



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

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

Ennio Willian Lima Silva

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientador

Prof. Dr. Bruno Luigi Macchiavello

Coorientador

Prof. Dr. Camilo Chang Dórea

Brasília
2016

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Pós-graduação em Informática

Coordenadora: Prof.^a Dr.^a Célia Ghedini Ralha

Banca examinadora composta por:

Prof. Dr. Bruno Luigi Macchiavello (Orientador) — CIC/UnB
Prof.^a Dr.^a Mylene Christine Queiroz de Farias — ENE/UnB
Prof. Dr. Wallace Correa de Oliveira Casaca — ICMC/USP

CIP — Catalogação Internacional na Publicação

Lima Silva, Ennio Willian.

Preenchimento de Buracos em Síntese de Vista Baseado em Mapa de Profundidade / Ennio Willian Lima Silva. Brasília : UnB, 2016.

99 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2016.

1. Vídeos Multivistas, 2. Televisão 3D, 3. Televisão de Ponto de Vista Livre, 4. Mapa de profundidade, 5. *Inpainting*.

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil



Universidade de Brasília

**Instituto de Ciências Exatas
Departamento de Ciência da Computação**

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

Ennio Willian Lima Silva

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Prof. Dr. Bruno Luigi Macchiavello (Orientador)
CIC/UnB

Prof.^a Dr.^a Mylene Christine Queiroz de Farias Prof. Dr. Wallace Correa de Oliveira Casaca
ENE/UnB ICMC/USP

Prof.^a Dr.^a Célia Ghedini Ralha
Coordenadora do Programa de Pós-graduação em Informática

Brasília, 14 de Julho de 2016

Resumo

A evolução dos sistemas e tecnologias de vídeo 3D nos últimos anos é uma consequência do empenho das produtoras de conteúdos multimídia no intuito de proporcionar a imersão 3D ao usuário. A disponibilidade dos conteúdos 3D estão voltadas para os mais variados tipos de tecnologias como dispositivos móveis (celulares e tablets), computadores pessoais, televisões, entre outras, o que aumenta ainda mais o número de usuários alcançados. Este fato, fez da televisão 3D (3DTV - 3D *Television*) e a televisão de ponto de vista livre (FVT - *Free ViewPoint Television*) tópicos relevantes em pesquisas relacionadas ao 3D.

O uso das abordagens vídeo+profundidade e multivista+profundidade fornece uma forma otimizada de transmissão do conteúdo nestas aplicações. A síntese de vista se faz um processo necessário para utilização destes formatos, sendo a renderização baseada em imagem de profundidade (DIBR - *Depth-Image-Based Rendering*) um dos métodos fundamentais para este processo. A DIBR produz vistas virtuais que possuem buracos também chamados desoclusões. O preenchimento destes buracos de maneira visualmente plausível é imprescindível para minimizar a degradação da qualidade dos resultados da síntese de vista.

Portanto este trabalho apresenta um algoritmo de *inpainting* que utiliza mapa de profundidade como auxílio para preenchimento de buracos presentes nas imagens virtuais que foram geradas no processo de síntese de vista. Os resultados objetivos e subjetivos mostram que o algoritmo proposto possui melhores performances em diversas sequências de imagens testadas em relação à outros algoritmos da literatura.

Palavras-chave: Vídeos Multivistas, Televisão 3D, Televisão de Ponto de Vista Livre, Mapa de profundidade, *Inpainting*.

Abstract

The recent evolution of 3D video system technologies is a consequence of multimedia content producers efforts to provide 3D immersion to the user. The 3D contents are available to a great range of technologies such as mobile devices (mobile phones and tablets), personal computers and televisions which increases even more the number of users. This fact has turned out made 3D television (3DTV) and Free ViewPoint Television (FTV) a promising topic regarding researches in 3D technologies.

The use of video+depth and multiview+depth approaches provides an optimized way of streaming 3D content on these applications. The view synthesis is a required procedure when using the aforementioned approaches, being the depth-image-based rendering (DIBR) a fundamental method in this context. The DIBR produces virtual views that have holes, which are also called disocclusion. Filling these holes in a visually plausible way is essential to minimise the degradation of quality of the view synthesis results.

Therefore, this work presents an inpainting algorithm that uses depth maps as aid for filling the holes present in virtual images, which are generated by the view synthesis process. Objective and subjective results show that the proposed algorithm has better performance in various image sequences in comparison to other state-of-art algorithms.

Keywords: Multiview Video, 3DTV, Free Viewpoint Television, Depth maps, Inpainting.

Sumário

1	Introdução	3
1.1	Definição do Problema	5
1.2	Objetivo geral	6
1.3	Objetivos específicos	6
1.4	Contribuições	7
1.5	Organização	7
2	Fundamentação Teórica	9
2.1	Modelo de câmera	9
2.1.1	Parâmetros intrínsecos da câmera <i>pinhole</i>	10
2.1.2	Parâmetros extrínsecos da câmera <i>pinhole</i>	11
2.2	Imagem Digital	11
2.3	Processamento de Imagens	14
2.3.1	Filtros no domínio espacial	14
2.4	Vídeo Digital	18
2.4.1	Vídeos digitais estéreo/Múltiplas Vistas	19
2.5	Mapa de profundidade	20
2.6	Síntese de Vista	23
2.7	<i>Inpainting</i>	26
2.8	Métricas de qualidade	29
2.8.1	Razão Sinal-Ruído de Pico (PSNR)	29
2.8.2	Índice de Similaridade Estrutural - SSIM	30
3	Revisão Bibliográfica	32
3.1	<i>Inpainting</i> digital baseado em exemplares	32
3.2	<i>Inpainting</i> digital para imagens virtuais	35
3.3	Outros trabalhos relevantes	38
4	Metodologia	40
4.1	Detecção da borda	41

4.2	Cálculo da Prioridade	43
4.2.1	Termo de Confiança	43
4.2.2	Termo de Dados	43
4.2.3	Prioridade	45
4.3	<i>Matching</i>	46
4.4	Preenchimento	47
5	Resultados Experimentais	49
5.1	Obtenção das imagens virtuais	49
5.2	Implementação do algoritmo proposto, execução e resultados	50
5.2.1	Ballet 5-4 (BA54)	51
5.2.2	Ballet 5-2 (BA52)	53
5.2.3	Breakdancers 5-4 (BR54)	55
5.2.4	Breakdancers 5-2 (BR52)	56
5.2.5	Ballons 5-4 (BL54)	58
5.2.6	Ballons 5-2 (BL52)	62
5.2.7	Lovebirds 8-7 (LB87)	66
5.2.8	Lovebirds 8-5 (LB85)	71
5.2.9	Poznan 5-4 (PZ54)	76
5.2.10	Poznan 3-4 (PZ34)	80
6	Conclusão	84
6.1	Trabalhos futuros	85

Lista de Figuras

2.1	Representação de Modelo de câmera <i>pinhole</i>	10
2.2	Geometria da câmera <i>pinhole</i>	10
2.3	Representação matricial de uma imagem digital.	12
2.4	Exemplo de imagem binária.	12
2.5	Exemplo de imagem em tons de cinza.	13
2.6	Canais R,G,B e imagem resultante após união destes canais.	13
2.7	Aproximações discretas para o filtro Laplaciano.	15
2.8	Exemplo de janela definida para o filtro da média.	16
2.9	Processamento com o filtro Gaussiano.	17
2.10	Representação de uma sequência de quadros de um vídeo digital.	18
2.11	Exemplo de um sistema de captura estereoscópico.	19
2.12	Exemplo de um sistema de captura multivista.	20
2.13	Mapa de profundidade (imagem (c)) gerado por uma triangulação dos pontos correspondentes das imagens (a) e (b).	21
2.14	Estimação da disparidade pela verificação de similaridade entre blocos (regiões destacadas na cor branca) da imagem de referência I_1 e da imagem I_2 ao longo da linha epipolar.	22
2.15	Etapas básicas da síntese de vista.	24
2.16	Configuração de câmeras para vistas estereoscópicas.	25
2.17	Representação genérica do processo de <i>inpainting</i>	27
3.1	Notação definida por Criminisi e Toyama(2004).	33
3.2	Síntese de textura baseada em modelos: (a) Imagem original. (b) Fragmento Ψ_p centralizado em $p \in \Phi$. (c) Candidatos mais prováveis $\Psi_{q'}$ e $\Psi_{q''}$. (d) Candidato mais provável é propagado para o fragmento alvo.	34
3.3	Princípio de Gautier.	37
3.4	Aplicação da máscara HRM para obtenção da borda de um único lado.	37
3.5	Resultado em vista virtual - Sequência Ballet - Deformação da câmera 5 para 4.	38

3.6	Resultado em vista virtual - Sequência Ballet - Deformação da câmera 5 para 2.	38
4.1	Fluxograma de processos do algoritmo proposto.	40
4.2	Diferentes posições dos buracos gerados na síntese de vista.	41
4.3	Diferentes sentidos para preenchimento dos buracos.	42
4.4	Detecção da borda.	42
4.5	Inicialização do termo de confiança para a borda detectada.	43
4.6	Resultado obtido para o frame 0 da sequência Ballet após difusão anisotrópica.	44
4.7	Realce de mapa de profundidade após filtro <i>unsharp</i>	45
4.8	Efeito do Termo de Relevância em parte da estrutura de uma imagem. . .	45
4.9	Imagem digital e seu respectivo mapa de profundidade.	46
4.10	Incômodo visual gerado pela estratégia gulosa.	47
5.1	Imagens finais da sequência BA54 (frames 0 e 11) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	53
5.2	Imagens finais da sequência BA52 (frames 0 e 14) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	54
5.3	Imagens finais da sequência BR54 (frames 0 e 52) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	56
5.4	Imagens finais da sequência BR54 (frames 61 e 75) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	57
5.5	PSNR frame a frame - Sequência Ballons 54.	58
5.6	SSIM frame a frame - Sequência Ballons 54.	58
5.7	PSNR - Boxplot - Sequência Ballons 54.	59
5.8	SSIM - Boxplot - Sequência Ballons 54.	59
5.9	Imagens finais da sequência BL54 (frames 0 e 60) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	61
5.12	PSNR frame a frame - Sequência Ballons 52.	63
5.13	SSIM frame a frame - Sequência Ballons 52.	63
5.14	PSNR - Boxplot - Sequência Ballons 52.	64
5.15	SSIM - Boxplot - Sequência Ballons 52.	64
5.16	Imagens finais da sequência BL52 (frames 06 e 57) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	65
5.19	PSNR frame a frame - Sequência Lovebirds 87.	67
5.20	SSIM frame a frame - Sequência Lovebirds 87.	67
5.21	PSNR - Boxplot - Sequência Lovebirds 87.	68

5.22 SSIM - Boxplot - Sequência Lovebirds 87.	68
5.23 Imagens finais da sequência LB87 (frames 02 e 18) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	70
5.26 PSNR frame a frame - Sequência Lovebirds 85.	71
5.27 SSIM frame a frame - Sequência Lovebirds 85.	72
5.28 PSNR - Boxplot - Sequência Lovebirds 85.	72
5.29 SSIM - Boxplot - Sequência Lovebirds 85.	73
5.30 Imagens finais da sequência LB85 (frames 08 e 53) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	74
5.33 PSNR frame a frame - Poznan 54.	76
5.34 SSIM frame a frame - Poznan 54.	76
5.35 PSNR - Boxplot - Poznan 54.	77
5.36 SSIM - Boxplot - Poznan 54.	77
5.37 Imagens finais da sequência PZ54 (frames 07 e 23) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	78
5.40 PSNR frame a frame - Poznan 34.	80
5.41 SSIM frame a frame - Poznan 34.	80
5.42 PSNR - Boxplot - Poznan 34.	81
5.43 SSIM - Boxplot - Poznan 34.	81
5.44 Imagens finais da sequência PZ34 (frames 07 e 44) após o <i>inpainting</i> com os algoritmos da literatura e o algoritmo proposto.	82

Lista de Tabelas

5.1	Vistas virtuais geradas	50
5.2	Valores médios de PSNR[dB] obtidos para cada sequência após <i>Inpainting</i> .	51
5.3	Valores médios SSIM obtidos para cada sequência após <i>Inpainting</i>	51

Lista de Abreviaturas

Abreviatura	Significado
2D	Duas dimensões (bi-dimensional)
3D	Três dimensões(tri-dimensional)
3DTV	<i>3D Television</i>
FTV	<i>Free ViewPoint Television</i>
DIBR	<i>Depth-image-based Rendering</i>
V+D	<i>Video-plus-depth</i>
MVD	<i>Multiview-plus-depth</i>
MSE	<i>Mean Squared Error</i>
HVS	<i>Human Visual System</i>
PSNR	<i>Peak Signal-to-noise Ratio</i>
SSIM	<i>Strucutural Similarity Index</i>
SAD	<i>Sum of Absolute Differences</i>
SSD	<i>Sum of Squared Differences</i>

Lista de Símbolos

Símbolo	Significado
Ω	Região da imagem que será preenchida (Buraco)
$\delta\Omega$	Borda do buraco
$\delta\Omega'$	Borda unilateral
Φ	Região de busca
\oplus	Operação morfológica de dilatação
\cap	Operação de interseção
Δ	Operador gradiente
∇	Operador Laplaciano
β	Número inteiro maior que 1
Ψ_p	Bloco a ser propagado
Ψ_q	Bloco candidato

Capítulo 1

Introdução

Sinais estão presentes em diversas situações do dia-a-dia do ser humano. Segundo Oppenheim e Willsky (2010) os sinais, que são funções de uma ou mais variáveis independentes, contêm informações sobre o comportamento ou natureza de algum fenômeno. Por exemplo, um sinal de fala pode ser representado matematicamente pela pressão acústica como uma função do tempo, uma imagem pode ser representada pelo brilho como uma função de duas variáveis espaciais, bem como um vídeo pode ser representado como uma função de duas variáveis espaciais e uma temporal.

Na categoria dos sinais produzidos pelo ou para o ser humano, especial atenção é dedicada aos sinais relacionados com os sentidos humanos da visão, contemplando a ampla gama dos diversos processamentos aplicáveis aos sinais de imagem e vídeo.

Historicamente falando, uma das primeiras aplicações relacionadas ao processamento de imagens remonta ao início do século XX, onde procuravam-se formas de melhorar a qualidade de impressão de imagens digitalizadas transmitidas por um cabo submarino entre Londres e Nova York. Entretanto, o grande impulso viria cerca de trinta anos depois com o surgimento dos primeiros computadores digitais que possibilitariam o uso de imagens e/ou vídeos digitais (FILHO; NETO, 1999).

Segundo Gonzalez e Woods (2011), uma imagem digital pode ser definida como uma função bidimensional $f(x, y)$, em que x e y são coordenadas espaciais, e a amplitude de f em qualquer par de coordenadas $f(x, y)$ é chamada de intensidade, onde esta possui valores discretos e finitos. Por outro lado, um vídeo digital refere-se à captura, manipulação e armazenamento de imagens digitais em movimento, sendo o tempo também um das variáveis contidas na função de representação de um vídeo.

Com o aumento da capacidade e poder de processamento dos computadores durante o passar dos anos, a área de processamento de imagens e vídeo digitais apresentou crescimento significativo. Atualmente suas aplicações estão presentes em diversos ramos da atividade humana abordando temas de caráter interdisciplinar em diversas áreas (FILHO;

NETO, 1999). Uma das áreas que tem sido alvo de diversos pesquisadores é a imersão na realidade tridimensional e por isso o vídeo 3D tem ganhado espaço na comunidade científica e comercial.

A evolução dos sistemas e tecnologias de vídeo 3D nos últimos anos é uma consequência do empenho das produtoras de conteúdos multimídia no intuito de proporcionar a imersão 3D ao usuário. Atualmente a disponibilidade dos conteúdos 3D estão voltadas para o mais variados tipos de tecnologias como dispositivos móveis (celulares e tablets), computadores pessoais e televisões, o que aumenta ainda mais o alcance no que diz respeito ao número de usuários.

O sistema 3D tradicional, também conhecido como sistema estéreo, permite que o usuário veja a cena com sensação de profundidade, mas somente a partir de um ponto de vista. Este sistema é baseado na captura e transmissão de dois sinais de vídeo, cada um associado a uma vista do sistema óptico humano.

O sistema multivistas, como o próprio nome já diz, é uma extensão do sistema estéreo que possui um número maior de vistas, normalmente obtido a partir de um conjunto de câmeras sincronizadas que capturam a mesma cena (CODING; GROUP, 2015). Este sistema pode ser usado em diversos tipos de aplicações, tais como a televisão 3D (3DTV - *3D Television*) e a televisão de ponto de vista livre (FTV - *Free-viewpoint Television*). Permitindo a observação de uma mesma cena a partir de diferentes pontos de vistas, o sistema multivistas melhora ainda mais a experiência visual do usuário (LUCAS et al., 2012).

O aumento do número de vistas disponíveis é uma tendência nos sistemas futuros, por melhorar a experiência oferecida ao usuário. A desvantagem do aumento do número de vistas é a grande quantidade de informação associada, pelo fato de que a quantidade de informação gerada é no mínimo duas vezes (se tivermos um sistema com apenas duas câmeras) a informação gerada por um vídeo monocular (gerado por apenas uma câmera), o que penaliza a transmissão ou armazenamento desta informação.

Uma alternativa para transmitir o conteúdo 3D requerido nos sistemas anteriormente citados é a utilização dos formatos vídeo+profundidade (V+D, do inglês *video-plus-depth*) e o multivista+profundidade (MVD, do inglês *multiview-plus-depth*), já que os mesmos garantem a renderização de vistas virtuais tanto para *displays* estereoscópicos como para *displays* autoestereoscópicos (MUDDALA; OLSSON; SJOSTROM, 2013).

O formato vídeo+profundidade é uma alternativa ao sistema de vídeo estéreo, onde apenas a informação de uma das vistas e o respectivo mapa de profundidade, o qual relaciona as distâncias das superfícies dos objetos de uma cena a partir de um ponto de vista, são transmitidos. A ideia é que a vista não transmitida possa ser sintetizada a partir das informações de textura e profundidade da outra vista. Uma vez que a profundidade

usa geralmente menos taxa do que uma vista de textura, esta representação pode ser mais eficiente que o sistema estéreo, sendo uma abordagem promissora. O fato de considerar apenas uma vista de textura também permite manter a compatibilidade com os sistemas 2D. Além disso, este sistema possibilita a síntese de um grande número de vistas, dentro de um intervalo angular limitado, o que aumenta a sua utilidade.

Por sua vez, o formato multivista+profundidade é uma extensão do sistema vídeo+profundidade. Neste sistema as vistas são sintetizadas a partir da informação de um conjunto limitado de vistas e mapas de profundidade. Isto significa que os pontos de captura das vistas transmitidas não correspondem necessariamente aos pontos de observação gerados pelas vistas virtuais sintetizadas. Desta forma, o número de vistas que é efetivamente transmitido pode ser muito inferior àquele que é requerido por um sistema multivistas que não usa informação de profundidade, permitindo um ganho significativo de desempenho.

Como citado anteriormente, um dos processos chaves que otimizam a utilização dos sistemas V+D e MDV consiste na síntese de vista, que geralmente é obtida através da *Renderização Baseada em Imagem de Profundidade* também chamada de Deformação 3D na Computação Gráfica (FEHN, 2004). Dados os parâmetros intrínsecos e extrínsecos das respectivas câmeras, esta síntese produz vistas virtuais usando pixel de textura densa e a informação de profundidade.

Um problema inerente dos algoritmos de deformação 3D é o fato de que cada pixel não necessariamente existe em outros pontos de vista. Consequentemente, a síntese de vista pode expor as áreas da cena que estão obstruídas na vista de referência e tornar-se visível na vista secundária (DARIBO; PESQUET-POPESCU, 2010). Estas áreas, também denominadas desoclusões ou buracos, devem ser preenchidas para melhor visualização da vista intermediária gerada. Devido a este fator, os algoritmos de deformação 3D normalmente utilizam-se de técnicas de *inpainting* em conjunto para preenchimento dos buracos resultantes da síntese de vista.

Em uma abordagem direta, pode-ser resumir o *inpainting* como o preenchimento de uma parte de uma imagem, quer gerada artificialmente ou por uma falha na própria imagem, baseando-se na vizinhança da área a ser preenchida. Uma boa técnica de *inpainting* é aquela que mantém a coerência da área preenchida artificialmente com sua vizinhança original, causando a impressão visual de que a área preenchida é realmente parte da imagem original.

1.1 Definição do Problema

Uma maneira prática de se obter um vídeo 3D é a através da utilização de sistemas estéreo e/ou multivistas. No entanto, a medida que o número de câmeras aumenta, uma

grande largura de banda se faz necessária para a transmissão e/ou armazenamento, o que limita seu uso. Uma alternativa para resolver este problema é a geração de vistas virtuais. A geração de vistas virtuais por sua vez pode ser realizada através da Renderização Baseada em Imagem de Profundidade que utiliza imagens de textura e suas respectivas imagens de profundidade (AHN; KIM, 2012). Um problema inerente que surge na geração de vistas virtuais é que as regiões abrangidas na exibição original podem causar buracos nas vistas virtuais.

O preenchimento destes buracos de maneira visualmente plausível é imprescindível para minimizar a degradação da qualidade dos resultados da síntese. Uma solução argumentada por Tauber et al. (2007) consiste em combinar síntese de vista com técnicas de *inpainting* para lidar com grandes buracos, já que há uma semelhança natural entre buracos danificados em pinturas e os buracos gerados na síntese de vista.

Portanto, o problema a ser abordado neste trabalho consiste no preenchimento de buracos provenientes da síntese de vista através de um algoritmo de *inpainting* que utilize mapas de profundidade como auxílio.

1.2 Objetivo geral

Considerando todos os aspectos apresentados, o *inpainting* é um campo com bastante relevância para a recuperação e/ou restauração de imagens. Dessa forma, o objetivo geral deste trabalho é propor um algoritmo de *inpainting* usando imagens de profundidade como auxílio para preenchimento de buracos em imagens resultantes da síntese de vista.

1.3 Objetivos específicos

- Identificar os principais algoritmos de *inpainting* na literatura;
- Identificar os principais algoritmos de *inpainting* que utilizem imagens de profundidade;
- Verificar melhorias nos algoritmos selecionados;
- Propor um algoritmo de *inpainting* que utiliza imagens de profundidade para preenchimentos de buracos em imagens resultantes da síntese de vista.

1.4 Contribuições

O algoritmo de *Inpainting* proposto neste trabalho compreende um conjunto de técnicas que somadas contribuem para a melhoria dos resultados obtidos em diversos algoritmos da literatura.

Diferentemente de Crimisi e Toyama (2004), de Daribo e Pesquet-Popescu (2010), Gautier (2011) e Ahn e Kim (2012), o algoritmo proposto inclui o Termo de Relevância integrando o mapa de profundidade no processo de cálculo de prioridade, levando em consideração as bordas como um dos principais fatores que definem a estrutura de uma imagem.

Computar a prioridade durante o processo de reconstrução das oclusões em imagens de sínteses de vista se faz necessário, uma vez que adotar mecanismos de *inpainting* não-sensíveis a esta questão pode levar à geração de artetafos e incongruências visuais na imagem, fazendo com que o resultado final perca o aspecto natural quanto à percepção visual humana.

