

UNIVERSIDADE DO VALE DO RIO DOS SINOS  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO  
EM COMPUTAÇÃO APLICADA

**Aplicação de Wavelets em Inpainting Digital**

por

UBIRATÃ AZEVEDO IGNÁCIO

Dissertação submetida a avaliação  
como requisito parcial para a obtenção do grau de  
Mestre em Computação Aplicada

Prof Dr. Cláudio Jung

Orientador

São Leopoldo, Janeiro de 2007

# **Agradecimentos**

Agradeço aos meus pais pelo apoio e a sabedoria com que me educaram. Ao professor Cláudio Jung orientador e amigo que me guiou com objetividade e empenho exemplar. Em especial agradeço à minha esposa Ana pelo companherismo e carinho em todos os momentos.

# Resumo

Inpainting Digital é uma técnica recente que permite completar a falta de informação em imagens, seja por falha ou por remoção intencional de alguma área ou objeto. Uma das atribuições importantes do inpainting digital é de que deve ser capaz de alterar uma imagem, de forma que não seja simples perceber que esta alteração foi feita; caracteriza uma modificação **indetectável**.

Os métodos para determinar como esta falta de informação será preenchida variam desde a criação do primeiro modelo de inpainting digital. Contudo, sempre deve ser mantida uma coerência no preenchimento, que fará com que a região preenchida automaticamente aparente como parte da imagem verdadeira. As técnicas atuais tratam este preenchimento como uma propagação da estrutura da área que está ao redor da região a ser preenchida, trabalhando diretamente no domínio das cores, utilizando abordagens como Variação Total e Equações Diferenciais Parciais. Neste trabalho, é feito o uso de transformada *Wavelet* para a aplicação de inpainting digital, propondo um modelo para definir o preenchimento de regiões de inpainting com síntese de texturas, atuando no domínio de Wavelets.

**Palavras-chave:** Processamento de imagens, Inpainting digital, Restauração de imagens, Transformada Wavelet, Completamento de imagens, Síntese de texturas, Filtragem de texturas.

**TITLE: “WAVELET TRANSFORM IN DIGITAL INPAINTING”**

## Abstract

Digital Inpainting is a recent technique that allows the filling of missing information in images. One important attribute of a digital inpainting technique is the ability of altering an image in such a way that it is not simple for the human observer to detect the modification, characterizing an **undetectable modification**.

The strategies for filling missing parts vary since the first inpainting model, but one thing that remains is the fact that the filled area must be coherent with the original part of the image. Current techniques handle the filling as a structure propagation problem, working directly in the image color domain, and based on concepts like Total Variation or Partial Differential Equations. In this work, we present a digital inpainting model that works exclusively in Wavelet domain, filling the target area with a texture synthesis mechanism using the properties of the Wavelet Transform.

**Keywords:** Image processing, Digital inpainting, Image restoration, Wavelet Transform, Image completion. Texture Sintesis, Texture Filteringing.

# Conteúdo

<b>Resumo</b>	<b>3</b>
<b>Abstract</b>	<b>4</b>
<b>Lista de Abreviaturas</b>	<b>7</b>
<b>Lista de Figuras</b>	<b>8</b>
<b>Lista de Tabelas</b>	<b>11</b>
<b>1 Apresentação</b>	<b>12</b>
1.1 Introdução . . . . .	12
1.2 Motivação . . . . .	16
1.3 Objetivos . . . . .	17
<b>2 Estado da Arte do Inpainting Digital</b>	<b>18</b>
2.1 Introdução . . . . .	18
2.2 Desenvolvimento das Técnicas de Inpainting Digital . . . . .	18
<b>3 Transformada Wavelet em Inpainting Digital</b>	<b>32</b>
3.1 Introdução à Transformada Wavelet . . . . .	32
3.2 A Funcionalidade da Transformada Wavelet no Inpainting Digital . . . . .	35
<b>4 Modelo de Inpainting Digital Utilizando a Transformada Wavelet</b>	<b>37</b>
4.1 Formulação do Modelo de Inpainting com o Uso de Wavelets . . . . .	37
4.1.1 Processamento dos dados de entrada . . . . .	40
4.1.2 Prioridade dos blocos no preenchimento . . . . .	41
4.1.3 Preenchimento da região de inpainting . . . . .	44

4.2	Completando Pequenas Áreas de Inpainting . . . . .	45
4.2.1	Solução para imagens coloridas . . . . .	49
4.3	Modelo de Inpainting por Blocos de Coeficientes . . . . .	51
4.3.1	Determinação da ordem de preenchimento . . . . .	53
4.3.2	Preenchimento de blocos por síntese de texturas . . . . .	57
<b>5</b>	<b>Resultados Experimentais e Discussão</b>	<b>65</b>
5.1	Implementação do Modelo de Inpainting Digital no Domínio de Wavelets . . .	65
5.2	Resultados Preliminares do Inpainting em Pequenas Áreas . . . . .	66
5.3	Resultados Preliminares do Inpainting por Blocos . . . . .	69
5.4	Avaliação Quantitativa . . . . .	75
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>80</b>
6.1	Extensões Futuras para o Modelo . . . . .	81
<b>Bibliografia</b>		<b>84</b>

# Listas de Abreviaturas

**PI** Processamento de Imagens

**TW** Transformada Wavelet

**CG** Computação Gráfica

**TV** *Total Variation*

**POCS** *Projection onto Convex Sets*

**EDP** Equações Diferenciais Parciais

**PSNR** *Peak Signal-to-Noise Ratio*

**MSE** *Mean Squared Error*

# Listas de Figuras

1.1	Recuperação de uma fotografia digitalizada . . . . .	13
1.2	Remoção de um objeto por inpainting . . . . .	14
1.3	Restauração de uma fotografia original danificada . . . . .	15
1.4	Fronteira da região de inpainting . . . . .	16
1.5	Resultado do trabalho de Sun <i>et al</i> em [SUN 2005] . . . . .	17
2.1	Aplicação de Inpainting no trabalho de Sapiro e Bertalmio. . . . .	19
2.2	Direção de propagação normal à borda . . . . .	20
2.3	Remoção de um objeto real por inpainting . . . . .	21
2.4	Decomposição da imagem dividindo em textura e estrutura, ilustrado em [BER 2003].	23
2.5	Reconstrução da textura em uma fotografia digitalizada. . . . .	24
2.6	Propagação de blocos de textura visando a preservação da estrutura da imagem.	24
2.7	Remoção de uma larga área em uma imagem . . . . .	26
2.8	Âncoras e copia dos blocos . . . . .	27
2.9	Falhas resultantes da perda de pacotes transmitidos . . . . .	28
2.10	Resultado da reconstrução dos blocos em pacotes perdidos em [RAN 2002].	30
2.11	Exemplo apresentado no trabalho de Patwardhan <i>et al.</i> . . . . .	31
3.1	Decomposição multi-resolução de um sinal unidimensional . . . . .	34
3.2	Quadrantes na decomposição wavelet em um nível . . . . .	35
3.3	Decomposição da transformada wavelet em 2 níveis . . . . .	35
3.4	Detecção da estrutura pela transformada wavelet . . . . .	36
4.1	Coeficientes wavelet de detalhe em uma imagem de análise. . . . .	39
4.2	Destaque da descontinuidade capturada pelo detalhe horizontal. . . . .	39
4.3	Preenchimento de coeficientes e seu resultado . . . . .	40

4.4	Transformada wavelet de um mapa de inpainting . . . . .	41
4.5	Importância da ordem de preenchimento. . . . .	42
4.6	Mapa com variação dos valores de K . . . . .	43
4.7	Ilustração do cálculo da energia na fronteira de inpainting . . . . .	47
4.8	Resultados da técnica pixel-a-pixel . . . . .	48
4.9	Reconstrução pixel-a-pixel de partes com texturas . . . . .	49
4.10	Inpainting pixel-a-pixel em várias áreas da figura . . . . .	49
4.11	Detalhe de uma reconstrução não satisfatória . . . . .	50
4.12	Metodo por pixel aplicado a uma imagem colorida danificada . . . . .	51
4.13	Método por pixel aplicado à desoclusão de uma imagem colorida . . . . .	52
4.14	Vetores gradiente no ponto de análise. . . . .	55
4.15	Comparação dos métodos de priorização por wavelet e gradiente. . . . .	57
4.16	Região de avaliação dos blocos do trabalho de Tonietto <i>et al.</i> [TON 2005]. . . . .	58
4.17	Determinação dos blocos base nos coeficientes de wavelet. . . . .	60
4.18	Região de busca ao redor do bloco base. . . . .	60
4.19	Seleção do bloco de textura em $\Theta$ . . . . .	61
4.20	Resultados do preenchimento com tamanho de bloco fixo e variável . . . . .	64
5.1	Reprodução de áreas com texturas, pixel-a-pixel. . . . .	67
5.2	Inpainting pixe-à-pixel em imagens coloridas com texturas homogêneas . . . . .	67
5.3	Restauração de uma imagem danificada . . . . .	68
5.4	Remoção de parte de uma imagem . . . . .	68
5.5	Execução passo a passo do inpainting pixel-a-pixel . . . . .	69
5.6	Inpainting por blocos sintetizando texturas simples . . . . .	70
5.7	Inpainting por blocos sintetizando texturas mais complexas . . . . .	70
5.8	Desoclusão em imagens coloridas e monocromáticas . . . . .	72
5.9	Remoção de objetos uma imagem colorida, com largas regiões de inpainting. . . . .	73
5.10	Remoção de objetos em uma região rodeada por textura simples . . . . .	73
5.11	Remoção de texto em imagens digitais . . . . .	73
5.12	Correção em falhas na informação digital de uma fotografia. . . . .	74
5.13	Correção de falhas indesejadas em uma cena fotografada . . . . .	74
5.14	Correção das falhas físicas de uma fotografia posteriormente digitalizada . . . . .	74
5.15	Variação do tamanho do bloco preenchido e resultados comparativos. . . . .	75

5.16	Detalhe do preenchimento por blocos . . . . .	76
5.17	Imagens submetidas a diversas técnicas de inpainting digital. . . . .	78
5.18	Relação entre a imagem gerada e seu PSNR . . . . .	79

# **Lista de Tabelas**

5.1 Comparação do valor do PSNR aplicado sobre resultados de várias técnicas. . . 77

# Capítulo 1

## Apresentação

Este capítulo apresenta o *Inpainting Digital*, abordando os aspectos que motivaram a realização do trabalho, bem como seus objetivos. O assunto é conduzindo ao estado da arte dos trabalhos e pesquisas sobre inpainting, que fundamentarão os conceitos desenvolvidos nos capítulos seguintes, bem como o modelo de inpainting digital proposto.

### 1.1 Introdução

*Inpainting* consiste na arte de alterar uma imagem ou escultura sem que esta alteração seja facilmente percebida por um observador. Técnicas de restauração de imagens passaram a ser desenvolvidas intensamente no período da renascença, quando artistas reconstruiam manualmente falhas em pinturas feitas na era medieval. Com o advento da tecnologia de imagens digitais, tornou-se interessante poder recuperar de forma mais automática estes mesmos materiais artísticos, uma vez que digitalizados. Dessa forma, o problema de inpainting também passou a ter importância nas áreas de Computação Gráfica (CG) e Processamento de Imagens (PI).

Apesar do problema já ter sido abordado anteriormente, o conceito de inpainting digital foi introduzido mais formalmente por Bertalmio e Sapiro [BER 2000] em 2000, estabelecendo o termo na comunidade científica. Em termos matemáticos, pode-se definir inpainting como um problema de interpolação em duas dimensões: são conhecidos os valores de pixels em torno de uma região desconhecida  $\Omega$ , e se deseja obter/estimar estes valores no interior de  $\Omega$ . As pesquisas em PI têm criado mecanismos automáticos utilizados para reparar falhas em imagens, ou ainda, remover digitalmente objetos de uma cena (ver figuras 1.2(a) e 1.2(b)), dando origem ao

termo **inpainting digital**. Em geral, as técnicas de inpainting digital utilizam o conhecimento da fisiologia da visão humana, visando gerar imagens sem contornos quebrados (ou seja, priorizando a continuidade das bordas). As regiões a serem preenchidas com alguma técnica de inpainting também são chamadas de regiões a serem completadas, ou áreas de inpainting.

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 (através de câmeras digitais) podem ser reparadas (ver figura 1.1) e objetos podem ser removidos em cenas estáticas, como mostra a figura 1.2. Esta última aplicação do inpainting também é conhecida como **Desoclusão**. Além de desoclusão e correção de danos, inpainting pode ser aplicado em situações como remoção automática de textos em imagens e melhoramento de resolução, bem como ser aplicado para completar regiões corrompidas em seqüências de vídeo [BER 2001]. No entanto, este trabalho dará maior atenção ao problema de inpainting para imagens estáticas, seja sua aplicação para desoclusão, correção de danos ou remoção de texto.



Figura 1.1: Recuperação de uma fotografia digitalizada por uma das primeiras técnicas de inpainting digital (Bertalmio *et al* [BER 2000]).

Em uma abordagem direta, pode-se resumir o inpainting digital como o preenchimento de uma parte de uma imagem (também chamado completamento), 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.

A **área de inpainting** ou **região de inpainting** 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



(a)



(b)

Figura 1.2: Resultado (b) da remoção de um objeto de uma imagem (a), gerado com o modelo desenvolvido neste trabalho.

de informação, que pode ser a corrupção de uma imagem digital (figura 1.3), ou uma fotografia digitalizada que possua algum tipo de falha física, entre outras possibilidades. Também pode-se definir uma região de inpainting que cubra um objeto que se deseja remover da imagem original, como a remoção de objetos (ver as figuras em 1.2).



(a)



(b)

Figura 1.3: Restauração de uma fotografia original danificada. Em (a) a fotografia original danificada, e em (b) a imagem recuperada pelo modelo desenvolvido neste trabalho.

Outro item importante nas técnicas de inpainting é a **fronteira da região**. A fronteira da região de inpainting é o conjunto de pontos ao redor da região de inpainting, que contém informações importantes para que a técnica de preenchimento tenha sucesso. Em geral, as técnicas de inpainting analisam essa fronteira (figura 1.4) e identificam, de alguma forma, por onde deve começar e qual informação será usada no preenchimento.

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 (inclusive a quantidade de pixels que será preenchida em cada ciclo). Esses fatores determinam a propagação da estrutura da imagem na região de inpainting, que impactam



Figura 1.4: A fronteira da região de inpainting é uma pequena área no limite entre a imagem e a própria região de inpainting. Na figura acima, a região de inpainting está destacada em verde, e sua fronteira em vermelho.

na continuidade dos pontos preenchidos em relação aos vizinhos já existentes. De fato, esse é o ponto de discussão e diferenciação entre as abordagens existentes, e é o ponto central de qualquer técnica de inpainting. 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.

## 1.2 Motivação

Completar falhas em imagens difere do processo de simples remoção de ruído. Em geral, os pixels de uma imagem contaminada com ruído aditivo ou multiplicativo contêm também informações sobre o sinal original. Já o ruído do tipo sal-e-pimenta [JAI 89] destrói completamente a informação de pixels isolados, que podem ser reparados através de operações locais como o filtro da mediana [JAI 89]. Entretanto, problemas de inpainting em geral envolvem regiões consideravelmente grandes a serem completadas, e que não possuem informação alguma (ver seção 2.2). Assim, fez-se necessário a criação de novos algoritmos e técnicas para eficientemente completar imagens, seja para correção de falhas ou para desocclusão de áreas, com um mínimo de artefatos visuais possível. Métodos para inpainting digital foram criados especificamente para este propósito, demonstrando sucesso na capacidade de completar áreas em imagens, com resultados como o da figura 1.5.

O trabalho na área é recente, trazendo oportunidades adequadas para novas pesquisas.

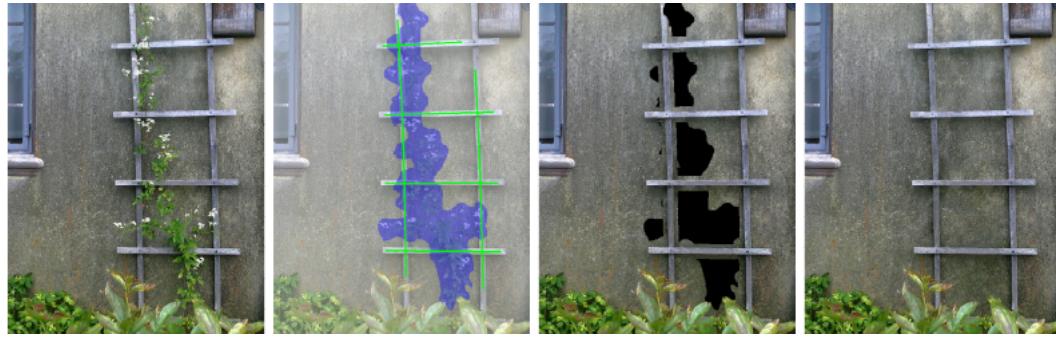


Figura 1.5: Resultado do trabalho de Sun *et al* em [SUN 2005]. A figura mostra as etapas da técnica apresentada pelos autores para desoclução de áreas.

