparidade são retificadas após a aquisição e, portanto, as linhas das imagens são correspondentes. Desta forma, a disparidade vertical é zero e somente o deslocamento horizontal dos *pixels* da imagem de referência são envolvidos no processo de projeção (3D *warping*) (ZHU; LI, 2016). Para a projeção de um ponto (x_O, y_O) da imagem de referência O, para um ponto de vista virtual V, aplica-se a seguinte equação:

$$x_V = x_O + s\frac{b_V d}{b_O}, \tag{2.3}$$

$$y_V = y_O,$$

onde $s = -1$ quando a vista estimada está do lado esquerdo da câmera de referência, e $s = 1$ quando está do lado direito. Os termos b_O e b_V representam, respectivamente, o *baseline* utilizado para obter o mapa de disparidades e a distância entre a câmera real e o ponto de vista virtual.

Muitos pontos podem ser mapeados para a mesma posição, o que pode levar à errônea oclusão de um *pixels* do *foreground* por *pixels* do *background* quando não tratado corretamente. Neste caso, utiliza-se como critério para a definição de qual *pixel* deve aparecer na imagem renderizada o valor de disparidade. Desta forma, o *pixel* com maior valor de disparidade é selecionado para manter-se à frente dos demais, uma vez que este encontra-se mais próximo da câmera.

2.1.2.2 Cracks

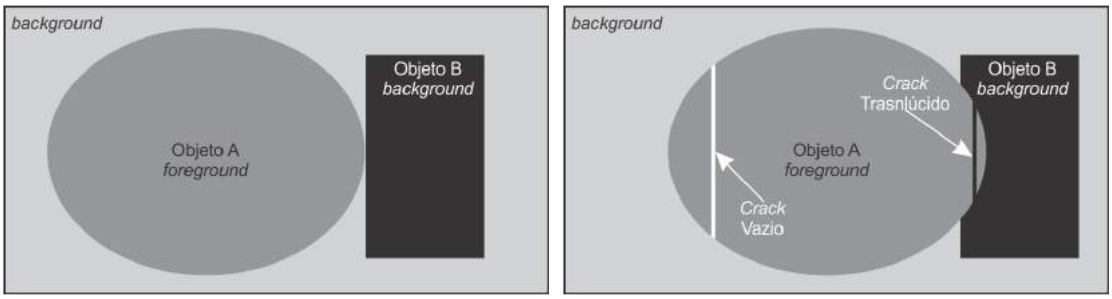
Os *cracks*, em geral, tem um ou dois *pixels* de largura e dispõe-se na forma de uma linha vertical (MUDDALA, 2015). O seu formato e largura são invariantes em relação ao tamanho da imagem. Estes são introduzidos pelo processo de projeção da imagem real para o ponto de vista virtual, em decorrência do cálculo de coordenadas. Neste processo,

cada ponto da imagem original é projetado de acordo com o *baseline* determinado para a vista virtual. Assim, o valor estimado para x_V é, na maioria das vezes, obtido em coordenadas de ponto flutuante, as quais precisam ser mapeadas para a posição mais próxima com um número inteiro correspondente (SMOLIC et al., 2008). Adicionalmente, erros de estimativa do mapa de disparidades podem dar origem a regiões similares.

Este artefato pode apresentar-se de duas formas distintas, translúcida e vazia (MUDDALA, 2015). O *crack* vazio caracteriza-se como uma fenda vertical fina sem informação de cor e textura na imagem virtual, a qual não possui informação de projeção. No entanto, em alguns casos estas regiões podem ser preenchidas por informação do *background* com projeção no mesmo ponto, configurando-se como *crack* translúcido.

A Figura 2.3(a) representa o ponto de vista real de uma cena, enquanto a imagem em (b) exibe o ponto de vista gerado através do processo de síntese definido pelo modelo DIBR. Neste exemplo, identifica-se na imagem sintética a ocorrência de um *crack* vazio (em branco), o qual não possui conteúdo em seu interior. Na mesma imagem, destaca-se a ocorrência de um *crack* translúcido (em preto), onde este apresenta em seu interior conteúdo do objeto B, que está situado no *background* em relação ao objeto A. Isto ocorre porque o objeto do *foreground* não possui informação de projeção nesta região, então a informação do *background* (que seria sobreposta no processo de síntese) é exibida no interior do *crack*.

Figura 2.3: Representação da ocorrência de *cracks*. (a) Imagem antes da projeção; (b) Imagem depois da projeção, com a sobreposição do *foreground* sobre o *background*, com *cracks* translúcido e vazio indicados por setas brancas.



(a) (b)

Fonte: O Autor.

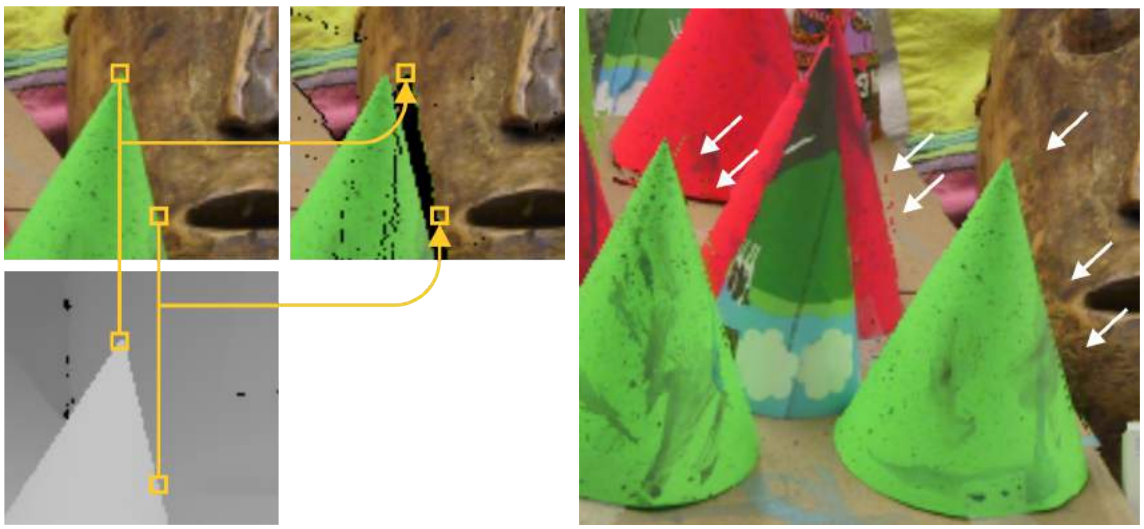
2.1.2.3 Ghosts

Os *ghosts* ocorrem quando tem-se uma descontinuidade de profundidade que não está bem definida no domínio da imagem, ou seja, existe uma transição de objetos em

cena e não é possível determinar os limites exatos. Estes podem ser definidos como uma mistura de cores das bordas de dois objetos que fazem fronteira na imagem original, que projeta-se para o *background* junto com o objeto de menor disparidade. Isto ocorre devido ao desalinhamento de cor e textura com a profundidade determinada pelo mapa (MUDDALA, 2015). Este artefato torna-se visível no processo de síntese da imagem sintética, e este processo é ilustrado na Figura 2.4(a).

Quando este artefato não é corretamente tratado, é comum encontrar a silhueta dos objetos do *foreground* em regiões do *background* da imagem gerada, como pode ser visto na Figura 2.4(b). Além de comprometer a qualidade da vista sintética, este artefato pode vir a prejudicar o algoritmo utilizado no preenchimento das regiões sem informação de projeção restantes. Uma vez que, estas regiões são preenchidas por algoritmos de *inpaintig*, os quais costumam utilizar informação de vizinhança para estimativa de conteúdo, estes podem inferir dados incorretos devido a presença dos *ghosts*.

Figura 2.4: Detalhe do exemplo de formação e disposição dos *ghosts*, seguindo o processo de síntese definido no modelo DIBR. Na imagem (a), do lado esquerdo superior, apresenta-se a imagem original (do ponto de vista real), a qual é projetada de acordo com o seu mapa de disparidades (lado esquerdo inferior), para o ponto de vista virtual, gerando a imagem sintética com o artefato (imagem do lado direito); Em (b), exibe-se os *ghosts* (destacados por setas brancas) na imagem final renderizada.



(a) (b)

Fonte: Adaptado de (MIDDLEBURY, 2016).

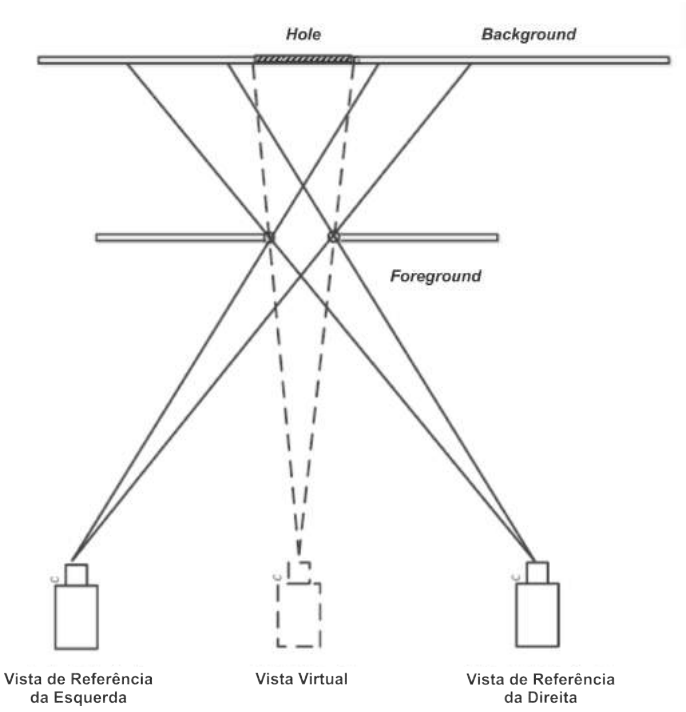
2.1.2.4 Holes

Um dos maiores problemas na renderização de imagens sintéticas com os algoritmos DIBR, é o fato de que áreas que são obstruídas na imagem original podem tornar-se

visíveis em vistas virtuais, o que é um problema conhecido como *exposure* ou *disocclusion* em computação gráfica (FEHN, 2004). A Figura 2.5 representa exatamente este processo, onde duas estruturas no *foreground* do cenário, não permitem a visualização de determinada região do *background*, que pode ser visualizada somente pelo ponto de vista da câmera virtual. As regiões obstruídas aparecem como buracos sem informação na vista virtual, denominadas *holes*.

Estas regiões aumentam proporcionalmente ao *baseline* determinado no momento de geração da imagem sintética, portanto não apresentam um tamanho pré-definido. Desta forma, faz-se necessário o emprego de técnicas de *inpainting* capazes de estimar além da cor destas regiões, também a informação de textura (pois estas regiões podem ser relativamente grandes). Neste contexto, este problema tem fomentado o desenvolvimento de inúmeras pesquisas destinadas especificamente a este fim, as quais tendem a explorar a informação de profundidade como meio de tornar mais precisa a estimativa de informação.

Figura 2.5: Configuração padrão de câmeras utilizada no modelo DIBR.



Fonte: Adaptado de (ZHU; LI, 2016).

Muddala (2015), separa uma determinada região dos *holes* em outra classe ainda, denominada *out-field areas*. Esta definição parte do princípio de que a câmera utilizada para a obtenção das imagens possui um campo de vista limitado. Desta forma, a informação utilizada para a geração da vista sintética é limitada pelo conteúdo obtido no

momento da aquisição da imagem. Assim, uma faixa lateral sem informação é formada quando a vista real é projetada para um ponto de vista virtual, em decorrência da limitação do campo de visão. Esta região pode ser observada na Figura 2.6 (em preto, destacada por um retângulo vermelho), a qual não pertence ao campo de vista da imagem original.

Figura 2.6: Imagem resultante do processo de síntese do modelo DIBR. Região que não pertence ao campo de vista da imagem original (em preto) delimitada por um retângulo vermelho.

Fonte: Adaptado de (MIDDLEBURY, 2016).

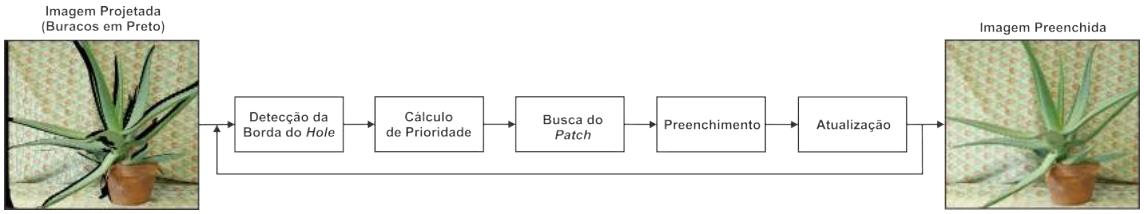
2.1.3 Algoritmos de *Inpainting*

Os *holes* são grandes regiões sem informação geradas nas imagens sintéticas, e necessitam do uso de técnicas que consigam preenchê-las corretamente. Neste ponto destacam-se as técnicas de *inpainting*, comumente utilizadas na restauração de pinturas e fotografias danificadas ou remoção/substituição de objetos selecionados em imagens (BERTALMIO et al., 2000). Observa-se, que quando áreas da imagem foram danificadas ou perdidas, o melhor a ser obtido é a reprodução de um resultado plausível, em vez de uma reconstrução perfeita dessas regiões (OLIVEIRA et al., 2001). Deste modo, métodos de *inpainting* costumam ser utilizados no preenchimento destas regiões (MORI et al., 2008; XU et al., 2013; MUDDALA; OLSSON; SJÖSTRÖM, 2013).

Métodos de *inpainting* baseados em *patches* costumam ser eficientes na reconstrução da estrutura e de detalhes da textura em buracos (MUDDALA, 2015). Neste contexto, algumas abordagens foram propostas para o preenchimento dos *holes*, como é o caso de (DARIBO; SAITO, 2011; GAUTIER; MEUR; GUILLEMOT, 2011), que desenvolve-

ram diferentes adaptações do método de *inpainting* de (CRIMINISI; PEREZ; TOYAMA, 2004). No entanto, os métodos presentes na literatura ainda apresentam limitações ao preencher grandes regiões sem informação, muitas vezes gerando artefatos ou recriando um padrão de textura inconsistente no interior dos buracos.

Figura 2.7: Ciclo de execução do método de *inpainting* baseado em *patches*.



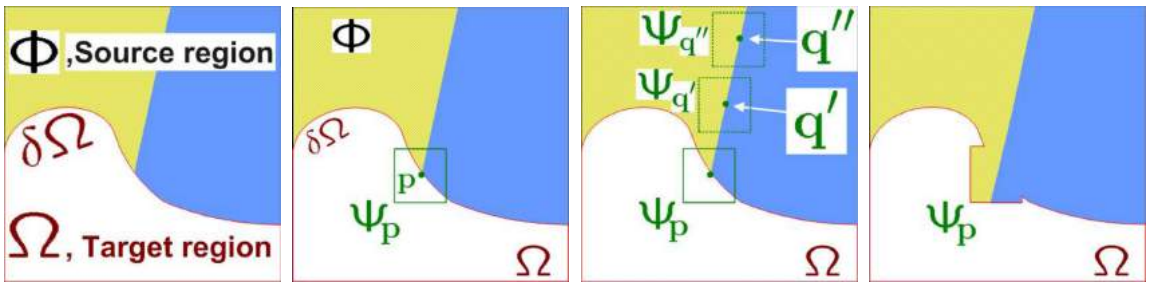
Fonte: O autor, baseado em (MUDDALA, 2015) utilizando imagens do *dataset* Aloe (MIDDLEBURY, 2016).

Dentre as técnicas propostas na literatura, destaca-se o trabalho de (CRIMINISI; PEREZ; TOYAMA, 2004), o qual propôs um método para a síntese de textura e preenchimento de buracos em imagens, que será descrito a seguir, visto fazer parte do método desenvolvido. O algoritmo de *inpainting* de (CRIMINISI; PEREZ; TOYAMA, 2004) apresenta um padrão de preenchimento gradativo, orientado a *pacthes* de tamanho fixo (9×9 *pixels*). A Figura 2.7 ilustra a sequência de passos do algoritmo, que é executada para o preenchimento de cada *patch*. Neste algoritmo, quando os *patches* são preenchidos na ordem correta (e o comparativo entre o ponto alvo e os candidatos a preenchimento é confiável), o resultado tende a ser mais exato (processo ilustrado na Figura 2.8).

Desta forma, partindo de uma região alvo a ser preenchida Ω, sua borda $\partial\Omega$ é detectada. Depois, é realizada a busca do *patch* Ψ_p com $p \in \partial\Omega$ que deve ser preenchido. Então, procura-se um *patch* Ψ_q na região de origem $\Phi = \mathcal{I} - \Omega$, onde $\mathcal{I}$ é a imagem a ser preenchida, e sua textura é copiada para Ψ_p. A ideia chave é usar Φ como uma base de dados de textura, e copiar pequenos *patches* Ψ_q para Ω de acordo com a informação local fornecida por Ψ_p.