No processo de propagação do *patch* analisado em questão, isto é, do bloco de pixels contendo parte da informação a ser sintetizada na área oclusa, o algoritmo realiza uma média ponderada entre os três melhores *patches*, gerando uma combinação linear para evitar transições abruptas nas bordas e descontinuidades.

O algoritmo proposto neste trabalho gerou uma publicação recentemente aceita para o Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT). O SBrT é o encontro de maior relevância acadêmica nacional da área de telecomunicações e processamento de sinais.

1.5 Organização

Este trabalho está organizado como se segue:

- Capítulo 2 - Apresenta os fundamentos teóricos necessários para o bom entendimento do trabalho como um todo, enfatizando a geometria do modelo de câmera, os conceitos imagem digital e vídeo digital, mapa de profundidade e o processo de síntese de vista.
- Capítulo 3 - Aborda os detalhes e conceitos da área de *Inpainting* Digital, bem como trabalhos relacionados ao tema do trabalho proposto.
- Capítulo 4 - Mostra os detalhes metodológicos pertinentes ao trabalho apresentado, tais como detecção de borda, cálculo de prioridade, *matching* e preenchimento da área do buraco.

- Capítulo 5 - Retrata os experimentos realizados no trabalho, bem como os resultados obtidos e suas discussões.
- Capítulo 6 - Descreve as conclusões do trabalho apresentado.

Capítulo 2

Fundamentação Teórica

Neste capítulo são descritos alguns conceitos teóricos necessários para o entendimento da pesquisa como um todo. Na primeira Seção 2.1, são abordados os princípios do modelo de câmera *pinhole*, para entendimento da representação de um espaço 3D em duas dimensões. Na Seção 2.2 são apresentados os conceitos de imagem digital e explanadas algumas de suas principais características. Na Seção 2.3 são expostos alguns dos principais tipos de processamento em imagens digitais e seus propósitos, bem como métricas de qualidade aplicadas nestas. Na Seção 2.4 são abordados os conceitos de vídeo digital, vídeo estéreo/multivistas e suas aplicações. Por último nas Seções 2.5 e 2.6 são abordadas as ideias básicas sobre mapa de profundidade e suas formas de estimação, tal como a síntese de vista e seus processos.

2.1 Modelo de câmera

Sendo um modelo simples e clássico de câmera de intensidade, conhecido desde épocas remotas da fotografia, o modelo em questão é conhecido como câmera *pinhole*. Esse modelo consiste de uma câmera escura com um pequeno orifício, por onde são captadas amostras de raios luminosos refletidos pelas superfícies da cena, e que atingem o plano oposto ao plano que contém o orifício, formando uma imagem invertida (ver Figura 2.1).

Se o plano de formação da imagem for revestido de uma superfície fotossensível como um papel fotográfico, a imagem é registrada. Em outras palavras, nesse tipo de projeção os raios de luz formam linhas concorrentes convergindo para o ponto O , que é chamado centro óptico ou centro de projeção, como pode ser observado na Figura 2.2.

A projeção de perspectiva altera medidas e formas reais de um objeto no mundo real - regiões mais próximas parecem maiores, e regiões mais longínquas parecem menores. O modelo que relaciona as coordenadas X , Y e Z de um ponto P no mundo real, aos pontos x' e y' no plano de imagem Π é o da Equação 2.1:

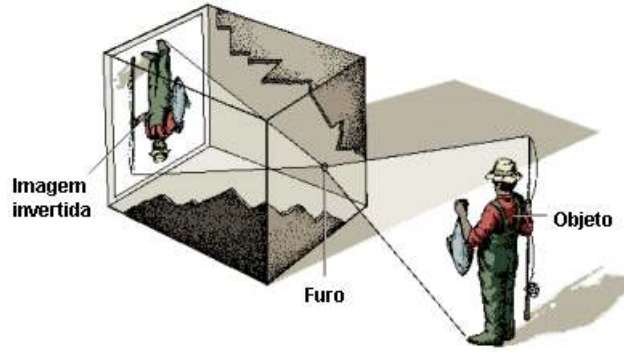


Figura 2.1: Representação de Modelo de câmera *pinhole*.

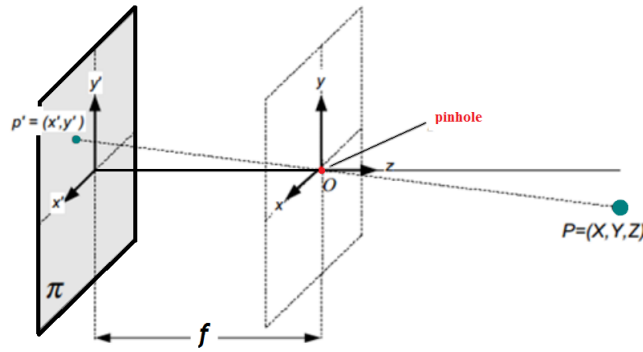


Figura 2.2: Geometria da câmera *pinhole*.

$$(x', y') = (f \frac{X}{Z}, f \frac{Y}{Z}), \quad (2.1)$$

na qual Z é a profundidade do objeto em relação à câmera e f é o comprimento focal ou distância do plano de imagem ao centro óptico O .

2.1.1 Parâmetros intrínsecos da câmera *pinhole*

Segundo Trucco e Verri (1998) os parâmetros intrínsecos podem ser definidos como um conjunto de parâmetros necessários para caracterizar óptica, geométrica e digitalmente o sistema de visão da câmera. Para uma câmera *pinhole* tem-se um conjunto de três parâmetros intrínsecos:

- A projeção perspectiva, para a qual o único parâmetro é o comprimento focal f .
- A transformação entre o sistema de referência da câmera e as coordenadas em pixels.
- A distorção geométrica introduzida pelos elementos ópticos.

2.1.2 Parâmetros extrínsecos da câmera *pinhole*

Os parâmetros extrínsecos são definidos como um conjunto de parâmetros geométricos que relacionam, de modo único, o sistema de referência da câmera desconhecido a um sistema de referência global conhecido (TRUCCO; VERRI, 1998). Para realizar a transformação entre estes dois sistemas de referência, são utilizados:

- Um vetor de translação 3D, vetor T , que descreve a posição relativa dos dois sistemas de referência.
- Uma matriz de rotação 3x3, R , que relaciona a orientação dos eixos entre os dois sistemas de referência.

2.2 Imagem Digital

A visão é um dos cinco sentidos fundamentais dos seres vivos que permite aprimorarem a percepção do mundo. Esta não permite somente detectar a luz e as imagens, mas também interpretá-las. Tendo em vista este fato, as imagens são consideradas importantes fontes de informações para os seres humanos.

Segundo Coser (2009), as imagens podem ser consideradas como o principal mecanismo de comunicação não-formal existente, permitindo que se tenha um entendimento imediato de determinadas situações, sem a necessidade de interpretações muito precisas ou formalização de regras.

Para Gonzalez e Woods (2011), uma imagem é definida como uma função $f(x,y)$ onde x e y são coordenadas em um plano instanciando um ponto luminoso. Esta é a definição matemática mais plausível de uma imagem do mundo real, ou seja, os pontos de luminosidade analógicos são contínuos, se $(x,y) \in \mathbb{R}^2$, conjunto dos reais.

Por outro lado, toda imagem tratada computacionalmente e numericamente é denominada imagem digital. Deste modo, as imagens digitais são imagens são funções discretas da forma $f(x_i, y_i)$ as quais são atribuídos valores numéricos discretos, pois como computadores não manipulam valores analógicos, é necessário realizar a conversão da forma contínua para a forma digital. Isto é feito através da divisão da área da imagem em uma matriz bidimensional de tamanho $m \times n$ (PARKER, 1996).

Na representação por matrizes bidimensionais, cada elemento do plano cartesiano é conhecido como pixel (*Picture element* ou elementos da figura), e a cada pixel ou elemento da matriz é atribuído um valor identificando uma intensidade/tonalidade de cor, como pode ser visto na Figura 2.3. Desta forma, o acesso à informação é feito em qualquer ponto da imagem. O pixel é também um elemento de dimensões finitas na representação

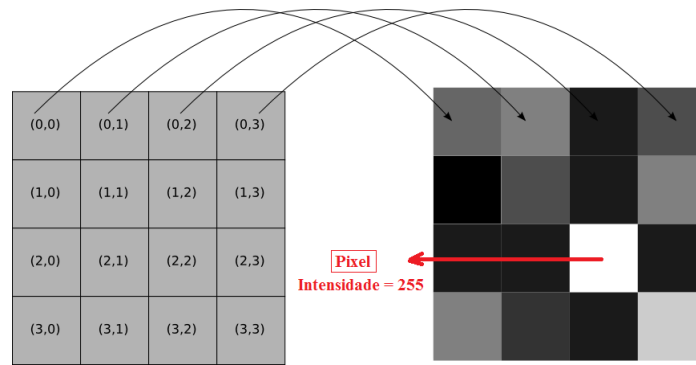


Figura 2.3: Representação matricial de uma imagem digital.

de uma imagem digital, sendo que a sua quantidade em uma determinada imagem reflete a sua resolução.

As imagens digitais podem ser classificadas em monocromáticas ou acromáticas (binárias ou tons de cinza) e coloridas. Uma imagem é binária ou binarizada quando cada pixel que a compõe só pode representar duas informações, a cor preta e a cor branca, como ilustra a Figura 2.4.



Figura 2.4: Exemplo de imagem binária.

Em imagem em tons de cinza, conforme em exemplo na Figura 2.5, os pixels possuem valores decimais de 0 à 255. As diferentes intensidades, partindo do escuro em direção ao claro, são representadas com um valor crescente, em que o valor 0 representa a cor preta e o 255 representa a cor branca.

Para as imagens coloridas, atualmente há diversos formatos de cores para sua representação. Entre os mais comuns estão os modelos RGB, YCbCr, HSV e CMYK. Dentre estes, o RGB abreviação de (*Red*, *Green* e *Blue* - Vermelho, Verde e Azul) é o mais comum, sendo utilizado por monitores de PC, pelas televisões e máquinas fotográficas digitais.

Em uma imagem digital colorida no formato RGB, um pixel pode ser visto como um vetor cujas componentes representam as intensidades de vermelho, verde e azul de sua cor.



Figura 2.5: Exemplo de imagem em tons de cinza.

A imagem colorida pode ser vista como a composição de três imagens monocromáticas, definida pela seguinte equação:

$$f(x, y) = f_r(x, y) + f_g(x, y) + f_b(x, y), \quad (2.2)$$

na qual $f_r(x, y)$, $f_g(x, y)$ e $f_b(x, y)$ representam respectivamente, as intensidades luminosas das componentes vermelha, verde e azul da imagem no ponto (x, y) .

Nas Figuras 2.6a e 2.6b, são apresentados os planos monocromáticos de uma imagem e o resultado da composição dos três planos. Os mesmos conceitos formulados para uma imagem digital monocromática aplicam-se a cada plano de uma imagem colorida.



Figura 2.6: Canais R,G,B e imagem resultante após união destes canais.

2.3 Processamento de Imagens

A evolução da área de processamento digital de imagens teve um crescimento significativo nos últimos anos, principalmente devido o aumento da capacidade de processamento dos dispositivos de computação. Por isso, sistemas de processamento de imagens têm conquistado mais espaço na comunidade científica, possibilitando que uma grande quantidade de aplicações possam ser desenvolvidas.

Dentro do processamento de imagens, os filtros são vistos como ferramentas indispensáveis. Atuando principalmente na etapa de pré-processamento de imagens, entre os objetivos dos filtros podemos citar a homogeneização da imagem ou de alvos específicos, extração de ruídos, detecção de bordas e formas, entre outros.

Pode-se entender filtragem de uma imagem, como sendo técnicas de transformações aplicadas a cada pixel da imagem, levando em conta os níveis de cinza de uma região vizinha de cada pixel desta imagem.

As técnicas de filtragem podem ser divididas em duas categorias: filtragem no domínio espacial e filtragem no domínio da frequência. Entretanto, neste trabalho são abordados somente filtros do domínio espacial.

2.3.1 Filtros no domínio espacial

A filtragem no domínio espacial aplicada a uma imagem digital é uma operação local que modifica o valor de cada pixel considerando o contexto atual do pixel. Em outras palavras, pela filtragem, o valor de cada pixel da imagem é modificado utilizando-se uma operação de vizinhança. Esta operação pode ser expressa pela seguinte equação:

$$g(x, y) = T[f(x, y)], \quad (2.3)$$

na qual $f(x, y)$ é a imagem de entrada, $g(x, y)$ é a imagem processada e T é um operador levando-se em conta alguns pixels vizinhos ao pixel central (x, y) .

A filtragem no domínio espacial normalmente baseia-se na utilização de máscaras. Essas máscaras são pequenas matrizes bidimensionais e os valores de seus coeficientes determinam o objetivo a ser alcançado durante o processamento (GONZALEZ; WOODS, 2001).

Dois filtros que são bastante utilizados no domínio espacial são os de realce e suavização.

Filtros de realce

Os filtros de realce, também chamados de filtros passa alta, tem como principal objetivo melhorar a qualidade visual geral da imagem digital, aumentando o contraste da imagem e realçando características específicas relacionadas aos alvos da imagem. O contraste entre dois objetos pode ser definido como a razão entre os seus níveis de cinza médios.

Entre alguns dos filtros de realce utilizados no domínio espacial podemos citar:

- Laplaciano
- *Unsharp masking* e *Highboost Filtering*

O filtro Laplaciano, $L(x,y)$ de uma imagem é dado pelo divergente do gradiente da função que descreve a intensidade dos seus pixels, $I(x,y)$, sendo representado por:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}. \quad (2.4)$$

Considerando que a imagem é armazenada como uma coleção de pixels discretos, é necessário produzir uma forma discreta do operador contínuo, a qual pode ser representada por uma máscara como a ilustrada na Figura 2.7.

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 2.7: Aproximações discretas para o filtro Laplaciano.

Este filtro possui resposta mais acentuada a detalhes finos como pontos isolados e linhas, destacando bordas ou descontinuidades na imagem. Todavia, ameniza regiões com nível de cinza constante.

O filtro *Unsharp masking*, por sua vez, utilizado por muitos anos em indústrias de impressão e publicação para realce de imagens, possui os seguintes passos:

1. Borramento da imagem original.
2. Subtração da imagem borrada pela original (a diferença resultante é chamada de máscara).
3. Adição da máscara à imagem original.

Chamando de $\overline{f(x, y)}$ a imagem borrada, a máscara é obtida através da seguinte operação:

$$g_{mask}(x, y) = \overline{f(x, y)} - f(x, y). \quad (2.5)$$

Em seguida, o resultado é ponderado e adicionado à imagem original:

$$g(x, y) = f(x, y) + k \cdot g_{mask}(x, y), \quad (2.6)$$

no qual k é um escalar positivo. Quando $k=1$, tem-se o *unsharp masking* definido anteriormente. Quando $k>1$, o processo é referido como *highboost filtering*. Se $k<1$ enfatiza-se pouco a contribuição da máscara.

Filtros de suavização

Também chamados de filtros passa baixa, os filtros de suavização são normalmente utilizados para borramento e redução de ruídos em imagens. O efeito visual de um filtro espacial passa-baixa é a redução do número de níveis de cinza da cena. As altas frequências, que correspondem às transições rápidas são atenuadas. Dentre os principais filtros de suavização podemos destacar:

- Filtro da média
- Filtro Gaussiano

Também conhecido como *Box Filter*, o filtro da média é expresso pelo somatório dos vizinhos ao pixel central. Uma janela que efetua uma filtragem passa-baixa, numa vizinhança de dimensão 3x3 está indicada na Figura 2.8:

1/9 x	1	1	1
	1	1	1
	1	1	1

Figura 2.8: Exemplo de janela definida para o filtro da média.

O filtro gaussiano por sua vez também é muito utilizado para suavização de imagens. Ele possui dois parâmetros, a dimensão da janela e um valor para o desvio padrão máximo sigma. O quanto a imagem será suavizada está relacionado ao desvio padrão sigma, isto é, quanto maior o sigma, mais a imagem é suavizada, não dependendo muito do parâmetro referente a dimensão da janela. Quanto maior o sigma, maior o número de pixels cujo

valor é diferente de zero, o que leva os pixels vizinhos a terem maior influência em cada ponto, realizando uma suavização maior na imagem. Sua definição é expressa pela equação a seguir:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (2.7)$$

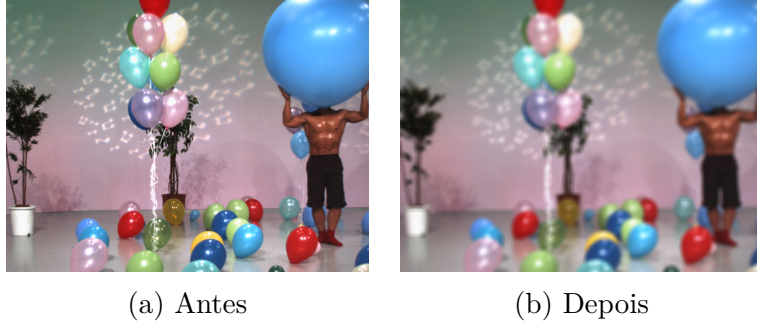


Figura 2.9: Processamento com o filtro Gaussiano.

Filtros gaussianos definem uma série de características que tornam o seu uso bem interessante:

1. A resposta do filtro é isotrópica, ou seja, igual em todas as direções.
2. A resposta do filtro é uma média ponderada dos vizinhos do pixel.
3. Funções gaussianas são separáveis, o que significa que a convolução pode ser aplicada em uma direção por uma máscara e outra máscara pode computar o resultado em outra direção, ao mesmo tempo.

Filtros gaussianos são importantes também por formarem a base em processos de difusão anisotrópica.

Segundo Saito et al. (2010) as técnicas de difusão anisotrópica, simulam, por meio do uso da equação de difusão do calor, o processo que cria um modelo do tipo espaço-escala, onde uma imagem gera uma família parametrizada de sucessivas imagens borradas. Perona e Malik (1990) formularam a equação de difusão anisotrópica discreta como:

$$I_s^{t+1} = I_s^t + \frac{\lambda}{|\eta_s|} \sum g(\nabla I_s, p) \nabla I_s, p, \quad (2.8)$$

no qual I_s^t corresponde a imagem discretizada e ∇I é o gradiente da imagem. A função $g(\cdot)$ é chamada de parada-na-aresta (*edge-stopping*) e o parâmetro t refere-se ao número de iterações. A constante $\lambda \in \mathbb{R}$, com $\lambda > 0$, corresponde à taxa de difusão e η_s define a vizinhança de um pixel na posição espacial $s = (x, y, z)$. O gradiente da imagem é calculado como:

$$\nabla I_{s,p} = I_p - I_s^t, \quad (2.9)$$

no qual $p \in \eta_s$ representa um voxel da vizinhança. O uso da direção do gradiente permite o borramento local apenas na direção paralela às bordas, o que faz com que as bordas sejam preservadas.

2.4 Vídeo Digital

Um vídeo digital é definido como uma sequência de quadros – imagens estáticas – capturados a uma determinada taxa de uma cena contínua no tempo que, quando exibidos a certa frequência (quadros por segundo) proporcionam a sensação de movimento para um expectador. A frequência da exibição de quadros está relacionada à suavidade do movimento. Vídeos digitais que são apresentados a uma taxa dentre 24 e 30 quadros por segundo são suficientes para que o olho humano tenha a impressão de movimento suave (RICHARDSON, 2003). Por sua vez, um quadro é definido como uma matriz de pixels, onde cada pixel armazena as informações de cor e de luminosidade de uma determinada posição. A Figura 2.10 ilustra uma sequência de imagens capturadas e exibidas de forma subsequentes como ocorre num vídeo digital.

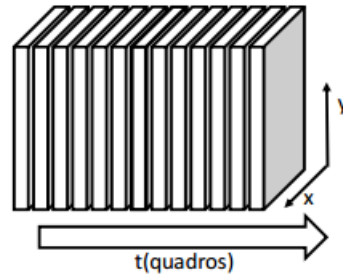


Figura 2.10: Representação de uma sequência de quadros de um vídeo digital.

Considerando um quadro uma matriz de pixels, existem inúmeras formas e sistemas para a representação de espaços de cores que visam a representação de um pixel numericamente. A representação de cores neste formato RGB é amplamente utilizada em equipamentos como televisores, projetores, monitores de tubo de raios catódicos, câmeras digitais portáteis e câmeras de vídeo. No entanto, RGB não é um espaço de cores adequado para ser utilizado por algoritmos de compressão de imagens, pois neste espaço existe uma alta correlação entre os componentes de cores.

Devido ao fator supracitado, em detrimento a utilização do sistema RGB, os padrões de codificação de vídeos mais populares como MPEG-2, MPEG-4 e H.264/AVC utilizam o

espaço de cores YCbCr. Neste espaço de cores as informações de um pixel estão separadas em matrizes de luminância (Y), croma azul (Cb) e croma vermelho (Cr). O componente de luminância representa a intensidade de luz do quadro. Os componentes de croma tratam das informações de cores propriamente ditas. Neste espaço, não há uma correlação significativa – como no sistema RGB – entre os três componentes. Desta forma, podem ser aplicadas técnicas distintas para cada matriz, explorando características de cada uma de forma apropriada (RICHARDSON, 2003).

2.4.1 Vídeos digitais estéreo/Múltiplas Vistas

A definição de sequências de vídeos de estéreo/múltiplas vistas, ou simplesmente vídeos multivistas, remete a um conjunto de vídeos capturados de forma síncrona de uma mesma cena a partir de câmeras dispostas em pontos distintos de observação. A 3DTV (3D Television) e a FVT (*Free Viewpoint Television*) são exemplos de aplicações que podem ser obtidas a partir de sistemas multivistas.

A 3DTV, que utiliza um sistema de captura estereoscópico semelhante ao que ilustra a Figura 2.11, consiste na produção e visualização de duas imagens (sequências de vídeo) correspondentes aos ângulos dos dois olhos humanos. Através da utilização de um dispositivo que permita a visualização de imagens separadas nos dois olhos, é criada uma noção de profundidade tridimensional da cena observada. Por outro lado, a FVT caracteriza-se pela possibilidade de livre seleção interativa (por parte do usuário) da posição e do ângulo de vista da cena filmada dentro de uma gama pré-determinada. A Figura 2.12 ilustra uma maneira de captura que viabiliza o uso da FTV. Por exemplo, caso a cena seja um jogo de futebol, o usuário pode escolher a posição tridimensional do estádio onde observa o jogo, ou seja, de que ângulo deseja assistir o jogo.



Figura 2.11: Exemplo de um sistema de captura estereoscópico.

Ambos os sistemas descritos são sistemas multivistas, visto que múltiplas vistas (posições e ângulos) do cenário filmado devem ser capturadas, codificadas, transmitidas e



Figura 2.12: Exemplo de um sistema de captura multivista.

visualizadas. O sistema utilizado na 3DTV é um caso particular de um sistema multivistas, onde apenas são necessárias apenas duas vistas.

As aplicações de sistemas multivistas vão além da televisão de entretenimento, onde o objetivo é maximizar a experiência audiovisual do usuário. De fato, pode-se utilizar estes sistemas, por exemplo, em aplicações médicas, aplicações de controle industrial, de estudo ambiental, em ambientes de *e-learning*, entre outros. Estes sistemas permitem executar com melhor desempenho tarefas que já eram executadas em sistemas de vista única, como por exemplo segmentação de imagem. Vídeos multivistas tem sido foco de diversas pesquisas, sejam elas voltadas para compressão e codificação ou para transmissão.