Nos últimos anos, o uso de wavelets em inpainting passou a ser objeto de estudo, como pode ser visto nos trabalhos [PAT 2003, CHA 2001]. Estes trabalhos aplicam técnicas que se aproveitam da decomposição proporcionada pela transformada wavelet, combinando com aplicações de Equações Diferenciais Parciais e Variação Total [CHA 2001] no domínio de wavelets. De fato, a utilização de filtros passa-baixas e passa-altas inerente à transformada wavelet pode ser explorada em diversos aspectos do inpaitning digital (como a aplicação eficiente em largas áreas e imagens com texturas complexas), como será visto ao longo deste trabalho.

### 1.3 Objetivos

O principal objetivo deste trabalho é propor um modelo de inpainting digital que possa ser aplicado no domínio de wavelets, aproveitando as características inerentes à transformada wavelet para o benefício deste modelo. Embora a transformada wavelet seja amplamente empregada em diversas áreas do processamento de imagens e computação gráfica, sua utilização para auxiliar o inpainting ainda é bastante escassa, com poucas referências bibliográficas. Assim, é necessário investigar as vantagens e desvantagens da utilização de técnicas baseadas em wavelets em problemas de inpainting digital, focando nas características da transformada wavelet que possam ser exploradas no contexto do inpainting digital.

O restante deste trabalho está estruturado da seguinte forma: o capítulo 2 apresenta o estado-da-arte sobre inpainting. Uma breve revisão sobre a transformada wavelet é descrita no capítulo 3 e a técnica proposta é apresentada no capítulo 4. O capítulo 5 é dedicado a ilustrar os resultados experimentais obtidos e as conclusões são descritas no capítulo 6.

# **Capítulo 2**

## **Estado da Arte do Inpainting Digital**

### **2.1 Introdução**

Este capítulo introduz mais formalmente alguns dos conceitos necessários para o melhor entendimento do problema de inpainting digital, através da descrição e análise de diversos algoritmos propostos na literatura para abordar o problema, juntamente com alguns conceitos de processamento de imagens necessários para a compreensão do tema. Ênfase maior será dada a técnicas consideradas mais relevantes, que forneceram base para a criação do modelo descrito neste trabalho, ou que contenham conceitos que possam ser explorados no domínio de wavelets.

### **2.2 Desenvolvimento das Técnicas de Inpainting Digital**

O processo de correção de uma imagem ou remoção de algum objeto em uma fotografia era inicialmente realizado com uma forte dependência do usuário (como em [HIR 96]). Este deveria informar quais áreas da imagem seriam utilizadas para preencher as partes desejadas (com defeito ou removidas), e a eliminação da dependência do usuário podia ser aplicada em alguns casos extremamente específicos. O artigo de Kokaram *et al.* [KOK 95] utiliza técnicas de estimativa de movimento para interpolar falhas em frames de um filme. A idéia básica do trabalho é copiar os pixels de frames adjacentes para preencher as falhas nos frames de interesse. Esta técnica contudo, não é possível em imagens estáticas.

Um dos primeiros trabalhos na linha de desoclusão foi apresentado por Nitzberg e Mumford [NIT 93], que consiste em conectar junções-T em um mesmo nível de escala de cinza,



(a)

(b)

Figura 2.1: Aplicação de Inpainting no trabalho de Sapiro e Bertalmio [BER 2000]. O texto da fotografia (a) é removido por inpainting (b).

com uma curva de minimização elástica. Esta técnica se aplica a imagens simples com diversos objetos em uma mesma escala de cinza, sendo pouco aplicável em imagens complexas. Mais tarde, Masnou e Morel [MAS 98] estenderam a técnica com um algoritmo que completa áreas, unindo com curvas geodésicas os pontos na área desejada, e que tenham o mesmo valor de cinza. A técnica de Nitzberg contudo se aplicava a imagens com a mesma topologia (não poderia ter buracos), e podia criar diversos artefatos. Estes trabalhos moldaram as primeiras intenções de reconstrução automática de partes de imagens, que levaram ao atual conhecimento sobre inpainting.

Em 2000, Bertalmio *et al.* [BER 2000] apresentaram uma técnica bastante eficiente, que estende o trabalho de Nitzberg e elimina alguns dos problemas. Os autores apresentam um algoritmo para realizar inpainting digital em imagens estáticas, de forma semi-automática, baseado em equações diferenciais parciais comumente utilizadas em dinâmica de fluidos. O usuário seleciona a área a ser trabalhada (não é necessário informar com o que será preenchida), e o algoritmo recupera áreas com falhas, ou remove objetos da cena (figuras 1.2 e 2.1).

O artigo de Bertalmio *et al.* concretiza o primeiro mecanismo automático para inpainting digital. O algoritmo de preenchimento utilizado analisa os isofotos que se projetam à borda da região de inpainting, preenchendo o conteúdo da região de tal forma que estes isofotos mantenham-se completos dentro da região. A direção de preenchimento considera a incidência destes isofotos sobre a fronteira da região de inpainting, e é aplicável a regiões que possuam alguma estrutura, mas os autores observaram que não gera bons resultados quando aplicado em largas áreas. Como fundamento, os autores utilizaram as técnicas e o conhecimento criado por artistas que fazem o inpainting manual, traduzindo-os em modelos matemáticos. Um dos problemas levantados pelos autores considera a necessidade de preencher largas áreas uti-

lizando alguma técnica de replicação de textura. Este problema foi resolvido em alguns artigos publicados subsequentemente (descritos mais à frente nesta seção), e é uma das contribuições desenvolvidas neste trabalho.

Aplicável em imagens em escala de cinza ou coloridas, a técnica de Bertalmio propaga as linhas de isofotos, mantendo seu ângulo de incidência sobre o contorno da região de inpainting (ver figura 2.2). A garantia de conservação dos ângulos dos isofotos, em regiões relativamente pequenas, garante o prolongamento da estrutura da área danificada. A propagação da informação externa à região de inpainting é feita suavemente em direção ao interior desta, medindo a variação da informação (nas regiões com informação, fora da região de inpainting) ao longo da direção de propagação. A evolução suave da informação é determinada processando os pixels vizinhos do pixel de análise, e para a correta direção de propagação ser atingida, é feita a avaliação do vetor gradiente do isofoto em um determinado ponto.

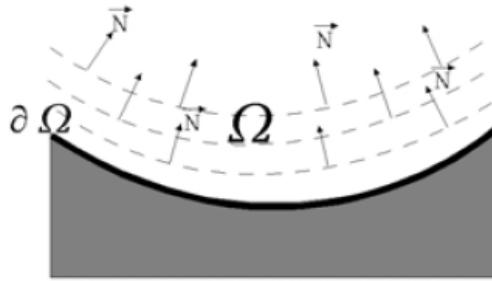


Figura 2.2: Direção de propagação em relação à normal da borda da região de inpainting, descrito em [BER 2000].

A equação abaixo fornece um esquema numérico genérico para a propagação de informação na região de inpainting, onde,  $I^n(i, j)$  é cada pixel da imagem na região de inpainting  $\Omega$ , no tempo  $n$ , com  $I_t^n(i, j)$  representando as atualizações dos pixels de saída em uma taxa  $\Delta t$ .

$$I^{n+1}(i, j) = I^n(i, j) + \Delta t I_t^n(i, j), \forall (i, j) \in \Omega \quad (2.1)$$

O modelo descrito pela equação (2.1) fundamenta o próprio conceito de inpainting digital, e é onde se sustentam a maioria das técnicas desenvolvidas posteriormente. É preciso observar os componentes da equação com atenção, uma vez que eles descrevem os pontos que devem ser resolvidos na aplicação de inpainting. Bertalmio *et al.* observaram que uma imagem  $I$  submetida a inpainting é melhorada até chegar a uma imagem  $I'$ , sendo processada em um número de passos  $n$ . Então, entende-se que a equação (2.1) processa, a cada passo, uma imagem

$I^n(i, j)$ , gerando uma versão melhorada desta em  $I^{n+1}(i, j)$ . Esta melhoria é dada por  $I_t^n(i, j)$ , que ocorre em uma taxa  $\Delta t$  a cada passo. Com isto, a forma de melhoramento em  $I_t^n(i, j)$  é um dos pontos fundamentais do mecanismo de inpainting digital, e foi um contribuição importante no trabalho de Bertalmio *et al.*. Observamos que também que a taxa  $\Delta t$  também deve ser observada com cuidado, e é discutida à frente neste trabalho.

O algoritmo de Bertalmio *et al.* também realiza uma difusão anisotrópica nas áreas onde há informação - intercalada com os passos de reconstrução - com o objetivo de manter a curvatura da linhas prolongadas sem perder a nitidez. A equação (2.2) define a difusão anisotrópica utilizada, onde,  $\Omega_e$  é a dilatação de  $\Omega$  em uma região de raio  $e$ ,  $k$  é a curvatura euclidiana dos isofotos da região de inpainting, e  $g_e(x, y)$  é uma função de suavização.

$$\frac{\partial I}{\partial t}(x, y, t) = g_e(x, y)k(x, y, t)|\nabla I(x, y, t)|, \forall(x, y) \in \Omega_e. \quad (2.2)$$

Em 2002, Chan *et al* apresentam um novo trabalho na linha de inpainting digital [CHA 2002] - baseado em variação total - seguindo um modelo para remoção de ruídos sustentados no modelo de Rudin [RUD 92, RUD 94]. A técnica aplica inpainting em segmentos locais em uma imagem, considerando propriedades específicas de simetria pertinentes a imagens não texturizadas.

Uma generalização das técnicas baseadas em Equações Diferenciais Parciais (EDP) é apresentada no trabalho de Tschumperlé e Deriche [TSC 2005], onde os autores apresentam um mecanismo poderoso para regularização de imagens (descritas por campos vetoriais). Seu algoritmo unificado permite realizar operações como remoção de objetos (figura 2.3), recuperação falhas em imagens e magnificação, com bons resultados.



Figura 2.3: Remoção de um objeto real utilizando a técnica de Tschumperlé e Deriche

Um problema comum às técnicas de inpainting se refere à preservação de regiões texturizadas. Vários autores propuseram alternativas para realizar o inpainting com preservação de texturas, combinando prolongamento de estrutura com síntese de texturas.

O trabalho de Hirani e Totsuka [HIR 96] combina análises em domínio de freqüência e espaço, para remover ruídos de uma determinada região, através de métodos de síntese de texturas. Wang *et al* [WAN 2005] apresentam um mecanismo também baseado em síntese de texturas [WOR 96, WIT 91] para remoção de objetos grandes em fotografias digitais. Tal trabalho formula uma teoria que consiste em preencher as regiões mais extensas (regiões do objeto a ser removido) através de síntese de texturas e interpolação. Um mecanismo automático determina a melhor forma de preencher as regiões de interesse, através de interpolação ponderada ou síntese de pequenas partes da imagem. Uma vez preenchida a região, eventuais artefatos na imagem são removidos na etapa final.

Um dos principais problemas da técnica proposta por Bertalmio em 2000 [BER 2000] foi a inabilidade do algoritmo em reconstruir grandes áreas de texturas, em grandes regiões de inpainting. Assim, Bertalmio, Sapiro e Osher [BER 2003] propuseram um mecanismo que também combina síntese de texturas com inpainting digital para reconstruir regiões em imagens. A técnica abordada decompõe a imagem em duas funções: uma representa a estrutura nos limites da região de interesse e outra captura a textura e eventuais ruídos (ver figura 2.4) [MEY 2001], estendendo o trabalho realizado em [VES 2003]. Um algoritmo de inpainting reconstrói a informação da estrutura na borda da região afetada, e as texturas no seu interior são sintetizadas. Ambas combinadas preenchem toda a área de interesse. Uma boa qualidade visual dos resultados depende fortemente da combinação das três partes (decomposição de imagens, síntese de texturas e inpainting) que formam a técnica apresentada.

A decomposição da imagem é feita utilizando procedimentos de minimização da variação total [MEY 2001, RUD 92] - muito aplicados à remoção de ruídos em imagens - e espaço de funções oscilantes [VES 2003] aplicado à construção de texturas e também redução de ruído. A saída obtida durante esta etapa é um conjunto de imagens inicialmente desejado, onde a informação principal se divide em informação de textura e de estrutura. As falhas relativas à estrutura da imagem origem são preenchidos com técnicas de inpainting, enquanto a textura é sintetizada utilizando um procedimento apresentado por Efros *et al.* [EFR 99], um método não paramétrico baseado em cadeias de Markov.

Técnicas de síntese de texturas baseadas em exemplos são, em geral, rápidas e eficientes em replicar texturas consistentes. Contudo, essas técnicas tem dificuldade em replicar texturas de fotografias de cenários em mundo real (ver figura 2.5), que geralmente necessitam da combinação de reprodução de estruturas lineares e composição de texturas [ZAL 2002]. Outra



Figura 2.4: Decomposição da imagem dividindo em textura e estrutura, ilustrado em [BER 2003].

técnica que entende o inpainting clássico, permitindo reparar largas áreas através de reprodução das texturas, foi desenvolvida por Criminisi *et al.* [CRI 2004]. O algoritmo proposto utiliza técnicas de síntese de texturas para reconstruir não apenas a estrutura das áreas danificadas, mas também como preencher seu conteúdo com a informação mais adequada, focando-se em cenas do mundo real.



Figura 2.5: Reconstrução da textura em uma fotografia digitalizada.

Geralmente, as bordas entre as regiões são um produto complexo da influência de texturas diferentes, que, em contraste com a natureza de texturas puras, podem ser consideradas como estrutura em uma imagem. Os trabalhos anteriores que direcionam a síntese de texturas em imagens com estas características geralmente inserem artefatos como borramento, resultado da aplicação do processo de difusão para preencher largas áreas. O algoritmo de Criminisi também considera a divisão da reconstrução da região de inpainting em duas fases: estrutura e textura, que apesar de baseado em [BER 2003], não divide a imagem original em duas, mas sim divide a tarefa de reconstrução.

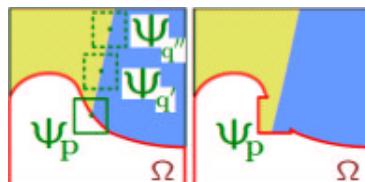


Figura 2.6: Propagação de blocos de textura visando a preservação da estrutura da imagem.

Uma das observações considerada importante por Criminisi é que uma síntese de textura que utiliza uma técnica simples de cópia - onde aplica-se a comparação dos erros quadráticos dos blocos a serem copiados - é suficiente para o inpainting de largas áreas texturizadas, desde que a ordem de preenchimento seja correta. Os autores deste artigo demonstram que é possível recuperar a informação da textura e a estrutura em partes de imagens, baseando-se apenas em reprodução de texturas por exemplo. O algoritmo utiliza o vetor gradiente na fronteira da região

de inpainting para a análise fundamental da informação nos pixels da imagem. A propagação dos pixels é uma questão de determinar qual pedaço da imagem original é mais adequado para a região falha, preservando a orientação dos gradientes calculados. Inicialmente, cada bloco a ser propagado necessita ter sua prioridade avaliada, que considera os vizinhos de cada bloco, sua direção de propagação e quantidade de contribuição (informação de cor). Esta prioridade é calculada de acordo com as equações (2.3) e (2.4), que definem a prioridade  $P(p)$ , o termo de confiança  $C(p)$  e o termo de dados  $D(p)$  para um certo pixel  $p$ :

$$P(p) = C(p)D(p), \quad (2.3)$$

$$C(p) = \frac{\sum_{q \in \Psi \cap (\mathcal{I} - \Omega)} C(q)}{|\Psi_p|}, \quad D(p) = \frac{|\nabla I_p^\perp \cdot \vec{n}_p|}{\alpha}, \quad (2.4)$$

onde  $\mathcal{I}$  é o domínio da imagem,  $\Omega$  é a região de inpainting,  $\Psi_p$  é um bloco de pixels centrado em  $p$  tendo com área  $|\Psi_p|$ ,  $\alpha$  é um fator de normalização (para uma imagem em tons de cinza, é recomendável usar  $\alpha = 255$ ),  $\vec{n}_p$  é um vetor unitário normal à fronteira  $\partial\Omega$  no ponto  $p$ , e  $\perp$  denota o operador ortogonal.