Para determinar a ordem correta de preenchimento do algoritmo, utiliza-se uma equação de prioridade. A escolha do ponto p do *patch* a ser preenchido, para cada iteração, é dada por $P(p) = C(p)D(p)$, onde $P(p)$ é a prioridade para um *pixel* $p \in \partial\Omega$, $C(p)$ é o termo de confiança e $D(p)$ é o termo de dados. O termo $C(p)$ impõe a ordem de preenchimento concêntrica desejável, mensurando a quantidade de informação confiável para cada candidato ao longo da borda. Já o termo de dados $D(p)$, aumenta a prioridade de estruturas lineares que fluem através do ponto analisado, para que estas sejam reconstruídas

Figura 2.8: Estrutura de propagação do algoritmo proposto por (CRIMINISI; PEREZ; TOYAMA, 2004). (a) Imagem original, com a região alvo Ω, a borda $\partial\Omega$ e a região de origem Φ; (b) A região a ser sintetizada delimitada por Ψ_p centrada no ponto $p \in \partial\Omega$. (c) O candidato mais provável para preencher Ψ_p encontra-se ao longo da borda entre as duas texturas na região de origem, por exemplo, Ψ'_q ou Ψ''_q. (d) O melhor *patch* substituto do conjunto de candidatos é copiado na posição ocupada por Ψ_p, conseguindo assim o preenchimento parcial de Ω.



(a) (b) (c) (d)

Fonte: (CRIMINISI; PEREZ; TOYAMA, 2004).

primeiro, como destacado na Figura 2.8(b).

Por fim, realiza-se a busca pelo melhor candidato a preencher o *patch* selecionado com os critérios de prioridade. Assim, Ψ_q é determinado pela procura em Φ do *patch* que melhor corresponde a Ψ_p, por meio de uma função de erro que computa a diferença entre os *patches* Ψ_q e Ψ_p. A similaridade entre dois *patches* é estipulada pela menor distância SSD (*Sum of Squared Differences*) no espaço de cores CIE Lab. Então, o candidato Ψ_q escolhido para preencher Ψ_p, é o que minimiza esta função de erro.

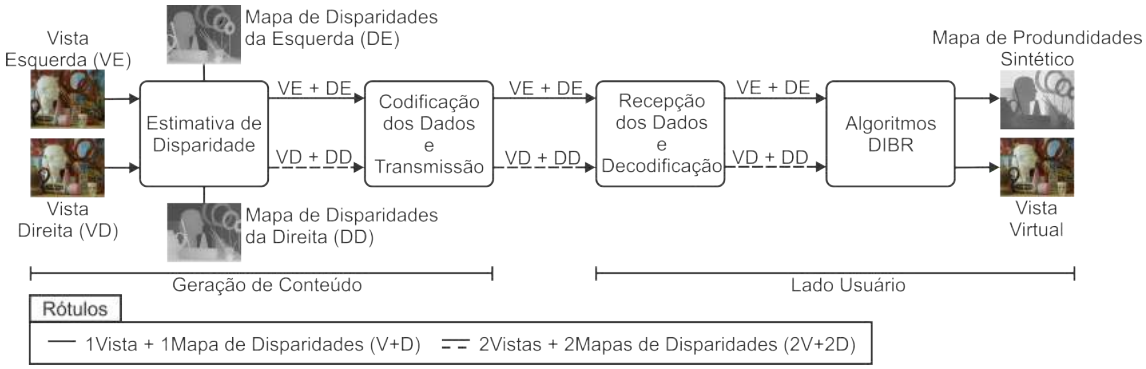
2.2 Trabalhos Relacionados

Nesta seção apresenta-se os principais trabalhos relacionados ao tema abordado nesta dissertação. Esses são brevemente sumarizados, destacando-se o contexto em que foram aplicados e as metodologias adotadas. Ao fim da seção é apresentado um comparativo em formato de tabela, o qual visa resumir os problemas abordados por cada uma das propostas.

Dois formatos de transmissão distintos – empregados em métodos de renderização com DIBR – podem ser encontrados na literatura (ilustrados na Figura 2.9). O primeiro formato segue os padrões de transmissão determinados no projeto do modelo, e transmite uma imagem colorida com o seu respectivo mapa de profundidades (V+D – *Video plus Depth*). Já o segundo, amplamente utilizado para aplicação em FTV, propõe submeter

duas imagens coloridas seguidas de seus mapas de disparidades (2V+2D). Observa-se, adicionalmente, que este segundo formato não se enquadra nos padrões atuais de transmissão televisiva, por enviar uma quantidade maior de dados e possivelmente exceder o limite de banda disponível. Inúmeros trabalhos seguem a abordagem 2V+2D, e tem apresentado avanços significativos no tratamento de artefatos e preenchimento dos *holes* e, portanto, também são relacionados com o método desenvolvido.

Figura 2.9: Diagrama de blocos que define o passo-a-passo para a transmissão e geração de vistas sintéticas com o modelo DIBR (V+D) e com múltiplas vistas (2V+2D) para TV 3D.



Fonte: O autor, com base em (MUDDALA, 2015).

Mori et al. (2008) propõem sintetizar vistas com um processo baseado na projeção e tratamento de mapas de profundidade. Inicialmente, os dois mapas obtidos com as câmeras mais próximas ao ponto de vista virtual são projetados em sua posição. Depois, pequenas regiões com 1 *pixel* de largura (*cracks*) são preenchidas por meio do filtro da mediana, seguido por uma filtragem bilateral para suavizar os mapas. Após o processamento, estes são reprojetados para os pontos de vista de origem e são utilizados para projetar as suas respectivas imagens coloridas para o ponto de vista alvo. Após, a imagem sintética é produzida através da mistura das duas imagens coloridas projetadas, utilizando um processo denominado *alpha blending*. Durante esta etapa, os *ghosts* são removidos por meio de uma dilatação dos *holes*. Por fim, as regiões sem informação restantes são preenchidas por meio do algoritmo de *inpainting* de (TELEA, 2004). Um problema desta abordagem é que não é prevista nenhuma etapa que tenha por objetivo equilibrar as diferenças de brilho e intensidade entre as imagens mescladas.

(OH; YEA; HO, 2009), apresentam uma abordagem semelhante à proposta por (MORI et al., 2008). No entanto, algumas adaptações foram adicionadas no algoritmo de preenchimento dos *holes*. Nesta proposta, a informação de *foreground* vizinha ao buraco é substituída temporariamente pela de *background*, para influenciar o algoritmo de

(TELEA, 2004) à propagar informação de cor da borda considerando apenas os pontos de maior profundidade. Esta abordagem tende a obter resultados superiores ao do algoritmo original, uma vez que os *holes* são compostos exclusivamente por informação de *background*. Contudo, observa-se que o bom comportamento do algoritmo depende do quão precisa é a seleção para substituição dos pontos de *foreground*.

(ZINGER; DO; WITH, 2010), utilizam um processo composto por quatro etapas para a geração de vistas sintéticas. Em um primeiro momento, as imagens e mapas de profundidades mais próximos ao ponto de vista virtual são projetados em sua posição. Depois, os *cracks* são preenchidos no mapa de disparidade com o filtro da mediana. A informação estimada nesta filtragem é utilizada como referência para a seleção de *pixels* na imagem original (antes da projeção), através do processo de projeção inversa (*inverse warping*). Após, realiza-se o *alpha blending* das duas imagens formando o ponto de vista virtual. Os *holes* restantes são preenchidos com um algoritmo de interpolação ponderada, o qual considera apenas informação de pontos vizinhos à região a ser estimada que apresentem baixo valor de profundidade. No entanto, destaca-se que o algoritmo empregado no preenchimento dos *holes* não produz bons resultados quando aplicado em grandes regiões, pois não é capaz de recuperar informação de textura.

Em (YANG et al., 2011), as duas imagens reais mais próximas são projetadas no ponto de vista virtual, onde uma é determinada com vista de referência e a outra vista auxiliar. Em um primeiro passo, os *cracks* translúcidos e vazios são identificados, e para cada pixel de seu conteúdo atribui-se o valor de intensidade do seu vizinho mais próximo (para a ocorrência de múltiplos vizinhos, o de maior profundidade é selecionado). Os pontos que não são visíveis na vista de referência e encontram-se disponíveis na vista auxiliar são copiados, seguindo o modelo 2V+2D. Após, aplica-se um procedimento para o ajuste de brilho da imagem para equilibrar as diferenças entre os dois pontos de vista. Posteriormente, os *ghosts* são removidos por um processo de dilatação. Para o preenchimento de cada *hole*, sua vizinhança é dividida em *foreground* e *background* utilizando o algoritmo *k-means clustering* auxiliado por um limiar com valor fixo. Por fim, aplica-se um algoritmo de preenchimento por dilatação assimétrica, o qual considera os pontos pertencentes ao *background*. Todavia, esta abordagem, assim como as demais que utilizam o modelo de transmissão 2V+2D não são comportadas pelo atual modelo de transmissão televisivo.

Para preencher os *holes*, (DARIBO; SAITO, 2011) propõem uma adaptação do algoritmo baseado em *patches* de (CRIMINISI; PEREZ; TOYAMA, 2004). Este, incorpora

um novo termo ao cálculo de prioridades do algoritmo original, baseado na estimativa de regularidade de profundidade. O termo inserido recebe peso proporcional aos termos originais do algoritmo. Desta forma, pontos candidatos com profundidade regular, bordas entrantes e informação confiável, tendem a receber maior prioridade de preenchimento. No cálculo de similaridade, passa-se a considerar adicionalmente a diferença absoluta de profundidade entre os *patches*, ponderada por um parâmetro. Contudo, a adição da similaridade em termos de profundidade compromete o comparativo de dois *patches*, que de maneira alguma é ligada a disposição dos objetos em cena.

(GAUTIER; MEUR; GUILLEMOT, 2011), apresentam outra variação do algoritmo baseado em *patches* de (CRIMINISI; PEREZ; TOYAMA, 2004). Nesta proposta, a prioridade para o preenchimento é estimada apenas em um dos lados do *hole* (*background*), de acordo com o sentido da projeção. Ainda, o termo de dados é calculado não somente na imagem colorida, mas no mapa de profundidades também. Na busca do melhor candidato para o preenchimento, realiza-se um comparativo de similaridade entre os *patches*, com base nos canais do espaço de cores RGB (*Red*, *Green*, *Blue*), e no mapa de disparidades, ponderando para que possuam o mesmo peso. Por fim, os cinco melhores *patches* – os quais se enquadram na estimativa de similaridade para o ponto alvo – são combinados e utilizados para o preenchimento da região alvo. No entanto, a combinação de *patches* pode ser um problema, pois a similaridade no comparativo geral dos pontos não garante que não exista uma grande diferença em subconjuntos específicos, o que pode acarretar o preenchimento impreciso.

Uma abordagem diferente foi desenvolvida por (SOLH; ALREGIB, 2012), denominada *Hierarchical Hole-Filling* (HHF). Esta produz uma estimativa de baixa resolução em forma de pirâmide para cada buraco na vista sintética, utilizando a média em blocos de 5×5 *pixels* válidos (ou seja, ignorando pontos sem informação). Estes são propagados para *pixels* em um próxima escala. Dentro de algumas escalas em multi-resolução é obtida uma estimativa de baixa resolução da vista sintética, sem buracos. Com a propagação desta imagem em baixa-resolução (ao longo das múltiplas escalas), é estimado o conteúdo dos buracos na imagem original. Os autores apresentam também uma variação deste algoritmo, denominada *depth adaptive* HHF, a qual visa dar prioridade para *pixels* com maior profundidade durante o processo de estimativa. Contudo, este algoritmo não possibilita a recuperação de textura nas regiões sem informação, produzindo para grandes regiões um resultado visual semelhante a um borramento, causado pela estimativa baseada no cálculo de média.

Xu et al. (2013) propõem reconstruir os *holes* em duas etapas. Primeiro, o mapa de profundidades é preenchido por meio de um processo de extrapolação. Este visa preencher o buraco com o maior valor de profundidade pertencente a sua borda. Em um segundo passo, a imagem é preenchida com uma adaptação do algoritmo de *inpainting* de (CRIMINISI; PEREZ; TOYAMA, 2004). Para tornar a técnica mais eficiente em termos de qualidade, foram inseridos termos que relacionam a profundidade nos cálculos de prioridade e similaridade. Neste processo, utiliza-se o mapa de profundidades totalmente preenchido, para auxiliar na seleção e preenchimento dos *holes*. Estas regiões são estimadas considerando somente informação pertencente ao *background*. Apesar do algoritmo apresentar bons resultados, observa-se que a qualidade de preenchimento está condicionada a correta segmentação da imagem em *background* e *foreground*, o que é um processo bastante complicado quando generalizado para toda a imagem.

Em (MUDDALA; OLSSON; SJÖSTRÖM, 2013), propõe-se uma adaptação do algoritmo de (CRIMINISI; PEREZ; TOYAMA, 2004). Neste método, o cálculo de prioridades é realizado apenas na região do contorno do buraco que pertence ao *background*. Assim, a imagem é particionada em *background* e *foreground* através de um limiar. Aqui, os termos de prioridade do algoritmo original são preservados, no entanto com algumas alterações. A confiança de um dado *patch* é medida pela quantidade de pontos pertencentes ao *background* que este possui. O termo de dados, o qual prioriza bordas entrantes, é estimado não somente sobre a imagem texturada, mas também no mapa de profundidades. No entanto, a similaridade entre *patches* é estimada com o mesmo cálculo descrito por (DARIBO; SAITO, 2011), e são considerados apenas *patches* pertencentes ao *background*.

Schmeing and Jiang (2015) propõem o preenchimento dos *holes* utilizando *superpixels* em um processo que considera informação de domínio espacial e temporal em vídeos. Primeiro, a imagem de referência é projetada de acordo com o mapa de profundidades para o ponto de vista virtual. Após, a imagem colorida é segmentada em *superpixels* utilizando o algoritmo proposto por (ACHANTA et al., 2012). Então, é determinado qual o próximo ponto a ser preenchido considerando as bordas dos *holes* vizinhas ao *background*. Nesta etapa são priorizados os pontos com mais informação preenchida em sua vizinhança, e também que possuem maior valor de profundidade na média dos pontos. Antes de realizar a busca pelo melhor candidato ao preenchimento é realizada uma filtragem em todos os *superpixels* disponíveis (no domínio espacial e temporal), para eliminar os que pertencem ao *foreground*. Por fim, um superpixel candidato é selecio-

nado dentro do conjunto disponível, considerando a similaridade de cor e profundidade para todos os pontos válidos. Este processo é repetido até que as regiões sem informação estejam completamente preenchidas. Entretanto, observa-se que mesmo utilizando *super-pixels* que são fracionados em regiões homogêneas, não é garantido que a similaridade de profundidade entre diferentes candidatos é válida, o que pode vir a prejudicar o processo de seleção.

Köppel, Müller and Wiegand (2016) apresentam um método híbrido para o preenchimento dos *holes*. O primeiro passo do algoritmo trata do pré-processamento da imagem colorida, o qual visa detectar os contornos. Em seguida, o mapa de profundidades e a imagem colorida são projetados para o ponto de vista virtual. Para iniciar o preenchimento, as bordas dominantes são utilizadas para separar a imagem em diferentes áreas de textura. Após, os contornos que fazem fronteira com as regiões a serem recuperadas são estimados através de uma adaptação do algoritmo de (CRIMINISI; PEREZ; TOYAMA, 2004), com um termo de prioridades criado exclusivamente para estimar a prioridade para bordas entrantes. No preenchimento, é considerada informação de *frames* anteriores para a recuperação desta estrutura. Depois, o segundo algoritmo de *inpainting* é aplicado, com o objetivo de estimar as áreas com textura homogênea. Para esta tarefa é utilizado um modelo espacial auto regressivo, o qual visa preencher estas regiões com informação de *background*. A qualidade obtida com este preenchimento é analisada de acordo com limites pré-estabelecidos de intensidade permitidos. Se o valor aferido estiver entre os limiares permitidos, o preenchimento é aceito. Caso contrário, a região é recuperada através do algoritmo de (CRIMINISI; PEREZ; TOYAMA, 2004). Por fim, é aplicado um filtro gaussiano entre o *foreground* e as áreas estimadas no *background*, para suavizar a transição entre as regiões.

Uma visão geral dos trabalhos descritos nesta seção é apresentada na Tabela 2.1, a qual detalha o formato de transmissão adotado e quais artefatos são tratados em cada abordagem. Os métodos *Selective Hole-Filling* e *Adaptative Feature-Oriented Hole-Filling* relacionados na tabela são propostos nesta dissertação, e são descritos posteriormente nas Subseções 3.4.1 e 3.4.2, respectivamente.

2.3 Sumário

Neste capítulo, foram apresentados inicialmente alguns conceitos chave para a compreensão do tema abordado nesta dissertação. Dentre estes, destaca-se a metodologia

Tabela 2.1: Visão geral dos algoritmos que compõem o atual estado da arte para o tratamento de artefatos e preenchimento de *holes* com o modelo DIBR. As notações CV, CT e G significam, respetivamente, *cracks* vazios, *cracks* translúcidos e *ghosts*.