A captura de múltiplas imagens implica um maior número de imagens para serem codificadas e transmitidas. Este fato implica em um problema relacionado a capacidade de transmissão de informações dos canais. A medida que o número de câmeras aumenta, uma grande largura de banda se faz necessária para a transmissão e/ou armazenamento, o que limita seu uso. Uma alternativa para resolver este problema é a geração de vistas virtuais. A geração de vistas virtuais por sua vez pode ser realizada através da Renderização Baseada em Imagem de Profundidade que utiliza imagens de textura e suas respectivos mapas de profundidade.

2.5 Mapa de profundidade

Um mapa de profundidade, também conhecido como imagem de profundidade, armazena os valores de profundidade estimados para cada ponto de uma imagem 2D, representando a estrutura 3D da cena. A Figura 2.13 (c) mostra o mapa de profundidade computado por uma triangulação dos pontos correspondentes e não ocluídos das imagens ilustradas na Figura 2.13 (a) e (b). As áreas mais claras do mapa de profundidade repre-

sentam as superfícies dos objetos mais próximos do plano da imagem, enquanto as áreas escuras, as superfícies mais distantes.



Figura 2.13: Mapa de profundidade (imagem (c)) gerado por uma triangulação dos pontos correspondentes das imagens (a) e (b).

A profundidade calculada pela triangulação requer que a correspondência de pontos entre um par de imagens seja estimada. Tradicionalmente, esta operação é realizada levando em consideração a restrição epipolar, a qual estabelece que, dado um ponto x da primeira vista, a busca por um ponto x' correspondente na segunda vista não precisa cobrir toda a imagem e restringe-se apenas a uma linha epipolar. A busca pode ser aprimorada empregando imagens retificadas. A retificação de um par de imagens transforma cada plano das imagens de forma que as linhas epipolares se tornem colineares e paralelas horizontalmente, permitindo que a busca seja realizada ao longo das linhas horizontais das imagens retificadas, aumentando o desempenho computacional (MEDIONI; KANG, 2004).

Na busca por pontos correspondentes, empregando um par de imagens retificadas, a medida de disparidade d , que é a diferença entre os pontos correspondentes das imagens da esquerda e direita, pode ser usada para obter o valor de profundidade z , pela relação definida em Okutomi e Kanade (1993) como:

$$d = bf \frac{1}{z}, \quad (2.10)$$

em que b é a distância entre os centros de projeção das câmeras (*baseline*) e f é a distância focal.

Um algoritmo básico para estimação de um mapa de profundidade entre um par de imagens retificadas é apresentado por Morvan (2009) e descrito a seguir.

Considerando o pixel p_1 da primeira imagem I_1 (imagem de referência), a busca pelo ponto correspondente p_2 na segunda imagem I_2 é realizada ao longo da linha horizontal, em que a similaridade entre p_1 e p_2 é medida pela comparação entre blocos W que cercam cada um dos pixels, como ilustra a Figura 2.14. A similaridade (correlação) dos blocos

pode ser medida pela soma das diferenças absolutas e a disparidade d de um pixel na posição (x, y) na imagem I_1 pode ser calculada como:

$$d = (x, y) = \arg \min \sum_{0 \leq d \leq d_{max}(i,j) \in W} |I_1(x + i, y + j) - I_2(x + i - \tilde{d}, y + j)| \quad (2.11)$$

em que d_{max} representa a disparidade máxima, limitando a busca, \tilde{d} é a disparidade candidata para um determinado par de blocos, i e j definem as coordenadas dos pixels pertencentes aos blocos e $d(x, y)$ é a disparidade computada para o pixel na posição (x, y) da imagem I_1 , com relação à menor diferença de similaridade entre os blocos comparados.

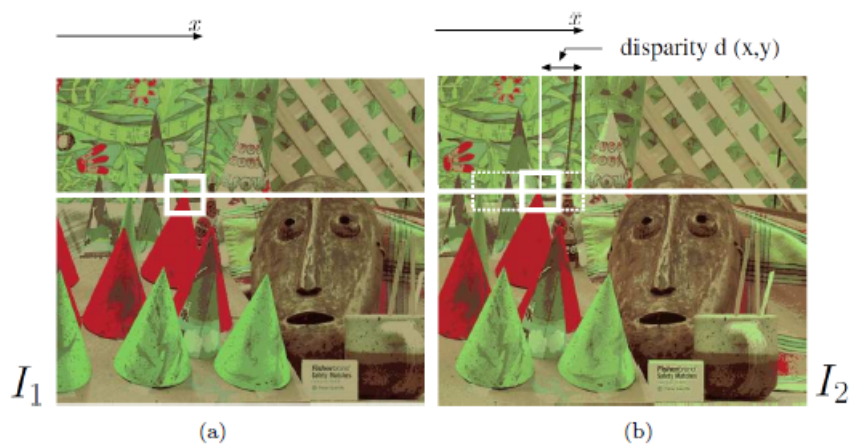


Figura 2.14: Estimação da disparidade pela verificação de similaridade entre blocos (regiões destacadas na cor branca) da imagem de referência I_1 e da imagem I_2 ao longo da linha epipolar.

Com base em Sun (2003), os principais problemas inerentes à estimação de mapas de profundidade são listados a seguir:

- Ruído: variações na iluminação e ruídos presentes nas imagens produzem diferentes intensidades de cores entre as vistas, fazendo com que a correspondência dos pontos estimada seja incerta.
- Regiões sem textura: em regiões com cor constante, a medida de similaridade calculada pode ser a mesma para todos os valores de disparidade estimados, resultando em valores de disparidade incorretos.
- Descontinuidades de profundidade: no mapa de profundidade, descontinuidades abruptas dos valores de profundidade estão tipicamente associadas com as bordas dos objetos. A determinação da medida de correlação de blocos próximos às descontinuidades possibilita que um bloco contenha porções de objetos com duas

profundidades diferentes (*foreground* e *background*), fazendo com que a medida de similaridade seja computada de forma incorreta.

- Oclusão: não é possível determinar a correspondência de pontos de regiões ocluídas.

Compreendido os conceitos de mapas de profundidade, podemos explicar o processo de síntese de vista que por sua vez utiliza-os na geração de vistas virtuais. Este processo pode ser melhor compreendido na Seção 2.6.

2.6 Síntese de Vista

O processo de síntese de vista consiste na criação de vistas virtuais a partir de uma ou mais imagens de referência e seus respectivos mapas de profundidade. Alguns trabalhos na literatura ((MÜLLER et al., 2008; OH; HO, 2009)) têm utilizado pelo menos duas imagens de referência. Por outro lado, para efeitos de experimentos, muitos outros trabalhos (MUDDALA; OLSSON; SJOSTROM, 2013; DARIBO; PESQUET-POPESCU, 2010; GAUTIER, 2011; AHN; KIM, 2012) têm utilizado somente uma imagem de referência. A Figura 2.15 ilustra o processos básicos da síntese de vista.

A Renderização baseada em Imagem de Profundidade (DIBR, do inglês *Depth-Based Image Rendering*), é uma das principais ferramentas utilizadas para a criação de vistas virtuais.

Conceitualmente, o DIBR pode ser dividido em três passos principais:

1. Pré-processamento do mapa de profundidade.
2. Deformação 3D.
3. Preenchimento de buracos.

Em (1) o mapa de profundidade é pré-processado com o objetivo de reduzir o grande número de buracos que são normalmente gerados pela deformação 3D (segundo passo). Além disso, o pré-processamento do mapa de profundidade também ajuda na redução do ruído geralmente presentes nesses mapas, diminuindo o erro no processo de deformação. Dependendo do algoritmo utilizado, este pré-processamento é capaz de reduzir a complexidade computacional do preenchimento dos buracos (terceiro passo). Como um exemplo, alguns trabalhos utilizam os filtros de Gauss como pré-processamento para remover completamente os buracos gerados pela deformação 3D, evitando assim a etapa de preenchimento de buracos.

Depois do pré-processamento no mapa de profundidade, a deformação 3D é usada para projetar os pixels da imagem de referência para o plano da imagem virtual. Dada

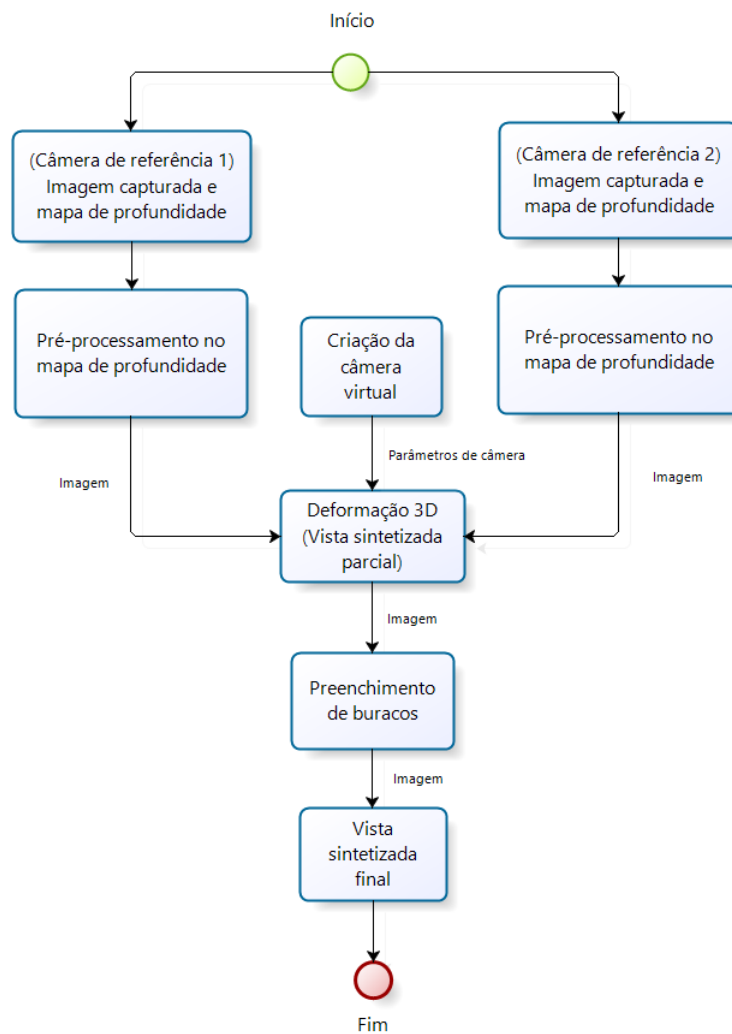


Figura 2.15: Etapas básicas da síntese de vista.

as imagens de referências, os correspondentes mapas de profundidade, e os parâmetros intrínsecos e extrínsecos da câmera (ver Seção 2.1) com os quais é criada a câmera virtual, os pixels de textura da imagem de referência são deformados para o plano de imagem virtual em duas etapas. Primeiro, os pixels em 2D no plano da imagem são projetados para pontos 3D no sistemas de coordenadas do mundo. Posteriormente, cada ponto 3D é reprojetoado no plano de imagem de destino. A Fig. 2.16 ilustra as configurações das câmeras virtuais em uma abordagem estereoscópica que pode ser utilizada na deformação 3D.

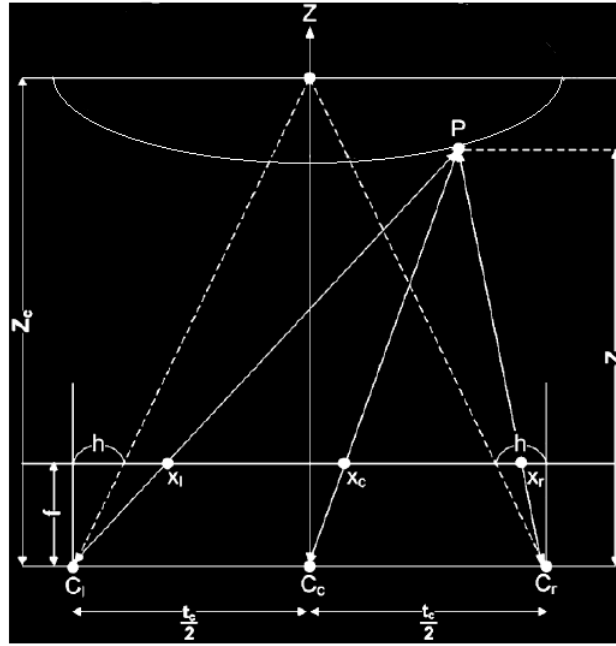


Figura 2.16: Configuração de câmeras para vistas estereoscópicas.

A partir da Figura 2.16 é possível determinar que as posições de pixels (x_c, y) , (x_l, y) , e (x_r, y) do original C_c , a vista esquerda virtual (C_L), e a visão virtual direita (C_r) respectivamente, estão relacionadas por:

$$x_l = x_c + \frac{t_c f}{2Z(x_c, y)} + h x_r = x_c + \frac{t_c f}{2Z(x_c, y)} - h \quad (2.12)$$

em que: f é a distância focal; t_c é a distância da linha de base a partir da visão virtual esquerda (C_l) para a exibição virtual direita (C_r); e h depende da distância de convergência selecionado (Z_c), e é dada por:

$$h = \frac{-t_c f}{2Z_c} \quad (2.13)$$

Quando usar imagens cinza para representar informações de profundidade, cada valor de pixel normalmente varia de 0 a 255. A distância profundidade original deve, então, ser

quantificado neste intervalo. Em vez de uma quantização linear, uma abordagem comum é a utilização de uma não-linear quantização que considera fatores humanos e melhora a profundidade percebida, tal como:

$$Z = \frac{1}{\frac{1}{Z_{min}}(\frac{Z}{255}) + \frac{1}{Z_{max}}(1 - \frac{z}{255})} \quad (2.14)$$

onde Z é a profundidade da cena real; Z_{min} e Z_{max} denotam a distância mais próxima e o mais longe possível, respectivamente; e z é o valor da profundidade em escala cinza no mapa de profundidade.

Na etapa de projeção pode ser que para uma determinada posição de pixel da câmera virtual, podemos ter as seguintes possibilidades:

- Não há projeção de pixels naquela posição, ocasionado por oclusão nas duas vistas. Tendo como consequência, a geração de buracos na imagem virtual, já que não há informação de pixels naquela posição.
- 1 ou mais pixels são projetados de apenas uma vista devido as oclusões.
- 1 ou mais pixels de diferentes vistas. Nesta possibilidade, diferentes pixels das vistas de referência podem ser projetados na mesma posição. O pixel de cada vista a ser inserido na posição é o de menor distância em relação a câmera virtual. Caso eles possuam distância muito próxima em relação à câmera virtual, significa que os pixels de cada cena pertencem ao mesmo objeto. Assim, o pixel a ser utilizado na imagem virtual será a interpolação dos dois pixels.

Após a etapa de deformação 3D, os pixels vazios na exibição de destino (buracos) são preenchidos por meio de interpolação utilizando informações dos pixels vizinhos. Contudo, as interpolações realizadas nem sempre conseguem manter uma qualidade visual desejada. Por conta disso, técnicas de *inpainting* têm sido utilizadas no preenchimento dos buracos resultantes do processo de síntese de vista.

2.7 *Inpainting*

Inpainting é o nome dado pelos especialistas em arte para o processo manual de restauração de pinturas de forma que fique imperceptível aos observadores tal restauração (COUTO, 2010). O processo de modificação de uma obra de arte danificada é uma prática tão antiga quanto a própria criação artística. Técnicas de restauração de obras passaram a ser desenvolvidas intensamente no período da renascença, quando artistas reconstruíam manualmente falhas em pinturas feitas na era medieval. Com o advento da tecnologia, as imagens e vídeos digitais passaram a ter o mesmo tratamento, mas de forma automatizada

por um computador, cuja a visão humana é utilizada com sucesso para julgar a qualidade final da reconstrução automática (RODRIGUES, 2006).

O conceito de *inpainting* digital foi introduzido mais formalmente por Bertalmio (2000), estabelecendo o termo na comunidade científica. Pode-se definir *inpainting* como um problema de interpolação em duas dimensões, onde são conhecidos os valores de pixels em torno de uma região desconhecida Ω , e se deseja obter/estimar estes valores no interior de Ω (ver Figura 2.17). As pesquisas em Processamento de Imagens têm criado mecanismos automáticos utilizados para reparar falhas ou buracos em imagens, ou ainda remover digitalmente objetos de uma cena. Em geral, tais técnicas utilizam o conhecimento da fisiologia humana, visando gerar imagens sem contornos quebrados, ou seja, priorizando a continuidade das bordas.



Figura 2.17: Representação genérica do processo de *inpainting*.

O problema de *inpainting* pode ser declarado como segue: seja I a imagem original (ou um *frame* em uma sequência de vídeo), que é composta por uma área de origem, indicada por Φ , cujos valores de pixel são conhecidos, e uma parte da imagem denominada área (ou região) de *inpainting*, denotada por Ω , representando a região danificada que deve ser reparada ou a região a ser preenchida. Como mostrado na Figura 2.17, estas não são áreas sobrepostas, ou seja, $I = \Phi \cup \Omega$, e $\delta\Omega$ representa o limite ou fronteira entre as áreas de origem e de *inpainting*.

A área de *inpainting* (Ω) é a parte da imagem que será preenchida, e é geralmente definida de forma manual pelo usuário. Esta região representa a parte da figura que se deseja completar, e pode representar uma parte da imagem que contém algum tipo de falha de informação, que pode ser a degradação de uma imagem digital, ou uma fotografia digitalizada que possua algum tipo de falha física, entre outras possibilidades. Também

pode-se definir uma área de *inpainting* que cubra um objeto que se deseja remover da imagem original.

Outro item importante nas técnicas de *inpainting* é a fronteira da região ($\delta\Omega$). A fronteira da área de *inpainting* é o conjunto de pontos ao redor da área a ser preenchida, que contém informações importantes para que a técnica de preenchimento tenha sucesso. Em geral, as técnicas de *inpainting* analisam essa fronteira e identificam, de alguma forma, por onde deve começar e qual informação será usada no preenchimento.

De uma forma geral o preenchimento da região de *inpainting* começa a ser realizado no sentido da fronteira para o interior da região. Cabe a técnica utilizada definir onde começar e como preencher esta região. Esses fatores determinam a propagação da estrutura da imagem na região de *inpainting*, que impactam na continuidade dos pontos preenchidos em relação aos vizinhos já existentes. De fato, esse ponto de discussão e diferenciação entre as abordagens existentes, é um fator crucial para o sucesso da restauração. Assim, os mecanismos de *inpainting* desenvolvidos geralmente consideram fortemente estes dois pontos: a ordem de preenchimento e a forma de preenchimento (propagação da informação) da região de *inpainting* (CASACA, 2014).

Para o problema de *inpainting* é essencial entender a diferença entre a estrutura e a textura de uma imagem. A estrutura, pode ser definida com sendo as principais partes - objetos de uma imagem, cuja superfície é homogênea sem quaisquer detalhes. Por sua vez a textura pode ser definida com sendo os detalhes sobre a superfície dos objetos que tornam as imagens mais realistas (SANGEETHA, 2011).

No que tange o *Inpainting* uma grande quantidade de métodos e técnicas já foram propostas na literatura. Estes métodos e/ou técnicas são baseados em:

- Equações Diferenciais Parciais (EDPs)
- Transformada de Wavelet
- Exemplar

Os algoritmos de *Inpainting* baseados em EDPs procuram preencher a área oclusa através da propagação de informação na direção de mudança mínima (SCHONLIEB, 2009; CASACA, 2010; YAO et al., 2010).

Os algoritmos baseados na Transformada de Wavelet (IGNACIO, 2007; PENEDO, 2014) tentam satisfazer o sistema visual humano utilizando análise de multi-resolução, separação de dados e propriedades estatísticas da imagem a ser preenchida.

Por outro lado, métodos baseados em exemplares como o proposto em Crimisi e Toyama (2004) tem sido mais comumente utilizados na realização do processo de *inpainting* (MACCHIAVELLO; DOREA, 2014; LEE; LEE; PARK, 2012; CHOI; HAHM,

2013; DARIBO; PESQUET-POPESCU, 2010; GAUTIER, 2011; AHN; KIM, 2012; MUD-DALA; OLSSON; SJOSTROM, 2013; CASACA et al., 2015).

2.8 Métricas de qualidade

Uma imagem pode sofrer degradação durante várias etapas de um sistema de processamento de imagens. Métricas de qualidade são técnicas empregadas para avaliar a similaridade de imagens que sofreram qualquer processo de transformação, em relação a uma imagem original sem ruído.

Segundo Pedrini e Schwartz (2007) algumas dessas métricas de qualidade são voltadas a avaliações subjetivas, as quais se baseiam na análise realizada por observadores humanos. Por outro lado, tem-se técnicas objetivas cuja função em geral é comparar a quantidade de degradação da imagem de entrada no sistema sem intervenção humana.

Dentre as diversas métricas objetivas presentes na literatura, podemos citar:

- PSNR
- SSIM

Tais medidas foram escolhidas para a aferição quantitativa dos resultados desta pesquisa com base em trabalhos correlatos da literatura de *inpainting* tais como (DARIBO; PESQUET-POPESCU, 2010; AHN; KIM, 2012; MUDDALA; OLSSON; SJOSTROM, 2013; CASACA, 2014; CASACA et al., 2015)

Cada uma desta são explanadas nas subseções a seguir:

2.8.1 Razão Sinal-Ruído de Pico (PSNR)

A razão sinal-ruído de pico (*peak signal-to-noise ratio* – PSNR) é uma métrica de referência completa bastante difundida por causa de sua simplicidade no cálculo e baixo custo computacional. Esta métrica basicamente avalia a energia da diferença entre o sinal de referência e o sinal-teste. A PSNR é definida por

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right] \quad (2.15)$$

para imagens codificadas com 8 bits. Quanto maior o valor da PSNR mais próximo é a imagem degradada da imagem original.

O MSE é calculado a partir do valor médio dos erros quadráticos entre os pixels da imagem original e da imagem degradada, sendo definido como:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2, \quad (2.16)$$

em que m e n representam as dimensões da imagem, $I(i, j)$ e $K(i, j)$ são valores de luminância dos pixels da imagem original e da imagem degradada na posição (i, j) , respectivamente. Valores pequenos de MSE indicam que a imagem degradada apresenta pouca distorção em relação à original. O valor de MSE igual a zero significa que as amostras de imagens submetidas à avaliação são idênticas.

Para uma imagem dada ou uma sequência de imagens, um PSNR alto geralmente indica uma alta qualidade e um PSNR baixo indica baixa qualidade.

Apesar das vantagens citadas, a metodologia de sensibilidade ao erro, ainda que seja coerente e simples, não é o método de avaliação ideal para reproduzir um índice de qualidade que se aproxime da qualidade visual percebida pelo Sistema Visual Humano (HVS - *Human Visual System*). Isto ocorre porque a medida fornecida por essas métricas não apresenta uma boa correlação com a qualidade realmente percebida (WANG; LU; BOVIK, 2004). Por exemplo, imagens com o mesmo valor de MSE podem possuir distorções totalmente distintas.