O termo  $C(p)$  representa a confiabilidade da informação no pixel  $p$ , que tende a diminuir à medida que se aproxima do centro da região de inpainting (já que novos blocos inseridos não fazem parte da informação original da imagem, sendo assim menos confiáveis). No processo de inicialização, define-se  $C(p) = 0$  quando  $p \in \Omega$ , e  $C(p) = 1$  quando  $p \in \mathcal{I} - \Omega$ . A propagação é então feita utilizando os blocos com maiores prioridades, em direção ao centro da região de inpainting, de forma guiada à estrutura da imagem (ver figura 2.6).

O termo  $D(p)$  indica a consistência entre a orientação local da região válida na fronteira com a geometria da fronteira. Assim, quanto maior esta consistência, mais o pixel na fronteira “penetra” na região de inpainting, significando que é um ponto importante na estrutura da imagem, e deve ser o primeiro a ser preenchido, de forma a preservar esta estrutura.

Para evitar problemas de borramento possíveis de serem causados por simples difusão de pixels, a propagação da textura da imagem é feita por amostragem direta da região de origem, onde é feita uma busca pelo bloco original que melhor se adequa à nova área a ser preenchida, similar ao que é feito em [EFR 99]. Matematicamente, tal bloco é obtido através de:

$$\Psi_{\hat{q}} = \underset{\Psi_q \in \Phi}{\operatorname{argmin}} d(\Psi_{\hat{p}}, \Psi_q), \quad (2.5)$$



Figura 2.7: Remoção de um objeto em uma imagem, mostrando as etapas onde se faz necessário a interação do usuário

onde  $d(\Psi_{\hat{a}}, \Psi_b)$  representa a distância entre os blocos  $\Psi_{\hat{a}}$  e  $\Psi_b$ , definida como a soma dos quadrados das diferenças (SSD) dos pixels já preenchidos em ambos blocos. Isto é suficiente para reproduzir tanto a estrutura quanto a estrutura das áreas processadas, bloco à bloco. À medida que os blocos preenchidos aproximam-se do centro da região de inpainting, estes passam ser considerados como cada vez menos significativos (de menor contribuição) para subsequentes propagações.

Novas técnicas propostas passam a utilizar uma informação fornecida pelo usuário, que serve de guia para o inpainting de estruturas complexas. Em 2005, Sun *et al.* [SUN 2005] propõem uma nova abordagem para desocclusão de áreas em uma imagem (onde os autores utilizam o termo *Image completion*). Com uma técnica denominada **Propagação de Estrutura**, o sistema apresentado permite que o usuário selecione áreas em uma imagem onde deseja-se remover algum objeto à frente de outro, e em seguida delineie a estrutura da imagem que deve ser reconstruída. O sistema sintetiza as partes necessárias para preencher a imagem, seguindo as linhas feitas pelo usuário. A abordagem procura pelas áreas mais adequadas para usar como preenchimento, utilizando algoritmos de programação dinâmica ou propagação de crédito (ver figura 2.7).

Se apenas uma linha de estrutura é definida pelo usuário, a propagação é feita através de programação dinâmica. Em se tratando de múltiplas linhas que se interceptam, um algoritmo de propagação por crédito é utilizado para selecionar os blocos mais adequados como fonte de informação para preenchimento. Os autores direcionaram o trabalho ao tópico de complementamento de imagens e desocclusão, aplicando isto ao preenchimento de largas áreas em imagens que possuem estruturas complexas e salientes.

A técnica de Sun *et al.* é amparada na propagação da estrutura previamente delineada manualmente. Sobre a curva traçada são definidos esparsamente alguns pontos, chamados de âncoras (ver figura 2.8). Os blocos de texturas que reconstroem a estrutura são centrados nestas

âncoras, que formam um grafo unidimensional  $G = (V, E)$ , onde  $V$  é o conjunto de vértices (âncoras) e  $E$  o conjunto de arestas que conectam estes vértices.

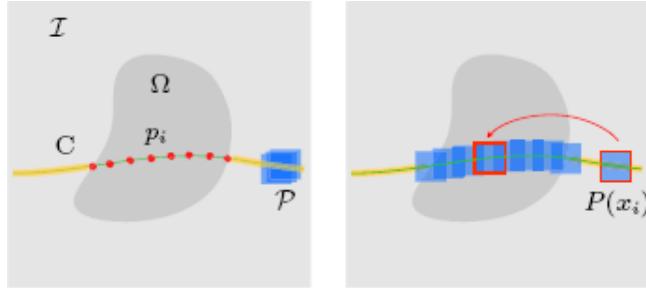


Figura 2.8: Linhas guias (à esquerda) e os pontos âncoras servindo de referência para a replicação dos blocos da estrutura da imagem (à direita).

Os pontos âncoras são amostrados em intervalos com metade do tamanho dos blocos replicados, e o conjunto  $P$  é dito o conjunto contendo todos estes blocos, centrados às âncoras. O algoritmo determina o bloco  $P(x_i)$  que será copiado para um próximo ponto  $p_i$  (figura 2.8). Uma vez definida a região de propagação das estruturas, escolher quais blocos deverão ser colados, e a sobreposição que terão sobre os existentes é um problema de minimização de energia que define a reconstrução da estrutura faltante da forma mais suave possível. Este é um problema caracteristicamente resolvido com técnicas de programação dinâmica. Os autores aplicaram estes conceitos, modelando a propagação dos blocos de estrutura como um problema de encontrar o menor caminho entre os nodos de um grafo [BEL 57] (grafo  $G$  citado anteriormente). O algoritmo de propagação por crédito é uma técnica de inferência probabilística definida por Pearl [PEA 88], e é utilizado quando existem diversas estruturas à serem recuperadas, e o resultado de uma interfere na outra. A natureza das soluções empregadas para a reconstrução das estruturas faz com que a complexidade dos algoritmos seja um ponto importante a ser considerado. Avaliando os resultados, os autores conseguiram um ótimo desempenho na maioria dos casos. Contudo, em algumas situações, resolver os problemas por programação dinâmica poderia levar horas, enquanto a aplicação de inferência probabilística levaria alguns segundos. Além disso, deve-se salientar que a técnica de Sun é bastante dependente da interação humana, para guiar o processo de recuperação da estrutura.

Recentemente, Barcelos e Batista [BAR 2006] apresentaram um método de restauração de imagens e desocclusão que também realiza remoção de ruído. O trabalho proposto consiste em propagar a informação na fronteira da região de inpainting, aplicando filtros que simultane-

amente suavizam a imagem, reduzindo o ruído. A tarefa de realizar inpainting em uma imagem aplica diferentes algoritmos, dependendo da informação da imagem em que está executando. Em uma etapa, apenas a restauração por inpainting é realizada, e em outra, uma suavização baseada em fluxo médio de curvatura, é aplicada.

Komodakis e Tziritas [KOM 2006] criam em 2006 um algoritmo que visa ser diferente dos métodos tradicionais, utilizando um modelo de otimização global discreta, baseada em cadeias de Markov, definindo funções energia apropriadas para o inpainting digital. O processo de inpainting é realizado por cópia de blocos de pixels que são dispostos através dos nós da cadeia. Um mecanismo de otimização, chamado pelos autores de *Priority-BP*, permite que o algoritmo seja computacionalmente viável, reduzindo o número de nós através de um processo de *poda dinâmica*, descrita no trabalho. Os resultados apresentados demonstram a aplicação do método a diversos tipos de imagens, com resultados interessantes. Entretanto, existem problemas ainda não resolvidos, inerentes à possibilidade de existência de um largo número de nós na cadeia de Markov.

A utilização de wavelets para preenchimento de falhas em imagens é o tema do trabalho de Rane *et al.* [RAN 2002], onde os autores apresentam uma técnica para reconstruir imagens (no formato de compressão JPEG) com blocos de informação perdidos por falhas em pacotes transmitidos em redes sem fio comutadas. A técnica descrita propõe um protocolo onde, em caso de pacotes perdidos, a retransmissão não faz-se necessária. A reconstrução da imagem é feita pelo software que a recebe. Assim, as falhas ocorridas nas imagens tem a área equivalente ao tamanho do bloco transmitido que foi perdido, como pode ser visto na figura 2.9.



Figura 2.9: Falhas resultantes da perda de pacotes transmitidos em redes sem fio. Imagem do trabalho de Rane *et al* [RAN 2002]

O mecanismo proposto no trabalho de Rane é vantajoso no que diz respeito à transmissão via redes sem fio, permitindo que uma boa taxa de transferência seja mantida. Uma vez que a transmissão esteja encerrada, o receptor identifica os pacotes perdidos, que refletem em blocos de pixels na imagem. Após identificados estes blocos, faz-se necessário então identificar o tipo do bloco com falha, e em seguida, efetivamente preenchê-lo.

A identificação dos tipos de blocos perdidos consiste em classificar este em um bloco que estava na fronteira do bloco afetado, ou não. Este primeiro passo decompõe a imagem com falhas, usando a transformada wavelet (os autores utilizaram blocos de tamanho fixo 8x8). Um limiar  $T_1$  foi determinado empiricamente, e este é então comparado com os coeficientes em uma área ao redor do bloco com informação perdida. Quatro testes específicos são realizados para verificar se a borda é horizontal ou vertical. Se esta comparação falha, o algoritmo então determina que o bloco perdido não faz parte de uma borda limite na imagem. A reconstrução do bloco perdido depende desta classificação, e permite que o algoritmo execute mais rápido e com melhores resultados.

Para reconstruir os blocos que foram classificados como blocos em uma fronteira, é considerado o tipo de detalhe sendo reconstruído (apenas horizontal ou vertical), e quais blocos vizinhos devem ser utilizados para propagar a informação (novamente, em uma vizinhança de 8 blocos). Desta forma, quando uma borda horizontal é detectada, a propagação é feita apenas no detalhe horizontal. Isto também vale para a propagação das bordas verticais. Esta propagação é feita avançando os blocos vizinho, tentando minimizar os erros quadrados das energias dos coeficientes. O resultado pode ser visto na figura 2.10.

A reconstrução dos blocos classificados como não estando em fronteiras consiste em simplesmente interpolar a vizinhança de todos os detalhes do bloco perdido (também considerando a minimização dos erros quadrados). Contudo, caso um dos blocos vizinhos for classificado como borda, este não entra na interpolação, evitando artefatos nas bordas. É importante salientar que a abordagem [RAN 2002], embora seja uma técnica de inpainting, é restrita ao domínio de blocos  $8 \times 8$ , em geral esparsos. De uma certa forma, essa técnica pode ser considerada uma remoção ruído do tipo sal-e-pimenta por blocos.

Chan *et al.* em 2001 [CHA 2001] apresentam os conceitos de inpainting com uma abordagem considerando variação total (*Total Variation Inpainting*), utilizando um procedimento para recuperação de coeficientes wavelets, assumindo que algumas informações da decomposição wavelet tenham sido perdidas. Os autores apresentam um algoritmo iterativo com solução nu-

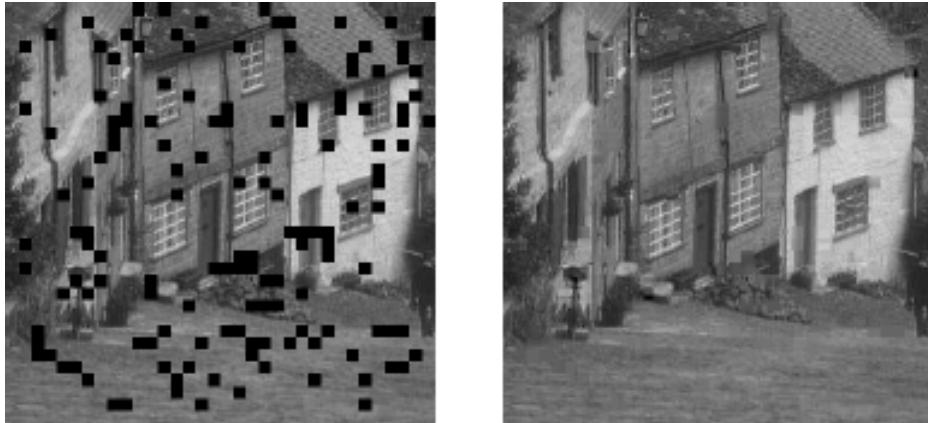


Figura 2.10: Resultado da reconstrução dos blocos em pacotes perdidos em [RAN 2002].

mérica para a extração dos coeficientes wavelet perdidos, analisando as curvas de nível em uma imagem. A técnica procura garantir o princípio da conectividade para realizar o preenchimento. Embora esta técnica seja aplicada no domínio wavelet, ela não considera uma região de inpainting com falta de dados na imagem sendo analisada: ela assume que alguns coeficientes wavelet podem ser perdidos em alguns pixels da imagens, enquanto que outros podem permanecer intactos (em situações como transmissão de dados em uma rede, por exemplo). Assim, o problema abordado não é completar pixels ou regiões sem informação, mas sim reconstruir alguns dos coeficientes wavelet perdidos.

Outro trabalho que explora a decomposição wavelet em inpainting digital é descrito em [PAT 2003], onde os autores propõem um algoritmo baseado na projeção em conjuntos convexos (POCS), que propaga a informação às áreas faltantes baseado na distribuição de freqüências possíveis de se obter com a transformada wavelet. O algoritmo estabelece restrições de freqüência no bloco em análise, considerando a energia da vizinhança no domínio de wavelets. A resolução da imagem determina o número de decomposições feitas (os autores utilizam a função wavelet Daubechies). A aproximação do último nível é mantida inalterada, de forma a preservar a informação de brilho da imagem. Para cada coeficiente wavelet pertencente a região de inpainting, os valores máximos e mínimos da vizinhança deste são encontrados, e o coeficiente a ser preenchido é restrinido entre estes valores, representado como  $C_w$ . Este novo valor é projetado no domínio da imagem, reconstruindo assim os pixels afetados. O processo de projeção e aplicação das restrições dos coeficientes pode ser visto na equação (2.6),

$$I_{n+1} = C_i P_{wi} C_w P_{iw} (I_n), \quad (2.6)$$

onde  $I_{n+1}$  é a janela de inpainting na iteração  $n + 1$ ,  $I_n$  é a janela de reparo em trabalho,  $P_{iw}$  representa a projeção do domínio da imagem para o domínio de wavelets,  $C_w$  é o coeficiente restringido,  $P_{wi}$  é a projeção do domínio de wavelet para o domínio da imagem e  $C_i$  representa as restrições no domínio de wavelet, normalizadas entre 0 e 1. O algoritmo itera, atualizando os pixels conhecidos na vizinhança a região de análise para o seu valor original, os novos pixels gerados na região de inpainting são então restringidos (entre 0 e 1). Esta técnica não explora explicitamente o caráter direcional da transformada wavelet, e os autores apresentaram apenas preenchimento de regiões com simples e poucas texturas (tal como a imagem mostrada na figura 2.11).



Figura 2.11: Exemplo apresentado no trabalho de Patwardhan *et al.*

A revisão bibliográfica apresentada neste capítulo indica uma grande variedade de abordagens para realizar o inpainting digital. Em particular, apenas algumas dessas abordagens utilizam a transformada wavelet, e mesmo assim, não parecem explorar todo potencial da transformada. A seguir, uma breve revisão sobre wavelets será vista, com foco nas propriedades interessantes no contexto de inpainting.

# Capítulo 3

## Transformada Wavelet em Inpainting Digital

Este capítulo apresenta os conceitos básicos de wavelets, focando nas características adequadas para a realização do inpainting. Inicialmente, uma visão geral à transformada e suas aplicações em processamento de imagens serão apresentados, seguindo com a descrição geral das importantes funcionalidades da transformada wavelet, como importantes recursos para completamento e desocclusão no inpainting digital.

### 3.1 Introdução à Transformada Wavelet

A **Transformada Wavelet** (TW) é um recurso matemático originalmente elaborado para análise de sinais. Herdando muito dos conhecimentos deixados no século 19 por Joseph Fourier [MEY 93, HAY 2002] para análise de freqüências, pesquisadores partiram da análise baseada em freqüência para baseada em escala, como propõe a teoria de wavelets. A verificação de sinais utilizando variação de escala provou ser menos sensível a ruídos, permitindo análises ajustáveis ao domínio do problema e consequentemente mais precisas.