Métodos	Formato		Artefatos			*Holes*
	V+D	2V+2D	CV	CT	G	
(MORI et al., 2008)		x	x		x	x
(OH; YEA; HO, 2009)		x	x		x	x
(ZINGER; DO; WITH, 2010)		x	x			x
(YANG et al., 2011)		x	x	x	x	x
(DARIBO; SAITO, 2011)		x	x			x
(GAUTIER; MEUR; GUILLEMOT, 2011)	x		x			x
(SOLH; ALREGIB, 2012)	x		x			x
(XU et al., 2013)	x		x			x
(MUDDALA; OLSSON; SJÖSTRÖM, 2013)	x		x			x
(SCHMEING; JIANG, 2015)	x		x			x
(KöPPEL; MüLLER; WIEGAND, 2016)	x		x			x
Selective Hole-Filling	x		x		x	x
Adaptative Feature-Oriented Hole-Filling	x		x	x	x	x

utilizada para a aquisição e representação da profundidade para imagens coloridas. Outro conceito importante trata da geração de imagens sintéticas com o modelo DIBR, através do processo de projeção (*image warping*). Ainda, foram definidos os diferentes tipos de artefato encontrados no processo de renderização das imagens, os quais foram descritos considerando especificamente os desafios impostos no tratamento de cada um.

Posteriormente, foram discutidos diversos trabalhos apresentados na literatura, os quais concentram seus esforços principalmente no preenchimento dos *holes*. No entanto, apresenta-se algumas abordagens que realizam o tratamento dos artefatos, as quais são utilizadas como um passo importante para a geração de vistas sintéticas. Nota-se que grande parte dos autores baseiam suas estrategias em algoritmos clássicos de *inpainting* (como de (CRIMINISI; PEREZ; TOYAMA, 2004) e (TELEA, 2004)) para o preenchimento dos *holes*. Neste sentido, observa-se um volume considerável de trabalhos científicos propondo melhorias nos métodos de preenchimento para os *holes*, os quais tem obtido melhoras significativas na qualidade da estimativa destas regiões. Apesar do grande esforço da comunidade cientifica, tais abordagens ainda não são suficientes para o preenchimento adequado de grandes regiões sem informação. Assim, destaca-se que a presente dissertação representa um passo significativo para a detecção de artefatos e preenchimento de grandes regiões sem informação em imagens sintéticas renderizadas com os algoritmos DIBR.

3 TÉCNICA PROPOSTA

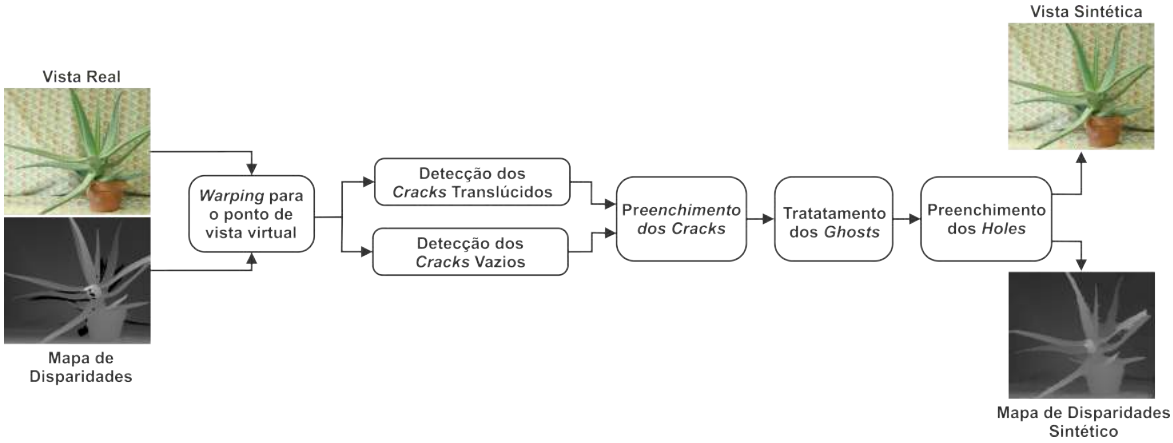
Neste capítulo são descritos os métodos desenvolvidos para a geração de vistas sintéticas utilizando o modelo DIBR. Na Seção 3.1, apresenta-se uma visão geral do algoritmo proposto, na qual descrevem-se as etapas que constituem o *pipeline* utilizado para a geração de vistas sintéticas. As seções seguintes detalham cada uma das técnicas desenvolvidas nesta dissertação. Mais especificamente, na Seção 3.2, apresentam-se os algoritmos para detecção e preenchimento dos *cracks*. A Seção 3.3, descreve os métodos desenvolvidos para a identificação e tratamento de *ghosts*. Por fim, na Seção 3.4, são propostas duas abordagens baseadas em *inpainting* para o preenchimento dos *holes*.

3.1 Visão Geral do Algoritmo Proposto

A geração de vistas sintéticas com o modelo DIBR compreende diversas etapas, as quais visam projetar a imagem para o ponto de vista virtual, remover artefatos e recuperar regiões sem informação de projeção. Neste contexto, a Figura 3.1 apresenta um diagrama de blocos com a sequência de passos empregados no *pipeline* proposto. A etapa inicial do algoritmo trata da projeção da imagem real para o ponto de vista virtual (*warping*). O processo é guiado pelo mapa de disparidades, e cada *pixel* da imagem real é projetado de acordo com o *baseline* definido pelo usuário. Esta primeira etapa compreende um procedimento padrão do modelo DIBR e, portanto, é detalhada na Subseção 2.1.2. Posteriormente, os *cracks* são detectados tanto em sua forma translúcida quanto vazia, em duas propostas distintas definidas por operadores morfológicos. Após a detecção dos *cracks*, aplica-se uma adaptação do algoritmo para preenchimento proposto por (OLIVEIRA et al., 2001), tanto no mapa de disparidades como na imagem colorida. O quarto passo do algoritmo consiste na identificação e tratamento dos *ghosts*. A identificação de candidatos a *ghost* é dividida em duas partes: (*i*) seleção de candidatos por meio de uma operação morfológica; (*ii*) avaliação ponto-a-ponto dos candidatos, relacionado-os com a informação RGB da vizinhança, para classificar se cada ponto pertence ou não ao artefato. Após, os *ghosts* identificados são reprojetados para a outra extremidade do *hole* (*foreground*), enquanto que os demais candidatos são mantidos em sua posição. Por fim, aplica-se um algoritmo para o preenchimento dos *holes*. Para esta tarefa foram propostos dois algoritmos de *inpainting*, ambos baseados no algoritmo proposto por (CRIMINISI; PEREZ; TOYAMA, 2004). Adicionalmente, observa-se que todas as etapas que constituem o *pi-*

peline proposto devem ser executadas em sequência, seguindo a ordem estabelecida no diagrama (Figura 3.1).

Figura 3.1: Diagrama geral da solução proposta. A interação entre os métodos é ilustrada através de setas.



Fonte: O Autor, com imagens adaptadas do *dataset* Aloe de (MIDDLEBURY, 2016).

3.2 Método para a Remoção dos *Cracks*

Por definição, os *cracks* possuem um ou dois *pixels* de largura e estão dispostos na forma de uma longa linha vertical. Ainda, podem apresentar-se nas formas vazia e translúcida e, portanto, faz-se necessário a utilização de métodos complementares de detecção. Nesta seção, apresentam-se as abordagens desenvolvidas para a identificação dos *cracks*, e posteriormente, descreve-se o algoritmo utilizado no preenchimento.

3.2.1 Detecção dos *Cracks* Vazios e Translúcidos

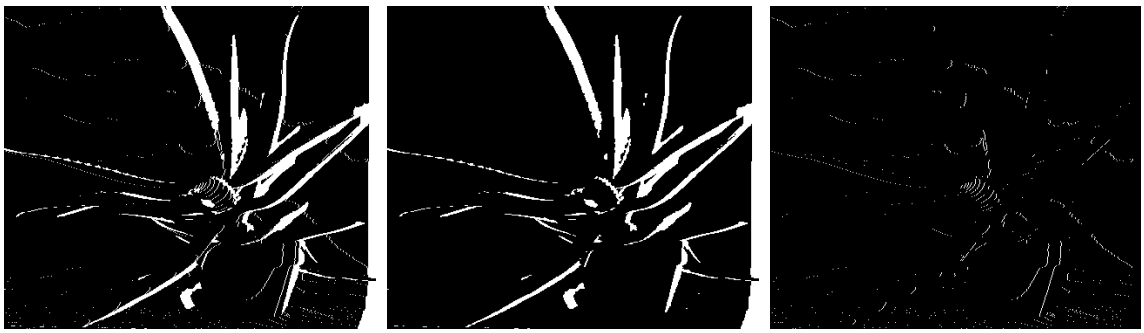
A primeira etapa para a remoção dos *cracks* é o processo de identificação. Para cada um dos formatos deste artefato foi desenvolvida uma abordagem diferente. Os métodos não possuem qualquer relação ou dependência, portanto podem ser aplicados em paralelo e os mapas com os pontos identificados unidos em uma etapa posterior. Para identificar os *cracks* vazios, calcula-se inicialmente uma imagem binária S – exibida na Figura 3.2 (a) – que contém todos os *pixels* sem informação de projeção na imagem sintética. Então, S é filtrada com um operador morfológico de abertura, por meio de um elemento estruturante H_{CV}, que resulta em uma imagem denominada $\hat{S}$ sem os *cracks*

vazios (Figura 3.2 (b)). Neste trabalho, H_{CV} é aplicado com um *pixel* de altura por dois de largura (ou seja, $H_{CV} = [1; 1]$), representando o padrão definido para os *cracks*. Ressalta-se que esta máscara possui exatamente um *pixel* de altura para que seja aplicada em todas as linhas da imagem. A largura do elemento estruturante é adequada ao máximo definido para um *crack* na literatura, abrangendo todas as ocorrências possíveis. O objetivo desta operação é detectar todos os pontos sem informação que não apresentam o padrão dos *cracks* vazios, isto é, $\hat{S}$ contém somente as ocorrências de *holes* na imagem. A máscara que contém somente os *cracks* vazios é dada através da remoção dos *pixels* associados a *holes* da máscara inicial S, por meio da seguinte equação:

$$C_V = S \setminus \hat{S}, \tag{3.1}$$

onde $\setminus$ representa o operador de complemento absoluto. O resultado desta operação pode ser visualizado na Figura 3.2 (c), na qual os pontos de C_V estão identificados em branco.

Figura 3.2: Passo-a-passo do algoritmo de detecção de *cracks* vazios. (a) Imagem binária S que contém todos os *pixels* sem informação de projeção na imagem sintética. (b) Imagem $\hat{S}$, resultante do processo de filtragem com operador morfológico de abertura. (c) Máscara C_V com os *cracks* vazios, obtida pela aplicação da Equação 3.1.



(a) (b) (c)

Fonte: O Autor.

Posteriormente, identificam-se os *cracks* em sua forma translúcida. Como pode ser visto na Figura 1.1(a), este apresenta o mesmo formato do *crack* vazio, no entanto é preenchido por informação de um objeto que está mais ao fundo na cena (no *background*). Quando a informação do objeto do *background* é inserida no interior do *crack*, ocorre a descontinuidade de disparidade no objeto do *foreground*, como pode ser visto na Figura 3.2(a). Assim, propõe-se um método para identificar estes padrões recorrentes de descontinuidade nos mapas de disparidade projetados.

A Figura 3.3(a) ilustra a ocorrência deste artefato – destacado em cinza – em termos de valores de intensidade no mapa de disparidades, o qual apresenta-se circundado

Figura 3.3: Representação do mapa de disparidades com a ocorrência de *cracks*. (a) Os pontos em cinza representam um *crack* translúcido, enquanto os pontos em preto representam uma região sem informação de projeção (parte *crack* vazio e o restante *hole*) da imagem sintética; (b) Representa o resultado obtido após a aplicação do operador de fechamento morfológico em (a).

62	62	62	62	0	63	63	47	63	63	63	63	63
62	62	62	62	0	63	63	63	47	63	63	63	63
62	62	62	61	61	0	63	63	47	63	63	63	63
62	62	61	61	61	0	63	63	47	63	63	63	63
62	62	61	61	61	0	63	63	47	63	63	63	63
62	62	61	61	0	0	0	63	47	63	63	63	63
62	61	61	61	0	0	0	63	63	47	63	63	63
62	61	61	60	0	0	0	0	63	47	63	63	63
62	61	61	60	0	0	0	0	63	47	63	63	63

(a)

62	62	62	62	0	63	63	63	63	63	63	63	63
62	62	62	62	0	63	63	63	63	63	63	63	63
62	62	62	61	61	0	63	63	63	63	63	63	63
62	62	61	61	61	0	63	63	63	63	63	63	63
62	62	61	61	61	0	63	63	63	63	63	63	63
62	62	61	61	0	0	0	63	63	63	63	63	63
62	61	61	61	0	0	0	63	63	63	63	63	63
62	61	61	60	0	0	0	0	63	63	63	63	63
62	61	61	60	0	0	0	0	63	63	63	63	63

(b)

Fonte: O Autor.

por valores significativamente superiores. Para identificar este padrão, aplica-se o operador de fechamento morfológico para imagens em tons de cinza no mapa de disparidades projetado D_p, obtendo a imagem $\hat{D}_p$. Nesta operação, o elemento estruturante utilizado (H_{CT}) tem um *pixel* de altura por três de largura (ou seja, $H_{CT} = [1; 1; 1]$). Esta técnica é aplicada com o objetivo de preencher as regiões inferiores a três *pixels* de largura com o maior valor de disparidade dentro de H_{CT}. Destaca-se que os pontos sem informação de preenchimento são desconsiderados neste processo (em preto na Figura 3.3). Desta forma, todos os *cracks* translúcidos – que possuem por definição no máximo dois *pixels* de largura – serão preenchidos em cada linha da imagem com o maior valor de disparidade vizinho.

O exemplo das Figuras 3.3 (a) e (b) ilustra o mapa de disparidades antes e depois da aplicação do operador de fechamento morfológico. Quando H_{CT} é posicionado sobre cada ponto pertencente ao artefato (com intensidade 47), seleciona-se o maior valor de disparidade em seu interior. A seleção considera o *pixel* analisado e os dois vizinhos mais próximos a direita, pois obedece o formato do elemento estruturante. Desta maneira, após esta operação, $\hat{D}_p$ deverá conter todas as ocorrências de *cracks* translúcidos preenchidas pelo maior valor de vizinhança, como ilustra a Figura 3.3(b). Assim, para obter a máscara com a identificação do artefato para cada coordenada (x, y) da imagem D_p, aplica-se a seguinte equação:

$$C_T(x, y) = \begin{cases} 1, & \text{se } \hat{D}_p(x, y) - D_p(x, y) \geq \lambda \\ 0, & \text{caso contrário} \end{cases} \tag{3.2}$$

onde λ é o limiar que controla a variação de intensidade aceita para um ponto após a

operação morfológica. Se λ for pequeno, vários mínimos locais do mapa de disparidade (devido a pequenas oscilações de intensidade) são considerados como *cracks*. Com λ grande demais, *cracks* verdadeiros podem não ser detectados. Com base em experimentos, o valor $\lambda = 5$ demonstrou um bom compromisso entre esses dois extremos. Por fim, os pontos com ocorrências de *cracks* de ambas as formas são computados formando a máscara C, onde $C = C_V \cup C_T$.

3.2.2 Preenchimento dos Cracks

Diferentes abordagens para o preenchimento dos *cracks* tem sido propostas na literatura. Algumas destas tem o propósito de preencher especificamente os *cracks*, como é o caso do filtro da mediana com (MORI et al., 2008) e *inverse warping* por (TRAN; HARADA, 2013; ZINGER; DO; WITH, 2010). Outros autores, preenchem este artefato juntamente com os *holes*, como é o caso de (SOLH; ALREGIB, 2012) com o HHF, e de (DARIBO; SAITO, 2011) e (OH; YEA; HO, 2009), com adaptações dos algoritmos de (CRIMINISI; PEREZ; TOYAMA, 2004) e (TELEA, 2004), respectivamente.

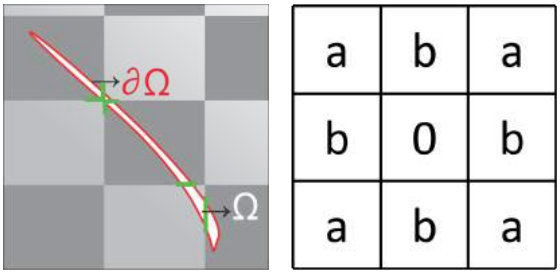
Validando o método proposto para a detecção dos *cracks* em um conjunto de 58 imagens de (MIDDLEBURY, 2016), constatou-se que estes compreendem em média cerca de 1,45% do total de *pixels* de uma vista sintética. Esta taxa de ocorrência é bastante significativa e indica portanto a necessidade de um tratamento individual do artefato, já que este não possui as mesmas características dos *holes*. Assim, foi proposta a adaptação do algoritmo de (OLIVEIRA et al., 2001) para executar especificamente esta tarefa. A decisão da escolha foi baseada nas suas característica de preenchimento, e nos promissores resultados obtidos em avaliações experimentais.

Oliveira et al. (2001) propuseram um método denominado *fast inpainting*, que tem por objetivo a recuperação de pequenas áreas sem informação em imagens defeituosas. O método é baseado em um processo de difusão isotrópica, que estima o conteúdo da região a ser preenchida Ω, através da propagação de informação disponível em sua borda $\partial\Omega$ (como representado na Figura 3.4(a)). Em nossa aplicação assumimos $\Omega = C$, pois C é a máscara que indica as ocorrências dos *cracks* a serem preenchidos na imagem.