Por este motivo, nas três últimas décadas, têm sido estudados, de forma intensiva, métodos objetivos para medir a qualidade de imagem e vídeo de modo que seus valores reproduzam a qualidade realmente percebida pelo HVS. Wang, Lu e Bovik (2004) apresenta um paradigma para métricas objetivas, denominado índice de similaridade estrutural (SSIM).

2.8.2 Índice de Similaridade Estrutural - SSIM

Um dos modelos mais recentes e já consolidados para as métricas objetivas de avaliação da qualidade de imagens e vídeos é o SSIM (*Structural Similarity Index*) ou Índice de Similaridade Estrutural (WANG; LU; BOVIK, 2004). A métrica SSIM foi proposta com base no pressuposto de que o Sistema Visual Humano é altamente adaptado para extrair informações estruturais de imagens.

No método SSIM os pixels possuem forte dependência entre si e ela aumenta consideravelmente de acordo com a proximidade entre os pixels. Wang, Lu e Bovik (2004) afirmam que essa dependência carrega informações importantes sobre a estrutura dos objetos na imagem e que quantificar a mudança estrutural de uma imagem pode fornecer uma boa aproximação para a qualidade percebida.

O SSIM é um algoritmo que utiliza a estatística da imagem para a avaliação da qualidade, ou seja, os atributos que constituem a informação estrutural dos objetos da imagem, que dependem da média da luminância e do contraste da imagem. Os seus valores são

dados em escalas entre 0 e 1, que representam, respectivamente, a mínima e máxima qualidade no processo de comparação entre duas imagens. Em outras palavras, quanto mais próximo de 1, menor a degradação e mais próxima é a imagem degradada da imagem original.

As fórmulas para esta métrica são descritas nas Equações 2.17, 2.18, 2.19, 2.20 :

$$SSIM(X, Y) = \frac{(2\mu_x\mu_y + C1)(2\sigma_{xy} + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2 + \sigma_y^2 + C2)} \quad (2.17)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (xi - \mu_x)(yi - \mu_y) \quad (2.18)$$

$$C1 = (k1 * L)^2; \quad (2.19)$$

$$C2 = (k2 * L)^2; \quad (2.20)$$

- μ_x , valor médio da imagem X;
- μ_y , valor média da imagem Y;
- L, faixa dinâmica dos valores de pixels (255 por 8 *bits* de escala cinza);
- σ_{xy} , correlação entre as amostras da imagem de referência e de comparação, ambas subtraídos das suas respectivas médias;
- σ_x , variância da imagem X;
- σ_y , variância da imagem Y.

Capítulo 3

Revisão Bibliográfica

Este capítulo introduz mais formalmente alguns dos conceitos necessários para o melhor entendimento da técnica de *inpainting* baseada em exemplares, bem como esta aplicada em algoritmos voltados para imagens resultantes do processo de síntese de vista.

3.1 *Inpainting* digital baseado em exemplares

Proposto por Crimisi e Toyama (2004), o *inpainting* baseado em exemplares, considera a divisão da reconstrução da região de *inpainting* em duas fases: estrutura e textura. Este método utiliza a síntese de textura para preencher as regiões que contém padrões texturais bidimensionais. Para isso, o algoritmo procura amostrar novas texturas a partir de áreas preservadas na imagem e faz uma cópia das mesmas para as regiões a serem preenchidas, o que define de fato a abordagem baseada em exemplares.

Crimisi e Toyama (2004) salientam que uma síntese de textura que utiliza uma estratégia do tipo copiar-e-colar é suficiente para o *inpainting* de largas áreas texturizadas, desde que a ordem de preenchimento seja considerada neste processo. Isso é demonstrado pelos autores através de seus resultados, baseando-se apenas em exemplares de texturas.

O algoritmo proposto utiliza o vetor gradiente nas bordas da região de *inpainting* para a análise da direção dos pixels. A propagação dos pixels é uma questão de determinar qual bloco da região não-danificada é o mais adequado para região a ser restaurada de forma a manter a estrutura e textura da imagem.

A princípio, cada bloco (ou *patch*) a ser propagado necessita ter sua prioridade avaliada. Esta prioridade é calculada de acordo com a Equação (3.1), que define a prioridade $P(p)$ multiplicando o valor do Termo de Confiança $C(p)$ com o Termo de Dados $D(p)$ para um bloco centrado em um pixel p :

$$P(p) = C(p).D(p), \tag{3.1}$$

na qual o termo $C(p)$ representa a confiabilidade da informação no pixel p e o termo $D(p)$ indica a o grau de incidência entre a orientação local da região válida na fronteira com a geometria da fronteira. Assim, quanto maior a contribuição deste termo, mais o pixel na fronteira penetra na região de *inpainting*, significando que é um ponto importante na estrutura da imagem, e deve ser o primeiro a ser preenchido, de forma a preservar esta estrutura.

Crimisi e Toyama (2004) define o Termo de Confiança e o Termo de Dados como segue:

$$C(p) = \frac{\sum C(q)_{q \in \Psi_p \cap (I - \Omega)}}{|\Psi_p|}, \quad (3.2)$$

$$D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha} \quad (3.3)$$

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

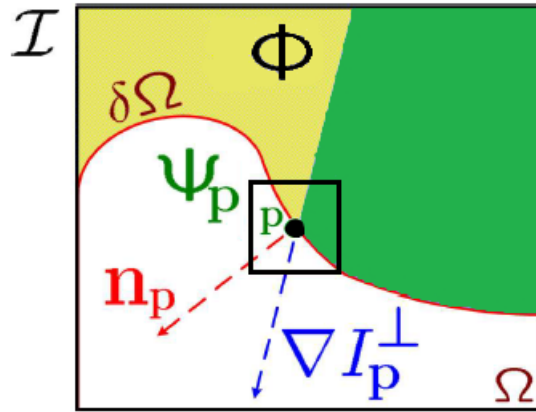


Figura 3.1: Notação definida por Criminisi e Toyama(2004).

A propagação da textura da imagem é feita por amostragem direta da região de origem. Nesta etapa é realizada uma busca pelo bloco original que melhor se adequa à nova área a ser preenchida, utilizando para isso a soma dos quadrados das diferenças (SSD - do inglês *Sum of Squared Differences*) dos pixels já preenchidos em ambos blocos.

Para melhor entendimento, na Figura 3.2 é apresentado o algoritmo, na qual se supõe que uma imagem em que existe uma área fonte Φ e uma área alvo Ω , contornada por $\delta\Omega$, claramente distinguível (Figura 3.2 (a)). O objetivo é sintetizar a área delimitada pelo fragmento Ψ_p , o qual é centralizado no ponto $p \in \Phi$, ilustrado na Figura 3.2.(b). Então, são computados os candidatos mais prováveis para preencher Ψ_p apresentados em $\delta\Omega$, por exemplo, $\Psi_{q'}$ e Ψ_q'' na Figura 3.2 (c). Entre os candidatos, existe um que mais se

aproxima do fragmento alvo, tendo seus pixels correspondentes copiados no fragmento Ψ_p . Este processo é repetido até o preenchimento completo de Ω . Na Figura 3.2 (d) pode-se notar que a textura, bem como a estrutura (a linha que separa as regiões cinzas, clara e escura), são propagadas sobre o fragmento Ψ_p .

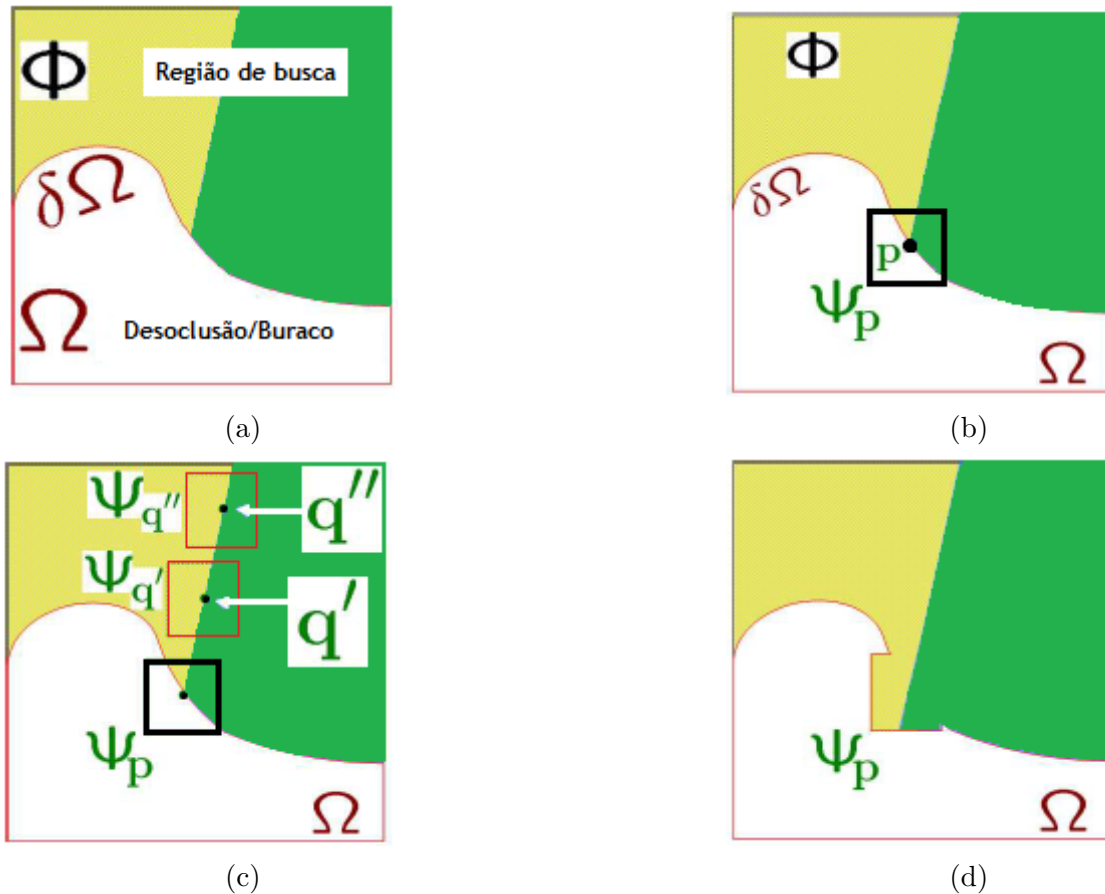


Figura 3.2: Síntese de textura baseada em modelos: (a) Imagem original. (b) Fragmento Ψ_p centralizado em $p \in \Phi$. (c) Candidatos mais prováveis $\Psi_{q'}$ e $\Psi_{q''}$. (d) Candidato mais provável é propagado para o fragmento alvo.

No algoritmo, cada pixel mantém um valor de cor (que é nulo nos pixels a serem preenchidos) e um valor de confiança, que reflete a confiança no valor de cor com base no momento em que este pixel foi definido. Durante a execução do algoritmo, os fragmentos localizados no contorno $\delta\Omega$ recebem um valor de prioridade temporário, baseados no Termo de Dados $D(p)$ e Termo de Confiança $C(p)$, definindo a ordem de preenchimento.

O algoritmo possui um processo iterativo é executado na seguinte sequência:

1. Computando prioridades dos fragmentos: O algoritmo realiza a tarefa de síntese através de uma estratégia de preenchimento *best-first*, que depende inteiramente dos valores de prioridade que são atribuídos a cada fragmento na frente de preen-

chimento. O cálculo da prioridade é tendenciosa para aquelas regiões que: a) estão sobre a continuação do bordas fortes e b) são cercadas por pixels de alta confiança.

2. Propagação das informações de estrutura e de textura: Uma vez que todas as prioridades na frente de preenchimento são computadas, o fragmento $\Psi_{\hat{p}}$ de maior prioridade é encontrado, sendo depois preenchido com dados extraídos da região fonte Φ .
3. Atualização de valores de confiança: Depois que o fragmento $\Psi_{\hat{p}}$ é redefinido com novos valores de pixel, o valor de confiança é atualizado na área delimitada por $\Psi_{\hat{p}}$.

3.2 *Inpainting* digital para imagens virtuais

Deve-se salientar que a aplicação de técnicas de *inpainting* digital não se restringe à recuperação de pinturas e fotografias digitalizadas. Imagens capturadas ou geradas digitalmente podem ser reparadas e objetos podem ser removidos em cenas estáticas. Por outro lado, uma abordagem onde o *inpainting* tem sido utilizado é no preenchimento de buracos provenientes da síntese de vista. Os algoritmos propostos por Bertalmio (2000) e Crimisi e Toyama (2004) tem servido como base para desenvolvimento de várias pesquisas na área.

Daribo e Pesquet-Popescu (2010) baseado nos trabalhos de Cheng (2008) e Oh e Ho (2009) em consonância também com os princípios de Crimisi e Toyama (2004) propuseram um algoritmo que estabelece uma nova forma de computar a prioridade dos pixels e realização da correspondência entre os *patches*.

Na computação da prioridade, Daribo e Pesquet-Popescu (2010) propõem a utilização do mapa de profundidade com objetivo de dar maior prioridade aos pixels pertencentes ao *background*, uma vez que as desocclusões normalmente aparecem ao lado dos objetos do primeiro plano (*foreground*). Para isso eles idealizaram uma forma diferente de calcular a prioridade do pixel, como segue na Equação 3.4:

$$P(p) = C(p).D(p).L(p), \quad (3.4)$$

na qual $L(p)$ é o Termo de Nível de Regularidade, definido como o inverso da variância do bloco de profundidade Z_p :

$$L(p) = \frac{|Z_p|}{|Z_p| + \sum_{q \in Z_p \cap \Phi} (Z_p(q) - \overline{Z_p})^2}, \quad (3.5)$$

na qual $|Z_p|$ é a área (em termos de número de pixels) do bloco da imagem de profundidade Z_p centralizado em p , $Z_p(q)$ é o valor da profundidade do pixel na localização de q sob Z_p

e $\overline{Z_p}$ o valor médio do pixel. O Termo de Dados $D(p)$ e Termo de Confiança $C(p)$ possuem a mesma definição do trabalho idealizado por Crimisi e Toyama (2004).

Na propagação da textura da imagem, é realizada a amostragem direta da região de origem, porém, no processo de *matching*, o mapa de profundidade é levado em consideração tendo um peso β que aumenta sua relevância neste processo, como evidencia a equação a seguir:

$$\Psi_{\hat{q}} = \arg_{\Psi_q \in \Phi} \min \{d(\Psi_{\hat{p}}, \Psi_q) + \beta \cdot d(Z_{\hat{p}}, Z_q)\} \quad (3.6)$$

É importante ressaltar que no algoritmo proposto por Daribo e Pesquet-Popescu (2010), leva em consideração o mapa de profundidade já preenchido através de PDEs.

Assim como em Daribo e Pesquet-Popescu (2010), Gautier (2011) utilizam os princípios contidos no algoritmo proposto por Crimisi e Toyama (2004), porém definindo uma nova atualização no Termo de dados ($D(p)$) como segue na Equação 3.7:

$$D(p) = \alpha + (1 - \alpha) \exp \left(\frac{-C}{(\lambda_1 - \lambda_2)^2} \right), \quad (3.7)$$

onde C é uma constante valor positivo e $\alpha \in [0, 1]$. Regiões planas (quando $\lambda_1 \approx \lambda_2$) não favorece nenhuma direção, é isotrópico. Por outro lado, com bordas fortes (quando $\lambda_1 \gg \lambda_2$), a propagação começa ao longo dos isofotos.

Os autores realizam o cálculo da prioridade dos pixels através da exploração da informação de profundidade, em primeiro lugar, definindo um tensor 3D, e depois restringindo o lado por onde começar o *inpainting*. O tensor 3D permite a difusão da estrutura, não só ao longo da cor, mas também da informação de profundidade. Por sua vez, a restrição do lado de início do *inpainting* pode ser definida observando o movimento horizontal da câmera, verificando se o mesmo foi para a direita ou esquerda.

O *matching* - como também pode ser definida a busca pelo bloco original que melhor se adequa à nova área a ser preenchida - utiliza as somas das diferenças quadradas (ver Equação 3.8 e Figura 3.3).

$$\Psi_{\hat{q}} = \arg_{\Psi_q \in \Phi} \min d(\Psi_{\hat{p}}, \Psi_q), \quad (3.8)$$

com $d = \sum_{p,q \in \Psi_{p,q} \cap \Phi} \alpha_l \|\Psi_{\hat{p}} - \Psi_p\|^2$.

A abordagem definida por Ahn e Kim (2012) também segue o conceito básico introduzido por Crimisi e Toyama (2004), porém, assim como em Daribo e Pesquet-Popescu (2010) e Gautier (2011), propõe um novo esquema de prioridade e uma nova forma para correspondência de blocos para preenchimento das desocclusões das vistas virtuais.

O novo esquema de prioridade é impulsionado pelo emprego de uma estrutura de tensor no Termo de Dados, e estabelece um novo Termo de Confiança, que produz bons resultados

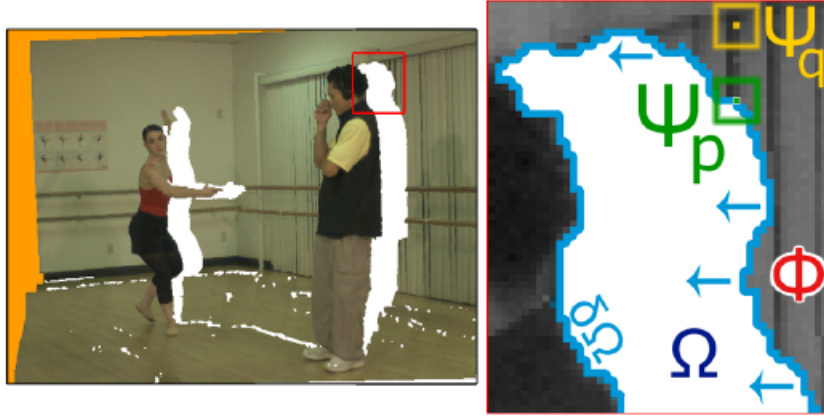


Figura 3.3: Princípio de Gautier.

de síntese, mesmo perto das fronteiras do primeiro plano da imagem (*foreground*). No esquema de correspondência, o mapa de profundidade é utilizado para distinguir as regiões fundo das regiões de primeiro plano, sendo que o bloco mais apropriado para ser copiado é escolhido através de uma nova medida de distância.

Tendo em vista que as desocclusões da imagem deformada para a vista esquerda (ou direita) aparece no lado esquerdo (ou direito) dos objetos em primeiro plano, Ahn e Kim (2012) propõe o preenchimento das desocclusões na direção oposta da deformação. Assim, o cálculo de prioridade é realizado apenas em um lado dos pixels do contorno da região de *inpainting*. Este conjunto único de pixels $\delta\Omega'$ é obtido através da aplicação da seguinte máscara HRM para a região dos buracos.

$$\delta\Omega' = (HRM \oplus B_1) - HRM, \quad (3.9)$$

onde \oplus denota um dilatação morfológica e B_1 é definido como:

$B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ se a deformação for para esquerda e $B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ se a deformação for para a direita. A aplicação da máscara HRM é expressada na Figura 3.4.

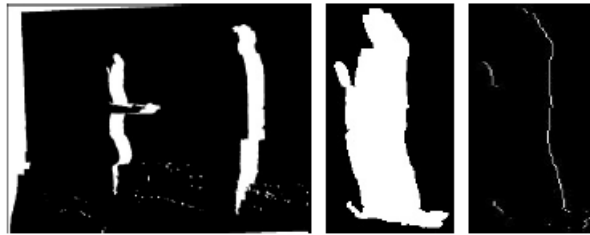


Figura 3.4: Aplicação da máscara HRM para obtenção da borda de um único lado.

Ahn e Kim (2012) também idealizaram uma nova forma de calcular a prioridade dos pixels pertencentes à borda baseando-se na propriedade de gradiente direcional da matriz Hessiana definida na Equação 3.10.

$$H = \begin{bmatrix} \sum_{(x,y) \in W} \frac{\partial^2 I_l(x,y)}{\partial x^2} & \sum_{(x,y) \in W} \frac{\partial^2 I_l(x,y)}{\partial x \partial y} \\ \sum_{(x,y) \in W} \frac{\partial^2 I_l(x,y)}{\partial x \partial y} & \sum_{(x,y) \in W} \frac{\partial^2 I_l(x,y)}{\partial y^2} \end{bmatrix} \quad (3.10)$$

onde I_l denota o valor do pixel no canais l que inclui R,G,B e também Z como canal de profundidade.

Os resultados experimentais, como ilustram as Figuras 3.5 e 3.6 demonstram que o método proposto fornece resultados visuais plausíveis na síntese de textura.



Figura 3.5: Resultado em vista virtual - Sequência Ballet - Deformação da câmera 5 para 4.



Figura 3.6: Resultado em vista virtual - Sequência Ballet - Deformação da câmera 5 para 2.

3.3 Outros trabalhos relevantes

Jawas e Suciati (2013) apresentam uma técnica de *inpainting* baseada em operações morfológicas de dilatação e erosão, que através da convolução de uma máscara da imagem com um elemento estruturante realiza o preenchimento das regiões de *inpainting*. A técnica proposta produziu bons resultados, mas identificou uma falha do algoritmo em danos horizontais.

Mais recentemente, Macchiavello e Dorea (2014) propuseram um algoritmo de preenchimento de desoclusões que inclui o valor da saliência no processo de busca do melhor bloco para síntese da textura. O algoritmo baseado em Crimisi e Toyama (2004) seleciona o bloco que possui ambos a menor distância e a menor saliência. Tendo em vista que regiões do fundo possuem menor saliência, os pixels do *background* foram priorizados e o processo de *inpainting* obteve bons resultados.

Capítulo 4

Metodologia

Considerando os diversos estudos mencionados na fundamentação teórica e na revisão bibliográfica, nesta parte do trabalho se descrevem as técnicas/e métodos que foram utilizados de tal forma a atingir o objetivo da pesquisa.

O algoritmo de *inpainting* proposto neste trabalho possui os processos representados na Figura 4.1 e estes são explanados nas secções subsequentes.

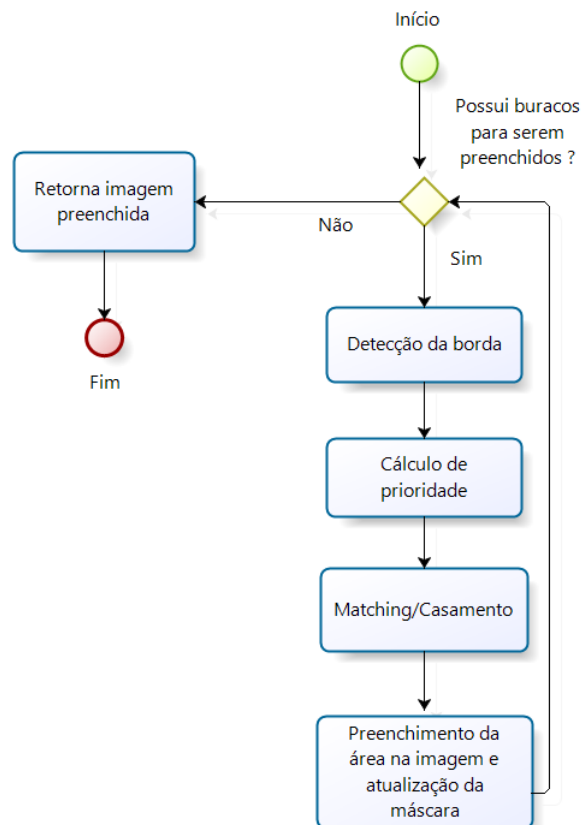


Figura 4.1: Fluxograma de processos do algoritmo proposto.