A definição formal de wavelets se dispersa, uma vez que o trabalho iniciado na década de 1930 foi se desenvolvendo ao longo dos anos em pesquisas independentes de cientistas nas áreas de física quântica, engenharia elétrica, matemática, entre outras. Ainda, as aplicações mais efetivas utilizando wavelets como ferramenta surgiram apenas cerca de 15 anos atrás, tornando-a importante instrumento na área de processamento de imagens (uma imagem também pode ser observada como um sinal bi-dimensional).

Wavelets são funções matemáticas utilizadas na decomposição de sinais em diferentes faixas de freqüências. As wavelets são conhecidas por permitirem não apenas a determinação das componentes de freqüência, mas também a localização espacial onde estas componentes aparecem. O conceito fundamental inserido nas funções wavelets é a noção de escala. Quanto maior a escala de análise, menos detalhes se captura no sinal (ou imagem), consequentemente escalas mais finas identificam uma maior quantidade de detalhes.

A análise através de wavelets utiliza-se de uma função conhecida como *função mãe*, geralmente denotada por  $\psi(x)$ . As altas freqüências do sinal podem ser analisadas através de uma versão contraída da wavelet-mãe ( $\psi(sx)$ , para  $s > 1$ ), enquanto que as baixas freqüências podem ser analisados através de uma versão dilatada da mesma wavelet-mãe ( $\psi(sx)$ , para  $0 < s < 1$ ). Considerando-se translações espaciais das contrações e dilatações da wavelet-mãe, um sinal  $f(x)$  pode ser representado através de uma expansão em wavelets, ou seja, uma combinação linear das dilatações e contrações da wavelet-mãe. A transformada wavelet pode ser aplicada em domínio contínuo, ou discreto. Neste trabalho, apenas o domínio discreto é de interesse.

Diversas funções mãe podem ser construídas, cada uma mais apropriada a algum domínio de análise. A wavelet de Haar é provavelmente a wavelet mais simples e com menor suporte, e consiste em um degrau unitário no intervalo  $[0, 1]$ . Entretanto, diversas outras funções wavelets foram definidas, entre elas encontram-se as wavelets de Daubechies, Coiflet e Meyer [HAY 2002]. Contudo, as funções devem obedecer um determinado comportamento, como descreve a teoria de Littlewood-Paley-Stein [MEY 93], onde uma função wavelet  $\psi(x) \in L^2(\mathbb{R})$  tem sua transformada de Fourier  $\hat{\psi}(\xi)$  que satisfaz a condição

$$\sum_{j=-\infty}^{\infty} |\hat{\psi}(2^{-j}\xi)|^2 = 1 \quad (3.1)$$

para quase todo o valor de  $\xi$ .

Assim como há diferentes definições de wavelet, há diferentes maneiras de denotar a transformada wavelet. Provavelmente a abordagem mais utilizada em processamento de sinais seja a proposta de Mallat [MAL 89], na qual um algoritmo piramidal baseado em filtros passa-baixas e passa-altas é utilizado para a decomposição de sinais de domínio discreto em múltiplas resoluções. Os filtros passa-baixas (associados a uma função de escala) aproximam um sinal em tempo discreto  $f[n]$  (no caso unidimensional) nas escalas diádicas  $2^j$ , para  $j = 1, 2, 3, \dots, J$ , e os filtros passa-altas (associados a uma wavelet) produzem os detalhes perdidos quando a

escala passa de  $2^j$  para  $2^{j+1}$ . Os coeficientes associados ao filtro passa-baixas são denotados por  $S_{2^j}f[n]$ , e os coeficientes de detalhe (associados ao filtro passa-altas) são denotados por  $W_{2^j}f[n]$ . A decomposição de um sinal  $f[n]$  através do algoritmo piramidal de Mallat é ilustrado na figura 3.1. Nesse diagrama, os filtros  $\tilde{H}$  e  $\tilde{G}$  são os filtros passa-baixas e passa-altas, respectivamente. O operador  $\downarrow 2$  denota dizimação, ou seja, uma reamostragem do sinal em que uma amostra da cada duas é mantida (resultando em um sinal com a metade do tamanho). O sinal original pode ser reconstruído usando filtros de reconstrução  $H$  e  $G$ , em um algoritmo piramidal similar ao da decomposição.

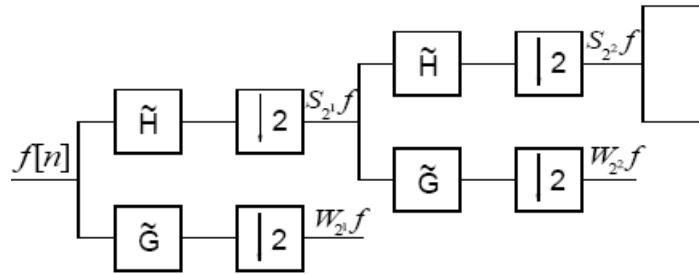


Figura 3.1: Decomposição multi-resolução de um sinal unidimensional

O modelo proposto por Mallat pode ser estendido para duas dimensões, permitindo então sua aplicação na análise de imagens digitais. Nesse caso, a decomposição produz, a cada escala, uma imagem suavizada (aproximação capturada pelo filtro passa-baixas) e três imagens de detalhe (horizontal, vertical e diagonal, capturados no filtro passa-altas), conforme indicado na figura 3.2. Essa imagem suavizada (quadrante superior esquerdo na figura 3.2) pode ser decomposta novamente, gerando outra imagem suavizada (de tamanho menor) e outras três imagens de detalhe, e assim sucessivamente. Um exemplo de decomposição em dois níveis é ilustrado na figura 3.3. Como pode-se perceber, a imagem aproximação é uma versão suavizada (e reduzida) da imagem original, enquanto que as imagens de detalhe fornecem os componentes horizontais, verticais e diagonais das bordas da imagem.

A decomposição da transformada wavelet em seus coeficientes pode ser revertida, restaurando o sinal original. Esta operação é chamada de **transformada wavelet inversa**.

Outra versão da transformada wavelet é a transformada **não dizimada**, na qual os operadores  $\downarrow 2$  e  $\uparrow 2$  não são utilizados [MAL 92]. Nesse caso, mantém-se o mesmo número de amostras em todos os níveis de decomposição (as resoluções se mantêm), o que não ocorre no tipo de transformada citado anteriormente (cada nível mantém apenas metade das amostras do nível anterior). No caso de imagens, a transformada não-dizimada produz três conjuntos de

Aproximação	$H_1$
$V_1$	$D_1$

Figura 3.2: Quadrantes na decomposição wavelet em um nível

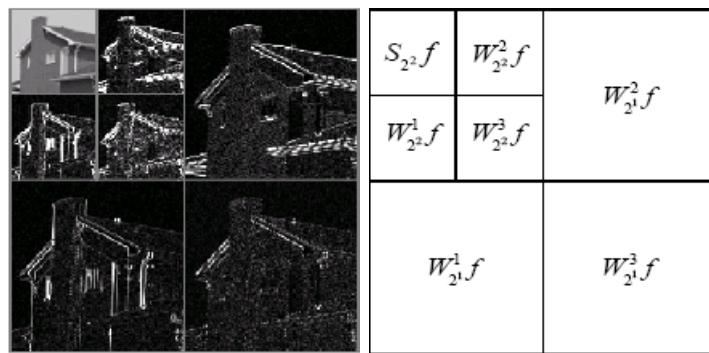


Figura 3.3: Decomposição da transformada wavelet em 2 níveis. As imagens resultantes (à esquerda) e a disposição dos coeficientes à direita.

saída: uma imagem suavizada, uma imagem de detalhes verticais e outra imagem de detalhes horizontais. Assim, a imagem de detalhes diagonal (que aparece na versão dizimada) não é necessária. Neste trabalho, será utilizada apenas a transformada wavelet *dizimada*. A dizimação reduz o tamanho da imagem, e, consequentemente da região de inpainting, diminuindo o custo computacional.

### 3.2 A Funcionalidade da Transformada Wavelet no Inpainting Digital

Quando visualizamos uma imagem que possua algum tipo de falha em alguma região, nosso sistema visual é capaz de “preencher” essa área, de acordo com o que imaginamos ser o mais adequado. Para tal, utilizamos como base nossa experiência e conhecimento sobre o que está sendo visualizado. Este processo é objeto de estudo da neuropsicologia [NOE 98], e é através desta habilidade que os artistas e restauradores realizam correções em pinturas e outros objetos de arte. Em particular, uma característica fundamental que tentamos preservar é a continuidade de estruturas, que também é desejada no desenvolvimento de inpainting digital.

A estrutura de uma região de uma imagem pode ser vista como um conjunto de blocos de textura que se propaga, com alguma continuidade. Este conjunto de texturas possui características que se destacam em sua região, ou seja, a sua coloração é de alguma forma diferente das regiões ao redor, sendo interpretado pelo sistema visual humano como uma estrutura (o encontro de diferentes texturas forma uma estrutura). Assim, esta diferença em sua área limite, forma uma **borda**. A preservação das estruturas no inpainting digital é talvez a etapa mais importante e mais complicada, e juntamente com a síntese de texturas, denotam os objetivos desta proposta.

Um das características da decomposição com transformada wavelet é que os coeficientes nas direções resultantes tendem a destacar a estrutura das imagens (ver figura 3.4), capturando a “forma” existente no encontro das texturas, e servindo como guia para a projeção dos coeficientes existentes em direção às áreas faltantes na região de inpainting. Em outras palavras, a borda das regiões da estrutura - descritas anteriormente - são capturadas inherentemente nos coeficientes de detalhe da transformada wavelet. Esta propriedade da TW é fortemente explorada neste trabalho.

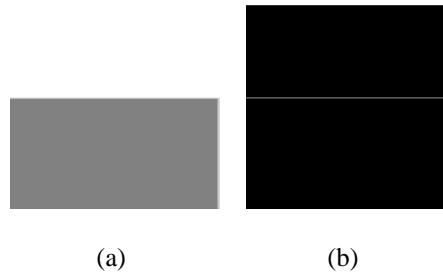


Figura 3.4: Figura (a) imagem original com uma borda horizontal, e na figura (b) a imagem do coeficiente wavelet com a informação do detalhe horizontal, com um nível na decomposição usando wavelets haar.

# Capítulo 4

## Modelo de Inpainting Digital Utilizando a Transformada Wavelet

O modelo de inpainting digital utilizando os coeficientes wavelet é apresentado nas seções seguintes. Inicialmente, o funcionamento geral do modelo é descrito, apresentando a forma com que o processamento de imagens no domínio wavelet é realizado. Em seguida, as construções feitas sobre este mecanismo criam variações no modelo, que aqui é aplicado com duas abordagens: o processamento de pequenas áreas de inpainting (propagação pixel-a-pixel) e de largas áreas de inpainting (propagação por blocos). Ambas abordagens podem ser utilizadas tanto para completamento de imagens, como para desoclução.

### 4.1 Formulação do Modelo de Inpainting com o Uso de Wavelets

Este modelo foi criado baseando-se nos fundamentos de inpainting discutidos anteriormente neste texto. As características básicas necessárias para o bom funcionamento de uma aplicação de inpainting foram consideradas e abordadas. Decidiu-se utilizar a informação contida nos coeficientes da decomposição wavelet de uma imagem para preservar sua estrutura no preenchimento da região de inpainting, preenchendo as partes faltantes. As técnicas aplicadas no posterior preenchimento correto da imagem variam conforme a necessidade, e vão desde simples cópia de pixels, até métodos de síntese de texturas mais elaborados [FOL 2000, MöL 2003]. Esta seção apresenta um algoritmo genérico que fundamenta a base do modelo criado, e as seções seguintes especializam este algoritmo de acordo com a configuração dos dados a serem processados.

Os dados de entrada iniciais necessários são a imagem original a ser trabalhada, e um mapa da região de inpainting, também chamado aqui de **máscara de inpainting**. Com estes dados disponíveis, o algorítimo itera até o término da realização de duas importantes funções: determinar a ordem mais adequada de preenchimento da região de inpainting e o preenchimento efetivo desta. O pseudo-algoritmo abaixo explica de forma geral este processamento.

---

```

para uma imagem  $I$  e um mapa de inpainting  $\Omega$  (1)
    aplicar a TW dizimada um nível em  $I$ , gerando coeficientes de aproximação  $S$  e detalhe  $W^1$ ,  $W^2$  e
     $W^3$  (2)
    aplicar a TW em  $\Omega$  gerando  $\Omega_w$  com a mesma resolução dos coeficientes wavelet (3)
while  $|\Omega_w| \neq 0$  do
    determinar a prioridade para preencher  $S$ ,  $W^1$ ,  $W^2$  e  $W^3$ , de acordo com critério pré-determinado
    (4.1)
    Identificar o bloco que melhor preenche a região com maior prioridade (4.2)
    preencher esta região em  $S$ ,  $W^1$ ,  $W^2$  e  $W^3$  com o bloco mais adequado (4.3)
    Atualizar  $\Omega_w$ , marcando os blocos preenchidos (4.4)
end while
    aplicar a transformada inversa em  $S$ ,  $W^1$ ,  $W^2$  e  $W^3$  (5)
    retornar a imagem completada  $I'$  (6)

```

---

A imagem de trabalho, decomposta através de uma transformada wavelet, fornece informação suficiente para que o algoritmo consiga identificar a geometria de uma imagem  $I$ , e manter a coerência dos coeficientes visando a presevação das estruturas. Os coeficientes  $S$ ,  $W^1$ ,  $W^2$  e  $W^3$  (os índices  $2^1$  da escala foram omitidos, já que apenas um nível é utilizado na decomposição) denotam respectivamente a aproximação e detalhes vertical, horizontal e diagonal, descritos no capítulo anterior. Estes coeficientes de detalhe são utilizados para preservar a estrutura da imagem (ver figura 4.1), uma vez que as descontinuidades (altas freqüências) na imagem são capturadas por tais coeficientes. O aproveitamento destas descontinuidades nos coeficientes de detalhe como determinador da geometria da imagem é um dos conceitos que identifica este modelo. Entende-se por **informação original** os coeficientes de aproximação e detalhes que estejam fora da região de inpainting, e sua informação pode ser usada com total confiança.

O conjunto de coeficientes gerados auxilia o algoritmo a identificar corretamente por onde começar a preencher a região de inpainting  $\Omega_w$ , e também qual região na vizinhança de  $\Omega_w$  é

mais adequada para o preenchimento. Um importante conceito estabelecido neste modelo é o de **energia dos coeficientes**, que representa a importância do coeficiente na estrutura da imagem, e é calculado simplesmente elevando o valor do coeficiente ao quadrado.

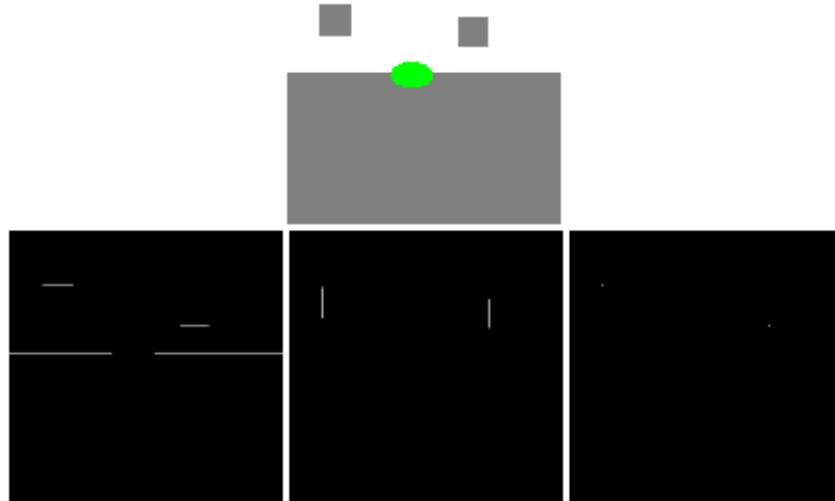


Figura 4.1: Coeficientes wavelet de detalhe em uma imagem de análise. A figura no topo é a imagem de trabalho não decomposta, destacando a região de inpainting em verde. Abaixo, são ilustradas as imagens de detalhe vertical, horizontal e diagonal, respectivamente.



Figura 4.2: Destaque da descontinuidade capturada pelo detalhe horizontal.