Inicialmente, a informação de Ω é removida para limpar qualquer dado que possa prejudicar o processo de preenchimento. Então, o processo de difusão é aproximado por repetidas convoluções na região a ser preenchida com o núcleo de difusão ilustrado na Figura 3.4(b). O núcleo de difusão têm seus valores setados como $a = 0.073235$

e $b = 0.176765$, seguindo a definição do algoritmo original. Esta operação resulta em uma imagem com os *cracks* totalmente preenchidos. Ressalta-se, que em geral poucas iterações são necessárias (no máximo 4) para o preenchimento completo de Ω.

Figura 3.4: (a) Ilustração de imagem com região sem informação Ω em branco e borda $\partial\Omega$ destacada em vermelho. Adicionalmente, estão destacadas em verde as regiões de alto contraste que fazem parte de Ω. (b) Núcleo de difusão, com $a = 0.073235$ e $b = 0.176765$.



(a) (b)

Fonte: O Autor em (a) e (OLIVEIRA et al., 2001) em (b).

Esta abordagem pode introduzir borramento quando Ω passa por regiões com bordas de alto contraste (em verde na Figura 3.4(a)). Contudo, os *cracks* tem geralmente apenas um ou dois *pixels* de largura (como definido pelo elemento estruturante H_{CV}), e dispõe-se no interior de objetos, o que limita bastante a ocorrência deste problema. Desta forma, o segundo processo de filtragem proposto originalmente no algoritmo não se faz necessário, visto destinar-se apenas a correção destas regiões.

A Figura 3.5 apresenta um exemplo da aplicação dos algoritmos de detecção e preenchimento dos *cracks*. A imagem (a) exibe o resultado obtido com o processo de projeção de uma imagem do ponto de vista real para o virtual. Em (b), os *cracks* translúcidos e vazios são identificados pelos métodos propostos, destacados em vermelho e verde, respectivamente. Estes foram detectados aplicando paralelamente as duas abordagens desenvolvidas (Subseção 3.2.1). Por fim, em (c), é apresentado o resultado obtido com o algoritmo de preenchimento de (OLIVEIRA et al., 2001).

3.3 Método para a Detecção e Tratamento dos Ghosts

Ghosts ocorrem quando não existe uma exata definição da região de transição (borda) dos objetos em cena. Os *ghosts* são compostos geralmente por informação do *foreground* que é propagada para o *background* no processo de projeção. Para solucionar este problema é preciso identificar as regiões que foram projetadas incorretamente, e aplicar o tratamento adequado. Desta forma, propõe-se uma abordagem em que primeiro

Figura 3.5: (a) Vista 1 do *dataset* Art projetada para o ponto de vista virtual (mesmo ponto da vista real 3). (b) *Cracks* identificados na imagem, em verde os vazios e em vermelho os translúcidos. (c) Imagem obtida após o preenchimento dos *cracks* com o algoritmo de (OLIVEIRA et al., 2001).

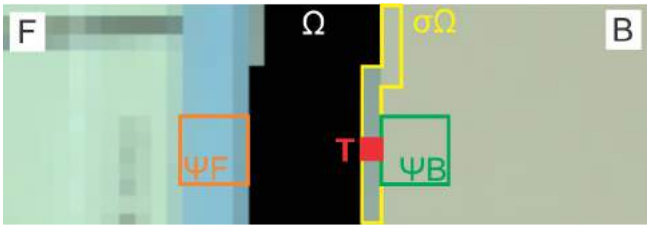
(a) (b) (c)

Fonte: O Autor adaptado de (MIDDLEBURY, 2016).

identificam-se as regiões que podem potencialmente conter *ghosts*, e segundo avaliam-se os pontos dessa região considerando a similaridade com as extremidades do *hole*. Os pontos identificados são reprojetados para a outra extremidade do *hole*, ou seja, são acoplados ao *foreground*.

Primeiramente, são identificadas as regiões que podem potencialmente conter *ghosts*. Para tal, calcula-se uma imagem binária G que é obtida pela exclusão dos *cracks* C de S, ou seja, $G = S \setminus C$. Depois, é aplicado um operador de dilatação morfológica com um elemento estruturante não simétrico H_G, para expandir as regiões oclusas da direção da câmera de referência no sentido da câmera virtual. Por exemplo, se a câmera virtual está do lado esquerdo da câmera de referência, a informação de *background* de todos os *holes* causados por problemas de oclusão devem estar do lado direito. Neste caso, utiliza-se um elemento estruturante (H_G) no formato de uma linha horizontal, com um pixel de altura por três de largura, seguindo a configuração $[0; 1; 1]$. Para projeções no outro sen-

Figura 3.6: Zoom em imagem do Dataset Monopoly (SCHARSTEIN; SZELISKI, 2003). Notação: σ_Ω são os candidatos a *ghost*. Ω, F e B representam o *hole*, *foreground* e *background* respectivamente. ψ_B e ψ_F são *patches* para a avaliação de similaridade do alvo (T) com F e B.



Fonte: O Autor.

tido, utiliza-se a máscara $[1; 1; 0]$. Posteriormente, separam-se os candidatos utilizando o operador de complemento absoluto sobre a imagem processada $\hat{G}$, obtendo σ_Ω, de acordo com a Equação 3.1. A Figura 3.6 ilustra este processo.

A próxima etapa consiste em avaliar cada ponto candidato $T \in \sigma_\Omega$, com base na similaridade dos *patches* da vizinhança de ambos os lados do *hole*. Neste momento, calcula-se a média de intensidade em *patches* de 3×3 ψ_B e ψ_F, que são vizinhos de T no *background* e *foreground*, respectivamente. Então, obtém-se a soma da diferença absoluta dos canais RGB entre as médias e o ponto analisado, resultando em dF para o *foreground* e dB para a outra extremidade, conforme definido a seguir:

$$dB = |T - \mu_{\psi_B}|, \tag{3.3}$$

$$dF = |T - \mu_{\psi_F}|, \tag{3.4}$$

onde μ_{ψ_B} e μ_{ψ_F} são vetores com a média dos canais RGB dos *patches* ψ_B e ψ_F, respectivamente.

Se $dB < dF$ e $dB < \alpha$, onde α é um limiar de similaridade, então T pertence ao *background*. Caso contrário, o ponto é considerado como pertencente ao *foreground* e movido horizontalmente até a outra extremidade do *hole*. Comparar apenas as médias dos *patches* vizinhos de T nas duas extremidades não garante que um ponto foi projetado de maneira correta, pois muitas vezes estes *pixels* são compostos por uma mistura de intensidades do objeto do *foreground* e do *background*. Nesse contexto, o valor determinado para o limiar α serve justamente para influenciar estes *pixels* a serem reprojetados para o *foreground*, que é o local correto para os pontos de transição entre os dois objetos. Um valor baixo para α tende a permitir que menos pontos sejam mantidos no seu local de projeção. Por outro lado, um valor alto tende a manter *pixels* mais similares ao *background* no ponto em que foram projetados. O resultado obtido com a aplicação do algoritmo proposto pode ser visualizado na Figura 3.7.

3.4 Preenchimento dos *Holes*

Após o tratamento dos artefatos, restam apenas regiões maiores sem informação, as quais precisam ser estimadas para a renderização final da vista sintética. Uma vez que é comum encontrar grandes áreas sem informação no processo de renderização com as

Figura 3.7: (a) Imagem resultante do processo de detecção e preenchimento dos *cracks*. (b) Pontos candidatos a *ghost* destacados em azul. (c) Imagem obtida após a avaliação e tratamento dos artefatos identificados.

(a) (b) (c)

Fonte: O Autor, adaptado de (MIDDLEBURY, 2016).

técnicas DIBR, causadas por oclusões e/ou problemas no mapa de disparidades, é preciso sintetizar não só uma cor plausível dentro dos buracos, mas também recriar uma textura localmente adequada. Para esta tarefa, foram propostas duas abordagens diferentes que têm como base o trabalho de síntese de textura de (CRIMINISI; PEREZ; TOYAMA, 2004). Os métodos desenvolvidos têm como principal característica priorizar a região do *background* no preenchimento dos *holes*, uma vez que as regiões obstruídas são, por definição, uma parte do *background* que é desobstruída após a projeção da imagem real.

Na próxima subseção, descreve-se a primeira adaptação proposta para o algoritmo baseado em *patches* de (CRIMINISI; PEREZ; TOYAMA, 2004), na qual foi remodelado o processo que determina a ordem de preenchimento dos buracos. Posteriormente, na subseção 3.4.2, apresenta-se a outra abordagem proposta, a qual segue o padrão de funcionamento do algoritmo anterior, no entanto adiciona passos de classificação para os *holes* e uma nova metodologia de busca por *patches* similares. Observa-se que a segunda abordagem foi proposta devido a limitações apresentadas pelo primeiro algoritmo, as quais exigiram uma remodelagem completa nos procedimentos que realizam o *inpainting* das regiões sem informação.

3.4.1 *Selective Hole-Filling*

Essa subseção apresenta um algoritmo de *inpainting* baseado em informação de profundidade para o preenchimento dos *holes*. O método proposto estende o trabalho de (CRIMINISI; PEREZ; TOYAMA, 2004), aprimorando suas características para a aplicação em imagens estéreo para DIBR. A ideia principal do algoritmo é dar prioridade na

ordem de preenchimento para pontos que tem maior valor de profundidade, preenchendo primeiramente a região conexa ao *background*.

Após selecionar a borda $\partial\Omega$ de cada um dos *holes* Ω, o próximo passo é definir a ordem de preenchimento dos *holes*. Para tal, foi proposto um novo termo $E(p)$ que determina para cada ponto p a média de profundidade nos *pixels* do *patch* ψ_p, considerando apenas os pontos pertencentes a Φ. Desta forma, a escolha de p dá-se por meio da seguinte equação:

$$P(p) = C(p)E(p), \tag{3.5}$$

onde $C(p)$ é o termo de confiança – determinado por (CRIMINISI; PEREZ; TOYAMA, 2004) – e $E(p)$ é o termo de profundidade, definidos como segue:

$$C(p) = \frac{\sum_{q \in \Psi_{p \cap (\mathcal{I}-\Omega)}} C(q)}{|\Psi_p|}, \tag{3.6}$$

$$E(p) = \frac{\sum_{q \in \Psi_{p \cap (\mathcal{I}-\Omega)}} d(q)}{|d(p)|}, \tag{3.7}$$

em que $|\Psi_p|$ é a área de Ψ_p, $|d(p)|$ é a area de $d(p)$ (em termos de números de *pixels*) e $d(q)$ é o valor de profundidade para cada ponto q do *patch*. A prioridade $P(p)$ é calculada para cada $p \in \partial\Omega$ em cada iteração, e o ponto com o maior valor é escolhido. Na configuração inicial, $C(p)$ é inicializado com zero $\forall p \in \Omega$, e $C(p) = 1 \; \forall p \in \mathcal{I} - \Omega$.

O termo de confiança $C(p)$ mensura a quantidade de informação confiável ao redor do *pixel* p e, por isso, prioriza o preenchimento dos *patches* que tem mais *pixels* válidos. O termo $E(p)$ prioriza a maior profundidade, a qual naturalmente favorece *pixels* do *background*. Depois de escolher o *patch* Ψ_p a ser preenchido, o último passo é procurar o candidato que melhor obedece o critério de similaridade em Φ. Assim, Ψ_q é escolhido pela busca em Φ do *patch* que é mais similar com Ψ_p:

$$\Psi_q = \arg \min_{\Psi_q \in \Phi} s(\Psi_p, \Psi_q), \tag{3.8}$$

$$s(\Psi_p, \Psi_q) \;\; = \sum_{x \in \Omega_v(\Psi_p)} \|\Psi_p(x) - \Psi_q(x)\|^2,$$

onde $\Omega_v(\Psi_p)$ denota o conjunto de *pixels* em Ψ_p contendo informação válida e $\Psi_p(x)$ é o vetor de cores RGB relacionado com o *pixel* x. Assim, Ψ_q deverá ser um *patch* que tem textura e cor similares com Ψ_p. O resultado obtido com a aplicação deste método

pode ser visto na Figura 3.8.

Figura 3.8: Resultado obtido com o método *Selective Hole-Filling* no preenchimento dos *holes*. (a) Imagem resultante dos processos de tratamento dos artefatos, com os *holes* destacados em vermelho. (b) Resultado obtido com o método de preenchimento proposto.

(a) (b)

Fonte: Adaptado de (MIDDLEBURY, 2016).

3.4.2 *Adaptative Feature-Oriented Hole-Filling*

Nesta Subseção apresenta-se outro método para o preenchimento dos *holes*. Este, baseia-se no algoritmo de síntese de texturas de (CRIMINISI; PEREZ; TOYAMA, 2004), assim como a abordagem descrita na subseção anterior. No entanto, nesta proposta são exploradas outras limitações encontradas no algoritmo original, que originaram novos métodos para o cálculo de prioridades e verificação de similaridade. Desta forma, desenvolveu-se uma nova abordagem com *patches* de tamanho adaptável, a qual seleciona os candidatos ao preenchimento do *hole* em um espaço de busca limitado da imagem original.

Como na abordagem original, a primeira etapa do algoritmo visa selecionar a borda de cada um dos *holes* Ω. No entanto, antes do cálculo de prioridades esta proposta inclui um estágio de classificação dos *holes*, baseado nos seguintes dados: (*i*) área A_Ω ocupada por cada *hole* em número de *pixels* e (*ii*) desvio padrão $\sigma_{\partial\Omega}$ de intensidade de disparidade dos pontos da borda $\partial\Omega$ de cada *hole*. A classificação é utilizada para determinar quais *holes* possuem informação suficiente para a segmentação de vizinhança em *background* e *foreground*. Além disso, utiliza-se $\sigma_{\partial\Omega}$ para verificar se cada *hole* possui uma variação significativa de disparidade em sua borda. Regiões que precisam ser preenchidas, e que apresentam disparidade homogênea na sua borda, não devem considerar

profundidade no preenchimento, pois dispõe-se no interior de um objeto e não na extremidade. Deste modo, para cada hole Ω com $A_\Omega > \alpha_A$ e $\sigma_{\partial\Omega} > \alpha_\sigma$ é definido um limiar t, que tem por objetivo dividir em duas partes a imagem utilizada para o preenchimento de Ω. Os valores de α_A e α_σ foram determinados de maneira empírica, visando manter apenas regiões grandes e que possuem variação de disparidades significativa para o processo segmentação. Se α_A é determinado com um valor baixo demais, pequenos *holes* que não possuem informação consistente na borda para segmentar a região vizinha são consideradas, gerando partições incorretas. Por outro lado, um valor alto demais faz com que *holes* com informação consistente na borda sejam desconsideradas. Ainda, é importante indicar um valor para α_σ que consiga identificar corretamente se o *hole* avaliado está circundado por disparidade uniforme ou não. Para este limiar, um valor baixo demais não consegue detectar *holes* que estão no interior de objetos homogêneos, e que não devem ser classificados, enquanto que um valor alto demais deixa de classificar outros que estão na extremidade de objetos, e que deveriam ter sua vizinhança segmentada. Para obter t, utiliza-se a técnica *Trimmed Mean* nos valores de disparidade de $\partial\Omega$. Nesta técnica, os valores do conjunto analisado são ordenados, e posteriormente é calculada a média de intensidades considerando apenas a faixa central de valores, removendo 10% no inicio e fim das amostras. Com isso, é descartada a presença de *outliers* na estimativa de t.

Com esta técnica, objetiva-se detectar para cada *hole* quais são os pontos da imagem que pertencem ao seu *background* ou *foreground*. Então, para gerar uma máscara bg_{map} com todos os pontos (x, y) da imagem Φ, indicando quais pertencem ao *background* de Ω, basta aplicar a seguinte equação:

$$bg_{map}(x, y) = \begin{cases} 1, & \text{se } d_p(x, y) > t \\ 0, & \text{caso contrário} \end{cases} \tag{3.9}$$

onde $d_p(x, y)$ é o valor de disparidade do ponto analisado. Os *holes* que apresentarem $A_\Omega \leq \alpha_A$ ou $\sigma_{\partial\Omega} \leq \alpha_\sigma$ tem 1 atribuído a todos os pontos da máscara.

A próxima etapa do algoritmo compreende a definição da ordem de preenchimento para cada *hole*. Neste momento, introduz-se um novo termo $B(p)$ que determina simultaneamente para cada ponto $p \in \partial\Omega$ sua confiabilidade e proximidade com o *background*. O cálculo de prioridades utiliza dois termos, $B(p)$ e $E(p)$, que são os termos de *background* e profundidade (definido na Subseção 3.4.1), respectivamente. Então, define-se

$B(p)$ como:

$$B(p) = \frac{\sum_{q \in \Psi_p \cap (\mathcal{I} - \Omega)} bg_{map}(q)}{|\Psi_p|}, \tag{3.10}$$

onde $bg_{map}(q)$ é a máscara que identifica quais pontos do *patch* pertencem ao *background*. Neste termo, *pixels* que estão predominantemente cercados por informação de *background* recebem valores altos. Por outro lado, candidatos pertencentes a região do *foreground* devem ser preteridos. Assim, escolhe-se o ponto que obtém o maior valor de prioridade na equação $P(p) = B(p)E(p)$.