É importante ressaltar que o algoritmo proposto possui como entradas a imagem virtual, seu respectivo mapa de profundidade, bem como a máscara que denota os buracos gerados na síntese de vista.

4.1 Detecção da borda

No algoritmo proposto por Crimisi e Toyama (2004), a detecção da borda é feita ao longo de todo o buraco a ser preenchido. Por outro lado, imagens virtuais geradas através da síntese possui uma peculiaridade em questão:

- Deformações 3D que utilizam somente a câmera de referência à esquerda geram buracos do lado esquerdo dos objetos que se encontram no primeiro plano da imagem ou *foreground*, conforme a Figura 4.2a.
- Deformações 3D que utilizam somente a câmera de referência à direita geram buracos do lado direito dos objetos que se encontram no primeiro plano da imagem ou *foreground*, conforme a Figura 4.2b.



(a) Buracos à esquerda dos objetos pertencentes ao *foreground*

(b) Buracos à direita dos objetos pertencentes ao *foreground*

Figura 4.2: Diferentes posições dos buracos gerados na síntese de vista.

Levando-se em consideração a peculiaridade dos buracos das imagens virtuais geradas e a necessidade de priorizar os elementos do *background* no preenchimento destes, percebe-se que há um sentido de preenchimento a ser adotado para a obtenção de um resultado mais adequado, como ilustra a Figura 4.3.

Tendo em vista esta singularidade relacionada ao sentido do preenchimento dos buracos, não há fundamento para o uso da borda inteira do buraco, já que a propagação dos pixels deve seguir somente um sentido. Portanto, para a detecção da borda neste trabalho foi utilizada a ideia proposta por Ahn e Kim (2012), que detecta a borda somente do lado



(a) Preenchimento da esquerda para direita

(b) Preenchimento da direita para esquerda

Figura 4.3: Diferentes sentidos para preenchimento dos buracos.

aposto aos objetos do primeiro plano. Logo, a borda ($\delta\Omega'$) é definida segundo a Equação 4.1;

$$\delta\Omega' = (Mascara \oplus E_1) - Mascara, \quad (4.1)$$

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

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

Na Figura 4.4 é ilustrado um exemplo de detecção de borda utilizando um elemento estruturante $E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$.



(a) Máscara



(b) Borda detectada

Figura 4.4: Detecção da borda.

4.2 Cálculo da Prioridade

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

4.2.1 Termo de Confiança

No algoritmo proposto por Crimisi e Toyama (2004), o termo de confiança tem seus valores inicializados de tal forma que todos os componentes da região de busca assumem o valor 1. Todavia essa forma de inicialização pode causar distorções no resultado do preenchimento. É importante ressaltar que o termo de confiança dos pixels vizinhos à borda deve ser maior do que os pixels que estão mais distantes do buraco para evitar distorções no preenchimento. Dado este fato, o algoritmo proposto utiliza a ideia de inicialização empregada por Ahn e Kim (2012), como se segue:

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

na qual $T_1 \in \mathbb{Z}$ e \neg é a operação de negação. A Figura 4.5 ilustra o resultado desta inicialização em uma determinada borda.

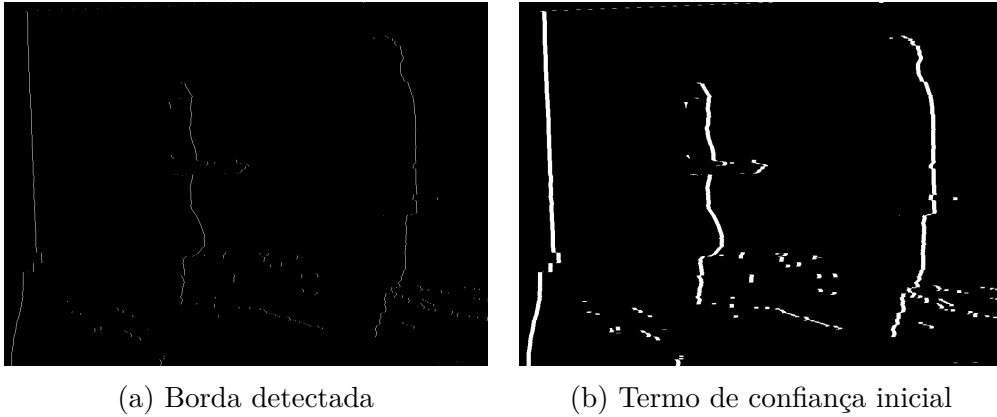


Figura 4.5: Inicialização do termo de confiança para a borda detectada.

4.2.2 Termo de Dados

Analisando o Termo de Dados proposto por Crimisi e Toyama (2004) e Daribo e Pesquet-Popescu (2010), verifica-se que o mesmo está exposto à ruídos na predição da direção da estrutura da imagem, uma vez que o operador gradiente é aplicado somente

nos pixels vizinhos com distância unitária do pixel em questão. Em outras palavras, o termo de dados obtido pode não refletir a estrutura da parte de uma imagem como todo, o que pode afetar no bom preenchimento do buraco.

Uma vez que o termo de dados é utilizado como meio de refletir a estrutura da imagem, e sabendo que esta última é evidenciada pelas bordas e descontinuidades da imagem em si, neste trabalho é proposto a substituição do termo de dados por um termo de relevância ($R(p)$). Esse termo de relevância é baseado no termo utilizado no trabalho de *inpainting* de imagens não diretamente relacionado à síntese de vista (CASACA et al., 2014).

Diferentemente das abordagens tradicionais, o termo de relevância aqui proposto computa a direção dos pixels com base em uma imagem auxiliar e o mapa de profundidade realçado. Esta imagem auxiliar é obtida através da difusão anisotrópica proposta por Perona e Malik (1990) ao qual visa suavizar a imagem, reduzindo consideravelmente a quantidade de ruído e outras informações de alta oscilação, mantendo grande parte das informações de borda e estrutura da imagem. Por outro lado, o mapa de profundidade que possui regiões segmentadas tem as bordas realçadas através de filtro *unsharp* para o cálculo do termo de relevância. As Figuras 4.6 e 4.7 ilustram os resultados obtidos após a difusão anisotrópica e o realce do mapa de profundidade. Já a Figura 4.8 evidencia o efeito do termo de relevância em estruturas de bordas.



(a) Frame 0 - Sequência Ballet



(b) Imagem processada

Figura 4.6: Resultado obtido para o frame 0 da sequência Ballet após difusão anisotrópica.



(a) Mapa de profundidade - Sequência Ballet

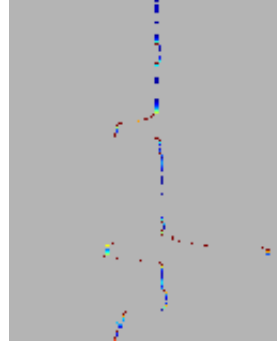


(b) Mapa de profundidade após filtro *unsharp*

Figura 4.7: Realce de mapa de profundidade após filtro *unsharp*.



(a) Corte de imagem



(b) Termo de relevância

Figura 4.8: Efeito do Termo de Relevância em parte da estrutura de uma imagem.

4.2.3 Prioridade

Tendo em vista os fatos apresentados neste processo, propõe-se para este algoritmo o cálculo de prioridade através da multiplicação entre o Termo de Confiança e o Termo de Relevância, como segue:

$$P(p) = C(p) \cdot R(p) \quad (4.2)$$

no qual $C(p)$ e $R(p)$ são definidos como:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I - \Omega)} C(q)}{|\Psi_p|}, \quad (4.3)$$

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

$$R(p) = \left| \nabla(\Delta u_p) \cdot \nabla(m_p) \cdot \vec{d}_p \right| \quad (4.4)$$

na qual ∇ é operador gradiente, Δ é o operador Laplaciano, u é a imagem auxiliar obtida com a difusão anisotrópica, m é o mapa de profundidade realçado, e \vec{d}_p é o vetor ortogonal ao gradiente definido como $\vec{d}_p = \frac{\nabla^\perp u_p}{|\nabla^\perp u_p|}$.

4.3 Matching

O *Matching*, ou casamento como também pode ser chamado, consiste na procura por blocos similares ao bloco do pixel que possui maior prioridade. Este processo em suas abordagens clássicas tem sido realizado através de funções de custo tais como SAD (*Sum of Absolute Differences*) ou SSD (*Sum of Square Differences*) nos três canais de cores das imagens (R,G,B). Entretanto, as cores em si não estão relacionadas diretamente com a estrutura da imagem, o que é uma das principais preocupações no processo de *inpainting*. Por outro lado, os mapas de profundidades como ilustra a Figura 4.9, trazem consigo uma distinção interessante entre objetos do primeiro plano para objetos do plano de fundo. Aproveitando esta peculiaridade, neste algoritmo é proposto a utilização do mapa de profundidade na busca do bloco similar.

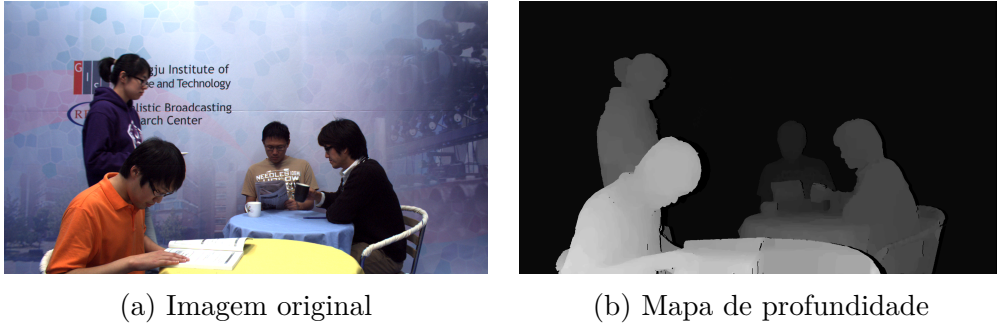


Figura 4.9: Imagem digital e seu respectivo mapa de profundidade.

Diferente das abordagens adotadas nos algoritmos proposto por Crimisi e Toyama (2004), Daribo e Pesquet-Popescu (2010), o algoritmo aqui proposto, assim como nas abordagens de Gautier (2011), Ahn e Kim (2012) e Casaca et al. (2014), adota uma região de busca menor e mais próxima ao bloco analisado. Isso aumenta o desempenho do algoritmo e a qualidade do *inpainting* realizado, já que os pixels vizinhos tendem a preencher mais adequadamente a região oclusa em questão mantendo a textura e estrutura da imagem.

Logo, para o processo de casamento, ou *Matching*, propõe-se:

$$\Psi_q = \arg_{\Psi \in \Phi'} \min d(\Psi_p, \Psi_q) + \beta \cdot d_m(\Psi_p, \Psi_q), \quad (4.5)$$

no qual Φ' é a região de busca do bloco, d é a distância SSD calculada para a imagem RGB, d_m é distância SSD calculada para o mapa de profundidade e β é um número inteiro maior ou igual a 1.

No método proposto neste trabalho, k candidatos são obtidos $\Psi_{q1}, \Psi_{q2}, \dots, \Psi_{qk}$ e posteriormente combinados linearmente. Os candidatos são os k blocos com menor distância segundo a Equação (4.5).

4.4 Preenchimento

Levando-se em consideração que há pequenos, médios e grandes buracos nas imagens resultantes em síntese de vista, estes pequenos buracos (área menor ou igual a 100 pixels) são preenchidos através de uma interpolação simples dos pixels vizinhos ao mesmos.

As abordagens da literatura utilizam normalmente o método *best-first*, que é uma estratégia gulosa. Entretanto, em alguns casos esta pode trazer alguns incômodos visuais, como pode ser visto na Figura 4.10. Por isso, neste algoritmo é proposto um preenchimento baseado em uma combinação linear de blocos com base em seus respectivos pesos.



Figura 4.10: Incômodo visual gerado pela estratégia gulosa.

Portanto, a combinação linear é expressa na Equação 4.6.

Dado os k candidatos, definidos anteriormente, são combinados para gerar a informação de preenchimento da seguinte forma:

$$\Psi'_p = (w_1 \Psi_{q1}) + (w_2 \Psi_{q2}) + \dots + (w_k \Psi_{qk}); \quad (4.6)$$

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

$$w_i = \frac{(\frac{1}{d_i})}{(\frac{1}{d_1}) + (\frac{1}{d_2}) + \dots + (\frac{1}{d_k})} \quad (4.7)$$

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

Após o preenchimento, a atualização da máscara é feita. O algoritmo continua o processamento até que toda a máscara esteja preenchida e, conseqüentemente todos os

buracos na imagem. Após a finalização do processo obtêm-se a imagem devidamente preenchida.

O Capítulo 6 detalha os resultados obtidos com os experimentos realizados.

Capítulo 5

Resultados Experimentais

Este capítulo apresenta os resultados obtidos com a implementação do algoritmo proposto neste trabalho, descrevendo maiores detalhes sobre os resultados obtidos discutindo as vantagens e oportunidades de melhoria no algoritmo. Também são apresentadas as formas de obtenção das imagens virtuais bem como algumas comparações entre o algoritmo proposto e outras abordagens disponíveis para realização do *inpainting* digital na literatura.

5.1 Obtenção das imagens virtuais

O software utilizado nesta etapa, o MPEG VSRS 3.5, utiliza uma ou duas imagens de entrada e, através de técnicas DIBR, a vista virtual que se deseja é gerada juntamente com seu respectivo mapa de profundidade e a máscara que denota os buracos existentes na vista.

Para fins de comparação com resultados publicados em trabalhos na comunidade científica, as imagens virtuais foram geradas a partir de uma única câmera de referência. Neste processo foram utilizadas sequências de imagens bastante difundidas na literatura:

- Ballet (MICROSOFT, 2016)
- Breakdancers (MICROSOFT, 2016)
- Ballons (UNIVERSITY, 2016a)
- Lovebirds1 (ELECTRONICS; INSTITUTE, 2016)
- Poznan (UNIVERSITY, 2016b)

Para cada uma destas sequências de imagens, foram geradas duas vistas virtuais. Cada sequência de vista virtual gerada contém 100 frames, o que representa um total de 1000

imagens virtuais geradas para os experimentos. Na Tabela 5.1 é possível ver os detalhes das sequências utilizadas no trabalho, bem como a porcentagem de degradação que denota a razão da quantidade de pixels que são considerados desoclusões e a resolução da imagem.

Tabela 5.1: Vistas virtuais geradas

Sequência	Vista de Referência	Vista virtual	Sigla	Resolução	Degradação (%)
Ballet	5	4	BA54	1024 x 768	12.22
Ballet	5	2	BA52	1024 x 768	30.78
Breakdancers	5	4	BR54	1024 x 768	5.5
Breakdancers	5	2	BR52	1024 x 768	14.41
Ballons	5	4	BL54	1024 x 768	2.22
Ballons	5	2	BL52	1024 x 768	8.28
Lovebirds	6	5	LV65	1024 x 768	0.75
Lovebirds	6	3	LV63	1024 x 768	2.04
Poznan	3	4	PZ34	1920 x 1088	1.79
Poznan	5	4	PZ54	1920 x 1088	1.85

5.2 Implementação do algoritmo proposto, execução e resultados

A implementação do algoritmo proposto foi desenvolvida com base na ideia chave descrita por Crimisi e Toyama (2004) - técnica baseada em exemplares - adotando também ideias que constam na literatura. Foi utilizado para isto a ferramenta MATLAB versão R2013b produzido pela companhia *The Math Works*.

Para a execução do algoritmo proposto foram utilizados 4 (quatro) computadores com Sistema Operacional Windows 8, processador Core I7 e com 8 gb de memória RAM. Estes executaram diferentes sequências simultaneamente, agilizando a obtenção dos resultados.

Para análise comparativa, as sequências de imagens virtuais formam o conjunto de entradas para o algoritmo de *inpainting* idealizado por Crimisi e Toyama (2004), Daribo e Pesquet-Popescu (2010), Gautier (2011) e Ahn e Kim (2012).

Neste sentido, foram realizadas dois tipos de análise: objetiva, que avaliam estatisticamente as imagens, e a qualitativa, realizada através de comparações entre a imagem no qual foi realizada o *inpainting* e a imagem original. Os resultados são discutidos e analisados sequência por sequência.

Para as sequências de imagens utilizadas no trabalho proposto obtivemos os valores de PSNR e SSIM expressos nas Tabelas 5.2 e 5.3. As células das tabelas que constam *** representam as sequências nas quais houveram erros na execução dos algoritmos analisados em questão.

Tabela 5.2: Valores médios de PSNR[dB] obtidos para cada sequência após *Inpainting*

Seq	Criminisi	Daribo	Gautier	Ahn	Proposto
BA54	27.09	28.91	28.70	32.43	32.12
BA52	25.29	24.96	24.51	25.78	26.23
BR54	29.46	24.71	28.59	29.52	29.98
BR52	26.82	26.82	26.26	26.96	25.42
BL54	32.75	32.70	***	***	32.94
BL52	25.98	26.32	***	***	26.68
LB87	28.95	28.94	***	26.06	29.03
LB85	30.10	30.17	***	23.74	30.27
PZ54	32.81	32.94	***	31.66	33.02
PZ34	32.58	32.87	***	26.13	32.97
GERAL	29.153	28.934	27.015	27.785	29.866

Tabela 5.3: Valores médios SSIM obtidos para cada sequência após *Inpainting*

Seq	Criminisi	Daribo	Gautier	Ahn	Proposto
BA54	0.9024	0.9040	0.9130	0.9369	0.9439
BA52	0.7759	0.7686	0.7654	0.7858	0.8197
BR54	0.9022	0.8430	0.8891	0.9136	0.9304
BR52	0.8455	0.8333	0.8411	0.8583	0.8329
BL54	0.9808	0.9811	***	***	0.8329
BL52	0.9269	0.9297	***	***	0.9324
LB87	0.9397	0.9395	***	0.9393	0.9403
LB85	0.9683	0.9680	***	0.9645	0.9699
PZ54	0.9800	0.9804	***	0.9792	0.9808
PZ34	0.9766	0.9772	***	0.9721	0.9780
GERAL	0.9198	0.9124	0.8521	0.9187	0.9161

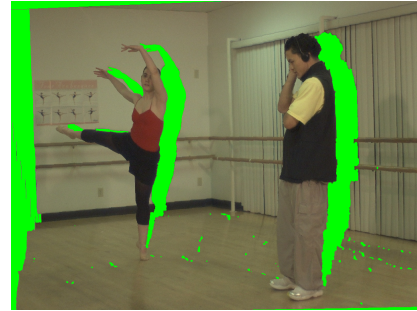
5.2.1 Ballet 5-4 (BA54)

Analisando os valores médios de PSNR e SSIM obtidos para a sequência BA54, verifica-se que o algoritmo proposto perde em 0.32 dB em PSNR para o algoritmo descrito por Ahn e Kim (2012). Por outro lado o valor médio de SSIM supera todos os algoritmos da

literatura em até 0.0415, o que representa um indicativo interessante dentro desta métrica. Os resultados subjetivos ilustrados na Figura 5.1 mostram, que apesar de possuir menor valor na métrica PSNR, as imagens preenchidas pelo algoritmo proposto chegam a possuir texturas melhores definidas, quando se observadas regiões ao lado da bailarina. Estes detalhes são recorrentes tanto no *frame* 0 quanto no *frame* 11. É interessante salientar que a estratégia de realizar o preenchimento de forma unilateral aumenta consideravelmente a qualidade visual do resultado final.



(a) BA54 - Frame 0 - Buracos/desoclusões



(b) BA54 - Frame 11 - Buracos/desoclusões



(c) BA54 - Frame 0 - Criminisi



(d) BA54 - Frame 11 - Criminisi



(e) BA54 - Frame 0 - Daribo



(f) BA54 - Frame 11 - Daribo



(g) BA54 - Frame 0 - Ahn



(h) BA54 - Frame 11 - Ahn



(i) BA54 - Frame 0 - Proposto



(j) BA54 - Frame 11 - Proposto

Figura 5.1: Imagens finais da sequência BA54 (frames 0 e 11) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.

5.2.2 Ballet 5-2 (BA52)

Com uma média de cerca de 30.78 % de degradação, a sequência BA52 é a que possui maior degradação dentre as demais sequências utilizadas neste trabalho. Nesta sequência, o algoritmo proposto demonstra robustez na manutenção da textura e estrutura das imagens preenchidas, mostrando que mesmo com grandes regiões de reconstrução consegue obter bons resultados, superando os demais algoritmos na métrica PSNR em até 1.71 dB e SSIM em até 0.0543 (veja Figura 5.2). Tanto o resultado ilustrado para *frame* 0 quanto para o *frame* 14, observa-se uma diferença nas regiões perto do primeiro plano da imagem, que notoriamente mostra que o algoritmo proposto consegue obter um melhor resultado visual, pois preserva de forma mais agradável as bordas e descontinuidades presentes do plano de fundo das imagens.



(a) BA54 - Frame 0 - Buracos/deso-clusões



(b) BA54 - Frame 14 - Buracos/deso-clusões



(c) BA54 - Frame 0 - Criminisi



(d) BA54 - Frame 14 - Criminisi



(e) BA54 - Frame 0 - Daribo



(f) BA54 - Frame 14 - Daribo



(g) BA54 - Frame 0 - Ahn



(h) BA54 - Frame 11 - Ahn



(i) BA54 - Frame 0 - Proposto



(j) BA54 - Frame 14 - Proposto

Figura 5.2: Imagens finais da sequência BA52 (frames 0 e 14) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.

5.2.3 Breakdancers 5-4 (BR54)

Esta sequência (a qual apresenta uma média 5.5% de degradação) possui buracos em diversas partes da imagem como ilustra na Figura 5.3, o que aumenta a complexidade do preenchimento. Todavia, ao se observar os resultados obtidos para a sequência BR54 é possível notar que o algoritmo proposto consegue sobressair em relação os demais algoritmos da literatura tanto na métrica PSNR quanto na métrica SSIM. As imagens ilustradas na Figura 5.3 elucidam os resultados obtidos pelas métricas avaliadas.



(a) BR54 - Frame 0 - Buracos/deso-clusões



(b) BR54 - Frame 52 - Buracos/deso-clusões



(c) BR54 - Frame 0 - Criminisi



(d) BR54 - Frame 52 - Criminisi



(e) BR54 - Frame 0 - Daribo



(f) BR54 - Frame 52 - Daribo



(g) BR54 - Frame 0 - Ahn



(h) BR54 - Frame 52 - Ahn



(i) BR54 - Frame 0 - Proposto



(j) BR54 - Frame 52 - Proposto

Figura 5.3: Imagens finais da sequência BR54 (frames 0 e 52) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.