A figura 4.1 demonstra o processamento de uma imagem teste, ilustrando como fica sua decomposição wavelet nos coeficientes de detalhe. A região de inpainting (em verde) interrompe a imagem em uma parte onde há uma mudança de intensidade. No domínio da transformada, percebe-se que há um contorno horizontal que foi interrompido pela máscara de inpainting (figura 4.2). A idéia geral do algoritmo proposto é prolongar esse contorno horizontal e, o resultado da propagação desta parte da estrutura pode ser visto na figura 4.3(a). Finalmente, o resultado após a reconstrução (aplicação da TW inversa) é ilustrado na figura 4.3(b). Os detalhes, após processados e preenchidos são partes importantes não apenas para a transformada inversa da imagem, mas também para os próximos passos de iteração do algoritmo, como será

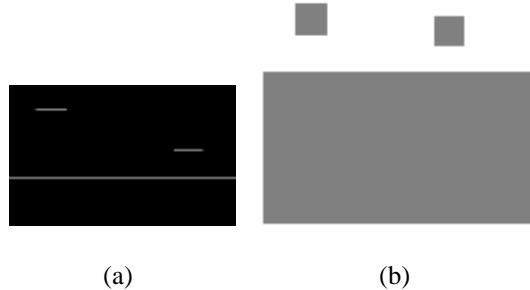


Figura 4.3: Preenchimento em destaque do detalhe vertical mantendo a estrutura (a) e o resultado final da imagem reconstruída (b).

visto nas seções seguintes.

#### 4.1.1 Processamento dos dados de entrada

A técnica desenvolvida neste trabalho permite que tanto imagens coloridas (no espaço de cores RGB) quanto imagens em escala de cinza sejam submetidas ao processo de inpainting. As imagens, em formato digital, podem tanto representar imagens nas quais se deseja recuperar alguma parte faltante em virtude de algum tipo de dano, quanto as que se deseja remover e/ou desobstruir alguma área. De fato, o modelo se aplica para qualquer finalidade de preenchimento, independente do objetivo desejado, e quem determinará o que será reconstruído é a máscara de inpainting informada.

Assim, os dados de entrada informados vêm na forma de uma imagem digital e uma máscara binária (criada pelo usuário), que demarca os pixels que deverão sofrer inpainting (máscara de inpainting), tendo valor 1 na região de inpainting e 0 no restante dos pixels. A informação da imagem original na posição da máscara de inpainting é então “apagada” (pixels preenchidos com zero), e calcula-se a transformada wavelet da imagem de entrada modificada usando apenas um nível na decomposição. A função wavelet escolhida como mais apropriada para este modelo é a wavelet de *Haar*, já que seu pequeno suporte afeta apenas uma região mínima em torno de cada pixel (wavelets com suporte maior tendem a espalhar o efeito da região de inpainting no domínio wavelet, como será visto a seguir).

Se a imagem de entrada for monocromática, calcula-se diretamente sua TW bi-dimensional. Para imagens coloridas, calcula-se a TW de cada canal de cor R, G e B separadamente (mais detalhes sobre o inpainting de imagens coloridas são dados nas seções 4.2.1 e 4.3). O resultado da transformação são os conjuntos de coeficientes de aproximação e detalhes, com resolução

equivalente a metade da resolução da imagem original. Também deve-se salientar que a borda criada na imagem original, na fronteira da máscara de inpainting tende a gerar uma borda ilusória na imagem (pois o valor dos pixels dentro da região de inpainting foi zerado), que acarreta em coeficientes wavelet de detalhe espúrios ao redor da região de inpainting. Tais coeficientes não fazem parte da informação original da imagem, e devem ser incorporados à região de inpainting  $\Omega_w$  no domínio de wavelet.

Para ajustar a resolução do mapa de inpainting fornecido e incluir coeficientes de detalhe espúrios gerados, calcula-se a TW do mapa de inpainting original (figuras 4.4(a) e 4.4(b)). Pixels nos quais ao menos um coeficiente wavelet (aproximação, detalhes horizontal, vertical ou diagonal) é diferente de zero foram efetivamente afetados pelo mapa de inpainting original, e são utilizados para formar o mapa de inpainting ajustado (figura 4.4(c)).

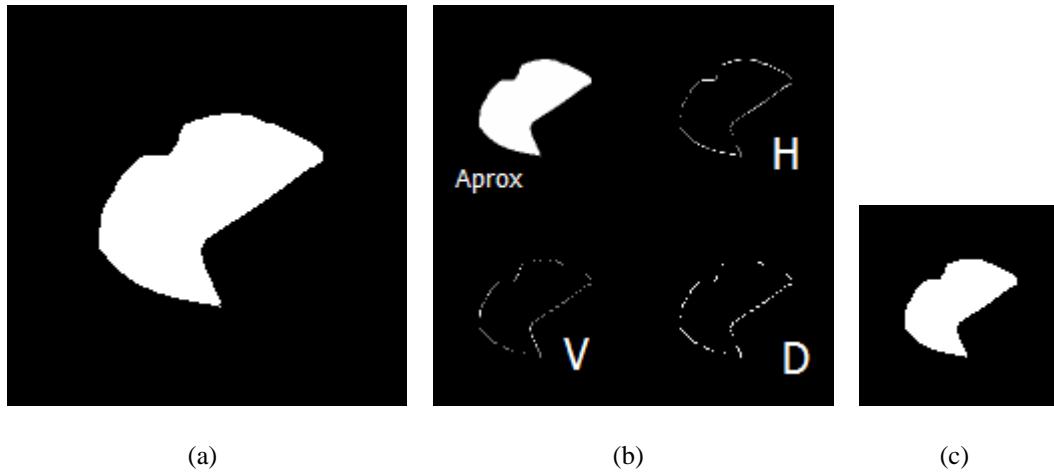


Figura 4.4: Processo de ajuste do mapa de inpainting. O mapa inicial em (a) tem a região de inpainting marcada como branco. A decomposição pela transformada wavelet gera os coeficientes em (b). A geração do mapa final como uma máscara binária está em (c).

#### 4.1.2 Prioridade dos blocos no preenchimento

O preenchimento da região de inpainting é feito por blocos (cada um conhecido como um **bloco a ser preenchido**), que, como opção de implementação, podem variar de tamanho sendo até mesmo unitários (1 pixel). Cada um destes blocos possui uma prioridade  $p$  associada, que determinará a ordem de preenchimento.

Bertalmio *et al.* observou em [BER 2000] que apenas preencher a região de inpainting, preenchendo blocos escolhidos espiralmente em direção ao seu centro não é suficiente. Neste trabalho, o primeiro importante uso dos coeficientes wavelet de detalhe é fornecer a ordem de

preenchimento da região de inpainting, selecionando corretamente em cada iteração, o bloco com maior prioridade de preenchimento (ordem de preenchimento). A determinação desta prioridade depende da forma escolhida para processar a imagem, e pode variar de acordo com o número de canais de cor e tamanho do bloco a ser preenchido, como será visto nas seções 4.2 e 4.3.

Pode-se avaliar o efeito da escolha da ordem de preenchimento da região de inpainting na figura 4.5. Na figura 4.5(c), o algoritmo implementado utilizou os coeficientes wavelet para determinar a ordem de preenchimento. Já na figura 4.5(d), foi retirada esta ordenação (o preenchimento foi feito sequencialmente, no sentido horário), fazendo com que o preenchimento ficasse incorreto. Esta ordem é extremamente importante na preservação da estrutura da imagem - no exemplo da figura 4.5(c), o mastro da placa tem sua estrutura vertical propagada corretamente.

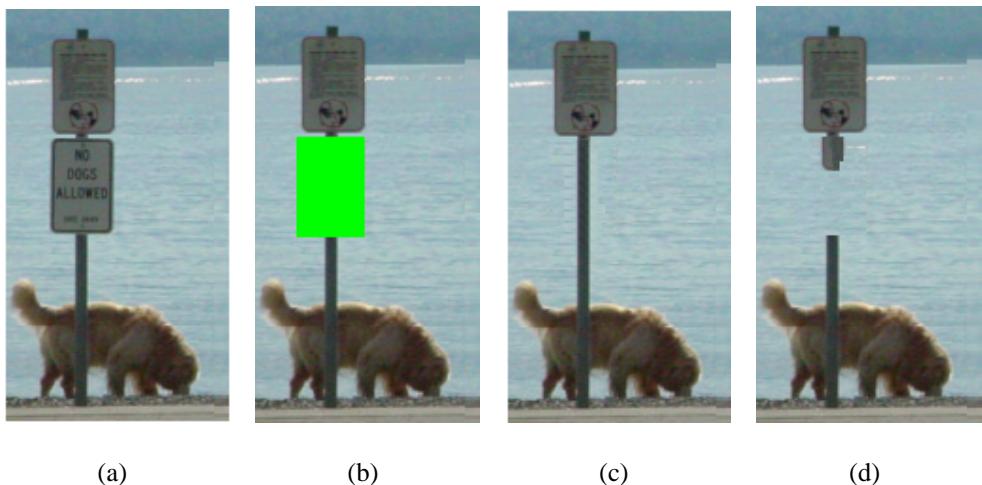


Figura 4.5: A figura original (a) processada com um mapa de inpainting (b). O resultado em (c) é obtido utilizando-se ordenação de preenchimento, já em (d), esta ordenação foi desconsiderada, gerando um resultado não satisfatório.

O algoritmo no início desta seção diz que, após encontrar o bloco com maior prioridade e o conteúdo para preenchê-lo (passos 4.1 e 4.2 do algoritmo), deve-se então completá-lo nos coeficientes e marcá-los como completos no mapa de inpainting. Uma vez que o bloco é preenchido nos coeficientes, sua região correspondente no mapa de inpainting passa a ser considerada como **válida**, significando que aqueles coeficientes podem ser utilizados para as próximas iterações. No entanto, uma vez que estes coeficientes recém marcados como válidos foram resultantes do processo de inpainting, estes não são tão “confiáveis” quanto os coeficientes que resultaram da

imagem original. Assim, faz-se necessário rotular cada coeficiente com um valor que indica o quanto ele é confiável, e pode contribuir para subsequentes preenchimentos. Isso deve ser considerado no momento de calcular a prioridade dos blocos.

Criminisi *et al.* criam em [CRI 2004] um mecanismo que associa **termos de confiança** aos pixels das imagens processadas, aqui chamados de  $K$ . Foi utilizado neste trabalho esta mesma abordagem, onde a confiança  $K$  de cada coeficiente está no intervalo  $(0, 1]$ . Inicialmente, todos os coeficientes fora da região de inpainting tem valor 1, por serem valores originais. Os coeficientes preenchidos vão tendo seus termos de confiança atualizados considerando os termos de confiança da sua vizinhança. Com isto, quanto mais ao centro da região de inpainting o bloco a ser preenchido se aproxima, menor o valor dos termos de confiança. A figura 4.6 mostra os termos de confiança dispostos em forma de mapa, que são atualizados a cada iteração. Além do termo de confiança, este trabalho emprega outros critérios para determinar a ordem do preenchimento dos pixels, como a energia dos coeficientes wavelet vizinhos e a geometria da região de inpainting. Mais detalhes são apresentados nas especializações da técnica, descritas nas seções 4.3 e 4.2.

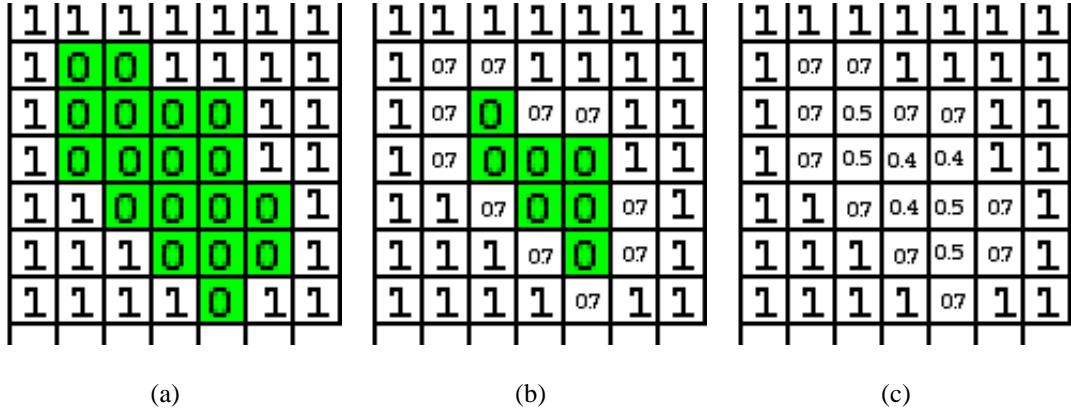


Figura 4.6: Mapa com variação dos valores de  $K$  à medida que o algoritmo vai preenchendo a região de inpainting (em verde), e atualizando este mapa conforme a equação pertinente (seções 4.3 e 4.2). Figura (a) apresenta o mapa em seu estado original, sem nenhum pixel preenchido, seguindo da figura (b) que mostra um passo intermediário, e a figura (c) o estado final do mapa. Note que no estado inicial (a) todas as posições fora da região de inpainting tem valor 1 e as dentro tem valor 0.

### 4.1.3 Preenchimento da região de inpainting

Determinar a prioridade de preenchimento nos diz qual parte da região de inpainting deve ser completada primeiramente, visando manter a correta estrutura da imagem conforme o **princípio da conectividade** [KAN 79, SAR 93]. A identificação do conteúdo com o qual deve-se preencher  $\Omega_w$  é uma outra tarefa definitiva, e que deve garantir também o prolongamento das estruturas, além da correta propagação das texturas para o interior da região de inpainting.

Para isso, são definidos o **mecanismo de preenchimento**, um bloco que será preenchido  $\beta$  e um bloco  $\theta$  que o preenche. O mecanismo de preenchimento é responsável por localizar na vizinhança de  $\beta$  o conteúdo que mais se apropria para o seu preenchimento. Como  $\beta$  está na fronteira da região de inpainting (tem coeficientes dentro, e coeficientes fora da região), devem ser preenchidos em  $\beta$  apenas aquelas posições que estejam dentro da região de inpainting, mantendo os outros intactos. A busca por  $\theta$  deve sempre ser realizada sobre os coeficientes wavelet, e sua utilização depende de detalhes específicos definidos para o mecanismo de preenchimento. É importante salientar que a determinação do bloco a ser preenchido e a determinação de qual bloco será utilizado no preenchimento depende, em geral, de todos os coeficientes wavelet (aproximação e detalhes). Entretanto, o preenchimento é realizado independentemente em cada tipo de coeficiente (por exemplo, um bloco no detalhe diagonal não pode ser preenchido com informação do detalhe vertical).

A definição do mecanismo de preenchimento depende da forma de busca desejada, e neste trabalho foram desenvolvidas duas maneiras: uma para blocos  $\beta$  de tamanho unitário (apenas 1 coeficiente preenchido por iteração) e blocos de tamanho maiores ou iguais a  $3 \times 3$ . Estes mecanismos estão definidos nas seções 4.2 e 4.3.

#### Preenchimento dos coeficientes de detalhe

Uma vez encontrado  $\theta$ , basta completar  $\beta$  com seu conteúdo. Vale lembrar que o modelo trabalha em domínio de wavelet. Então, ao contrário das abordagens que preenchem a região de inpainting diretamente no domínio de intensidade (ou cor), deve-se preencher os coeficientes wavelet da aproximação e dos três detalhes (horizontal, vertical e diagonal). Esta tarefa é naturalmente necessária, pois a imagem final é resultado da transformada inversa destes coeficientes, e é necessário que todos esses coeficientes estejam completos para que o resultado seja coerente.

Além da necessidade de ter dados completos para a realização da transformada inversa,

foi visto que os próprios coeficientes preenchidos influenciam nos passos seguintes da iteração. Então, na etapa de preenchimento, o algoritmo deve:

- Certificar-se de que os coeficientes de aproximação e detalhes horizontal, vertical e diagonal, sejam preenchidos;
- Marcar a área de  $\beta$  no mapa de inpainting como uma região válida (não mais região de inpainting);
- Atualizar os valores de confiança  $K$  correspondente a região de  $\beta$ , de acordo com a regra especificada.

Satisfeitos esses itens, o algoritmo está pronto para a próxima iteração, e executará até que  $|\Omega_w| = 0$ .

Foi visto até este ponto o modelo genérico de inpainting digital no domínio de wavelets, abordando os passos básicos necessários em cada etapa. A seguir, serão apresentadas as duas especializações deste modelo, descrevendo com detalhes o método de determinação da prioridade e mecanismo de preenchimento para o completamento de regiões pequenas (pixel-a-pixel) e de regiões maiores (blocos). A idéia inicial na elaboração deste trabalho era realizar o preenchimento da região de inpainting um pixel de cada vez (chamado de preenchimento pixel-a-pixel), e foi desenvolvida inicialmente. Essa técnica produz resultados visualmente interessantes para pequenas áreas de inpainting, e sua simplicidade permite a rápida geração de resultados. Entretanto, regiões de inpainting com texturas mais elaboradas não são eficientemente preenchidas por essa abordagem. Assim partiu-se para a síntese por blocos propriamente dita, que é mais elaborada, e leva em consideração a geometria da região de inpainting. Essa segunda abordagem utiliza blocos de tamanho  $3 \times 3$  ou superiores, completando diversos pixels a cada iteração, e permitindo uma melhor reconstrução da textura.