Posteriormente, realiza-se a busca pelo melhor *patch* Ψ_q para o preenchimento de Ψ_p. Nesta proposta, a procura por Ψ_q é realizada na imagem original (antes da projeção), com o objetivo de evitar a propagação de possíveis artefatos introduzidos por incoerências do mapa de disparidades no momento do *warping*. Adicionalmente, definiu-se uma região de busca Φ limitada, com base na premissa de que pontos com textura e cor similares tendem a encontrar-se agrupados e, consequentemente, próximos da região de origem de p. Contudo, a coordenada (x, y) de p foi alterada após o processo de *warping* da imagem real para o ponto correspondente na vista virtual. Então, para centralizar corretamente a janela de busca Φ, o ponto p é reprojetado ao ponto de origem p' realizando o *inverse warping* do ponto em questão. Ainda, pontos pertencentes ao *foreground* são eliminados da região de busca, com base no limiar t definido para o *hole*. Desta forma, Φ é definida como uma janela quadrada de $N \times N$ *pixels* centrada no ponto p' da imagem obtida pelo ponto de vista real, na qual pontos com disparidade superior a t são desconsiderados. O tamanho determinado para janela de busca foi baseado no algoritmo de (AZZARI et al., 2011), o qual visa acelerar o processo de preenchimento através de um espaço de busca limitado. Observa-se ainda, que a estimativa de t para a segmentação da borda do *hole* em *background* e *foreground*, não deve ser generalizada para toda imagem, uma vez que é estimada apenas na sua vizinhança.

Por fim, inicia-se o processo de busca do *patch* Ψ_q, adaptando o seu tamanho de acordo com o valor de erro obtido pela função $s(\Psi_p, \Psi_q)$ no cálculo de similaridade. A função fornece como resultado o valor médio da diferença absoluta de intensidade entre os pontos válidos de Ψ_p e do *patch* analisado, somando os três canais do espaço de cores RGB. Então, a busca é iniciada com um *patch* de 9×9 *pixels*. Quando o valor obtido em $s(\Psi_p, \Psi_q)$ exceder o limiar β de intensidade, o tamanho de Ψ_q e Ψ_p é reduzido em uma razão de dois *pixels*. No caso do *patch* chegar ao tamanho de 3×3 *pixels*, este deve ser aceito mesmo excedendo o erro máximo permitido. Objetiva-se com isto, carregar o

mínimo de erro quando não é possível localizar Ψ_q com a similaridade adequada em Φ. Se o limiar β possuir um valor muito alto, *patches* com baixa similaridade são aceitos, podendo influenciar o algoritmo de preenchimento a procurar futuramente outros *patches* similares a ele, que já possuía baixa similaridade com a região a ser preenchida. Por outro lado, se o limiar for baixo demais, o algoritmo acaba preenchendo boa parte dos *holes* apenas com *patches* de tamanho 3×3, tornando o processo muito mais lento e menos confiável. Em adição, observa-se que dificilmente será encontrado um candidato para o preenchimento com valores de intensidade exatamente iguais, portanto, é plausível permitir uma pequena margem de erro. Para os testes, o valor de erro permitido foi de $\beta = 35$, estipulado de maneira empírica com base em sucessivas verificações. O resultado obtido com a aplicação deste método é exibido na Figura 3.9.

Figura 3.9: Resultado obtido com o método *Adaptative Feature-Oriented Hole-Filling* no preenchimento dos *holes*. (a) Imagem resultante dos processos de tratamento dos artefatos, com os *holes* destacados em vermelho. (b) Resultado obtido com o método de preenchimento proposto.

(a) (b)

Fonte: Adaptado de (MIDDLEBURY, 2016).

3.5 Sumário

Neste capítulo foi apresentado o *pipeline* proposto para a geração de vistas sintéticas utilizando o modelo DIBR. Os algoritmos propostos para a detecção dos *cracks* (em suas duas formas), exploram as características morfológicas do artefato, permitindo uma detecção mais precisa e confiável. Além disso, destaca-se a análise de abrangência dos *cracks*, a qual estimou por meio da análise de 58 imagens diferentes, que o artefato compreende em média cerca de 1,45% dos *pixels* de uma imagem gerada com o modelo.

Este percentual destaca a necessidade do uso de técnicas especializadas no preenchimento deste artefato (uma vez que este não têm as mesmas características dos *holes*), o que não é comum nos trabalhos encontrados na literatura. Para o tratamento dos *ghosts*, foi desenvolvida uma técnica diferente das encontradas na literatura, que baseia-se na origem do artefato, o qual possui em sua intensidade de cor uma mistura de dois objetos que fazem fronteira ou apenas a informação do objeto de menor profundidade entre eles. Desta forma, analisa-se pontos candidatos a *ghost* de acordo a sua informação de vizinhança em ambos os lados do *hole*, e quando o artefato é identificado, o ponto é reconectado ao *foreground*. Por fim, esta dissertação apresenta duas novas abordagens para o preenchimento dos *holes*, as quais baseiam-sem em informação de profundidade para recuperação de informação. A primeira abordagem, baseia-se no preenchimento gradativo dos buracos partindo das regiões de maior profundidade com sentido as de menor, considerando que os *holes*, quase que na sua totalidade são regiões do *background* que eram oclusas por objetos do *foreground*. Este algoritmo, juntamente com as demais técnicas desenvolvidas para o tratamento de artefatos, exceto o algoritmo de detecção dos *cracks* translúcidos, foi publicado em um congresso internacional e o artigo completo está disponível no Apêndice A desta dissertação (OLIVEIRA et al., 2015). Outra abordagem, explorando as características individuais de cada *hole* foi descrita posteriormente, a qual possuí maior controle sobre o preenchimento das regiões sem informação, considerando apenas informação do *background* no processo de preenchimento. Esta proposta reduz significativamente a possibilidade do preenchimento dos *holes* com informação incorreta, devido ao controle mais preciso na busca de conteúdo para a estimativa das regiões sem informação.

4 RESULTADOS EXPERIMENTAIS

Neste capítulo são apresentados os resultados obtidos com as soluções propostas para o preenchimento dos *holes* e tratamento de artefatos para a geração de vistas sintéticas com o modelo DIBR. Na primeira Seção, apresentam-se as métricas e imagens utilizadas para a avaliação dos algoritmos. A Seção 4.2 exibe o resultado visual obtido com os algoritmos de detecção dos *cracks*, e apresenta um comparativo entre diferentes técnicas de preenchimento através de uma análise quantitativa. Posteriormente, na seção 4.3, o método proposto para a identificação e tratamento dos *ghost* é comparado visualmente com outras técnicas apresentadas na literatura. Por fim, na Seção 4.4, apresenta-se um comparativo quantitativo e qualitativo entre as técnicas propostas e os principais algoritmos de *inpainting* que constituem o atual estado da arte no preenchimento dos *holes*. Em todos os experimentos, os valores de parâmetros requeridos pelas técnicas propostas foram aqueles descritos nas seções anteriores, e sumarizados na Tabela 4.1.

Tabela 4.1: Relação de parâmetros dos algoritmos desenvolvidos, com os respectivos valores utilizados nos testes.

Notação	Valor	Descrição
H_{CV}	[1; 1]	Elemento estruturante utilizado na detecção dos *cracks* vazios.
H_{CT}	[1; 1; 1]	Elemento estruturante utilizado na detecção dos *cracks* translúcidos.
λ	5	Limiar que controla a variação máxima de disparidade permitida para *pixels* vizinhos no processo de detecção dos *cracks* translúcidos.
H_G $(D \rightarrow E)$	[0; 1; 1]	Elemento estruturante utilizado na detecção de candidatos a *ghost* quando a imagem é projetada da direita para a esquerda.
H_G $(E \rightarrow D)$	[1; 1; 0]	Elemento estruturante utilizado na detecção de candidatos a *ghost* quando a imagem é projetada da esquerda para a direita.
α	11	Limiar de similaridade, que determina aceitação ou não de um ponto candidato a *ghost*.
α_A	3	Limiar que determina a classificação ou não de um *hole*, de acordo com a área em número de *pixels*.
α_σ	2	Limiar que determina a classificação ou não de um *hole*, de acordo com o desvio padrão de intensidade de disparidade em sua borda.
N	100	Tamanho da janela Φ utilizada na busca por *patches* candidatos ao preenchimento de um *patch*.
β	35	Limiar de erro máximo permitido no comparativo entre dois *patches*.

4.1 Imagens e Métricas para Avaliação

Esta seção tem como objetivo apresentar as imagens utilizadas para a avaliação dos algoritmos, bem como descrever superficialmente as métricas utilizadas para a análise quantitativa. Para a avaliação dos métodos desenvolvidos nesta dissertação foram utilizadas as imagens estéreo dos *datasets* de Middlebury (MIDDLEBURY, 2016), as quais são amplamente utilizadas pela comunidade acadêmica em trabalhos similares. A qualidade das vistas sintéticas geradas foi avaliada quantitativamente utilizando como medidas o PSNR (*Peak Signal-to-Noise Ratio*)[1] e o SSIM (*Structural Similarity Index*)[2] , comparando-se a imagem real no ponto de vista projetado e a imagem sintetizada. O cálculo do PSNR baseia-se em um comparativo ponto a ponto entre os *pixels* das duas imagens, enquanto que o SSIM considera adicionalmente fatores globais, como iluminação e o gradiente, visando identificar mais fielmente a qualidade visual e estrutural da imagem. Adicionalmente, observa-se que foram utilizados os parâmetros *default* no cálculo do SSIM. Para o computo das informações estatísticas e do index SSIM, foi utilizada uma janela 11×11 circular simétrica, com desvio padrão de $1,5$. As constantes $K1$ e $K2$ foram setadas com os valores $0,01$ e $0,03$, respectivamente.

Os *datasets* de imagens estéreo de Middlebury possuem entre 7 e 9 imagens de diferentes pontos de vista para cada cena, obtidas com um conjunto de câmeras alinhado verticalmente, separadas por uma pequena distância horizontal (4cm, mais especificamente). Estas imagens têm entre 413×370 e 465×370 *pixels* de dimensão. Para cada *dataset* são disponibilizados dois mapas de disparidades *ground truth*, em geral para a segunda e sexta vistas. Deste modo, foram selecionados 29 conjuntos de imagens estéreo (*datasets*) do site do projeto Middlebury[3], para análise dos métodos desenvolvidos. As imagens dos pontos de vista que possuem mapa de disparidades *ground truth* (1 e 5 ou 2 e 6, dependendo do *dataset*), são projetadas para a vista intermediária (3 e 4, respectivamente), e posteriormente são aplicados os algoritmos que compõem a solução proposta.

[1] Página para *download* do *software*: <http://www.mathworks.com/matlabcentral/fileexchange/45236-the-maximum-spacing-noise-estimation-in-single-coil-background-mri-data/content/MSP_estimate_version1/QuantitativeMeasures/ImagePSNR.m>

[2] Página para *download* do *software*: <http://www.cns.nyu.edu/lcv/ssim/>

[3] <http://vision.middlebury.edu/stereo/data/>

4.2 Detecção e Preenchimento dos *Cracks*

Para analisar a efetividade dos métodos propostos para a detecção dos *cracks* vazios e translúcidos, realizou-se uma análise visual de sua aplicação nos *datasets* de Middlebury. Observa-se que não existe uma definição de *ground truth* com as ocorrências dos artefatos identificados, portanto não é possível realizar uma avaliação quantitativa dos algoritmos propostos. Desta forma, apresenta-se apenas uma avaliação visual dos resultados obtidos. Posteriormente, realiza-se um comparativo entre o método proposto e os principais algoritmos utilizados no preenchimento deste artefato.

Primeiramente, foi avaliado o resultado obtido na detecção dos *cracks* vazios. Nesta etapa o elemento estruturante H_{CV} foi utilizado com um *pixel* de altura para detectar pontos até mesmo nas extremidades da imagem, por dois de largura que é o padrão máximo definido para um *crack*. A Figura 4.1 apresenta as imagens (a) e (c) antes do processo de identificação (em preto as regiões sem informação de projeção) e, posteriormente, em (c) e (d) estão destacados em azul os pontos com o artefato identificado pelo algoritmo proposto. Ressalta-se, que com a aplicação da técnica proposta não foram encontradas ocorrências do artefato não identificadas.

Posteriormente, avaliou-se o algoritmo proposto para a identificação dos *cracks* translúcidos. Utilizando um elemento estruturante H_{CT} com um *pixel* de altura por três de largura. O tamanho de H_{CT} foi definido para selecionar o maior valor de disparidade entre três *pixels* adjacentes pertencentes a mesma linha. Desta maneira, mesmo que um *crack* ocupe dois *pixels* (largura máxima) no interior do elemento estruturante, o valor selecionado não será o seu, e sim do seu vizinho no *foreground* (com maior disparidade). Os resultados obtidos com o método proposto podem ser visualizados na Figura 4.2. As ocorrências identificadas do artefato estão indicadas em vermelho nas imagens (b) e (d), as quais são obtidas pela aplicação do método proposto nos mapas de disparidade das imagens (a) e (c), respectivamente.

Após a validação dos métodos propostos para identificação, avaliou-se a representatividade do artefato no conteúdo das imagens geradas. O objetivo consiste em medir o seu impacto no processo de geração das imagens sintéticas. Neste passo, foram utilizados os métodos propostos para a detecção de *cracks* translúcidos e vazios, visando medir a quantidade de *pixels* ocupados pelo artefato no conteúdo total da vista sintética. Esta avaliação revelou que 1.45% dos *pixels* na média das 58 imagens analisadas são ocupados pelos *cracks*, destacando-se como uma fração considerável em relação ao conteúdo total.

Figura 4.1: Resultados obtidos com o método de detecção dos *cracks* vazios. (a) Imagem do *dataset* Cones depois do processo de projeção (região sem informação em preto); (b) Resultado obtido com o método proposto, com os pontos detectados indicados em azul; (c) Imagem do *dataset* Teddy projetada; (d) *Cracks* vazios identificados em azul.

(a) (b)

(c) (d)

Fonte: Adaptado de (MIDDLEBURY, 2016).

Visando determinar a melhor abordagem para o preenchimentos de *cracks*, foram avaliados os principais métodos propostos na literatura, comparado-os com a adaptação proposta na Subseção 3.2.2. Em uma primeira etapa, os *Cracks* são detectados (vazios e translúcidos) utilizando os métodos propostos. Após, estes são preenchidos com cada uma das diferentes técnicas e avaliados quantitativamente, considerando apenas os pontos pertencentes ao artefato, utilizando como medida o PSNR. O SSIM não pode ser utilizado nesta etapa, pois analisa a imagem em blocos e não poderia ser medido nestas regiões finas e assimétricas.

Em geral, os *cracks* são caracterizados por possuir em sua vizinhança (de ambos os lados) informação de mesma disparidade. Assim, a sua ocorrência se dá no interior de regiões homogêneas e dificilmente em pontos de transição de objetos. Por tratar-se de uma fenda pequena com informação consistente na vizinhança, abordagens que uti-

Figura 4.2: Resultados obtidos com o método de detecção dos *cracks* translúcidos. (a) Imagem do *dataset* Aloe depois do processo de projeção; (b) Resultado obtido com o método proposto, com os pontos detectados indicados em vermelho; (c) Imagem do *dataset* Wood1 projetada; (d) *Cracks* translúcidos identificados em vermelho.

(a) (b) (c) (d)

Fonte: Adaptado de (MIDDLEBURY, 2016).

lizam dados locais tendem a obter melhores resultados do que métodos que consideram o conteúdo total da imagem para o preenchimento. Este indício é confirmado através da Tabela 4.2, que apresenta melhores resultados para técnicas que consideram informação de vizinhança (como o HHF e o *Fast Inpainting*), em relação aos métodos que utilizam informação global da imagem (como o algoritmo baseado em *patches* de (CRIMINISI; PEREZ; TOYAMA, 2004)).

Tabela 4.2: Desvio padrão (σ) e média (μ) da métrica PSNR do preenchimento dos *cracks*, obtido para 29 *datasets* de (MIDDLEBURY, 2016), gerados com a projeção das vistas 1 e 5 para a vista intermediária 3, somando um total de 58 imagens analisadas. Os melhores resultados encontram-se destacados em negrito.

Método	Vista 1		Vista 5	
	$\mu PSNR$	$\sigma PSNR$	$\mu PSNR$	$\sigma PSNR$
(SOLH; ALREGIB, 2012)	**26,4798**	3,4562	**26,0177**	3,5582
Método Proposto	25,9988	3,3003	25,3359	2,9900
(MORI et al., 2008)	24,5845	3,4169	23,0117	2,7702
(CRIMINISI; PEREZ; TOYAMA, 2004)	24,0003	3,0738	23,5205	2,6961
(ZINGER; DO; WITH, 2010)	23,8006	4,3277	22,7364	3,1797
(TELEA, 2004)	23,1527	2,6430	22,9419	2,0891

Na análise dos resultados da Tabela 4.2 é possível identificar uma pequena vantagem (pouco mais de 0,5dB) obtida pelo método HHF de (SOLH; ALREGIB, 2012) em relação ao algoritmo proposto. Ressalta-se que este valor é pouco expressivo considerando a diferença em relação às outras técnicas. Contudo, a abordagem proposta apresenta um desempenho computacional superior ao método HHF. Em média, o método proposto

consome 0,18 segundos para preencher os *cracks* em cada imagem (com apenas 4 iterações), enquanto o algoritmo HHF, 5,2 segundos, ou seja, o método proposto consome apenas $3,46\%$ do tempo utilizado pela técnica concorrente. Portanto, ressalta-se que a adaptação do método de (OLIVEIRA et al., 2001) apresenta bons resultados qualitativos para no preenchimento dos *cracks*, além de ser significativamente mais rápido no comparativo com o seu principal concorrente, que é o método HHF. Destaca-se ainda, que artefatos visuais não foram encontrados nas imagens que utilizaram o método proposto.