5.2.4 Breakdancers 5-2 (BR52)

Os resultados obtidos para a sequência BR52, assim como a sequência BA54, possuem valores interessantes. Enquanto o algoritmo proposto perde para o algoritmo idealizado por Ahn e Kim (2012) na métrica PSNR, o mesmo acaba atingindo um melhor resultado quando comparado com os demais algoritmos da literatura. É possível notar que todas possuem alguns preenchimentos inadequados no que se concerne o preenchimento de partes das regiões com elementos do primeiro plano, principalmente os algoritmos propostos por Crimisi e Toyama (2004), Daribo e Pesquet-Popescu (2010). Neste sentido, subjetivamente falando, o algoritmo proposto conseguiu em alguns pontos se sobressair em relação aos demais.



(a) BR52 - Frame 61 - Buracos/desoclusões



(b) BR52 - Frame 75 - Buracos/desoclusões



(c) BR52 - Frame 61 - Criminisi



(d) BR52 - Frame 75 - Criminisi



(e) BR52 - Frame 61 - Daribo



(f) BR52 - Frame 75 - Daribo



(g) BR52 - Frame 61 - Ahn



(h) BR52 - Frame 75 - Ahn



(i) BR52 - Frame 61 - Proposto



(j) BR52 - Frame 75 - Proposto

Figura 5.4: Imagens finais da sequência BR54 (frames 61 e 75) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.

5.2.5 Ballons 5-4 (BL54)

Os resultados das médias das métricas PSNR e SSIM expressos nas Tabelas 5.2 e 5.3, bem como os gráficos nas Figuras 5.7 e 5.8 remetem a robustez do algoritmo proposto em relação aos demais algoritmos da literatura. Entretanto, nos gráficos ilustrados nas Figuras 5.5 e 5.6, os quais mostram as medidas de PSNR e SSIM *frame a frame*, é possível verificar que nos primeiros 50 *frames* o algoritmo proposto possui uma média inferior ao algoritmo descritos por Criminisi e Toyama (2004), Daribo e Pesquet-Popescu (2010) provavelmente devido à qualidade dos mapas de profundidades da sequência, enquanto que nos últimos 50 *frames* o algoritmo proposto se mantém superior na maioria dos frames.

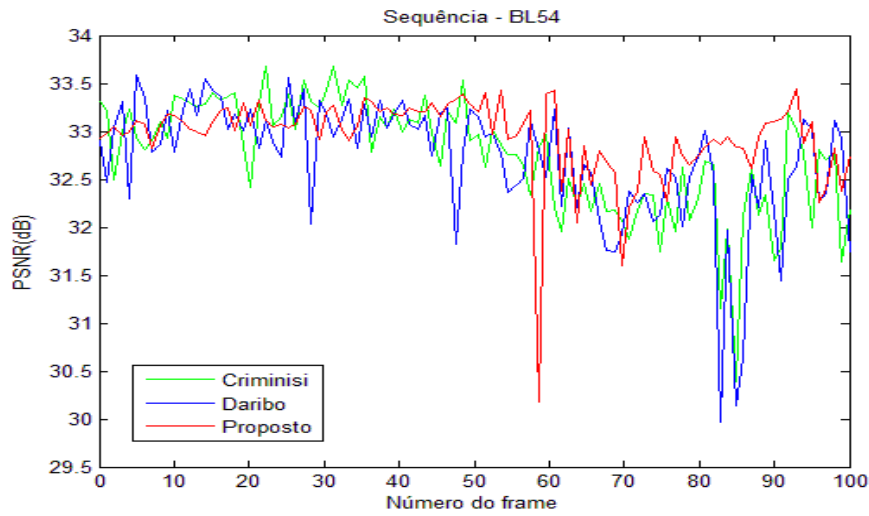


Figura 5.5: PSNR frame a frame - Sequência Ballons 54.

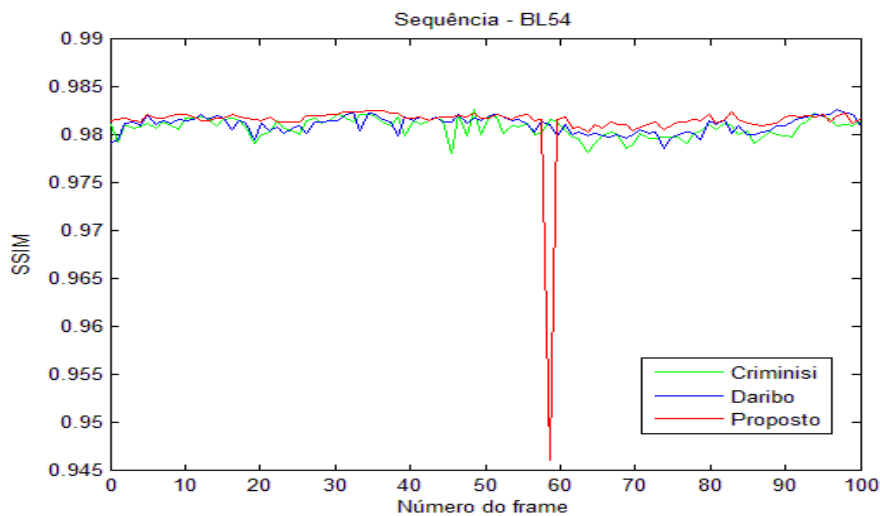


Figura 5.6: SSIM frame a frame - Sequência Ballons 54.

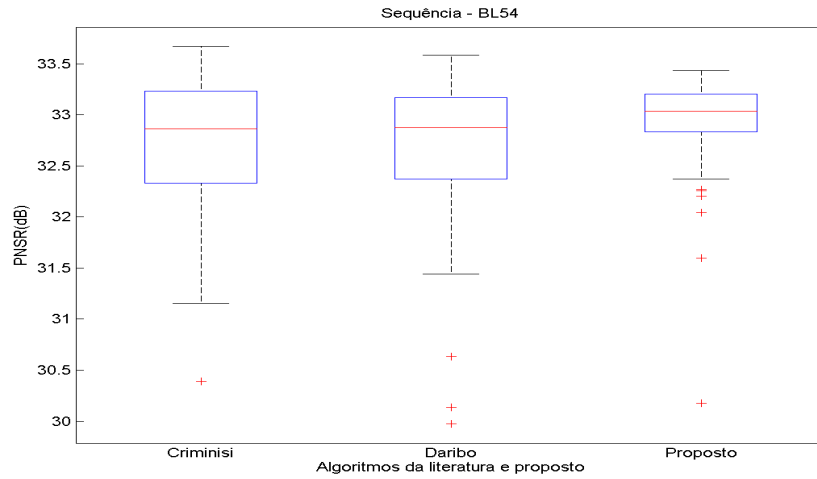


Figura 5.7: PSNR - Boxplot - Sequência Ballons 54.

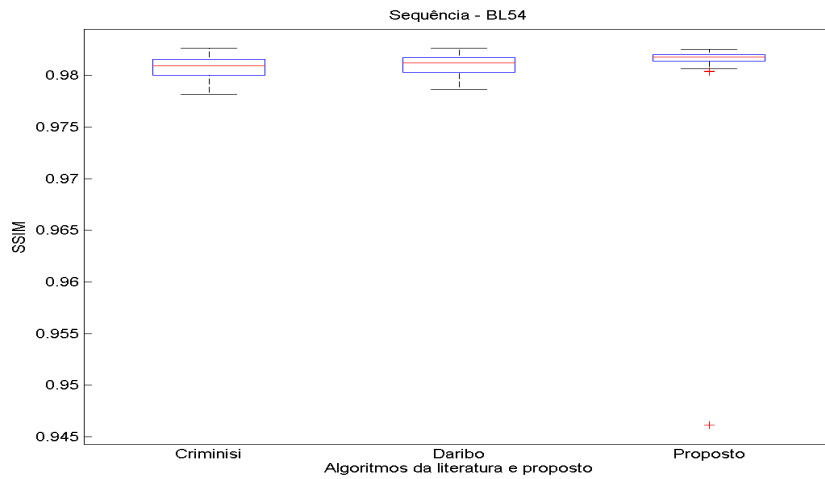


Figura 5.8: SSIM - Boxplot - Sequência Ballons 54.

Por outro lado, como ilustra a Figura 5.9, pode-se notar que os resultados são bem similares, já que os buracos são pequenos. Ao se fazer uma análise mais acentuada, como as ilustradas nas Figuras 5.10 e 5.11, é possível notar as pequenas diferenças entre as abordagens definidas em cada algoritmo da literatura e o aqui proposto.

Enquanto os algoritmos da literatura acabam preenchendo o buraco com informações inadequadas, o algoritmo proposto consegue manter a estrutura e texturas em uma condição mais regular.



(a) BL54 - Frame 0 - Buracos/desoclu-
sões



(b) BL54 - Frame 60 - Buracos/deso-
clusões



(c) BL54 - Frame 0 - Criminisi



(d) BL54 - Frame 60 - Criminisi



(e) BL54 - Frame 0 - Daribo



(f) BL54 - Frame 60 - Daribo



(g) BL54 - Frame 0 - Proposto



(h) BL54 - Frame 60 - Proposto

Figura 5.9: Imagens finais da sequência BL54 (frames 0 e 60) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.



(a) Criminisi

(b) Daribo

(c) Proposto

(d) Original

Figura 5.10: Comparações qualitativas de um segmento da sequência BL54 (Frame 0).

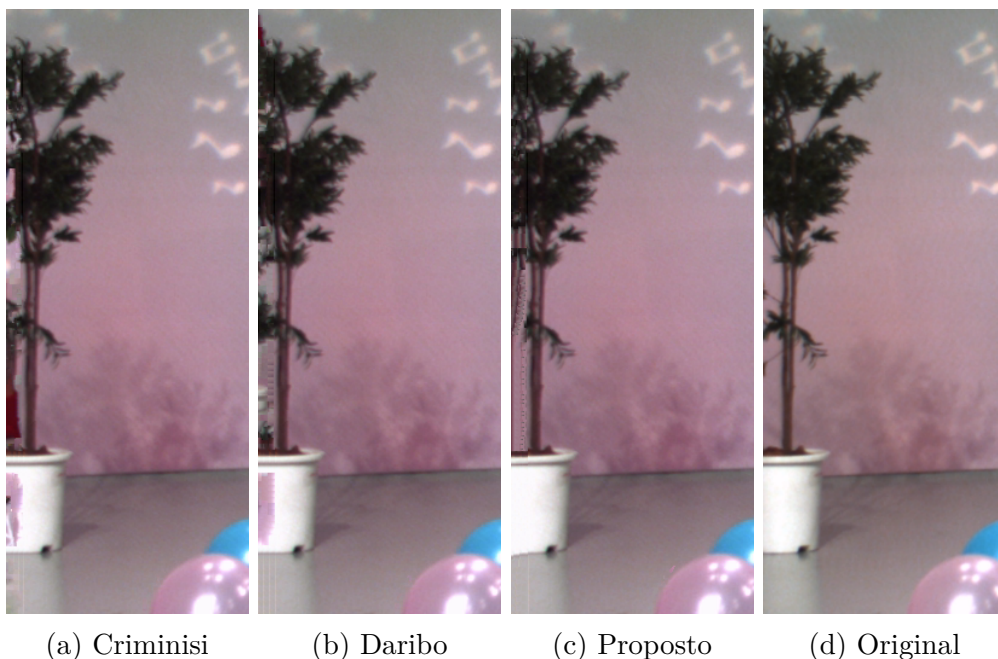


Figura 5.11: Comparações qualitativas de um segmento da sequência BL54 (Frame 60).

É possível notar nas bordas referentes dos balões na Figura 5.10 que o algoritmo proposto é que mais se aproxima da imagem original, mantendo melhor o contorno e preservando a textura de forma mais adequada.

Na Figura 5.11 é notável a diferença dos preenchimentos quando observado o vaso presente nas imagens. Os algoritmos de Criminisi e Daribo acabam por propagar informações de textura não condizentes com a cor natural do vaso, diferentemente do algoritmo proposto que mantém o padrão de cor necessário para uma textura mais harmonizada.

5.2.6 Ballons 5-2 (BL52)

Os resultados obtidos para a sequência BL52 expressos nas Tabelas 5.2 e 5.3, bem como os gráficos ilustrados nas Figuras 5.12 e 5.12 mostram que o algoritmo proposto possui um ganho considerável em relação os demais algoritmos da literatura, mesmo que alguns *frames* obtiveram restaurações não tão satisfatórias. Os gráficos ilustrados nas Figuras 5.14 e 5.15 revelam um menor variação nos resultados apresentados pelo algoritmo proposto, bem como uma maior mediana no conjunto dos resultados analisados para esta sequência.

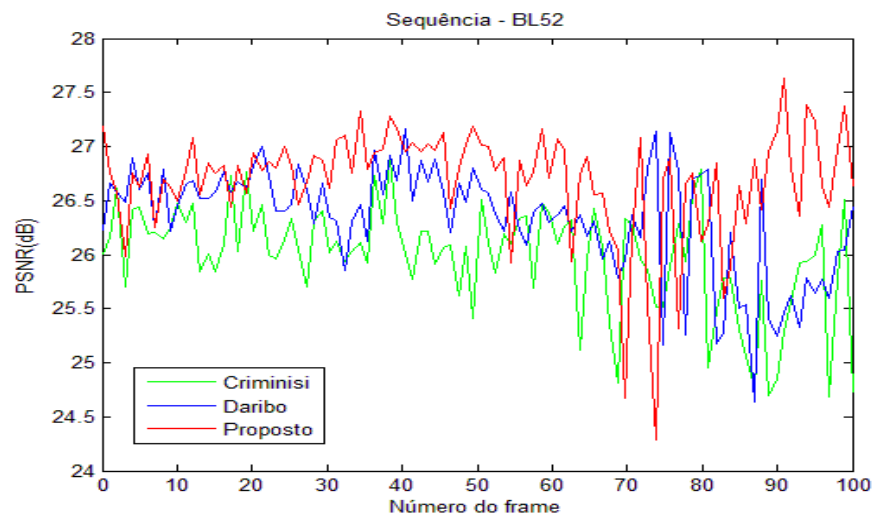


Figura 5.12: PSNR frame a frame - Sequência Ballons 52.

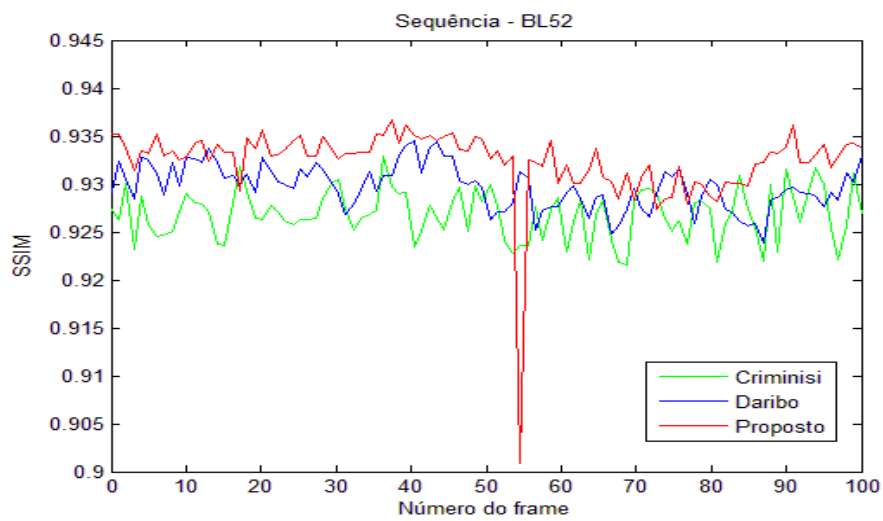


Figura 5.13: SSIM frame a frame - Sequência Ballons 52.

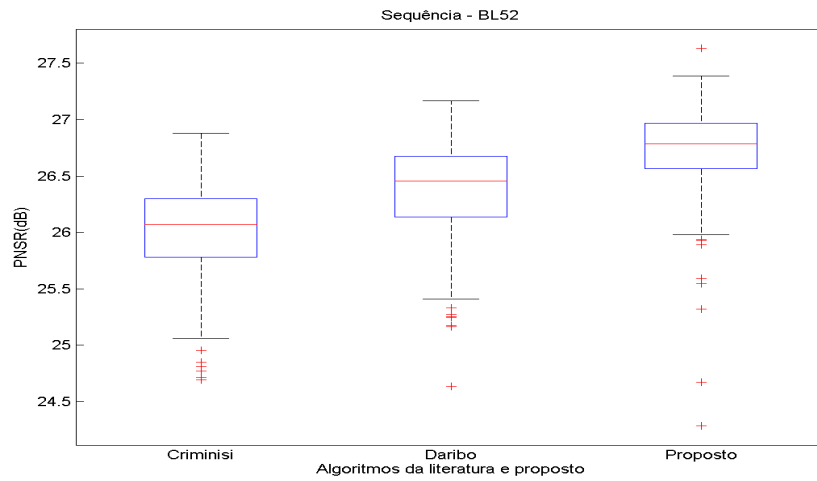


Figura 5.14: PSNR - Boxplot - Sequência Ballons 52.

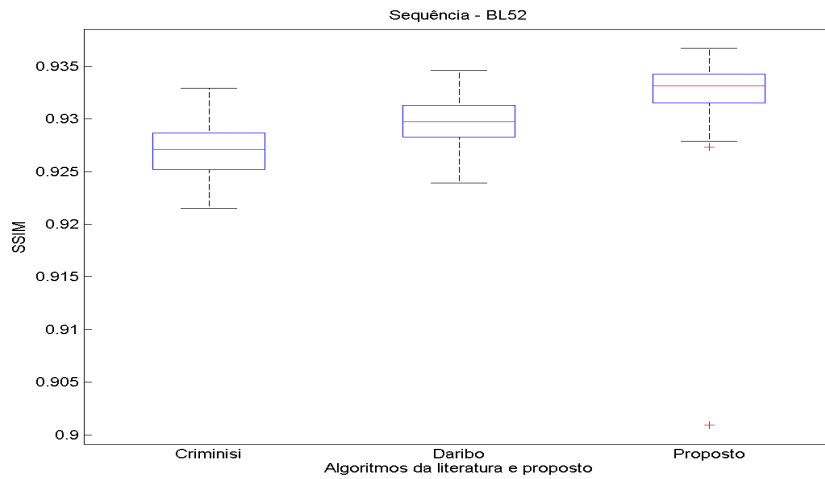


Figura 5.15: SSIM - Boxplot - Sequência Ballons 52.

Analisando os resultados subjetivos ilustrados na Figura 5.16 percebe-se diferenças de preenchimento principalmente no canto esquerdo da imagem, próximo ao vaso. Outros detalhes podem ser vistos nas Figuras 5.17 e 5.18.



(a) BL52 - Frame 06 - Buracos/deso-
clusões



(b) BL52 - Frame 57 - Buracos/deso-
clusões



(c) BL52 - Frame 06 - Criminisi



(d) BL52 - Frame 57 - Criminisi



(e) BL52 - Frame 06 - Daribo



(f) BL52 - Frame 57 - Daribo



(g) BL52 - Frame 06 - Proposto



(h) BL52 - Frame 57 - Proposto

Figura 5.16: Imagens finais da sequência BL52 (frames 06 e 57) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.

Nas Figuras 5.17 e 5.18 percebe-se que diferentemente dos demais, o algoritmo proposto consegue manter uma melhor harmonia de cores e de estrutura com relação ao *frame* original. É importante ressaltar que, quanto mais a estrutura da imagem são preservadas,

melhor qualidade visual alcançada.

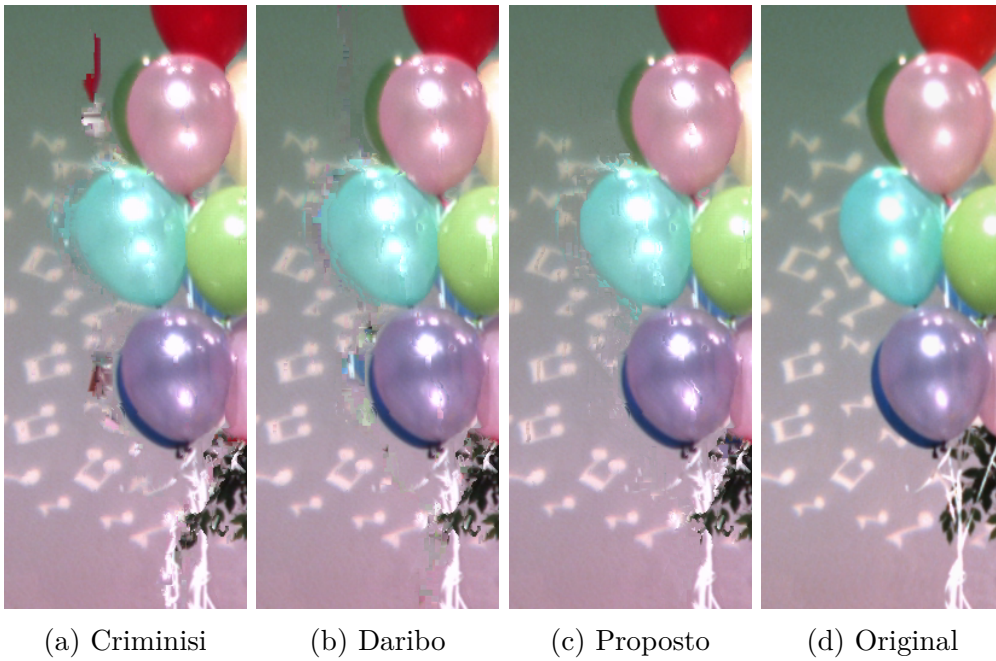


Figura 5.17: Comparações qualitativas de um segmento da sequência BL52 (Frame 06).

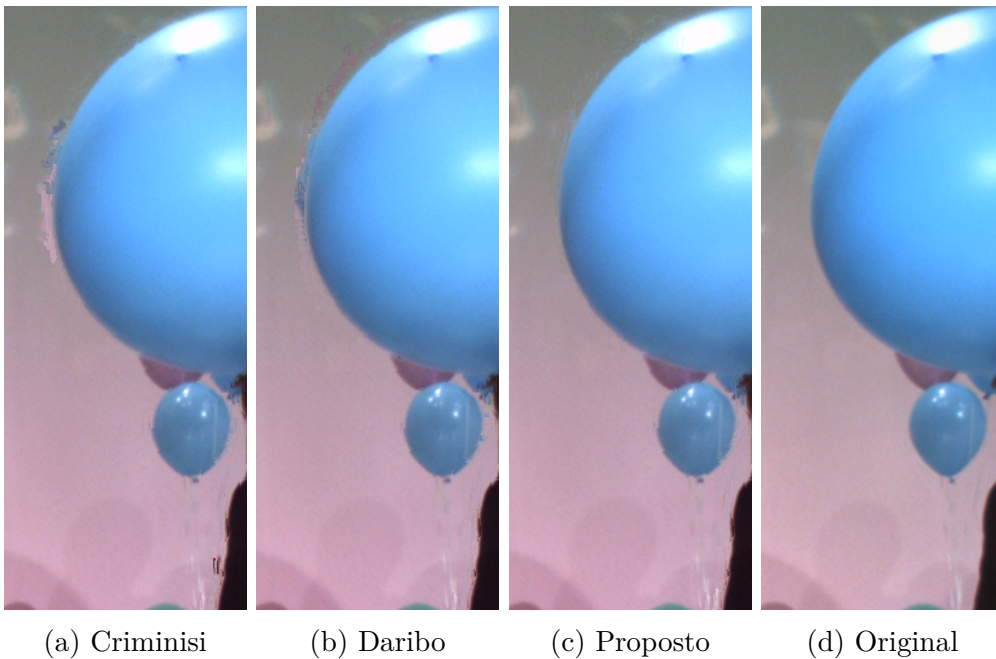


Figura 5.18: Comparações qualitativas de um segmento da sequência BL54 (Frame 57).