## 4.2 Completando Pequenas Áreas de Inpainting

A especialização do modelo descrita nesta seção cria um mecanismo de inpainting onde o preenchimento é realizado propagando cada coeficiente individualmente, que será referido por **preenchimento pixel-a-pixel**. O processo é um detalhamento do modelo definido na seção anterior, e como tal, é dividido em 3 etapas: decomposição da imagem, definição da prioridade de preenchimento dos coeficientes, finalizando com a propagação para as regiões de inpainting em

espaço de coeficientes de wavelet. Aqui, são definidos o método de determinação da prioridade de preenchimento e o mecanismo de preenchimento, especializados para este procedimento: completar pequenas regiões de inpainting.

A decomposição com a wavelet *Haar* em um único nível gera os quatro conjuntos de coeficientes que serão manipulados. Mais especificamente, gera-se a imagem de aproximação  $S$ , e os detalhes  $W^2$  (horizontal),  $W^1$  (vertical) e  $W^3$  (diagonal) (os coeficientes de detalhes são tidos aqui também como **direções**, por representarem as direções em que as estruturas aparecem). Para utilizar uma notação mais intuitiva, serão utilizadas as variáveis  $C_a$ ,  $C_h$ ,  $C_v$  e  $C_d$  para representar os coeficientes  $S$ ,  $W^2$ ,  $W^1$  e  $W^3$ , respectivamente. Primeiramente será descrito o mecanismo utilizado para inpainting de imagens em escala de cinza, seguindo das modificações necessárias para processar imagens coloridas.

O cálculo da prioridade dos blocos é feito através da análise dos coeficientes de wavelet na fronteira da região de inpainting. Além de encontrar o local com maior prioridade, identifica-se a direção que mais se salienta (considerando  $C_h$ ,  $C_v$  e  $C_d$ ), e pode significar uma estrutura a ser propagada. Para determinar a prioridade de preenchimento utiliza-se a **energia** dos coeficientes de wavelet de detalhe, o que nada mais é do que o valor dos coeficientes elevado ao quadrado. Assim, esta energia  $E_{dir}$ , para  $dir \in \{h, v, d\}$ , é dada por  $E_{dir} = C_{dir}^2$ . A direção na qual a energia é máxima indica a direção de projeção dos coeficientes na região de inpainting. De fato, cada direção  $dir$  possui uma **máscara direcional** associada  $\gamma^{dir}$ , que é utilizada para calcular o valor do pixel a ser preenchido. Essa máscara tem o objetivo de preencher a região de inpainting preservando a estrutura da imagem (por exemplo, uma borda capturada predominantemente pela wavelet vertical tende a se prolongar na direção horizontal, como mostrado nas figuras 4.2 e 4.3(a)). Ainda, deve-se considerar o termo de confiança  $K$ , inicializado como 0 para todos pixels correspondentes a região de inpainting e 1 para disto.

A prioridade é definida pelo detalhe que possui maior energia, em uma pequena vizinhança de cada pixel da fronteira da região de inpainting. Mais formalmente, seja  $\Omega_w$  a região de inpainting no domínio wavelets, e seja  $\partial\Omega_w$  sua fronteira. Para cada pixel  $(x, y) \in \partial\Omega_w$ , consideramos uma janela  $J$  de tamanho  $3 \times 3$  centrado em  $(x, y)$ . Sejam  $\gamma^h$ ,  $\gamma^v$ ,  $\gamma^d$  máscaras direcionais de tamanho  $3 \times 3$ , dadas por:

$$\gamma^h = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \gamma^v = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \gamma^d = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad (4.1)$$

que são utilizadas para restringir o tipo de coeficiente de detalhe que será considerado com base na direção de borda esperada. Dentro da região  $J$ , a soma das energias dos coeficiente de detalhe é calculada utilizando a respectiva máscara direcional:

$$E_{dir}(x, y) = \sum_{(u,v) \in J \cap \Omega_w^c \cap \gamma^{dir}} C_{dir}(u, v)^2 K(u, v), \quad (4.2)$$

para  $dir \in \{h, v, d\}$ ,  $\Omega_w^c$  como o complemento da região de inpainting, e termo de confiança  $K$ . A figura 4.7 ilustra o procedimento.

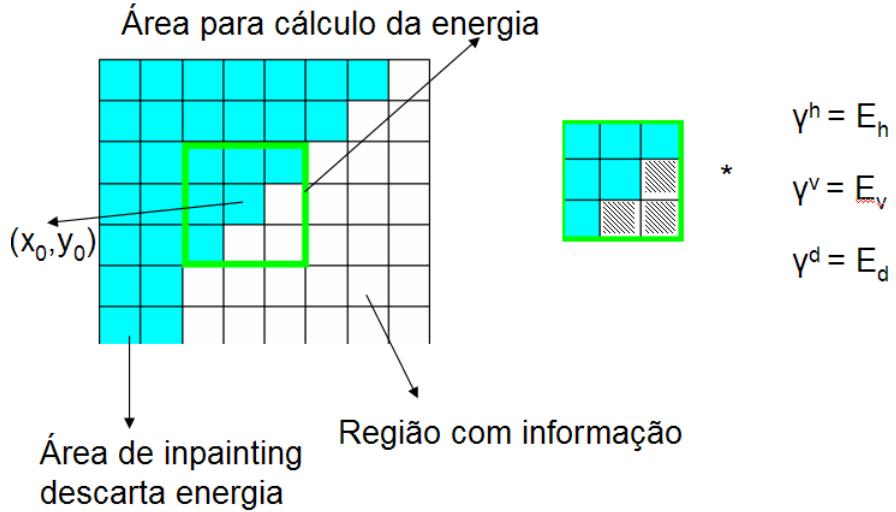


Figura 4.7: Cálculo da energia na fronteira região de inpainting (em verde) . A área hachurada será computada considerando as máscaras  $\gamma^{dir}$ , para  $dir \in \{h, v, d\}$ .

Para cada pixel  $(x, y) \in \partial\Omega_w$ , é calculada a maior energia direcional  $E(x, y)$ , e também é obtida a direção  $dir_{max}$  na qual a energia é máxima, ou seja:

$$E(x, y) = \max_{dir \in \{h, v, d\}} E_{dir}(x, y), \quad dir_{max}(x, y) = \operatorname{argmax}_{dir \in \{h, v, d\}} E_{dir}(x, y). \quad (4.3)$$

Os valores de  $E(x, y)$  são utilizados para calcular a ordem do preenchimento. De fato, o completamento da região de inpainting começará por pixels com alta energia, ou seja, o pixel  $(x_0, y_0)$  onde começará o inpainting é dado por:

$$(x_0, y_0) = \operatorname{argmax}_{(x, y) \in \partial\Omega_w} E(x, y). \quad (4.4)$$

A direção  $dir_{max}(x_0, y_0)$  é então utilizada para determinar o método de preenchimento do pixel  $(x_0, y_0)$ , levando em consideração a máscara direcional  $\gamma^{dir_{max}}(x_0, y_0)$  associada. Mais precisamente, o valor  $C(x_0, y_0)$  usado para preencher esse pixel é calculado fazendo a média

dos coeficientes na janela  $J$  de tamanho  $3 \times 3$  centrada em  $(x_0, y_0)$ , considerando a região de inpainting e a máscara  $\gamma^{dir_{max}}(x_0, y_0)$  encontrada na equação (4.3). Isso deve ser aplicado para preencher  $(x_0, y_0)$  tanto nos coeficientes de detalhes quanto nos de aproximação. Esta descrição qualifica o **mecanismo de preenchimento** para esta solução, que formalmente resulta em  $C_{dir}(x_0, y_0)$  como definido na equação (4.5) abaixo:

$$C_{dir}(x_0, y_0) = \frac{\sum_{(x,y) \in \Omega^c \cap J \cap dir_{max}} C_{dir}(x, y)}{|\Omega^c \cap J \cap dir_{max}|}, \text{ para } dir \in \{a, h, v, d\}, \quad (4.5)$$

utilizando apenas os pixels fora da região de inpainting ( $\Omega^c$ ) e que tenham valor 1 na máscara  $\gamma^{dir}$  da direção preferencial encontrada. O termo de confiança  $K$  em  $(x_0, y_0)$  é atualizado (equação (4.6)) também considerando os coeficientes na sua vizinhança que contribuiram para o preenchimento do mesmo e um **fator de atenuação**  $\alpha$ . Este fator de atenuação determina o quanto decai a confiança na região preenchida (analisando os resultados, observou-se que um valor de  $\alpha = 0.8$  é suficiente, e que também não deve ser inferior a 0.5, para não atenuar em excesso coeficientes com energia proporcionalmente muito alta).

$$K(x_0, y_0) = \frac{\sum_{(x,y) \in \Omega^c \cap J \cap dir_{max}} \alpha K(x, y)}{|\Omega^c \cap J \cap dir_{max}|}. \quad (4.6)$$

Um sistema protótipo para esta técnica foi construído e alguns resultados foram gerados, demonstrando os casos de bom funcionamento do modelo, e pontos onde esta técnica deve ser melhorada. A figura 4.8 demonstra a capacidade de reproduzir corretamente estruturas contínuas bem definidas, como a linha intangível que separa dois tons de cinza diferentes na figura.



Figura 4.8: Resultado gerado pelo sistema protótipo. À esquerda, a imagem com a região de inpainting em preto, e à direita, a reconstrução feita pelo sistema.

A figura 4.9 demonstra um outro bom resultado da técnica, recuperando a estrutura na borda da parede da casa. A textura da parede da casa também é recuperada satisfatoriamente, contudo, deve-se observar que a região de inpainting é bastante fina, a linha original tem espessura de dois pixels.



Figura 4.9: Borda reconstruída com sucesso em uma região de inpainting com maior composição de texturas.

Um problema do algoritmo proposto pode ser visto na figura 4.10. A área relativa ao telhado da casa (no detalhe da figura 4.11) possui um erro de decisão do algoritmo com relação a direção de propagação escolhida. Detalhes sobre a causa do problema e possíveis melhorias podem ser vistos na seção 5.2.



Figura 4.10: Uma das falhas da técnica, em não reconstruir corretamente pixels quando nenhuma direção não se destaca frente às outras.

#### 4.2.1 Solução para imagens coloridas

A extensão do método para imagens coloridas é bastante simples. A idéia é aplicar a TW a cada canal de cor independentemente, e utilizar a informação conjunta dos três canais de cor no cálculo da ordem de preenchimento. De fato, a energia direcional em cada pixel é a soma das energias nos três canais de cor.

O processo é bastante simples, e realizado em três etapas: 1) dividir a imagem RGB em três matrizes, uma para cada canal; 2) aplicar a transformada wavelet desejada em cada um dos



Figura 4.11: Detalhe de uma reconstrução não satisfatória

canais separados; 3) compor uma matriz que unifique os três canais, resultante da soma dos três canais, elevados ao quadrado. Ou seja, para cada canal  $I_R$ ,  $I_G$ ,  $I_B$ , suas transformada wavelet correspondentes são obtidas independentemente, gerando um conjunto de coeficientes  $C_{dir}^c$ , onde  $c \in \{R, G, B\}$  denota o canal de cor, e  $dir \in \{a, v, h, d\}$  denota o tipo de coeficientes (aproximação, vertical, horizontal ou diagonal).

As energias  $E_{dir}^{RBG}$  dos coeficientes wavelet da imagem colorida são compostas simplesmente somando as contribuições nos três canais de cor para cada tipo de coeficiente wavelet, como indicado na equação abaixo.

$$E_{dir}^{RBG} = \sum_{c \in \{R, G, B\}} (C_{dir}^c)^2, \text{ para } t \in \{v, h, d\} \quad (4.7)$$

Assim temos as energias necessárias para determinar a ordem e preencher a região de inpainting. A atualização no entanto é feita três vezes para cada conjunto de coeficientes (aproximação, horizontal, vertical e diagonal) associado a cada canal. Isto faz com que, ao final da iteração do preenchimento, tenha-se 12 blocos (os 4 resultados da transformada wavelet para cada canal RGB), e reconstrução da imagem também precisa ser feita por etapas.

Para a reconstrução, primeiramente é aplicada a transformada inversa para os coeficientes de cada canal, individualmente, resultando em 3 matrizes, cada uma com a informação de um único canal. Em seguida, a imagem final é reconstruída, agrupando estas 3 matrizes de canais em apenas uma imagem RGB final.

Alguns resultados podem ser vistos nas figuras 4.12(b) e 4.13(c). Nesta última foi possível remover uma grande área da figura original em 4.13(a). No entanto, pode-se notar que a textura da área ao redor do suporte da placa (área que representa a água ao fundo) foi reconstruída com um padrão que não representa muito bem a textura real, gerando muitos artefatos visuais. Isto

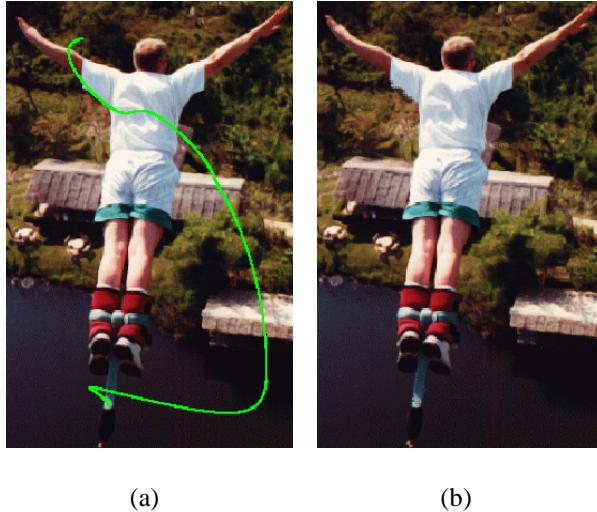


Figura 4.12: Restauração de uma imagem colorida com o método pixel-a-pixel. A figura (a) destaca a região com um dano simulado em verde, que é restaurada e apresentada em (b).

se dá pelo fato de que apenas pixels individuais são preenchidos por ciclo, o que não é suficiente para capturar toda a informação da textura, e conseguir sintetizá-la corretamente.

A síntese das texturas em inpainting passou a ser um tópico considerado importante nas técnicas desenvolvidas mais recentemente. Inicialmente, inpainting tradicional e síntese de texturas eram conceitos bem separados, mas hoje se confundem como uma única metodologia de inpainting digital. Métodos para replicação de textura baseada em pixel como o de [EFR 99] poderiam ser adaptados e utilizados para o método de inpainting proposto no domínio de wavelets. Porém, o desenvolvimento deste trabalho voltou-se a solução e aplicação de síntese de texturas baseadas em técnicas mais recentes, que manipulam blocos inteiros de pixels [TON 2005, LIA 2001, TON 2006]. A seção seguinte apresenta a técnica estendida do modelo base, que realiza inpainting digital no domínio de wavelets por blocos de pixels de dimensões  $3 \times 3$  ou superiores.

### 4.3 Modelo de Inpainting por Blocos de Coeficientes

Esta implementação do método de inpainting digital com o uso de wavelets visa criar um mecanismo de inpainting capaz de reconstruir tanto a estrutura como as texturas na região de inpainting. Isto torna possível tanto completar largas áreas danificadas em uma imagem, quanto prover desocclusão. O método de inpainting proposto segue a idéia básica apresentada em [CRI 2004], onde as áreas de inpainting são completadas por propagação dos isofotos e

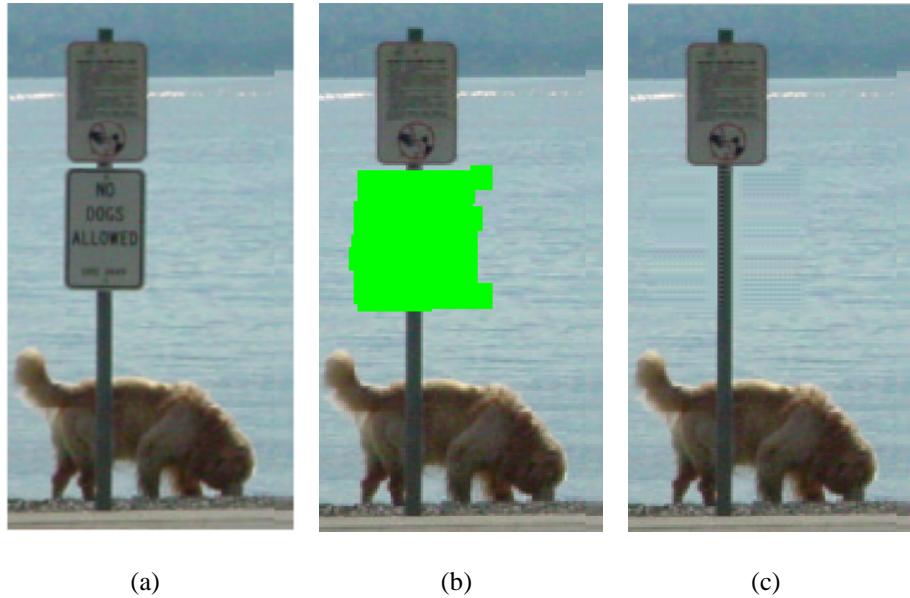


Figura 4.13: Desocclusão em (a) pelo método pixel-a-pixel de uma imagem colorida. A região a ser removida é representada na figura (b) em verde, e a figura em (a) após processada, com a região removida.

síntese de texturas. A abordagem continua sendo reconstruir as regiões de inpainting no domínio de wavelets, projetando conjuntos de blocos de informações, ao contrário da abordagem por unidade descrita na seção anterior. Esta técnica facilita a identificação da estrutura a ser propagada, e pretende capturar os aspectos geométricos locais, utilizando os coeficientes na decomposição wavelet.