4.3 Identificação e Tratamento dos *Ghosts*

O algoritmo proposto para a identificação e tratamento dos *ghosts* é avaliado visualmente quanto ao seu impacto na geração de vistas sintéticas. Apresenta-se nesta seção, um comparativo entre o método desenvolvido e as principais abordagens encontradas na literatura. Além disso, analisa-se os diferentes tipos de tratamento do artefato, relacionando suas características.

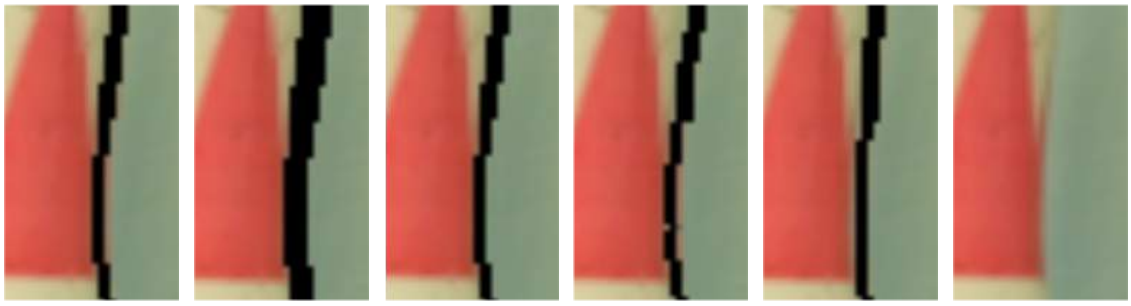
Os *ghosts* são decorrentes da imperfeita definição da borda dos objetos em cena, que resulta na projeção incorreta desta região de transição para a extremidade errada do *hole*. Este erro mantém a silhueta do objeto junto a região do *background*, quando na verdade deveria estar conexa com o *foreground*. Portanto, a melhor abordagem é remanejar este pontos para o local correto, pois trata-se apenas de uma inconsistência do mapa de disparidades. Nesse contexto, o método proposto baseia-se em um processo de análise e tratamento de um conjunto de pontos candidatos a *ghost*. Quando é confirmada a ocorrência (mediante avaliação) do artefato, o *pixel* identificado é reprojetado para a outra extremidade do *hole* (*foreground*). Em todos os experimentos o algoritmo proposto foi utilizado com $\alpha = 11$ como *threshold*, o qual foi determinado de forma empírica.

A presença de informação inválida na extremidade do *hole* pode prejudicar o processo de preenchimento dos buracos, pois os algoritmos de *inpainting* convencionalmente utilizam informação vizinha a região alvo como base para suas operações. Deste modo, abordagens como a proposta por (ZINGER; DO; WITH, 2010), que podem falhar na detecção das regiões pertencentes ao *ghost* (como no caso exibido pela Figura 4.3(d)), podem comprometer o resultado dos algoritmos de preenchimento, pois não detectam o artefato em sua completude. Por outro lado, algoritmos que visam remover qualquer candidato a *ghost*, como no trabalho de (OH; YEA; HO, 2009) (Figura 4.3(b)), acabam impactando no resultado final da vista sintética por retirar informação válida da imagem,

o que acarreta no uso desnecessário de algoritmos de preenchimento que precisam estimar ainda mais conteúdo. Ainda, mesmo que o *hole* seja preenchido com informação correta, a transição dos objetos em cena acaba perdendo a suavidade.

(MUDDALA, 2015) estima valores para as regiões onde podem existir *ghosts* ainda na imagem original, alterando seu valor de intensidade de acordo com valores selecionados da vizinhança. Esta abordagem apresenta resultados similares aos obtidos com o algoritmo proposto, no entanto altera o valor de intensidade original dos pontos de borda, e torna mais abrupta a transição entre os objetos na cena. A Figura 4.3 apresenta um comparativo entre os resultados obtidos com as diferentes técnicas analisadas, onde pode ser observado que o algoritmo proposto representa selecionar todos os pontos pertencentes ao *ghost*, mantendo a suavidade da borda do objeto. Por fim, destaca-se que o resultado quantitativo obtido pela técnica proposta é superior às demais, quando empregada no *pipeline* de geração de vistas sintéticas proposto.

Figura 4.3: Comparativo entre os algoritmos para remoção e/ou tratamento de *ghosts*. (a) Imagem da esquerda projetada para a vista sintética com a presença de um *ghost*; (b) Resultado obtido com o método proposto por (OH; YEA; HO, 2009); (c) Resultado do algoritmo de (MUDDALA, 2015); (d) Resultado da técnica proposta por (ZINGER; DO; WITH, 2010); (e) Resultado obtido com o algoritmo proposto; (f) *ground truth*, imagem real do ponto em que a vista sintética foi projetada.



(a) (b) (c) (d) (e) (f)

Fonte: Adaptado de (MIDDLEBURY, 2016).

4.4 Preenchimento dos *Holes*

Nesta seção, analisa-se o resultado obtido pelas duas abordagens propostas para o preenchimento dos *holes*. Estas são comparadas com o resultado produzido pelos métodos de geração de vistas sintéticas de (SOLH; ALREGIB, 2012) e (DARIBO; SAITO, 2011). Adicionalmente, compara-se todos os métodos com o algoritmo de *inpainting* de (CRIMINISI; PEREZ; TOYAMA, 2004), visto que as abordagens propostas e o método

desenvolvido por (DARIBO; SAITO, 2011) derivaram desta técnica. Para quantificar os resultados obtidos com a aplicação dos diferentes métodos utilizou-se como métricas o PSNR e o SSIM.

Neste trabalho, os resultados foram mensurados desconsiderando a faixa lateral sem informação da imagem (destacada em preto na Figura 4.4 (a)), a qual encontra-se fora do campo de vista da câmera utilizada para obter a imagem de referência para projeção. Muddala (2015), define estas regiões como *out-of-field areas*, e trata-as sem considerar profundidade em seu algoritmo de *inpainting*, devido a falta de informação do *foreground*. Partindo do pressuposto de que estas regiões não possuem conteúdo consistente na vizinhança para a recuperação de cor e textura, adicioná-las no comparativo tornaria inconsistente a avaliação dos métodos. Em adição, observa-se que estas áreas não apresentam as mesmas características dos *holes*, os quais encontram-se circundados por informação consistente, que pode ser utilizada como orientação para o preenchimento. Daribo and Saito (2011), propõem o envio adicional de uma parte da imagem obtida por um segundo ponto de vista da cena (lado oposto ao transmitido), justamente para preencher (mesmo que parcialmente) esta e outras regiões grandes sem informação. As Figuras 4.4 (c) e (d) exibem o conteúdo transmitido pela abordagem de (DARIBO; SAITO, 2011). Com o mesmo objetivo, outras abordagens utilizam um modelo que transmite simultaneamente duas imagens. Estas são projetadas para o ponto de vista virtual e posteriormente combinadas formando uma vista com pequenos buracos, reduzindo consideravelmente as regiões sem informação, como é o caso de (ZINGER; DO; WITH, 2010) e (MORI et al., 2008).

Desta forma, para a avaliação dos resultados quantitativos, executam-se os métodos de preenchimento nas 58 imagens dos *datasets* analisados e, posteriormente, aplica-se uma máscara para remover os pontos pertencentes a faixa lateral de cada uma delas. A faixa lateral é determinada de acordo com a composição da região sem informação na extremidade inversa ao sentido de projeção de cada uma das imagens geradas, ou seja, se a imagem foi projetada para a esquerda, esta região é identificada no lado direito e vice-versa. Por tratar-se de uma métrica que calcula as diferenças ponto-a-ponto entre duas imagens, o PSNR é medido apenas nos *pixels* não pertencentes a máscara que compreende a faixa lateral (identificada em preto na Figura 4.4(a)). No entanto, para medir o SSIM é necessário recortar parte da imagem (como pode ser visto na Figura 4.4(b)), pois este considera algumas informações globais da imagem como, por exemplo, distribuição da iluminação, assim não permitindo a estimativa de similaridade para cada ponto

individualmente.

Figura 4.4: As imagens (a) e (b) representam a aplicação da máscara com os pontos que são desconsiderados – em preto – na medição do PSNR e SSIM, respectivamente. Em (c) é apresentada uma imagem do *dataset* Balet de (ZITNICK et al., 2004) projetada para um ponto de vista virtual. Em (d) é exibida a camada residual que é transmitida em complemento para o preenchimento dos buracos de (c), seguindo a abordagem definida por (DARIBO; SAITO, 2011).

(a) (b) (c) (d)

Fonte: As imagens (a) e (b) foram adaptadas de (MIDDLEBURY, 2016), enquanto as outras duas foram retiradas de (DARIBO; SAITO, 2011).

As Tabelas 4.3 e 4.4 apresentam os resultados quantitativos obtidos pelos métodos analisados, para as medidas PSNR e SSIM, exibindo a média e desvio padrão da aplicação em 58 imagens estéreo dos *datasets* de (MIDDLEBURY, 2016). Nestas tabelas, é possível identificar a superioridade dos métodos desenvolvidos neste trabalho, nas quais estes apresentam os melhores resultados considerando as duas medidas analisadas. Mais especificamente, destaca-se que os melhores resultados são apresentados pelo método *Adaptative Feature-Oriented Hole-Filling*, o qual é decorrente de uma evolução do primeiro método proposto para o preenchimento dos *holes*, denominado *Selective Hole-Filling*. Justifica-se esta vantagem pela manipulação de ordem e seleção de *patches* desenvolvida nesta segunda abordagem, que não permite a inserção de informação do *foreground* no interior dos *holes*, baseando-se nas características individuais de cada região sem informação.

Tabela 4.3: Média (μ) e desvio padrão (σ) das métricas PSNR e SSIM, obtidos com a aplicação dos diferentes métodos analisados em 29 *datasets* de (MIDDLEBURY, 2016). Os resultados foram medidos projetando a imagem real do ponto de vista 1 para o ponto de vista virtual 3. A imagem gerada é comparada com o *ground truth* (imagem real do ponto de vista 3). Os melhores resultados encontram-se destacados em negrito.

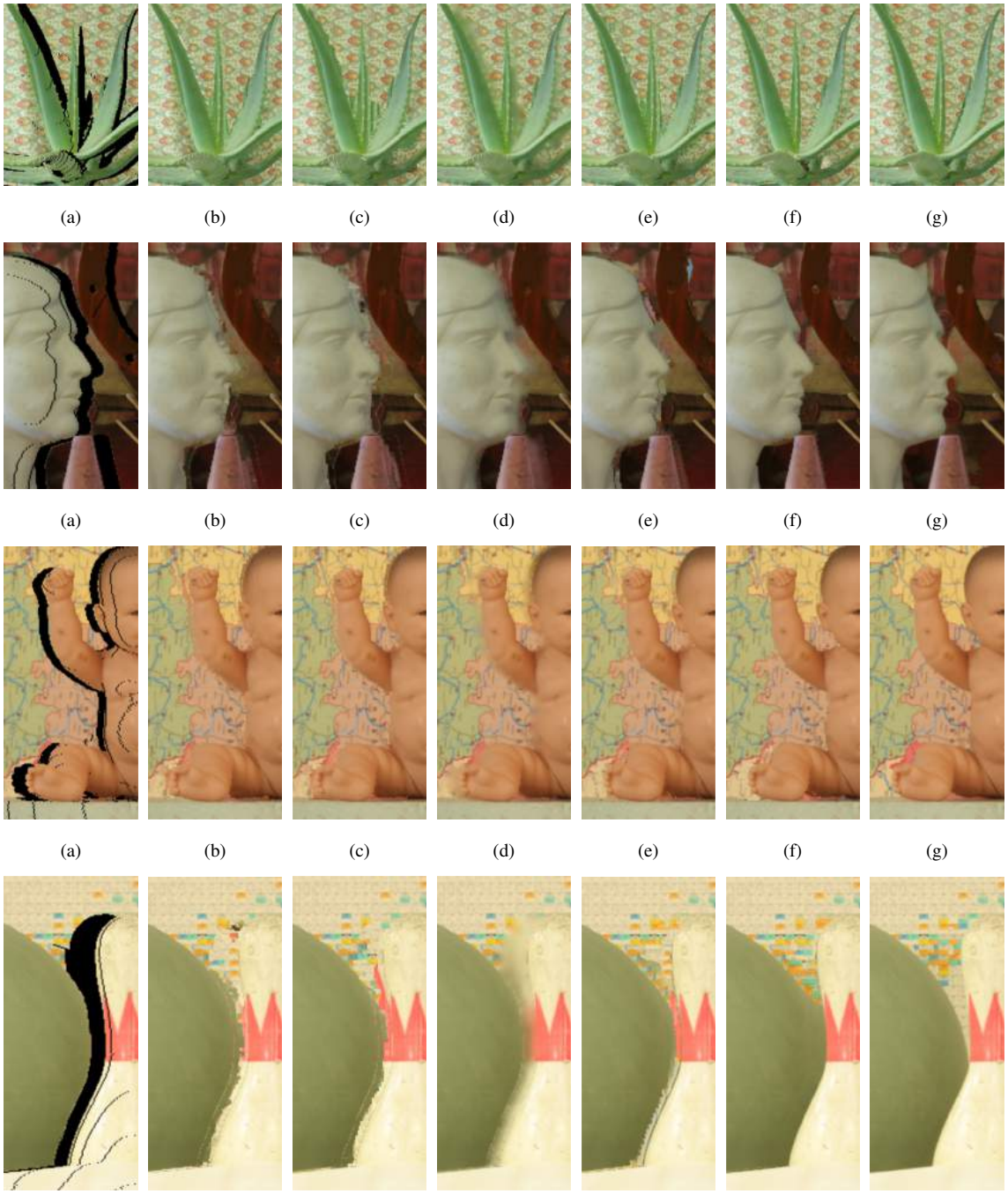
Método	$\mu PSNR$	$\sigma PSNR$	$\mu SSIM$	$\sigma SSIM$
Adaptative Feature-Oriented Hole-Filling	**30,4749**	3,2526	**0,9404**	0,0239
Selective Hole-Filling	30,1078	3,0956	0,9358	0,0258
(SOLH; ALREGIB, 2012)	29,8489	3,1498	0,9352	0,0267
(CRIMINISI; PEREZ; TOYAMA, 2004)	28,8250	3,3996	0,9233	0,0307
(DARIBO; SAITO, 2011)	28,8247	3,3650	0,9229	0,0304

Tabela 4.4: Média (μ) e desvio padrão (σ) das métricas PSNR e SSIM, obtidos com a aplicação dos diferentes métodos analisados em 29 *datasets* de (MIDDLEBURY, 2016). Os resultados foram medidos projetando a imagem real do ponto de vista 5 para o ponto de vista virtual 3. A imagem gerada é comparada com o *ground truth* (imagem real do ponto de vista 3). Os melhores resultados encontram-se destacados em negrito.

Método	$\mu PSNR$	$\sigma PSNR$	$\mu SSIM$	$\sigma SSIM$
Adaptative Feature-Oriented Hole-Filling	**30,2676**	3,3734	**0,9407**	0,0250
Selective Hole-Filling	29,9231	3,2704	0,9373	0,0259
(SOLH; ALREGIB, 2012)	29,4403	3,0805	0,9373	0,0256
(CRIMINISI; PEREZ; TOYAMA, 2004)	28,1992	3,0850	0,9245	0,0297
(DARIBO; SAITO, 2011)	28,2198	3,1947	0,9242	0,0301

A Figura 4.5 apresenta o comparativo visual entre as diferentes técnicas analisadas nesta seção. Identifica-se na primeira coluna (a) a imagem do ponto de vista real projetada para o virtual (ainda com os *holes* e artefatos), enquanto os resultados obtidos pelos métodos de (CRIMINISI; PEREZ; TOYAMA, 2004), (DARIBO; SAITO, 2011), (SOLH; ALREGIB, 2012), e dos algoritmos propostos nas Subseções 3.4.1 e 3.4.2, são exibidos nas colunas (b), (c), (d), (e) e (f), respectivamente. A imagem da coluna (g) exibe o *ground truth* para cada uma das linhas. Nestas figuras são exibidos, especificamente, os resultados obtidos para os *datasets* Aloe, Art, Baby1 e Bowling1 em cada linha, nesta ordem. Observa-se ainda, que as figuras representam apenas parte das imagens processadas, para enfatizar o resultado obtido no preenchimento de grandes regiões sem informação. Visualmente, é possível identificar nas colunas (b) e (c), com os resultados gerados pelos algoritmos (CRIMINISI; PEREZ; TOYAMA, 2004) e (DARIBO; SAITO, 2011), a presença recorrente de artefatos. Isto se dá porque estes algoritmos tem como objetivo manter o formato dos objetos dentre suas prioridades (através do termo de dados). Desta forma, o termo de dados pode conduzir ao preenchimento incorreto dos *holes*, pois podem ser preenchidos com base em informação *foreground*, o que produz um resultado incorreto, uma vez que estas regiões pertencem ao *background*. A coluna (c) exibe o resultado obtido pelo método de (SOLH; ALREGIB, 2012), no qual a informação de preenchimento gerada por seu algoritmo assemelha-se a um borramento, proveniente da mistura de informação das extremidades dos *holes*. Por fim, nas colunas (d) e (e) são apresentadas as imagens produzidas com as técnicas propostas, descritas nas Subseções 3.4.1 e 3.4.2, respectivamente. No resultado exibido na coluna (d), com o primeiro método proposto, pode-se visualizar a ocorrência de artefatos em alguns casos, onde mesmo priorizando a região do *background*, o mesmo acaba relacionando informação do *foreground* quando as duas extremidades do *hole* encontram-se próximas. Por outro lado, a proposta apresentada

Figura 4.5: Resultados obtidos com a aplicação de diferentes técnicas no preenchimento dos *holes*. Apresenta-se em destaque os resultados obtidos com os *datasets* Aloe, Art, Baby1 e Bowling1 em cada linha, respectivamente. Os métodos avaliados encontram-se distribuídos na colunas utilizando a seguinte ordem: (a) imagem projetada (*holes* em preto); (b) (CRIMINISI; PEREZ; TOYAMA, 2004); (c) (DARIBO; SAITO, 2011); (d) (SOLH; ALREGIB, 2012); (e) método proposto na Subseção 3.4.1; (f) técnica apresentada na Subseção 3.4.2 e (g) *ground truth*.