5.2.7 Lovebirds 8-7 (LB87)

Os resultados objetivos logrados na sequência LB87 apresentados nas Tabelas 5.2 e 5.3 da mesma maneira que os gráficos ilustrados nas Figuras 5.19 , 5.20, 5.21 e 5.22 mostram

que o algoritmo proposto possui melhores resultados no preenchimento dos buracos nas imagens resultantes da síntese de vista para esta sequência. É importante ressaltar que para esta sequência, o algoritmo proposto obteve maior média e mediana PSNR e SSIM em todos os *frames* em relação aos algoritmos da literatura aqui comparados.

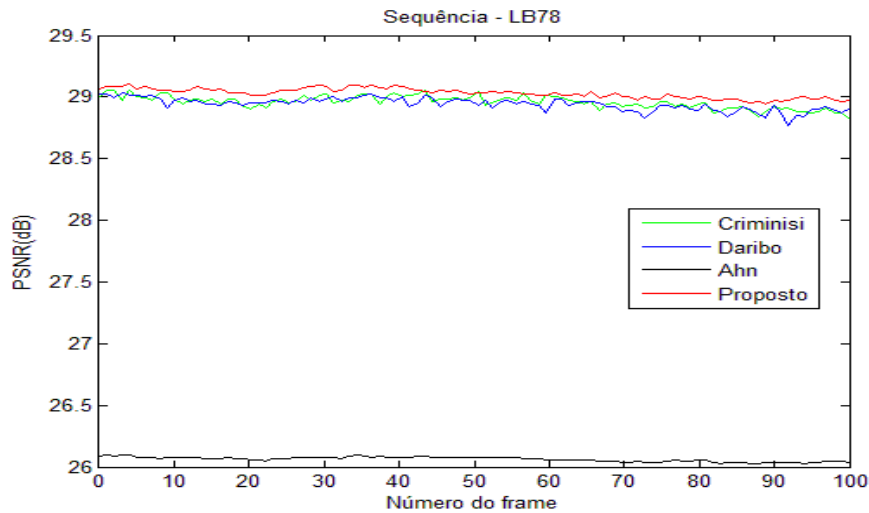


Figura 5.19: PSNR frame a frame - Sequência Lovebirds 87.

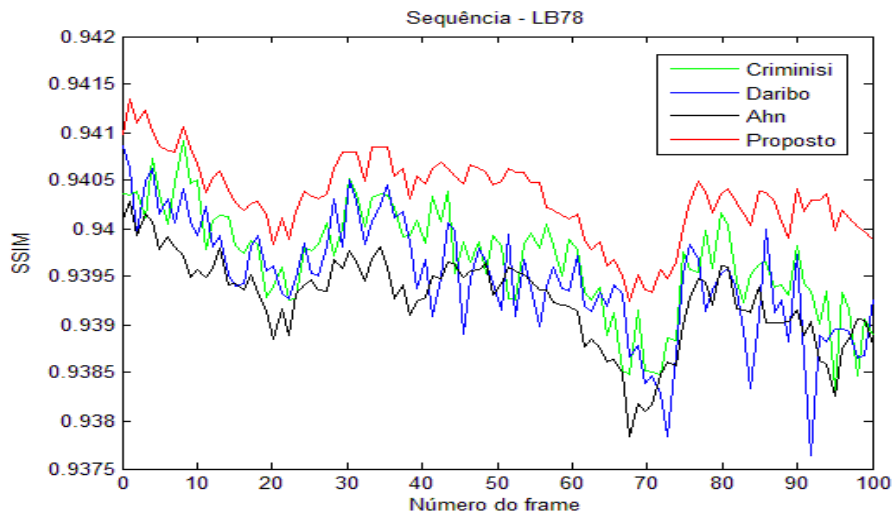


Figura 5.20: SSIM frame a frame - Sequência Lovebirds 87.

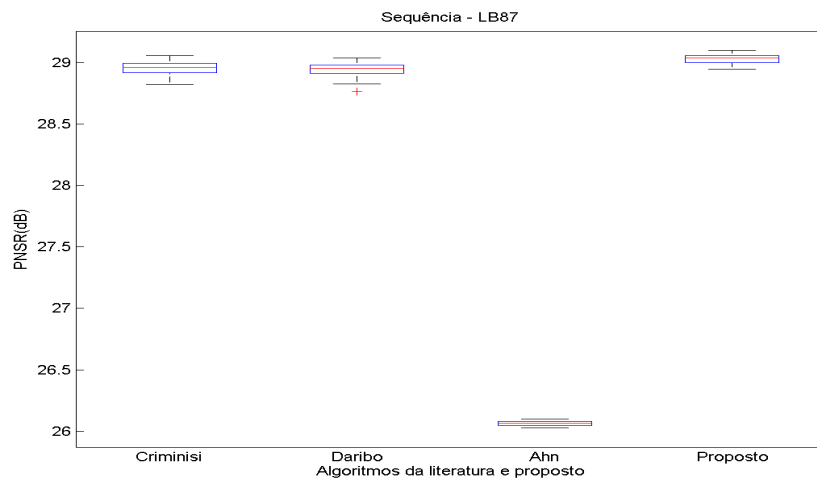


Figura 5.21: PSNR - Boxplot - Sequência Lovebirds 87.

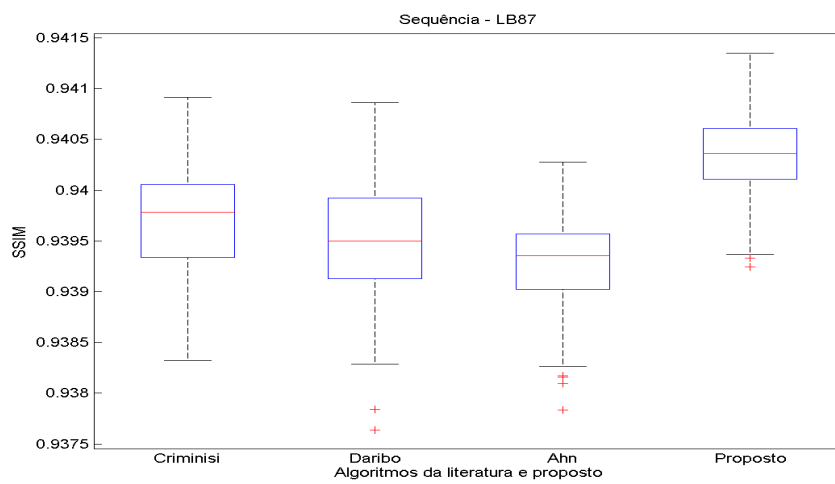


Figura 5.22: SSIM - Boxplot - Sequência Lovebirds 87.

Na Figura 5.23 não é possível observar com grande clareza as diferenças entre as diferentes abordagens, por conta dos tamanhos dos buracos existentes nas sínteses, os quais são relativamente pequenos.



(a) LB87 - Frame 02 - Buracos/desoclusões



(b) LB87 - Frame 18 - Buracos/desoclusões



(c) LB87 - Frame 02 - Criminisi



(d) LB87 - Frame 18 - Criminisi



(e) LB87 - Frame 02 - Daribo



(f) LB87 - Frame 18 - Daribo



(g) LB87 - Frame 02 - Ahn



(h) LB87 - Frame 18 - Ahn



(i) LB87 - Frame 02 - Proposto



(j) LB87 - Frame 18 - Proposto

Figura 5.23: Imagens finais da sequência LB87 (frames 02 e 18) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.

De outro ponto de vista, analisando as imagens mais profundamente, é possível perceber as diferenças existentes, principalmente ao lado esquerdo da pessoa de vermelho nas imagens ilustradas nas Figuras 5.24 e 5.25. O algoritmo proposto consegue manter as bordas e discontinuidades de maneira plausível em relação os demais algoritmos, o justifica as medidas PSNR e SSIM serem superiores para o algoritmo proposto.



(a) Criminisi

(b) Daribo

(c) Ahn

(d) Proposto

(e) Original

Figura 5.24: Comparações qualitativas de um segmento da sequência LB87 (Frame 02).

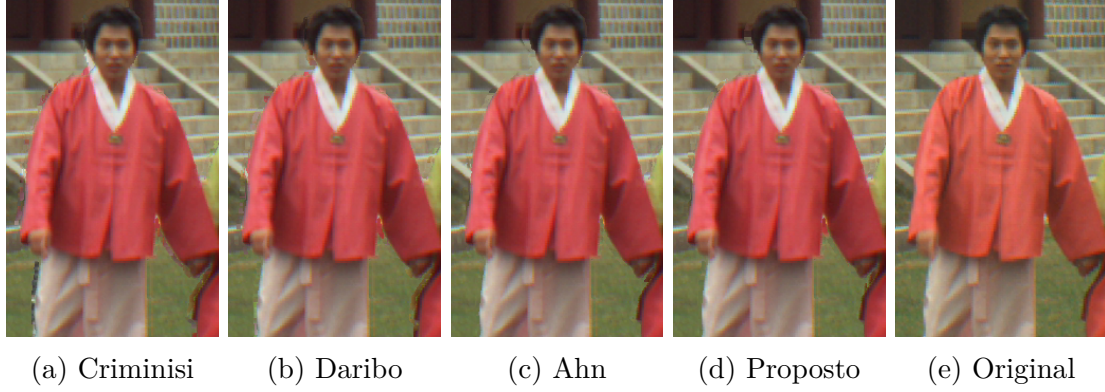


Figura 5.25: Comparações qualitativas de um segmento da sequência LB87 (Frame 18).

5.2.8 Lovebirds 8-5 (LB85)

É factível esperar que os valores médios de PSNR e SSIM para cada um dos algoritmos testados sejam bem próximos um dos outros. Contudo, em grande parte dos frames o algoritmo proposto consegue superar os algoritmos da literatura como verifica-se nos gráficos das Figuras 5.26 , 5.27 , 5.28 e 5.29. A diferença é mais notável na métrica SSIM, onde a desigualdade é mais acentuada como é observada no gráfico da Figura 5.29.

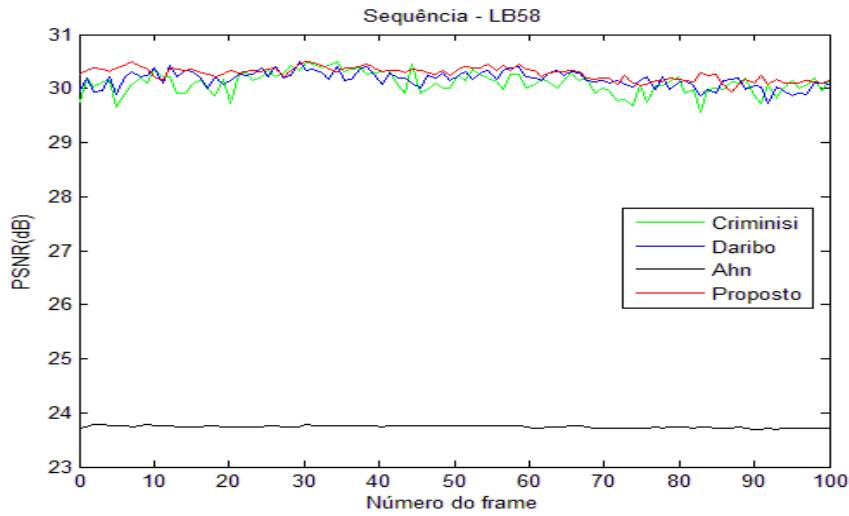


Figura 5.26: PSNR frame a frame - Sequência Lovebirds 85.

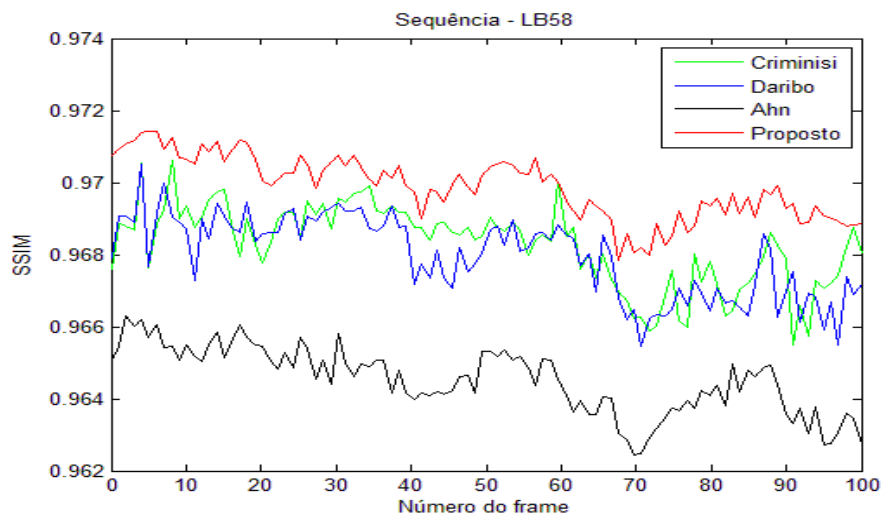


Figura 5.27: SSIM frame a frame - Sequência Lovebirds 85.

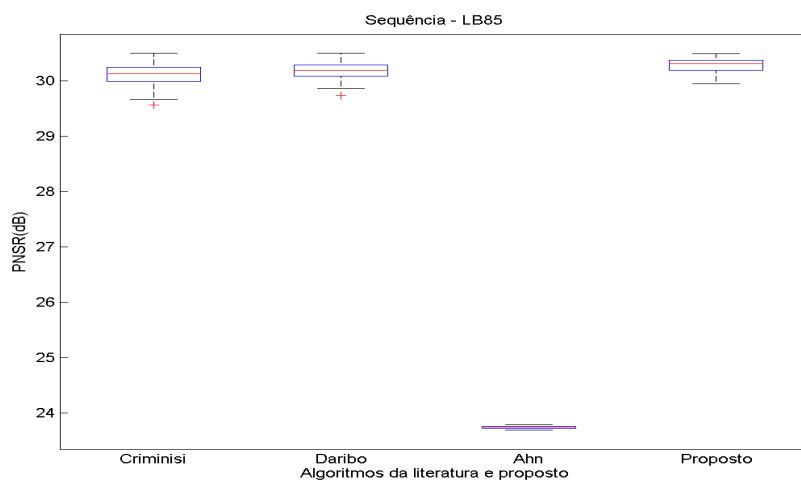


Figura 5.28: PSNR - Boxplot - Sequência Lovebirds 85.

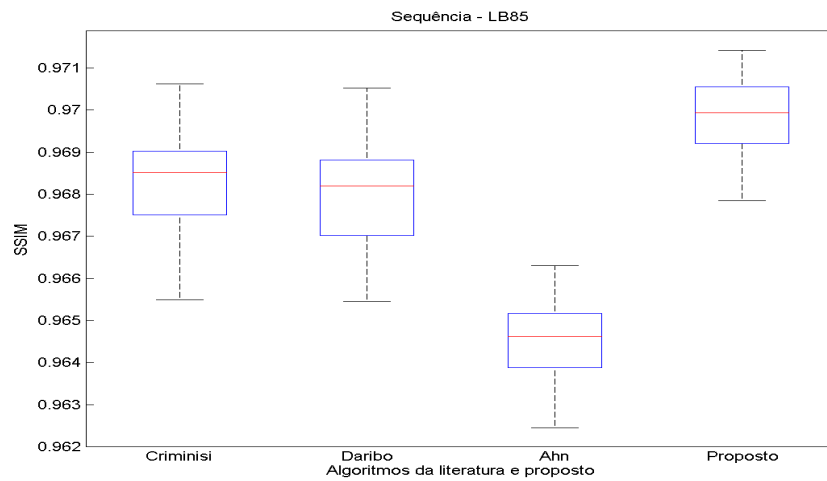


Figura 5.29: SSIM - Boxplot - Sequência Lovebirds 85.

Ao se observar as imagens resultantes na Figura 5.30, as diferenças visuais não são tão notáveis. Por isso, as Figuras 5.31 e 5.32 ilustram de forma mais detalhada as desigualdades entre os resultados.



(a) LB85 - Frame 08 - Buracos



(b) LB85 - Frame 53 - Buracos



(c) LB85 - Frame 08 - Criminisi



(d) LB85 - Frame 53 - Criminisi



(e) LB85 - Frame 08 - Daribo



(f) LB85 - Frame 53 - Daribo



(g) LB85 - Frame 08 - Ahn



(h) LB85 - Frame 53 - Ahn



(i) LB85 - Frame 08 - Proposto



(j) LB85 - Frame 53 - Proposto

Figura 5.30: Imagens finais da sequência LB85 (frames 08 e 53) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.

Assim como na sequência LB87, as principais diferenças podem ser vistas ao lado esquerdo da pessoa de vermelho presente nas imagens. Pode-se afirmar que mais uma vez o algoritmo proposto alcançou uma eficácia mais acentuada no que concerne a manutenção dos contornos e textura da imagem, quando se comparada com a imagem original.



(a) Criminisi (b) Daribo (c) Ahn (d) Proposto (e) Original

Figura 5.31: Comparações qualitativas de um segmento da sequência LB85 (Frame 08)



(a) Criminisi (b) Daribo (c) Ahn (d) Proposto (e) Original

Figura 5.32: Comparações qualitativas de um segmento da sequência LB85 (Frame 53).

5.2.9 Poznan 5-4 (PZ54)

Os resultados objetivos logrados para esta sequência, tanto na métrica PSNR como na métrica SSIM, revelam uma diferença pouco acentuada. Os gráficos ilustrados nas Figuras 5.33 e 5.34 enfatizam este ponto. Da mesma forma, os gráficos ilustrados nas Figuras 5.35 e 5.36 mostram um melhor variação entre os frames nas medidas PSNR e SSIM, tal como uma melhor mediana apresentada nos resultados do algoritmo proposto.

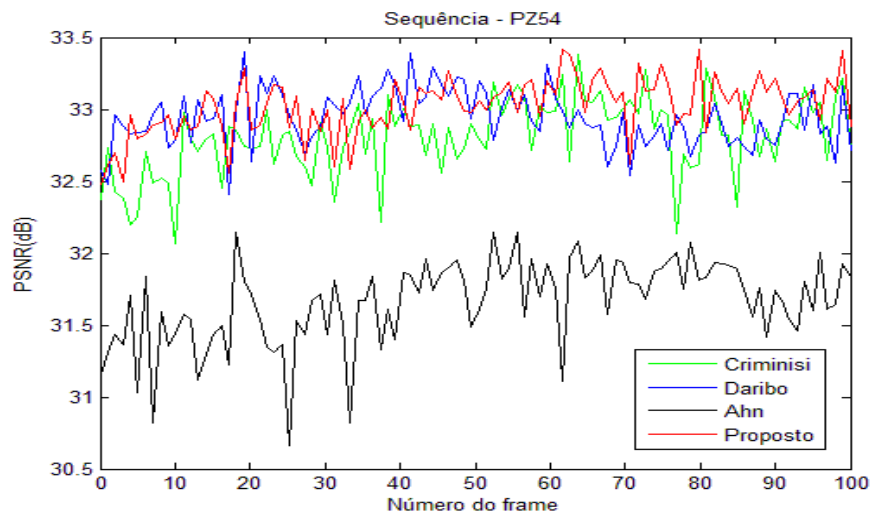


Figura 5.33: PSNR frame a frame - Poznan 54.

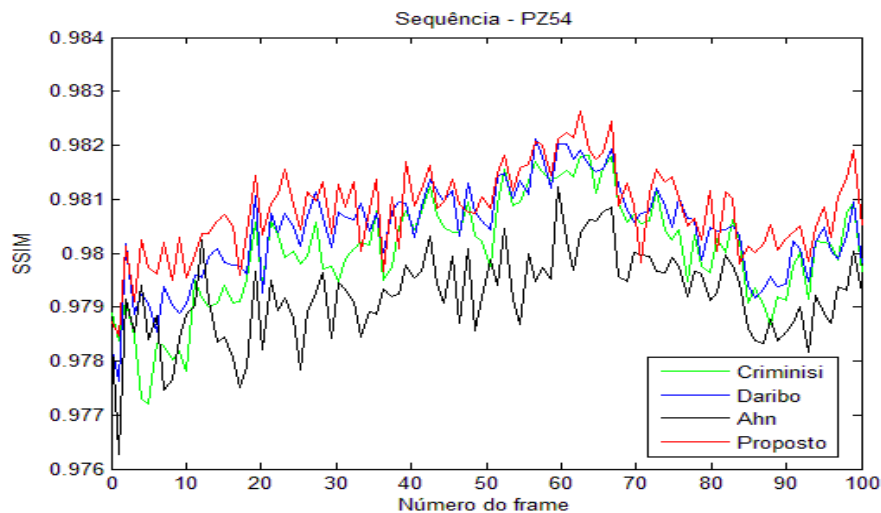


Figura 5.34: SSIM frame a frame - Poznan 54.

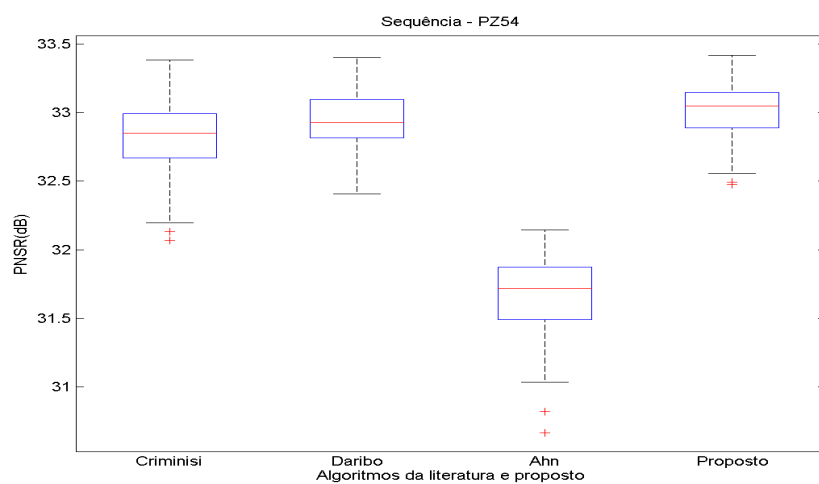


Figura 5.35: PSNR - Boxplot - Poznan 54.

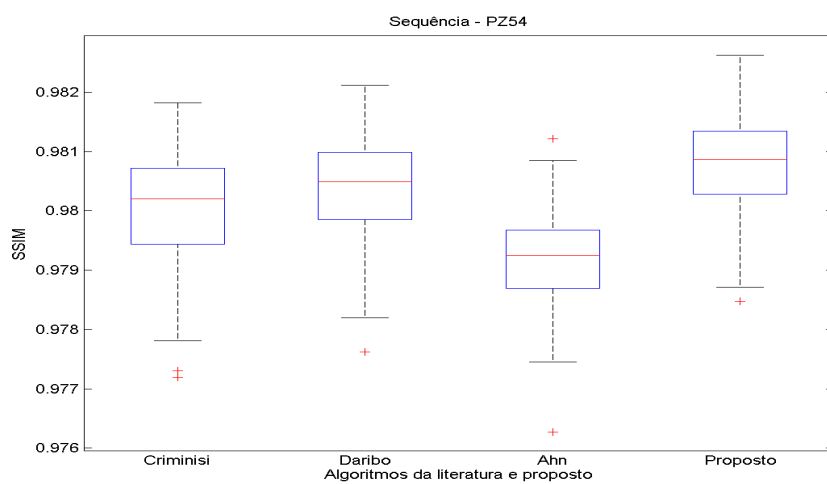
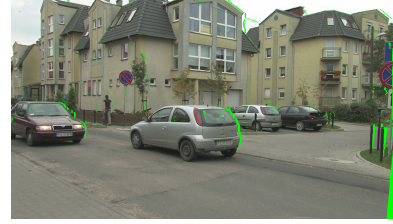


Figura 5.36: SSIM - Boxplot - Poznan 54.