O completamento feito por blocos segue o mesmo algoritmo definido na seção 4.1, onde a idéia básica de determinar a prioridade e preencher as áreas com dados da vizinhança mantém-se fortemente presente. As alterações necessárias são realizadas no método de determinação da prioridade do preenchimento e, principalmente, no preenchimento em si, que agora aplica técnicas de síntese de texturas mais avançadas.

Para isso, utiliza-se então a informação dos coeficientes de wavelet, na seleção e preenchimento de blocos na região de inpainting, e o trabalho de Tonietto et al. [TON 2006] demonstra que o uso dos coeficientes wavelet pode melhorar a qualidade deste preenchimento por síntese de texturas em blocos. O mecanismo de inpainting com síntese de texturas utilizando o domínio de wavelets como espaço de atuação é uma das importantes contribuições deste modelo de inpainting por blocos de pixels, e finaliza este capítulo.

### 4.3.1 Determinação da ordem de preenchimento

A determinação da ordem de preenchimento necessita desta vez considerar regiões de inpainting maiores, com geometrias bem definidas. As variáveis de entrada são trabalhadas em domínio de wavelets, como a região de inpainting  $\Omega_w$ , sua fronteira  $\partial\Omega_w$  e o conjunto de coeficientes de aproximação e detalhes, contendo a informação original.

A preparação dos dados de entrada é feita da mesma forma que no método pixel-a-pixel (seção 4.2). As imagens em escala de cinza são processadas diretamente, e as coloridas são convertidas em mapas de energia e vetores gradiente, utilizadas para a definição da ordem e a seleção do conteúdo que preencherá a região alvo. As seções seguintes definem detalhadamente o método para determinação da ordem de preenchimento e o mecanismo de preenchimento, específicos para este modelo de inpainting por blocos.

A seguir, será apresentada a forma de determinação da prioridade de preenchimento, que inicialmente introduz a técnica do gradiente, descrita por Criminisi *et al.* em [CRI 2004], e que fornece importante conhecimento para a abordagem mista criada neste trabalho. Esta abordagem mista é utilizada na implementação do mecanismo de inpainting por blocos de pixels.

#### Determinação da prioridade do preenchimento com estimativa do gradiente da imagem

A estimativa do **vetor gradiente** é eficiente na determinação da geometria local de imagens digitais [GON 2001, MCA 2004]. Para o inpainting digital, o vetor gradiente pode ser utilizado tanto para determinar a orientação local das estruturas na parte válida da imagem, quanto para analisar a geometria local da região de inpainting. A idéia apresentada em [CRI 2004] é priorizar regiões nas quais a direção local do seu contorno se choca ortogonalmente com a fronteira da região de inpainting, conforme ilustrado na figura 4.14. Equivalentemente, o vetor ortogonal ao gradiente deve ser paralelo ao vetor normal à fronteira da região de inpainting. De fato, tal medida de prioridade pode ser quantificada através do produto interno entre o vetor normal à fronteira de inpainting e o vetor perpendicular ao gradiente na região válida (que corresponde à orientação local do contorno). Quanto maior o valor do produto interno (em módulo), maior deve ser a prioridade de inpainting.

Desenvolveu-se então neste trabalho um método para determinação de prioridade de preenchimento da região de inpainting, baseada na idéia proposta em [CRI 2004], porém adaptando o processamento do espaço de imagem para o domínio de wavelets. Assim, para a aplicação do mecanismo da estimativa do gradiente seguindo a proposta de Criminisi *et al.* seria necessário

calcular sobre a imagem de aproximação, resultante da decomposição pela transformada wavelet. Sendo a aproximação uma versão suavizada (devido à aplicação dos filtros passa-baixa da transformada wavelet) da imagem original, ela é bastante adequada para o cálculo do vetor gradiente, pois o ruído de alta freqüência atrapalham a estimativa do gradiente.

Nesta abordagem, a determinação da prioridade de preenchimento encontra o vetor gradientes da imagem, calculando sobre os coeficientes de aproximação. O vetor normal à fronteira da região de inpainting é calculado sobre a própria fronteira (utilizando a máscara de inpainting). Esta determinação do vetor gradiente pode ser feita utilizando métodos tradicionais. Assim, para encontrar a região cujo preenchimento é prioritário, verifica-se o grau de paralelismo entre o vetor  $\nabla I(x, y)^\perp$  (ortogonal ao gradiente da imagem) e a normal  $\vec{n}$  à região de inpainting, ambos vetores ponderados pelo termo de confiança  $K$ . De fato, tal grau de paralelismo pode ser calculado através do produto escalar, e então o pixel  $(x_0, y_0)$  onde deve começar o preenchimento é dado por:

$$(x_0, y_0) = \underset{(x,y) \in \partial\Omega_w}{\operatorname{argmax}} |\nabla I_n(x, y)^\perp \cdot \vec{n}| K(x, y), \quad (4.8)$$

onde

$$\nabla I_n(x, y)^\perp = \frac{\nabla I(x, y)^\perp}{|\nabla I(x, y)^\perp|} \quad (4.9)$$

representa a normalização de  $\nabla I(x, y)^\perp$

Apesar de ser possível utilizar a imagem aproximação para estimar o gradiente da imagem, um método mais apropriado para isto utiliza-se dos coeficientes já existentes, resultantes da transformada wavelet, formulando uma abordagem mista descrita a seguir.

### Método misto para determinação da prioridade de preenchimento

Uma das propriedades básicas inerentes à TW é que coeficientes horizontal e vertical ( $C_h$  e  $C_v$  respectivamente) capturam altas freqüências nas direções horizontal e vertical, respectivamente. De fato, tais coeficientes podem ser utilizados como uma estimativa do vetor gradiente, sendo uma alternativa ao mecanismo descrito anteriormente (que precisa estimar o gradiente na imagem de aproximação). Em outras palavras, o vetor gradiente  $\nabla I$  pode ser aproximado via:

$$\nabla I = \begin{bmatrix} C_h \\ C_v \end{bmatrix}, \quad (4.10)$$

e tal estimativa pode ser utilizada no cálculo da prioridade. Além disso, como foi utilizado na versão para preenchimento pixel-a-pixel, a energia dos coeficientes wavelet é incluída no

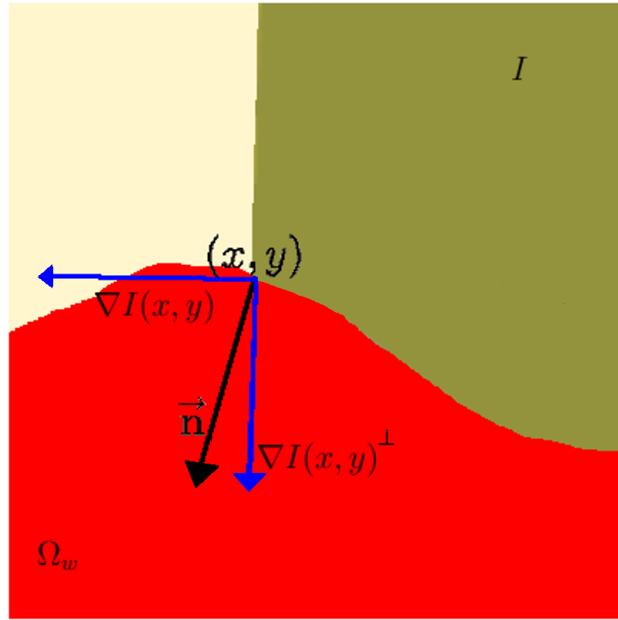


Figura 4.14: Vetores gradiente no ponto de análise (destaque em vermelho). Na figura, os vetores gradiente  $\nabla I(x, y)$  e seu ortogonal  $\nabla I(x, y)^\perp$ . O vetor  $\vec{n}$  representa o vetor normal à fronteira da região de inpainting.

cálculo da prioridade de preenchimento. Entretanto, ao contrário do método pixel-a-pixel, nenhuma máscara direcional é aplicada, pois o vetor gradiente já fornece informação necessária sobre como a estrutura da imagem penetra na região de inpainting.

Continua-se utilizando o produto interno entre o ortogonal do gradiente e o vetor normal à fronteira da região de inpainting como determinante da prioridade. Acrescentou-se no entanto, um fator que leva em consideração a energia dos coeficientes de wavelet no pixel de análise. Assim, o uso do gradiente fornece informação sobre a geometria da imagem, e as energias dos coeficientes wavelet destacam o pixel mais significativo entre geometrias analisadas na fronteira da região de inpainting. Será descrito primeiramente como utilizar os coeficientes wavelet para determinar o vetor gradiente, seguindo pelo cálculo de prioridade de preenchimento considerando a perpendicularidade dos vetores gradiente e a energia dos coeficientes wavelet.

A energia dos coeficientes wavelet utilizada para complementar a determinação da prioridade depende dos três conjuntos coeficientes wavelet de detalhe. Conforme descrito na equação (4.3), a energia adotada  $E(x, y)$  é o máximo das energias entre os três coeficientes de detalhe, constituindo o **mapa de energia** utilizado na equação.

Com os vetores e mapa de energias disponíveis, a equação (4.8) foi adaptada para considerar também as energias  $E$  dos coeficientes e encontrar o ponto central do bloco com maior

prioridade de preenchimento. A equação abaixo descreve como encontrar o ponto  $(x_0, y_0)$  com maior prioridade, utilizando o gradiente normalizado da imagem  $\nabla I_n(x, y)$  estimado utilizando os coeficientes  $C_h$  e  $C_v$ , e cada vetor normal à fronteira da região de inpainting representado como  $\vec{n}$  (figura 4.14).

$$(x_0, y_0) = \underset{(x,y) \in \partial\Omega_w}{\operatorname{argmax}} |\nabla I_n(x, y)^\perp \cdot \vec{n}| K(x, y) E(x, y), \quad (4.11)$$

onde  $K(x, y)$  denota o termo de confiança, definido na seção anterior e atualizado conforme a equação (4.6), considerando janela direcional com 1 em todas as posições.

Novamente, caso a imagem for em escala de cinza, pode-se utilizar os dados diretamente. Contudo, no caso de imagens coloridas RGB, deve-se compor os canais como citado anteriormente, utilizando a equação (4.7). Isto é fundamental para que todos os canais possam ser corretamente incluídos na determinação da energia. No caso específico do uso dos coeficientes de detalhe para determinar o gradiente, os termos  $C_h$  e  $C_v$  empregados na equação (4.10) denotam a soma dos respectivos coeficientes  $C_h^c$  e  $C_v^c$  ao longo dos três canais de cor ( $c \in \{R, G, B\}$ ). Esta solução considera que o vetor resultante da soma dos gradientes de cada canal pode ser seguramente utilizado para calcular a prioridade, pois carrega a contribuição da geometria da imagem, existente em cada canal. Ainda, como a máscara  $\Omega_w$  é binária, nenhuma composição de dados é necessária, e o vetor gradiente normal  $\vec{n}$  pode ser determinado através de métodos tradicionais.

Aplicando a equação que determina a ordem de preenchimento, obtém-se a coordenada  $(x_0, y_0)$  do centro do bloco que será completado primeiramente, e essa coordenada é passada à próxima etapa do processo (etapa 4.2 do pseudo-algoritmo definido na seção 4.1) que efetivamente preenche a região ao redor de  $(x_0, y_0)$ , assunto que será abordado na seção seguinte.

A figura 4.15 apresenta uma comparação de um procedimento desocclusão, utilizando apenas a informação do gradiente para determinar a prioridade de preenchimento (idéia original de Criminisi et al [CRI 2004]), comparando com a técnica construída neste trabalho, que utiliza a informação do gradiente e a energia dos coeficientes wavelet. Os resultados apresentam uma melhor reconstrução da estrutura da imagem no processo que utiliza o método por wavelets. Utilizar a energia do coeficiente de wavelet tem vantagens na reconstrução da estrutura, principalmente nos casos onde existam estruturas que sejam bem detectadas nos coeficientes de detalhe da transformada wavelet (estruturas horizontais, verticais e diagonais).

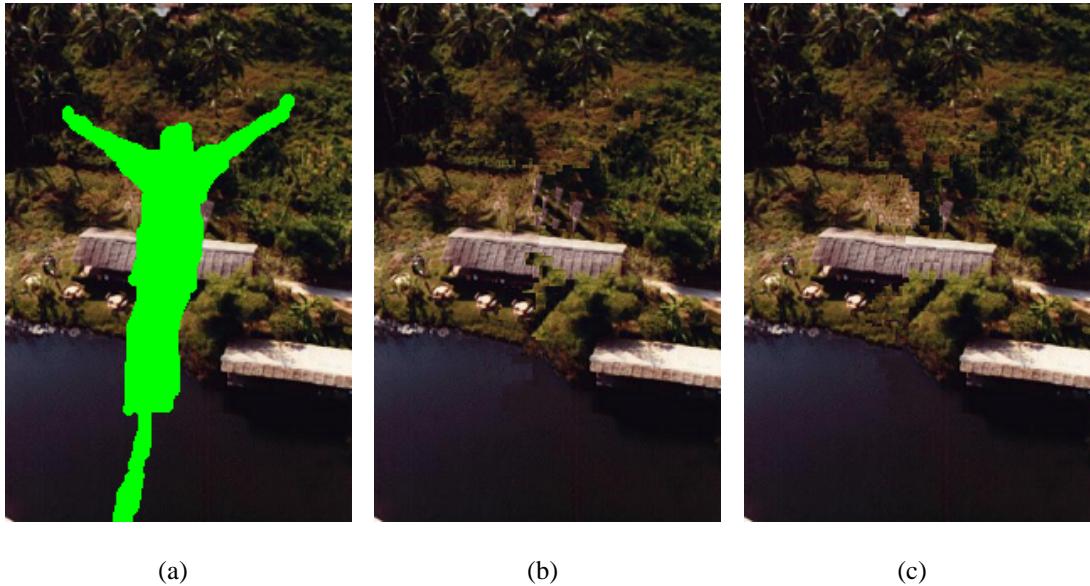


Figura 4.15: A figura original (a) com sua máscara de inpainting (em verde) é processada utilizando o método do gradiente (b) para determinar a prioridade de preenchimento. Um melhor resultado, utilizando wavelet aparece na figura (c).

#### 4.3.2 Preenchimento de blocos por síntese de texturas

Para preencher largas áreas de uma imagem, é necessário preservar não somente as estruturas, mas também a informação de textura. Técnicas de síntese de texturas tem sido utilizadas para tal tarefa com sucesso.

Num processo de síntese de textura tradicional uma imagem chamada **padrão** é utilizada como base, e é replicada para formar uma imagem maior. Esta replicação possui uma **métrica** que define como esse padrão se encaixa durante sua replicação: quanto melhor essa métrica, melhor o resultado final. Para o inpainting digital, o processo é semelhante: métricas similares podem ser aplicadas com sucesso, mas o padrão de textura pode ser diferente a cada iteração (ou seja, os blocos a serem encaixados podem ter padrões de textura diferentes em posições diferentes da região de inpainting). Assim, ao invés de ter-se um padrão que deve ser replicado para formar uma imagem, existe um **bloco base**  $\beta$  a ser preenchido, e devemos procurar na imagem qual bloco realiza um melhor encaixe neste bloco base.