Fonte: Imagens adaptadas de (MIDDLEBURY, 2016).

na coluna (e) evidencia a presença pouco significativa (praticamente imperceptível) de artefatos visuais incoerentes. No entanto, em alguns casos, como o do *dataset* Bowling1 (quarta linha), este acaba distorcendo objetos – como a bola verde – em direção ao outro lado da região sem informação, devido ao menor valor de profundidade, sendo assim considerado como o objeto a ser prolongado para a completude do *hole*, perdendo o formato. Todos os resultados obtidos com a aplicação das técnicas nas 58 imagens analisadas podem ser visualizados no endereço <www.inf.ufrgs.br/~mwalter/dissertacaoadriano.html>.

4.5 Sumário

Neste capítulo foram apresentados os resultados obtidos com os métodos propostos para a geração de imagens sintéticas utilizando o modelo DIBR. Inicialmente, destaca-se que o método proposto para a detecção dos *cracks* mostrou-se eficaz, uma vez que não foram identificadas visualmente ocasiões onde o artefato não foi localizado – considerando suas duas formas (vazia e translúcida) – em todas as imagens testadas. Além disso, não foram identificadas falsas detecções, o que justifica-se pela análise morfológica empregada pelos métodos. Já o algoritmo proposto para o preenchimento deste artefato, foi comparado com os métodos que formulam o atual estado da arte para tal tarefa. No comparativo, a técnica utilizada apresentou resultados satisfatórios, principalmente considerando a sua relação velocidade $\times$ qualidade de preenchimento. Posteriormente, analisou-se a abordagem proposta para a detecção e tratamento dos *ghosts*. O método foi elaborado com base nas características que originam o artefato, apresentando-se como a melhor metodologia de tratamento quando comparada com as outras abordagens apresentadas na literatura. Isto ocorre devido a sua capacidade de manter a suavidade de transição entre os objetos da cena, sem remover nenhum *pixel* da imagem. Por fim, os métodos desenvolvidos para o preenchimento dos *holes* foram comparados com os algoritmos que compõem o atual estado da arte. No comparativo com os demais métodos, a primeira abordagem desenvolvida (Subseção 3.4.1) mostrou-se superior as técnicas concorrentes, obtendo uma superioridade de $0,37db$ no PSNR e $0,0003$ considerando o SSIM na média das 58 imagens analisadas. Contudo, a segunda abordagem (Subseção 3.4.2) apresenta resultados ainda melhores, obtendo uma vantagem de $0,36db$ no PSNR e $0,004$ no SSIM sobre a primeira proposta. Observa-se adicionalmente, que a abordagem proposta na Subseção 3.4.2 apresenta os melhores resultados visuais, por não gerar artefatos no preenchimento, além de estimar de forma mais ordenada e precisa o conteúdo dos *holes*, o

que justifica os resultados quantitativos obtidos.

5 CONCLUSÕES E TRABALHOS FUTUROS

A geração de imagens sintéticas com o modelo DIBR tem recebido bastante atenção tanto da indústria quanto da comunidade acadêmica, resultando em uma série de trabalhos focados principalmente no preenchimento de regiões sem informação (*holes* e *cracks*). No entanto, boa parte destes trabalhos desconsidera os limites de banda disponíveis para a transmissão televisiva atual, mostrando-se infactíveis para tal aplicação. Estes, justificam o uso de duas ou mais imagens (com os respectivos mapas de disparidade) pela dificuldade encontrada no preenchimento de grandes regiões sem informação, as quais são preenchidas (parcial ou completamente) pela combinação dos outros pontos de vista da cena. Contudo, observa-se que alguns trabalhos apresentados na literatura demonstram ser possível reproduzir diferentes pontos de vista de uma mesma cena (mantendo a qualidade), utilizando apenas a imagem do ponto de vista real da cena acompanhado do seu respectivo mapa de disparidades.

Desta forma, foi apresentado nesta dissertação um novo método para a geração de vistas sintéticas com o modelo DIBR, o qual utiliza como entrada uma imagem real e o respectivo mapa de disparidades (abordagem V+D). A partir deste modelo de entrada, o algoritmo é capaz de renderizar inúmeros pontos de vista para a mesma cena, apenas variando o *baseline* entre a câmera real e o ponto escolhido (considerando apenas deslocamentos horizontais). O primeiro passo do algoritmo trata da projeção da imagem real para o ponto de vista virtual. Posteriormente, aplica-se diferentes abordagens para a detecção dos *ghosts* e *cracks*, as quais são baseadas em operadores morfológicos definidos de acordo com o padrão de cada artefato. Em seguida, os *ghosts* identificados são reprojetados para o local correto e os *cracks* preenchidos com o algoritmo de (OLIVEIRA et al., 2001). Por fim, aplica-se uma adaptação do algoritmo de *inpainting* baseado em exemplares de (CRIMINISI; PEREZ; TOYAMA, 2004) nos *holes*.

O método proposto foi avaliado em cada uma das suas etapas de execução. A detecção dos *cracks* foi analisada visualmente quanto a sua capacidade de identificação do artefato nos dois formatos, comprovando sua eficácia. Em adição, o processo avaliativo destacou o percentual significativo representado pelo artefato no processo de geração de imagens sintéticas, enfatizando técnicas que realizam seu preenchimento em um processo separado dos *holes*. Após, exibiu-se uma análise quantitativa de diferentes técnicas propostas para o preenchimento dos *cracks*. A identificação e tratamento dos *ghosts* foi examinada visualmente, destacando a coerência e eficiência apresentadas pela técnica

desenvolvida, perante as demais propostas encontradas na literatura. Por fim, os resultados obtidos com as técnicas propostas para o preenchimento dos *holes* foram avaliadas de forma individual e no contexto da geração de imagens sintéticas, onde mostraram-se superiores aos resultados apresentados no atual estado da arte.

Alguns desafios foram encontradas durante o desenvolvimento deste trabalho. Dentre estes, destaca-se inicialmente a ausência de uma definição precisa das características dos artefatos gerados após o *warping* da imagem (*ghosts* e *cracks*). Poucos trabalhos abordam estes problemas, pois consideram apenas o preenchimento dos *holes*. Ainda, notou-se durante a pesquisa que estes variam de tamanho e disposição de acordo com a qualidade do mapa de disparidades fornecido como entrada para o algoritmo. Desta forma, foi necessário o desenvolvimento de técnicas adaptáveis as diferentes condições que os artefatos podem apresentar. Outro problema encontrado, foi a dificuldade de uma estimativa plausível de cor e textura para o preenchimento dos *holes*. Então, para tal problema foi proposta a adaptação do algoritmo de *inpainting* baseado em exemplares. Contudo, a determinação da melhor maneira de influenciar a ordem de preenchimento dos *holes* com informação de profundidade foi bastante complexa, onde foram avaliadas diversas combinações de termos de prioridade, até o estabelecimento de uma combinação ideal. Adicionalmente, realizaram-se testes com diversas variações do emprego da diferença de profundidade para o cálculo de similaridade entre exemplares, no entanto constatou-se que esta não se comporta bem quando inserida neste processo.

Como trabalho futuro espera-se o aprimoramento do *pipeline* proposto para aplicação em vídeos, com a renderização de imagens sintéticas para pontos arbitrários (da cena visualizada) em tempo de execução. O primeiro passo para isto, consiste na implementação paralela do algoritmo proposto em CUDA (*Compute Unified Device Architecture*), para execução em GPU (*Graphics Processing Unit*). Adicionalmente, deve ser desenvolvida uma técnica com o objetivo de manter a consistência temporal durante a troca de *frames* na execução de vídeos.

REFERÊNCIAS

ACHANTA, R. et al. Slic superpixels compared to state-of-the-art superpixel methods. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 11, p. 2274–2282, Nov 2012.

AZZARI, L. et al. A modified non-local mean inpainting technique for occlusion filling in depth-image-based rendering. In: . [S.l.: s.n.], 2011. v. 7863, p. 78631C–78631C–13.

BERTALMIO, M. et al. Image inpainting. In: **Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques**. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000. (SIGGRAPH '00), p. 417–424.

CRIMINISI, A.; PEREZ, P.; TOYAMA, K. Region filling and object removal by exemplar-based image inpainting. **IEEE Transactions on Image Processing**, v. 13, n. 9, p. 1200–1212, 2004.

DARIBO, I.; SAITO, H. A novel inpainting-based layered depth video for 3dtv. **IEEE Transactions on Broadcasting**, v. 52, n. 2, p. 533–541, 2011.

FEHN, C. Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. In: . [S.l.: s.n.], 2004. v. 5291, p. 93–104.

FICKEL, G. P. **Video view interpolation using temporally adaptive 3D meshes**. Thesis (PhD) — Federal University of Rio Grande do Sul, Institute of Informatics, Sep 2015.

GAUTIER, J.; MEUR, O. L.; GUILLEMOT, C. Depth-based image completion for view synthesis. In: **3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011**. [S.l.: s.n.], 2011. p. 1–4.

KöPPEL, M.; MüLLER, K.; WIEGAND, T. Filling disocclusions in extrapolated virtual views using hybrid texture synthesis. **IEEE Transactions on Broadcasting**, PP, n. 99, p. 1–13, 2016.

MARK, W. R. **Post-Rendering 3D Image Warping: Visibility, Reconstruction, and Performance for Depth-Image Warping**. Thesis (PhD) — University of North Carolina, Department of Computer Science, April 1999.

MCMILLAN, L. **An Image-Based Approach to Three-Dimensional Computer Graphics**. Thesis (PhD) — University of North Carolina, Department of Computer Science, 1997.

MIDDLEBURY. **Middlebury's Database**. 2016. Disponível em: <vision.middlebury.edu/stereo/>. Acessado em: 31 maio 2016.

MORI, Y. et al. View generation with 3d warping using depth information for ftv. In: **2008 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video**. [S.l.: s.n.], 2008. p. 229–232.

MUDDALA, S. M. **Free View Rendering for 3D Video: Edge-Aided Rendering and Depth-Based Image Inpainting**. Thesis (PhD) — Mid Sweden University, Department of Information and Communication Systems, June 2015.

MUDDALA, S. M.; OLSSON, R.; SJÖSTRÖM, M. Disocclusion handling using depth-based inpainting. In: **Fifth International Conferences on Advances in Multimedia; MMEDIA 2013; Apr 21-26 2013; Venice, Italy**. [S.l.: s.n.], 2013. p. 136–141.

OH, K.-J.; YEA, S.; HO, Y.-S. Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video. In: **Picture Coding Symposium, 2009. PCS 2009**. [S.l.: s.n.], 2009. p. 1–4.

OLIVEIRA, A. et al. Selective hole-filling for depth-image based rendering. In: **2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2015. p. 1186–1190.

OLIVEIRA, M. M. et al. Fast digital image inpainting. In: **Proceedings of the Internacional Conference on Visualization, Imaging and Image Processing (VIIP 2001)**. [S.l.]: ACTA Press, 2001. p. 261–266.

PURICA, A. I. et al. Improved view synthesis by motion warping and temporal hole filling. In: **2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.: s.n.], 2015. p. 1191–1195.

SCHARSTEIN, D.; SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. **International journal of computer vision**, Springer, v. 47, n. 1-3, p. 7–42, 2002.

SCHARSTEIN, D.; SZELISKI, R. High-accuracy stereo depth maps using structured light. In: **Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on**. [S.l.: s.n.], 2003. v. 1, p. I–195–I–202 vol.1.

SCHMEING, M.; JIANG, X. Faithful disocclusion filling in depth image based rendering using superpixel-based inpainting. **IEEE Transactions on Multimedia**, v. 17, n. 12, p. 2160–2173, Dec 2015.

SMOLIC, A. et al. Intermediate view interpolation based on multiview video plus depth for advanced 3d video systems. In: **2008 15th IEEE International Conference on Image Processing**. [S.l.: s.n.], 2008. p. 2448–2451.

SOLH, M.; ALREGIB, G. Hierarchical hole-filling for depth-based view synthesis in ftv and 3d video. **IEEE Journal of Selected Topics in Signal Processing**, v. 6, n. 5, p. 495–504, 2012.

TELEA, A. An image inpainting technique based on the fast marching method. **Journal of Graphics Tools**, v. 9, n. 1, p. 23–34, Jan 2004.

TRAN, A.; HARADA, K. Depth based view synthesis using graph cuts for 3dtv. Journal of Signal and Information Processing, v. 4, p. 327–335, 2013.

XU, X. et al. Depth-aided exemplar-based hole filling for dibr view synthesis. In: **2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)**. [S.l.: s.n.], 2013. p. 2840–2843.

YANG, X. et al. Dibr based view synthesis for free-viewpoint television. In: **3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011**. [S.l.: s.n.], 2011. p. 1–4.

ZHU, C.; LI, S. Depth image based view synthesis: New insights and perspectives on hole generation and filling. **IEEE Transactions on Broadcasting**, v. 62, n. 1, p. 82–93, March 2016.

ZINGER, S.; DO, L.; WITH, P. H. N. de. Free-viewpoint depth image based rendering. **Journal of Visual Communication and Image Representation**, v. 21, n. 5-6, p. 533–541, 2010.

ZITNICK, C. L. et al. High-quality video view interpolation using a layered representation. In: **ACM SIGGRAPH 2004 Papers**. New York, NY, USA: ACM, 2004. (SIGGRAPH '04), p. 600–608.

APÊNDICE A – ARTIGO PUBLICADO - ICASSP 2015

- Título: SELECTIVE HOLE-FILLING FOR DEPTH-IMAGE BASED RENDERING
- Conferência: 40th^{o} IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)
- Seção: 3D Processing
- URL: http://icassp2015.org/
- Data: 19-24 de Abril de 2015
- Local: Brisbane, Australia

SELECTIVE HOLE-FILLING FOR DEPTH-IMAGE BASED RENDERING

Adriano Oliveira, Guilherme Fickel, Marcelo Walter, Cláudio Jung

Institute of Informatics, Federal University of Rio Grande do Sul

ABSTRACT

One of the biggest challenges in view interpolation is to fill the regions without projective information in the synthesized view. In this paper, we present a new approach that identifies and corrects different types of missing information. In the first stage, we propose a fast solution to tackle the problems of cracks and ghost, common artifacts in the view interpolation process. Then, we complete larger holes by exploring the disparity map as an additional cue to select the best patch in a patch-based inpainting procedure. Our experimental results indicate that we were able to outperform current state of the art hole filling techniques for view interpolation.

Index Terms— View interpolation, hole filling, DIBR, view synthesis, inpainting

1. INTRODUCTION

With the recent development of 3D displays, much more content is now generated using multiple cameras. This fostered the research of DIBR (Depth Image Based Rendering) techniques, which consists of using a single reference image (usually from the left camera) with its respective disparity map to generate another synthesized view, usually a reconstruction of the second camera.

DIBR techniques are of great importance for stereo video encoding since they can greatly reduce the required bandwidth by only encoding one color (reference) image and its respective grayscale disparity map, instead of two color images. And by having the ability to generate other synthesized views with arbitrary baselines, many applications may be possible such as free viewpoint camera where the spectator can change in real time the desired view of the scene, baseline retargeting to adapt the stereo baseline depending on the characteristics of each display, etc.

However, the generation of interpolated views presents several challenges, and visual artifacts are common. In this paper we directly deal with three of the most common classes of visual artifacts: cracks, which are generated due to the quantization of the disparity map, ghosts, which occur when we have a disparity discontinuity that is not well defined in the image domain, and holes, that are larger areas of unprojected data due to occlusions and/or errors in the disparity map. The pipeline of our approach can be seen in Figure 1.

2. RELATED WORK

The crack holes are long and thin, usually 1 to 2 pixels wide. To solve this problem some techniques estimate the disparity of the synthetic view and fill the cracks in the disparity with some simple filtering procedures [1, 2], such as a median filter. Another solution is to apply a simple filtering or inpainting algorithm directly to the synthetic view cracks [3].

The holes caused by disocclusions, on the other hand, are usually large both in length and width. Inpainting methods that propagate information through diffusion [4], even if they propagate incoming edges such as [5], are not able to propagate complex textures. To coherently fill those holes a more robust solution is needed. Mori and colleagues [2] proposed to project both the left and right views in the synthetic view position. Both views are combined using an alpha blending procedure, and the holes from one projection are mostly completed by the information of the other. The remaining holes are usually small, and a simple inpainting algorithm can be used to estimate them. However, since this procedure uses both stereo views and depth maps, it is not suitable for DIBR.