(a) PZ54 - Frame 07 - Buracos



(b) PZ54 - Frame 23 - Buracos



(c) PZ54 - Frame 07 - Criminisi



(d) PZ54 - Frame 23 - Criminisi



(e) PZ54 - Frame 07 - Daribo



(f) PZ54 - Frame 23 - Daribo



(g) PZ54 - Frame 07 - Ahn



(h) PZ54 - Frame 23 - Ahn



(i) PZ54 - Frame 07 - Proposto



(j) PZ54 - Frame 23 - Proposto

Figura 5.37: Imagens finais da sequência PZ54 (frames 07 e 23) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.

A Figura 5.37 exibe resultados relativamente bem similares os quais possuem apenas pequenas diferenças. Estas pequenas diferenças são evidenciadas nas Figuras 5.38 e 5.39. É perceptível na Figura 5.38 que nenhum algoritmo obteve um preenchimento satisfatório para a estrutura e textura da imagem. O próprio algoritmo proposto obteve um resultado subjetivo inferior que os demais algoritmos.

Na Figura 5.39 é possível verificar que o algoritmo proposto alcançou um resultado subjetivo mais convincente, mantendo a textura e estrutura de forma mais satisfatória quando comparado com os demais algoritmos da literatura testados neste trabalho.



Figura 5.38: Comparações qualitativas de um segmento da sequência PZ54 (Frame 07).

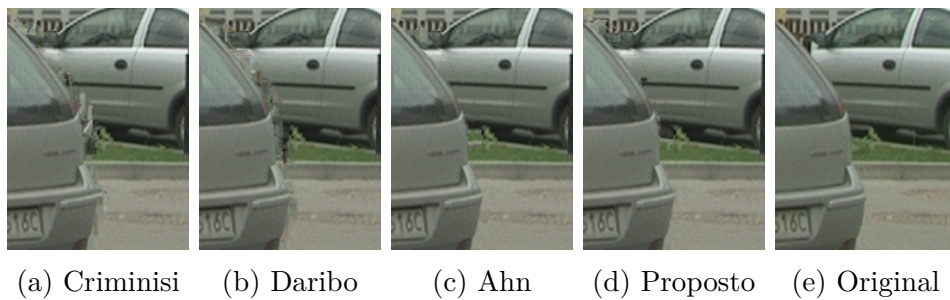


Figura 5.39: Comparações qualitativas de um segmento da sequência PZ54 (Frame 23).

5.2.10 Poznan 3-4 (PZ34)

Assim como a sequência PZ54, os resultados objetivos alcançados não possuem grandes discrepâncias, excetuando pelo resultado obtido pelo algoritmo de Ahn que não consegue preencher completamente os buracos presentes nas imagens, como pode ser visto na Figura 5.44. As demais imagens possuem grandes semelhanças quando comparadas superficialmente. Por outro lado, os gráficos ilustrados nas Figuras 5.42 e 5.43 denotam uma mediana superior alcançada pelo algoritmo proposto em relação aos algoritmos comparados.

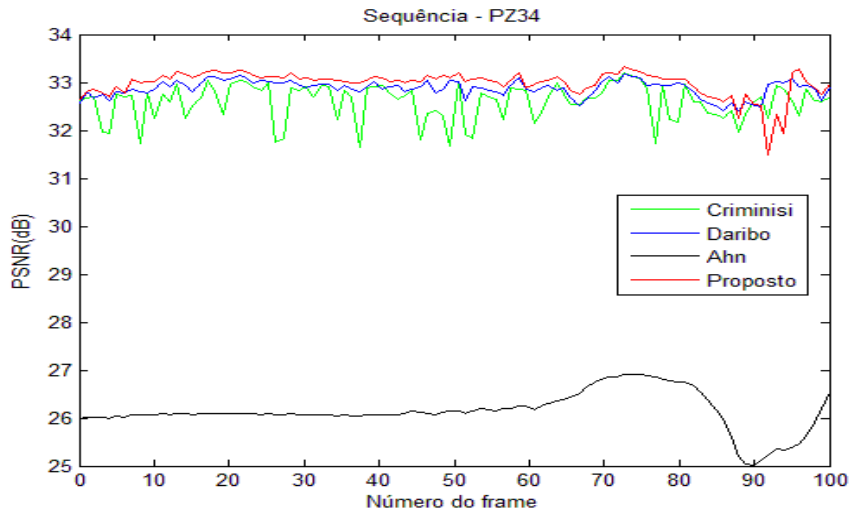


Figura 5.40: PSNR frame a frame - Poznan 34.

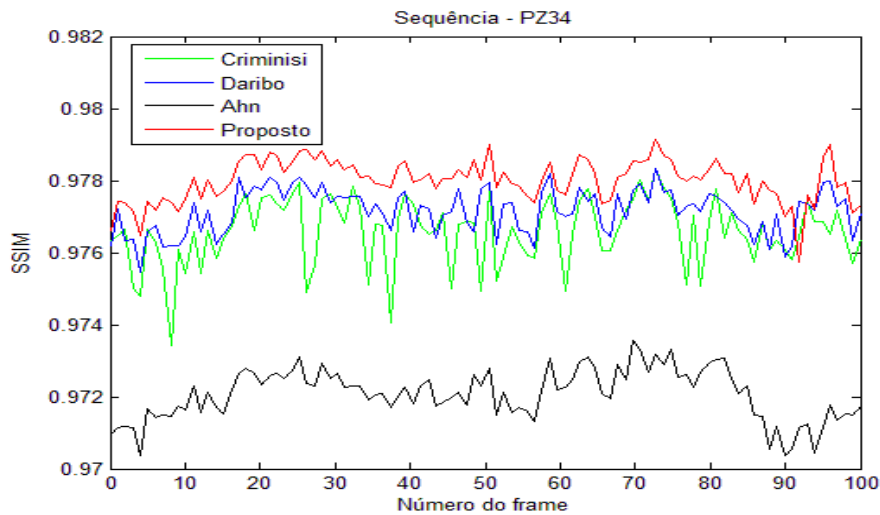


Figura 5.41: SSIM frame a frame - Poznan 34.

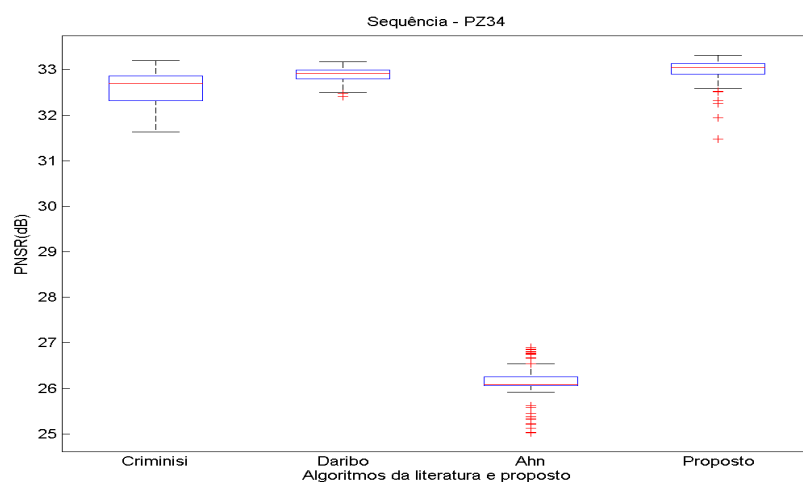


Figura 5.42: PSNR - Boxplot - Poznan 34.

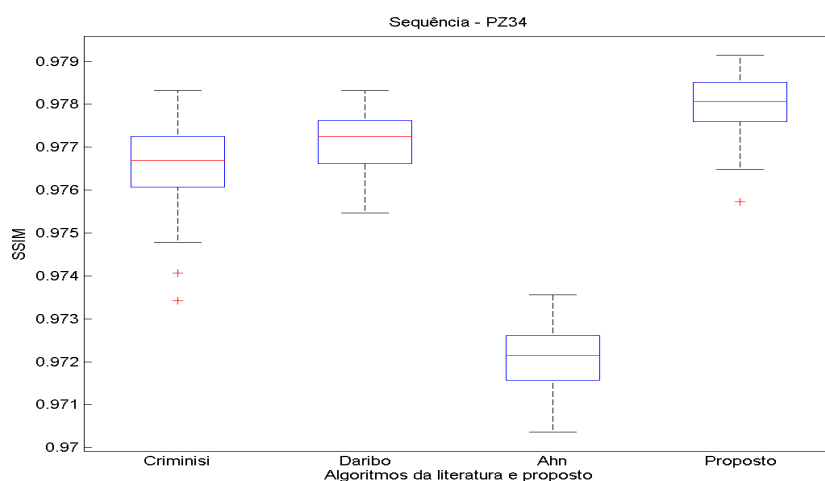


Figura 5.43: SSIM - Boxplot - Poznan 34.

Realizado-se um corte em cada uma destas imagens e analisando estes cortes de maneira mais profunda, é perceptível a visualização das diferenças entre o preenchimento dos algoritmos testados.



(a) PZ34 - Frame 07 - Buracos



(b) PZ34 - Frame 44 - Buracos



(c) PZ34 - Frame 07 - Criminisi



(d) PZ34 - Frame 44 - Criminisi



(e) PZ34 - Frame 07 - Daribo



(f) PZ34 - Frame 44 - Daribo



(g) PZ34 - Frame 07 - Ahn



(h) PZ34 - Frame 44 - Ahn



(i) PZ34 - Frame 07 - Proposto



(j) PZ34 - Frame 44 - Proposto

Figura 5.44: Imagens finais da sequência PZ34 (frames 07 e 44) após o *inpainting* com os algoritmos da literatura e o algoritmo proposto.



(a) Criminisi (b) Daribo (c) Ahn (d) Proposto (e) Original

Figura 5.45: Comparações qualitativas de um segmento da sequência PZ34 (Frame 07).



(a) Criminisi (b) Daribo (c) Ahn (d) Proposto (e) Original

Figura 5.46: Comparações qualitativas de um segmento da sequência PZ34 (Frame 44).

Capítulo 6

Conclusão

As tecnologias de imersão 3D tem crescido de forma bastante vertiginosa, uma vez que as produtoras de conteúdos multimídias têm buscado aumentar a experiência do usuário com seus produtos. Neste sentido, a 3DTV e FTV têm recebido atenções consideráveis. Como a forma de transmissão de tais conteúdos 3D ainda encontram-se em busca de otimização, os pesquisadores encontraram na síntese de vista uma saída bastante promissora. Um problema inerente no processo supracitado são os buracos existentes nas vistas virtuais geradas.

Este trabalho apresentou uma proposta de um algoritmo de *inpainting* digital para preenchimento de buracos em imagens virtuais resultantes do processo de síntese de vista utilizando o mapa de profundidade como uma ferramenta para auxiliar no processo de reconstrução.

Um modelo genérico foi desenvolvido, o qual destaca os pontos chaves que podem melhorar a forma do preenchimento dos buracos ou regiões de restauração. Apesar de ter o foco voltado para imagens virtuais resultantes do processo de síntese de vista, o modelo proposto pode ser utilizado para outros tipos de imagens que possuem um respectivo mapa de profundidade.

Entre as principais contribuições de trabalho podemos enfatizar uma publicação recentemente aceita para o Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT). O SBrT é o encontro de maior relevância acadêmica nacional da área de telecomunicações e processamento de sinais.

Os resultados tanto objetivo como qualitativos mostraram que o algoritmo proposto mostrou-se na maioria dos casos superior em relação à algumas abordagens da literatura. Contudo, este também pode ser alvo de melhorias que são descritas na Seção 6.1.

6.1 Trabalhos futuros

Os resultados apresentados no capítulo anterior demonstraram o funcionamento do algoritmo proposto para o preenchimento de buracos em síntese de vista com bons resultados se comparados a algoritmos representativos na literatura, porém com alguns pontos passíveis de melhoria. Algumas alterações de funcionalidade e novas técnicas podem ser incorporadas ao modelo, visando resolver problemas e/ou melhorar a qualidade dos resultados gerados. Estas alterações podem ser pequenas, mas interessantes, como utilizar o vetor gradiente para direcionar a busca do bloco homólogo no processo de *matching*.

A utilização da direção do gradiente para a busca do bloco ideal para o preenchimento em cada iteração pode auxiliar na diminuição do tempo computacional. A ideia consiste em restringir a região de busca ao longo de uma linha na qual reside o contorno local (que é ortogonal ao vetor gradiente). Isto fará com que a busca seja direcionada para aqueles blocos que tem mais chance de estar de acordo com a estrutura da região ao ser preenchida.

Outra ideia que pode contribuir para a melhoria do algoritmo consiste na inclusão do mapa de profundidade no cálculo direto da prioridade. O mapa de profundidade tem em seu conteúdo as bordas e descontinuidades mais visíveis, e neste há a possibilidade de se distinguir mais facilmente o *background* do *foreground*.

Um fator percebido durante a execução do algoritmo é que o Termo de Confiança dos pixels vai caindo drasticamente à medida que as iterações são incrementadas, o que faz com que o mesmo deixe de ter o mesmo valor de contribuição no cálculo de prioridade, em outras palavras, o Termo de Dados acaba por ter maior peso no processo de restauração. Um fator de balanceamento poderia ser utilizado no Termo de Confiança para manter seus valores mais estáveis, o que poderia melhorar o cálculo de prioridade e consequentemente o preenchimento do algoritmo.

Referências Bibliográficas

- AHN, I.; KIM, C. Depth-based disocclusion filling for virtual view synthesis. In: IEEE. *Internaciol Conference on Multimedia and Expo*. [S.l.], 2012. p. 109–114. 6, 7, 23, 28, 29, 36, 37, 38, 41, 43, 46, 50, 51, 56
- BERTALMIO, M. e. a. Image inpainting. In: *ACM Computer Graphics Proceedings (SIGGRAPH)*. [S.l.: s.n.], 2000. p. 417–424. 27, 35
- CASACA, W. et al. Combining anisotropic diffusion, transport equation and texture synthesis for inpainting textured images. *Pattern Recognition Letters*, v. 36, p. 36–45, 2014. Disponível em: <<http://dx.doi.org/10.1016/j.patrec.2013.08.023>>. 44, 46
- CASACA, W. et al. A user-friendly interactive image inpainting framework using laplacian coordinates. In: *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*. [S.l.: s.n.], 2015. p. 862–866. 28, 29
- CASACA, W. C. de O. *Restauração de Imagens Digitais com Textura utilizando Técnicas de Decomposição e Equações Diferenciais Parciais*. Dissertação (Mestrado) — Universidade Estadual Paulista, 2010. 28
- CASACA, W. C. de O. *Graph Laplacian for spectral clustering and seeded image segmentation*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2014. 28, 29
- CHENG, S. e. a. C. Improved novel view synthesis from depth image with large baseline. In: *International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2008. p. 1–4. 35
- CHOI, J. H.; HAHM, C. H. An exemplar-based image inpainting method with search region prior. In: *Consumer Electronics (GCCE), 2013 IEEE 2nd Global Conference on*. [S.l.: s.n.], 2013. p. 68–71. 28, 29
- CODING, V.; GROUP, A. R. *Multi-view 3D video acquisition, coding and rendering*. 2015. [Online; acessado dia 02 de Janeiro de 2015]. Disponível em: <<http://vca.ele.tue.nl/demos/mvc/>>. 4
- COSER, L. *Filtro de Difusão Anisotrópico Orientado por Evidência de Borda*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2009. 11
- COUTO, N. M. V. *Inpainting-based Image Coding: A Patch-driven Approach*. Dissertação (Mestrado) — Instituto Superior Técnico - Universidade Técnica de Lisboa, 2010. 26

- CRIMISI, P. P.; TOYAMA, K. Region filling and object removal by exemplar-based image inpainting. In: IEEE. *IEEE TRANSACTIONS ON IMAGE PROCESSING*. [S.l.], 2004. 7, 28, 32, 33, 35, 36, 39, 41, 43, 46, 50, 56, 58
- DARIBO, I.; PESQUET-POPESCU, B. Depth-aided image inpainting for novel view synthesis. In: IEEE. *Multimedia Signal Processing (MMSP)*. [S.l.], 2010. p. 167–170. 5, 7, 23, 28, 29, 35, 36, 43, 46, 50, 56, 58
- ELECTRONICS; INSTITUTE, T. R. *Sequences from ETRI Korea*. 2016. <ftp://203.253.128.142>. Acesso:06-06-2016. 49
- FEHN, C. Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. In: *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XI*. [S.l.: s.n.], 2004. p. 93–104. 5
- FILHO, O. M.; NETO, H. V. *Processamento Digital de Imagens*. [S.l.]: Brasport, 1999. ISBN 8574520098. 3, 4
- GAUTIER, J. e. a. Depth-based image completion for view synthesis. In: *The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*. [S.l.: s.n.], 2011. p. 1–4. 7, 23, 28, 29, 36, 46, 50
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2nd. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001. ISBN 0201180758. 14
- GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. [S.l.]: Pearson Education, 2011. ISBN 9788576054016. 3, 11
- IGNACIO, U. A. *Aplicação de Wavelets em Inpainting Digital*. Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos, 2007. 28
- JAWAS, N.; SUCIATI, N. Image inpainting using erosion and dilation operation. In: *International Journal of Advanced Science and Technology*. [S.l.: s.n.], 2013. p. 127–134. 38
- LEE, J.; LEE, D. K.; PARK, R. H. Robust exemplar-based inpainting algorithm using region segmentation. *IEEE Transactions on Consumer Electronics*, v. 58, n. 2, p. 553–561, May 2012. ISSN 0098-3063. 28, 29
- LUCAS, L. F. et al. Codificação eficiente de mapas de profundidade com base em predição e aproximação linear. In: *Simpósio Brasileiro de Telecomunicações - SBrT*. [S.l.: s.n.], 2012. 4
- MACCHIAVELLO, B.; DOREA, C. e. a. Low-saliency prior for disocclusion hole filling in dibr-synthesized images. In: *IEEE Transactions on Multimedia*. [S.l.: s.n.], 2014. p. 711–725. 28, 29, 39
- MEDIONI, G.; KANG, S. B. *Emerging Topics in Computer Vision*. [S.l.]: Prentice Hall, 2004. ISBN 0131900757. 21

- MICROSOFT. *MSR 3D Video Dataset*. 2016. <<http://research.microsoft.com/en-us/downloads/5e4675af-03f4-4b16-b3bc-a85c5bafb21d/>>. Acesso:06-06-2016. 49
- MÜLLER, K. et al. View synthesis for advanced 3d video systems. *EURASIP J. Image and Video Processing*, v. 2008, 2008. Disponível em: <<http://dblp.uni-trier.de/db/journals/ejivp/ejivp2008.html#MullerSDMKW08>>. 23
- MORVAN, Y. *Acquisition, Compression and Rendering of Depth and Texture for Multi-View Video*. Dissertação (Mestrado) — Eindhoven University of Technology, 2009. 21
- MUDDALA, S. M.; OLSSON, R.; SJOSTROM, M. Disocclusion handling using depth-based inpainting. In: IARIA. *The Fifth International Conferences on Advances in Multimedia*. [S.l.], 2013. p. 136–141. 4, 23, 28, 29
- OH, S. K.; HO, Y. Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video. In: *Proc. of the Picture Coding Symposium (PCS)*. [S.l.: s.n.], 2009. p. 1–4. 23, 35
- OKUTOMI, M.; KANADE, T. A multiple-baseline stereo. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [S.l.: s.n.], 1993. p. 353–363. 21
- OPPENHEIM, A. V.; WILLSKY, A. S. *Sinais e Sistemas*. [S.l.]: Pearson Education, 2010. ISBN 9788576055044. 3
- PARKER, J. R. *Algorithms for Image Processing and Computer Vision*. 1st. ed. New York, NY, USA: John Wiley & Sons, Inc., 1996. ISBN 0471140562. 11
- PEDRINI, H.; SCHWARTZ, W. R. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. [S.l.]: Editora Thomson Learning, 2007. 528 p. ISBN 978-85-221-0595-3. 29
- PENEDO, S. R. M. *Inpainting de imagens digitais por estimação de curvas de decaimento dos coeficientes wavelet via regularidade e espaços de Besov*. Dissertação (Mestrado) — Universidade de São Paulo, 2014. 28
- PERONA, P.; MALIK, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 12, p. 629–639, 1990. 17, 44
- RICHARDSON, I. *H.264/AVC and MPEG-4 Video Compression - Video Coding for Next-Generation Multimedia*. [S.l.]: John Wiley and Sons, 2003. 18, 19
- RODRIGUES, M. L. *Modelos de Retoque Digital Utilizando Equações Diferenciais Parciais*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, 2006. 27
- SAITO, Y. et al. Modelo estatístico para a determinação do número Ótimo de iterações do filtro de difusão anisotrópica aplicado à redução de ruído de imagens de ressonância magnética. *Revista de Informática Teórica e Aplicada - RITA*, 2010. 17
- SANGEETHA, K. Combined structure and texture image inpainting algorithm for natural scene image completion. In: *Journal of Information Engineering and Applications*. [S.l.: s.n.], 2011. p. 7–12. 28

- SCHONLIEB, C.-B. *Modern PDE Techniques for Image Inpainting*. Dissertação (Mestrado) — University of Cambridge, 2009. 28
- SUN, J. e. a. Stereo matching using belief propagation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [S.l.: s.n.], 2003. p. 787–800. 22
- TAUBER, Z. et al. Review and preview: Disocclusion by inpainting for image-based rendering. In: IEEE. *Systems, Man, and Cybernetics, Part C: Applications and Reviews*. [S.l.], 2007. p. 527–540. 6
- TRUCCO, E.; VERRI, A. *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN 0132611082. 10, 11
- UNIVERSITY, N. *Nagoya University Multi-view Sequences*. 2016. <<http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/>>. Acesso:06-06-2016. 49
- UNIVERSITY, P. *Poznan Multiview Video Test Sequences and Camera Parameters*. 2016. <<ftp://multimedia.edu.pl/3DV>>. Acesso:06-06-2016. 49
- WANG, Z.; LU, L.; BOVIK, A. C. Video quality assessment based on structural distortion measurement. *Sig. Proc.: Image Comm.*, v. 19, n. 2, p. 121–132, 2004. Disponível em: <<http://dblp.uni-trier.de/db/journals/spic/spic19.html#WangLB04>>. 30
- YAO, W. et al. Pde image inpainting with texture synthesis based on damaged region classification. In: *Advanced Computer Control (ICACC), 2010 2nd International Conference on*. [S.l.: s.n.], 2010. v. 5, p. 447–450. 28