An interesting inpainting solution with texture synthesis was proposed by Criminisi and colleagues [6], which fills the holes by copying patches from the available image. They showed that by filling the holes within a certain order, prioritizing first complete regions that had strong edges intersected with the hole, their approach was able to correctly synthesize textures while propagating the edges. By recognizing that holes generated by occlusion typically belong to the background, Daribo and Saito [7] proposed to change the priority function from [6] to fill the holes starting from the background. By using the disparity information, they were able to correctly propagate the background texture information to fill the holes and achieve more coherent results. Oh and colleagues [1] also proposed to use the disparity information in the inpaint procedure. They adapted the inpaint order from Telea's algorithm [8], but the results are not very good in large areas. This is due to the limitations of Telea's algorithm that, similarly to Bertalmio's approach, propagates color information as well as edges but is not able to propagate complex textures. Hervieu and colleagues [9] presented a two-stage process: in the first one, the disparity map is inpainted, and used as a basis to inpaint the stereo pair using an extension of [6]. Mao et al. [10] presented an approach for

identifying expansion holes, and two methods for correcting them: the first one, based on linear interpolation, is very simple and fast; the second one, based on graphs with a sparsity prior, is better but more expensive computationally.

A different approach was developed by Solh and AlRegib [11], called Hierarchical Hole-Filling (HHF). They produce pyramid-like lower resolution estimates of the synthetic view with holes by taking the mean between blocks of 5×5 of the valid pixels (i.e. ignoring pixels without projection information), and propagating it to one pixel in the next scale. Within a few multi-resolution scales they obtain a low resolution estimative of the synthetic view without holes. By propagating this low resolution image along the multi-scale structure they estimate the holes in the original image.

Solh and AlRegib also proposed to use a pre-processed image as the input for the HHF algorithm. This algorithm is called depth adaptive HHF, and the pre-processed image is the synthetic view with holes weighted according to the disparity. The main idea is to give a higher importance (weight) for lower disparity regions since they belong to the background, and holes ideally should be completed by background pixels. However the depth adaptive HHF has a minor impact in the final results over the original HHF.

3. THE PROPOSED APPROACH

3.1. Cracks Removal

To tackle the artifacts caused by cracks, our first step is to correctly identify them. We compute a binary image S that contains all the pixels in the synthetic view that do not have any projection information. Then, S is filtered with a morphological opening operation using a structuring element H_C, resulting in a filtered image called $\hat{S}$. In this work, we used H_C with a horizontal line format, with length of 1 pixel and width of 2 pixels, and the inverse is used to verify vertical small holes.

Image $\hat{S}$ fills out thin vertical lines of S, so that the binary mask C containing all the cracks can be found by:

$$C = S \setminus \hat{S}, \tag{1}$$

with $\setminus$ being the absolute complement operator in set terminology. The result of this identification process can be seen in the block diagram (Fig. 1), where the cracks are painted red in the images that illustrates the "fill cracks" stage.

After identifying all the cracks in the synthesized image, we use a fast inpainting procedure proposed by Oliveira and colleagues [4]. Let Ω be the crack to be inpainted and $\partial\Omega$ its boundary, the inpainting procedure is approximated by an isotropic diffusion that propagates the information from $\partial\Omega$ to Ω. Initially, the color information of Ω is cleared and the diffusion process is approximated by repeatedly convolving the region to be inpainted with the diffusion kernel shown in Fig. 2.

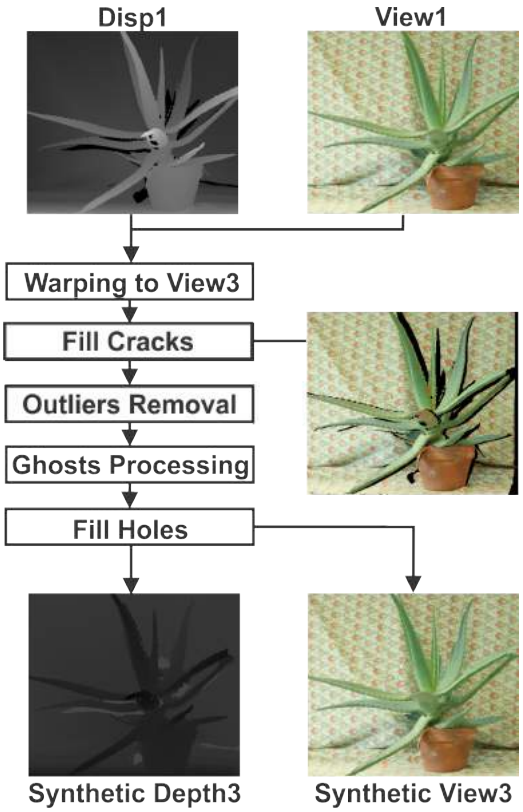


Fig. 1. Block diagram of the proposed algorithm.

a	b	a
b	0	b
a	b	a

Fig. 2. Diffusion kernel used, with $a = 0.073235$ and $b = 0.176765$.

This simple approach may introduce blurring when Ω crosses the boundaries of high contrast edges. In practice, however, the cracks are usually only 1 or 2 pixels wide (as defined by the structuring element), so only a small number of iterations is needed, and the resulting blurring artifacts are not noticeable.

Additionally, we also detect small isolated "islands" of projected pixels within holes of unknown data, typically due to errors in the disparity map. These outliers are identified and removed using morphological opening with linear structuring elements (1×2 for horizontal outliers, and 3×1 for vertical ones).

3.2. Ghosts Removal

Due to the finite size of image sensors and imprecisions of the disparity map, pixels around an image boundary are usually

composed by the foreground and background objects. The ghost artifacts consist in foreground information being propagated to background regions due to the lack of information of the disparity map in representing those smooth boundaries. It is important to notice that those artifacts can greatly impact the inpaint algorithm, so it is necessary to deal with them first.

In order to identify the regions potentially related to ghosts, we calculate the binary image G that is computed by excluding the crack regions C from S as $G = S - C$. After we have the regions G that may contain ghosts, we use the morphological dilation operator with a non-symmetric structural element H_G to expand the occluded regions in the direction of the reference camera to the virtual camera. For example, if we have one reference camera and generate a synthetic view using a virtual camera on the right side, the background information of all the holes caused by occlusion problems will be on their right side. Therefore, depending on the side, a different mask H_G is generated. In all of our tests, the structural element H_G used was a horizontal line, one pixel tall and 3 pixels wide, whose configuration varies depending on the projection orientation (left or right). Then we separate candidates using the absolute complement operator in the processed image $\hat{G}$, obtaining σ_Ω, as in the cracks removal approach. Fig. 3 illustrates this process.

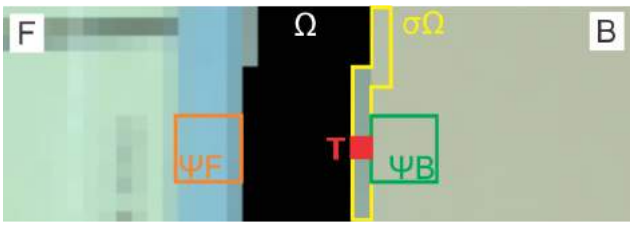


Fig. 3. Zoom of monopoly dataset [12]. Notation: σ_Ω are ghost candidates. Ω, F and B represent the hole, foreground and background respectively. ψ_B and ψ_F are patches for evaluation of the target (T) similarity with F and B.

The next step is to evaluate each candidate point $T \in \sigma_\Omega$ based on its similarity with neighboring patches. For that purpose, we compute the mean intensity within 3×3 patches ψ_B and ψ_F, which are neighbors of T in the background and foreground, respectively, and compute the differences dB and dF from T to these mean values. If $dB < dT$ and $dB < \alpha$, where T is a similarity threshold, then T is kept attached to the background. Otherwise, it is considered as belonging to the foreground, and moved horizontally to the other extremity of the hole. In our tests, we used $\alpha = 11$ as the threshold in all experiments.

3.3. Hole Filling

Due to the sampling theorem, there are constraints to the spatial frequency content of an image that cannot be reconstructed once lost. In those cases of missing or damaged areas, the best we can achieve is to produce a plausible result rather than a perfect reconstruction [4].

Since it is not uncommon to have big holes in DIBR techniques caused by occlusions and/or disparity problems, we need to synthesize not only a plausible color within the holes but also to recreate a locally adequate texture. For this task we propose to extend the texture synthesis work of [6] to prioritize the background regions in the hole filling algorithm, since the occluded regions are by definition a portion of the background that has been disoccluded.

Given a hole Ω and its boundary $\partial\Omega$, the first step is to find the patch Ψ_p with $p \in \partial\Omega$ that must be inpainted. We then search for a patch Ψ_q in the source region $\Phi = \mathcal{I} - \Omega$, where $\mathcal{I}$ is the image to be inpainted, and copy its texture to Ψ_p. The main idea is to use Φ as a texture database, and copy small patches Ψ_q to Ω according to the local information provided by Ψ_p.

The first step is to define the hole filling order, aiming to both preserve incoming edges and prioritize the background. The choice of p for each iteration is given by the following priority equation $P(p) = C(p)E(p)$ where $P(p)$ is the priority for a given pixel $p \in \partial\Omega$, $C(p)$ is the confidence term (described in [6]) and $E(p)$ is the depth term. They are defined as follows:

$$C(p) = \frac{\sum_{q \in \Psi_{p \cap (\mathcal{I}-\Omega)}} C(q)}{|\Psi_p|}, \tag{2}$$

$$E(p) = \frac{\sum_{q \in \Psi_{p \cap (\mathcal{I}-\Omega)}} d(q)}{|d(p)|}, \tag{3}$$

where $|\Psi_p|$ is the area of Ψ_p, $|d(p)|$ is the area of $d(p)$ (in terms of number of pixels), $d(q)$ is the depth value for each point of the patch. The priority $P(p)$ is calculated for every $p \in \partial\Omega$ for each iteration, and the point with the biggest value is chosen. In the initial configuration, $C(p)$ is set to zero $\forall p \in \Omega$, and $C(p) = 1$ $\forall p \in \mathcal{I} - \Omega$.

The confidence term $C(p)$ measures the amount of reliable information around the pixel p, so it prioritizes filling patches which have more pixels already filled. The depth term $E(p)$ prioritizes the greatest depths, which naturally favor background pixels.

After choosing the destination patch Ψ_p to be filled, the last step is to find the origin patch Ψ_q that is obtained from Φ. We choose Ψ_q by searching in Φ the patch that is the most similar to Ψ_p:

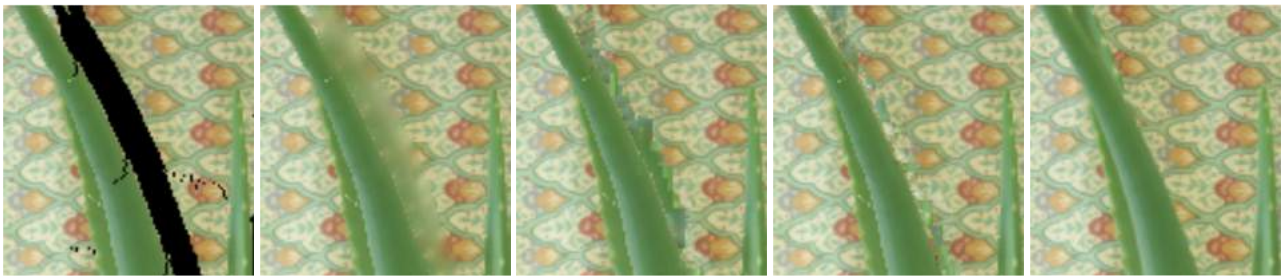
$$\Psi_q = \arg \min_{\Psi_q \in \Phi} s(\Psi_p, \Psi_q), \tag{4}$$

$$s(\Psi_p, \Psi_q) = \sum_{x \in \Omega_v(\Psi_p)} \|\Psi_p(x) - \Psi_q(x)\|^2,$$

where $\Omega_v(\Psi_p)$ denotes the set of pixels in Ψ_p containing valid information, $\Psi_p(x)$ is the RGB color vector related to pixel x.

Method	Aloe1	Aloe5	Art1	Art5	Books1	Books5	Monopoly1	Monopoly5	Mean
Criminisi	26.8196	26.9094	23.6227	23.7331	27.2535	29.3389	27.8316	23.6397	26.1436
HHF	26.7166	27.5551	24.1260	24.9558	27.7551	29.2626	29.2352	26.7045	27.0389
Proposed	**27.4329**	**27.7281**	**26.3628**	**25.2687**	**29.6710**	**29.7081**	**29.6776**	**28.7181**	**28.0709**

Table 1. Quantitative evaluation of PSNR.



Warped View1 to View3 | HHF | Criminisi | Proposed approach | Ground truth

Fig. 4. Zoomed region from the Aloe dataset.

Thus, Ψ_q should be a patch that has similar texture and colors with Ψ_p.

4. EXPERIMENTAL RESULTS

To evaluate the proposed approach we use the datasets and disparity ground truth from the well known Middlebury dataset [12]. We also compare our results with the traditional exemplar-based inpainting approach [6] and the Hierarchical Hole-Filling (HHF) from Solh and AlRegib [11] qualitatively, through visual inspection, and quantitatively, using the Peak Signal-to-Noise Ratio (PSNR). It is important to note that the boundaries of the interpolated images (either left or right, depending on the reference image used) do not contain any valid information. Although our approach, as well as [6] and [11] are able to fill out those regions, they extrapolate image information. These portions should be visually coherent, but they are not taken into account when evaluating the PSNR.

Results for the proposed method and competitive approaches are shown in Table 1. For the tests we used the ground truth disparity from views 1 and 5, putting the synthetic camera in the location of view 3. As it can be observed, our method outperforms both approaches with respect to the PSNR metric for all tested datasets. Figure 4 shows a cropped and zoomed region of the Aloe dataset. Visual inspection indicates that the use of the disparity information to guide the hole filling algorithm was able to correctly propagate the background information within the holes. In contrast, the results from HHF are much blurrier, which is a expected result from the used multi-resolution approach that fills the hole with a combination from all the surrounding pixels, regardless of their disparity. More results are available at `http://www.inf.ufrgs.br/~aqoliveira/research/`.

5. CONCLUSION

In this work we first propose a simple solution for the removal of both ghosts and cracks, common visual artifacts in DIBR view interpolation techniques. For the cracks problem, we first classify crack regions using simple morphological operators, followed by the fast inpaint algorithm proposed by Oliveira et al. [4]. And to eliminate the ghost artifacts we identify the possible ghost regions and move them if necessary. By eliminating the ghosts we also help the hole filling algorithm, since they will not propagate those artifacts to the hole.

For the hole filling problem we propose to extend the work from Criminisi and collaborators [6] by changing the hole filling order using the depth information. By enforcing the texture propagation from the background to the foreground we were able to outperform the hole filling algorithm from Solh and AlRegib [11] in most of the tests, and obtain a much more sharp interpolated view.

As future work, we would like to investigate other disparity-based penalty metrics. We also intend to explore the disparity within the to-be-inpainted patches to reduce the search area for good patches, thus reducing the computational cost. Another possible path for future work is the extension of the patch selection scheme for video view interpolation, in which temporal coherence is an additional constraint.

Acknowledgments

We gratefully acknowledge the partial financial support from CAPES through grant BRANETEC 013/2013 and CNPq through grant 478730/2012-8.

6. REFERENCES

[1] Kwan-Jung Oh, Sehoon Yea, and Yo-Sung Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video," in *Picture Coding Symposium, 2009. PCS 2009*, 2009, pp. 1–4.

[2] Y. Mori, N. Fukushima, T. Fujii, and M. Tanimoto, "View generation with 3d warping using depth information for ftv," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, 2008, pp. 229–232.

[3] Karsten Mller, Aljoscha Smolic, Kristina Dix, Philipp Merkle, Peter Kauff, and Thomas Wiegand, "View synthesis for advanced 3d video systems.," *EURASIP J. Image and Video Processing*, vol. 2008, 2008.

[4] Manuel M. Oliveira, Brian Bowen, Richard McKenna, and Yu sung Chang, "Fast digital image inpainting," in *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON VISUALIZATION, IMAGING AND IMAGE PROCESSING*. 2001, pp. 261–266, ACTA Press.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2000, SIGGRAPH '00, pp. 417–424, ACM Press/Addison-Wesley Publishing Co.

[6] Antonio Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, 2004.

[7] I. Daribo and H. Saito, "A novel inpainting-based layered depth video for 3dtv," *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 533–541, 2011.

[8] Alexandru Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[9] Alexandre Hervieu, Nicolas Papadakis, Aurélie Bugeau, Pau Gargallo, and Vicent Caselles, "Stereoscopic image inpainting: distinct depth maps and images inpainting," in *IEEE International Conference on Pattern Recognition*. IEEE, 2010, pp. 4101–4104.

[10] Yu Mao, Gene Cheung, Antonio Ortega, and Yusheng Ji, "Expansion hole filling in depth-image-based rendering using graph-based interpolation.," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 1859–1863.

[11] M. Solh and G. AlRegib, "Hierarchical hole-filling for depth-based view synthesis in ftv and 3d video," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 6, no. 5, pp. 495–504, 2012.

[12] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, vol. 1, pp. I–195–I–202 vol.1.