Uma vez definida a ordem de preenchimento da região de inpainting, a posição escolhida terá seu conteúdo completado com um bloco de pixels que melhor se encaixa no local. Um dos métodos mais utilizados para realizar este completamento (quando por blocos), é baseado no conceito de **síntese de textura por exemplos** [ASH 2001, EFR 2001, HER 2001, LIA 2001], cuja métrica avalia a variação entre as cores de cada pixel em análise.

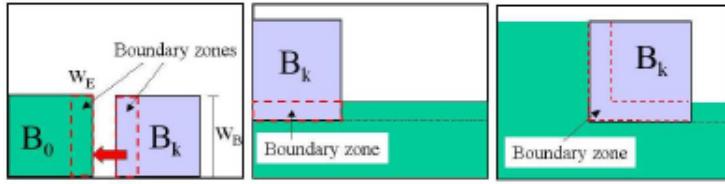


Figura 4.16: Região de avaliação dos blocos do trabalho de Tonietto *et al.* [TON 2005].

Assim, foi desenvolvido para este trabalho um método que copia na região de inpainting conjuntos de coeficientes com dimensões  $3 \times 3$  ou maiores, disponíveis na vizinhança de um bloco base. A região que deve ser completada fornece informações que são utilizadas para escolher o melhor bloco de pixels que a completa, semelhante a síntese de texturas por exemplos. Como o método é desenvolvido no domínio de wavelets, o conceito de síntese baseada em exemplos deve ser aplicado tanto nos coeficientes de detalhe quanto de aproximação. A métrica utilizada definirá qual bloco (da vizinhança de um bloco tomado como original) encaixa melhor na região a ser preenchida.

Uma grande vantagem da reprodução de textura no domínio de wavelets é que a busca dos blocos passa a levar em conta também a estrutura da região, capturada pelos coeficientes de wavelet de detalhe [TON 2005, TON 2006]. O encaixe do novo bloco com a região que contém informação deve levar em consideração não apenas a comparação direta entre as respectivas cores/intensidades (coeficientes de aproximação), mas também devem visar a preservação das bordas e estruturas relevantes (coeficientes de detalhes) no encaixe. De fato, em [TON 2005, TON 2006] os autores salientam que uma métrica de síntese de texturas que verifique a soma dos erros quadrados apenas em espaço RGB (métrica  $L^2$  aplicada a cada canal de cor) não é suficiente para considerar a estrutura dos blocos de imagem analisados. A inserção dos coeficientes wavelet de detalhe melhoram esta métrica, criando um mecanismo altamente apropriado para propagação de blocos de imagens em inpainting digital. Neste trabalho, é utilizada uma técnica de síntese de texturas similar à proposta em [TON 2005, TON 2006], mas com algumas diferenças que serão explicitadas a seguir.

Tonietto *et al.* definem também de zonas de fronteira retangulares (ou em formato de ‘L’), situadas nas intersecções do bloco a ser encaixado e a região já preenchida, que são utilizadas para verificar o encaixe (figura 4.16). Neste trabalho, a região de inpainting pode ter formatos variados, e então as zonas de fronteiras podem ter geometrias distintas. Salienta-se que todos os pixels na intersecção entre o bloco a ser encaixado e a região já completada são empregados na

métrica de similaridade neste trabalho. Ainda, os autores de [TON 2005, TON 2006] fornecem a possibilidade de aplicar a transformada wavelet 1D ou 2D na métrica de similaridade; na proposta deste trabalho, já dispõe-se dos coeficientes wavelets 2D em todos os pixels válidos (fora da região de inpainting), que serão utilizados para fazer o casamento dos blocos de textura.

Toma-se então como base a métrica de síntese de textura criada em [TON 2006], porém, utilizando os coeficientes de wavelet 2D já existentes e incorporando blocos de tamanho variável. A técnica desenvolvida é apresentada a seguir, e é separada em 3 estágios: definir o bloco base e área de busca, encontrar o bloco com melhor encaixe no bloco base, e completar a região de inpainting com o bloco encontrado.

A definição do bloco base inicia na determinação da prioridade de preenchimento, o que neste trabalho é feita utilizando o método misto descrito anteriormente. Com as coordenadas  $(x_0, y_0)$  (do pixel com maior prioridade de preenchimento) definidas, cria-se o bloco base  $\beta$  centrado em  $(x_0, y_0)$ , que é uma região quadrada de tamanho variável  $M \times M$ . Como o processo é realizado no domínio de wavelets, esta janela delimita os blocos base para os coeficientes de aproximação ( $C_a$ ) e de detalhes ( $C_h$ ,  $C_v$  e  $C_d$ ), obtendo-se no final 4 blocos base, como mostra a figura 4.17. O bloco base dos coeficientes de aproximação permitirá a métrica de síntese de texturas semelhante ao realizado no espaço RGB, e os blocos dos coeficientes de detalhe adicionarão a informação de estrutura da imagem à métrica.

A procura do bloco ideal para o preenchimento da região de inpainting leva em consideração os quatro blocos base, comparando-os com blocos situados na sua vizinhança (com métrica descrita abaixo). Essa vizinhança pode se estender à toda a imagem, mas como a informação mais adequada para o preenchimento provavelmente estará próxima da região de inpainting, é mais adequado reduzir a área de busca (diminuindo, assim, o custo computacional). No modelo criado, utilizamos uma região de **busca quadrada**  $\Theta$  de pelo menos 3 vezes o tamanho do bloco base ( $3M$ ), o que demonstrou conter informação suficiente para dar opções de busca, sendo suficiente para a geração de bons resultados (ver figura 4.18). Como  $M$  pode ser variável,  $\Theta$  deve ser sempre definido em função de  $M$ .

Resumidamente, a métrica para o encaixe dos blocos considera as diferenças entre os coeficientes wavelet do bloco candidato e do bloco a ser preenchido, restrita aos pixels não pertencentes à região de inpainting. Mais formalmente, sejam  $\beta^{dir}$  os coeficientes wavelet do bloco  $\beta$  (centrado em  $(x_0, y_0)$ ) de tamanho  $M \times M$  a ser preenchido, onde  $dir \in \{a, h, v, d\}$  representa o tipo de coeficiente. Seja  $\theta \subset \Theta$  o bloco candidato, também com dimensões  $M \times M$ ,

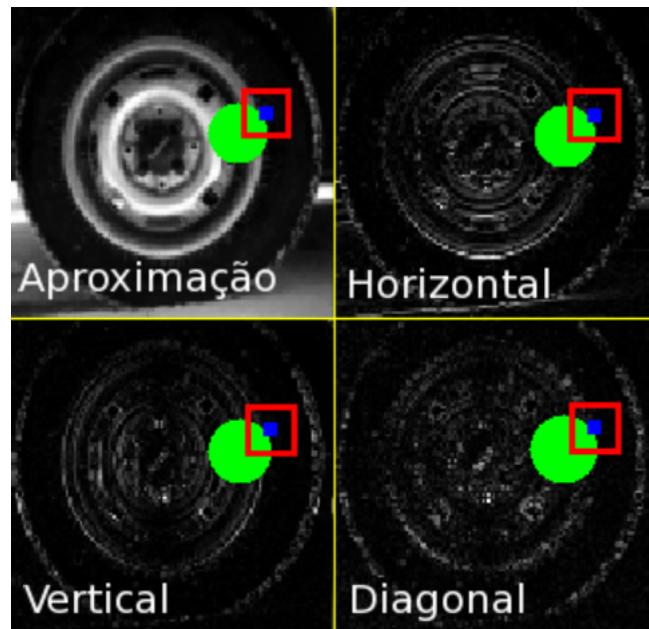


Figura 4.17: Determinação dos blocos base  $\beta$  (em vermelho), de tamanho  $M \times M$ , centrado em  $(x_0, y_0)$  (em azul) na fronteira da região de inpainting (em verde), para as quatro resultantes da decomposição wavelet.

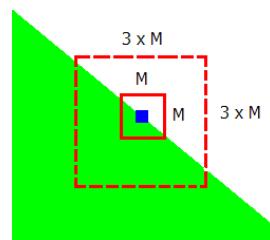


Figura 4.18: Região de busca (tracejado em vermelho) ao redor de um bloco base centrado no ponto  $(x_0, y_0)$  (em azul). O tamanho utilizado para a região de busca é de pelo menos 3 vezes as dimensões do bloco base.

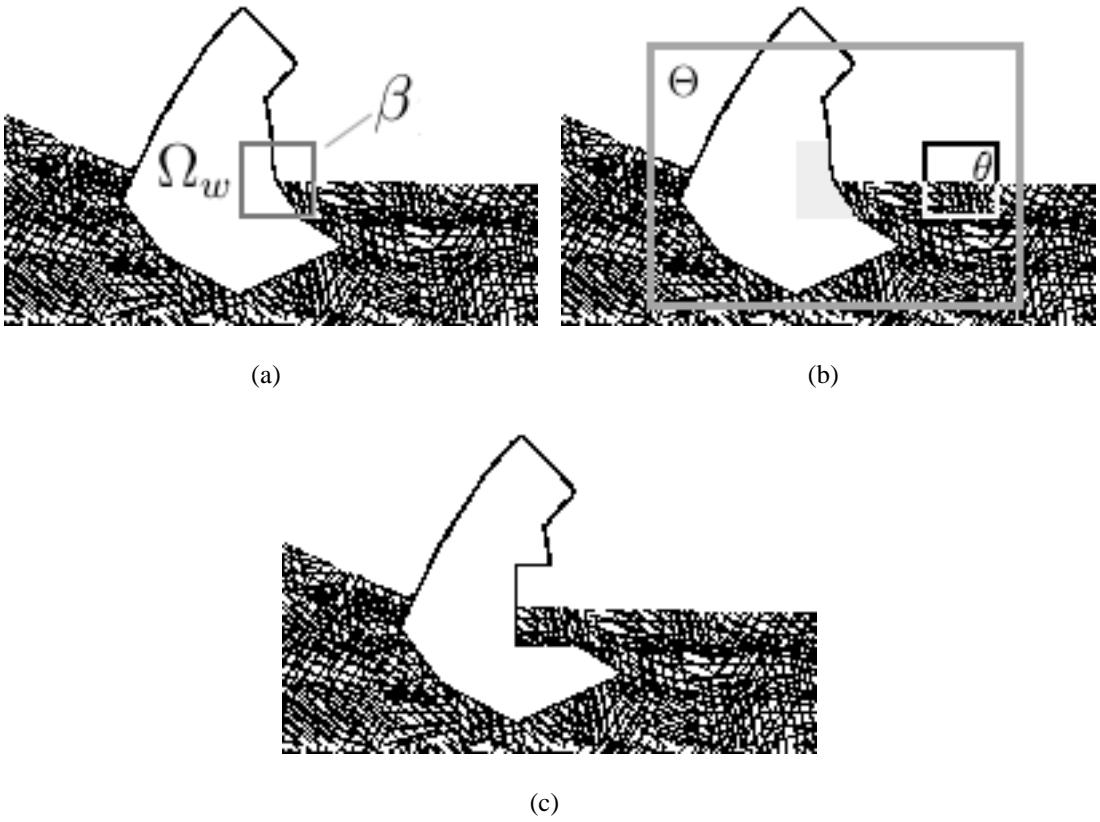


Figura 4.19: O preenchimento por blocos de  $\beta$  (a) é feito procurando um bloco  $\theta$  em uma vizinhança  $\Theta$  (b). O bloco encontrado é aquele que minimiza o erro entre os coeficientes de wavelet, e preenche apenas os pixels dentro da região de inpainting  $\Omega_w$  (c).

cujos coeficientes wavelet são dados por  $\theta^{dir}$ . A métrica proposta é a soma do erro quadrático nos 4 coeficientes wavelet:

$$err(\theta) = \sum_{dir \in \{a, h, v, d\}} \sum_{(x,y) \in \beta \cap \Omega_w^c} (\theta^{dir}(x,y) - \beta^{dir}(x,y))^2, \quad (4.12)$$

onde  $\beta \cap \Omega_w^c$  representa os pixels no bloco  $\beta$  que estão fora da região de inpainting (conseqüentemente, pertencem à região válida da imagem).

O bloco a ser inserido é aquele que minimiza o erro proposto na equação (4.12). Em outras palavras, o bloco  $\theta$  que deve preencher  $\beta$  é encontrado utilizando a seguinte equação:

$$\theta = \operatorname{argmin}_{l \in L} err(\theta_l), \quad (4.13)$$

onde  $l \in L$  representa todas as coordenadas possíveis de serem obtidas deslizando a janela  $M \times M$  em  $\Theta$  (de modo que  $\theta \subset \Theta$ ), e  $\theta_l$  é a respectiva janela. Uma vez determinado o bloco  $\theta$  que minimiza o erro, a região de inpainting em cada conjunto de coeficiente é preenchida separadamente, copiando a informação contida na posição onde se encontra  $\theta$  para os pixels de

$\beta$  que estejam dentro da região de inpainting (figura 4.19). Em outras palavras, o bloco  $\beta$  é substituído pelo bloco  $\beta'$ , calculado via:

$$(\beta')^{dir}(x, y) = \begin{cases} \beta^{dir}(x, y) & \text{se } (x, y) \in \beta \cap \Omega_w^c \\ \theta^{dir}(x, y) & \text{se } (x, y) \in \beta \cap \Omega_w \end{cases}. \quad (4.14)$$

Em alguns casos, a textura que deve ser sintetizada pode ter padrões bem específicos, em dimensões específicas. Um exemplo seria uma parede de tijolos, onde o padrão para a textura se repetiria em blocos do tamanho de cada tijolo. Para tomar proveito destes comportamentos, foi criado um recurso que consiste em variar a dimensão  $M$  de  $\beta$  (e consequentemente,  $\theta$ ) na equação (4.12), de forma que o erro  $err(\theta)$  - calculado com vários tamanhos de bloco - seja mínimo, e que o padrão da textura consiga ser melhor reproduzido. Este método considera que o tamanho do bloco  $\theta$  possa variar em uma única iteração na etapa de preenchimento da região de inpainting. Assim, erro  $err(\theta)$  deve ser calculado para estes diversos tamanhos de bloco dentro da janela  $L$  (equação (4.13)), não apenas considerando a posição de  $\theta$  em  $\Theta$ , mas também as dimensões de  $\theta$ . No entanto, quando  $M$  for pequeno, o erro também será pequeno por conter menos pixels no somatório. Logicamente quando  $M$  for grande, o erro será maior por conter mais pixels no somatório. Para corrigir este comportamento, criou-se uma forma de considerar o erro local por pixel utilizado, dividindo o erro final pelo número de pixels de  $\beta$  que efetivamente são utilizados no somatório. Consideramos as janelas de tamanho variável  $\beta^M$  e  $\theta^M$  com dimensões  $M \times M$ , onde  $M$  é variável, e o erro normalizado  $nerr(\theta^M)$  é calculado para cada valor de  $M$  (com  $M_{\min} \leq M \leq M_{\max}$ ) levando em consideração o número de pixels do somatório, como mostra a equação abaixo:

$$nerr(\theta^M) = \frac{err(\theta^M)}{|\beta^M \cap \Omega_w^c|}, \quad (4.15)$$

onde  $|\beta^M \cap \Omega_w^c|$  é o número de pixels de  $\beta^M$  utilizados no cálculo do erro. Assim, à medida que  $M$  aumenta, o erro é atenuado, pois o número de pixels válido também é maior. Com  $M$  pequeno, o número de pixels é menor, e o erro tende a manter-se perto do original. Esta solução com  $M$  variável não apenas permite que texturas com padrões bem definidos sejam sintetizadas gerando um bom resultado visual, mas também replica bem texturas sem padrão bem definido.

Na verdade,  $M$  deve assumir valores inteiros, sendo que  $M_{\min} \geq 3$ , e  $M_{\max}$  pode ser um valor arbitrariamente grande (até a dimensão total da imagem). O padrão utilizado neste modelo considera  $M$  entre 3 e 9, mas pode ser aberto à parametrização pelo usuário. De fato, a faixa de variação de  $M$  é o único parâmetro definido pelo usuário no modelo, o que não o

deixa menos automático pois seus valores pré-determinados conseguem adequar a síntese de texturas a diversos tipos de padrões. A figura 4.20 exemplifica como valores diferentes de  $M$  em uma mesma iteração do preenchimento dos blocos pode melhorar a qualidade do inpainting, por melhor identificar texturas com padrões bem definidos.

Após a região de inpainting ter sido totalmente preenchida nos coeficientes de detalhes e aproximação, a transformada wavelet inversa é aplicada, gerando a imagem final  $I'$  (passo 6 do pseudo-algoritmo na seção 4.1).

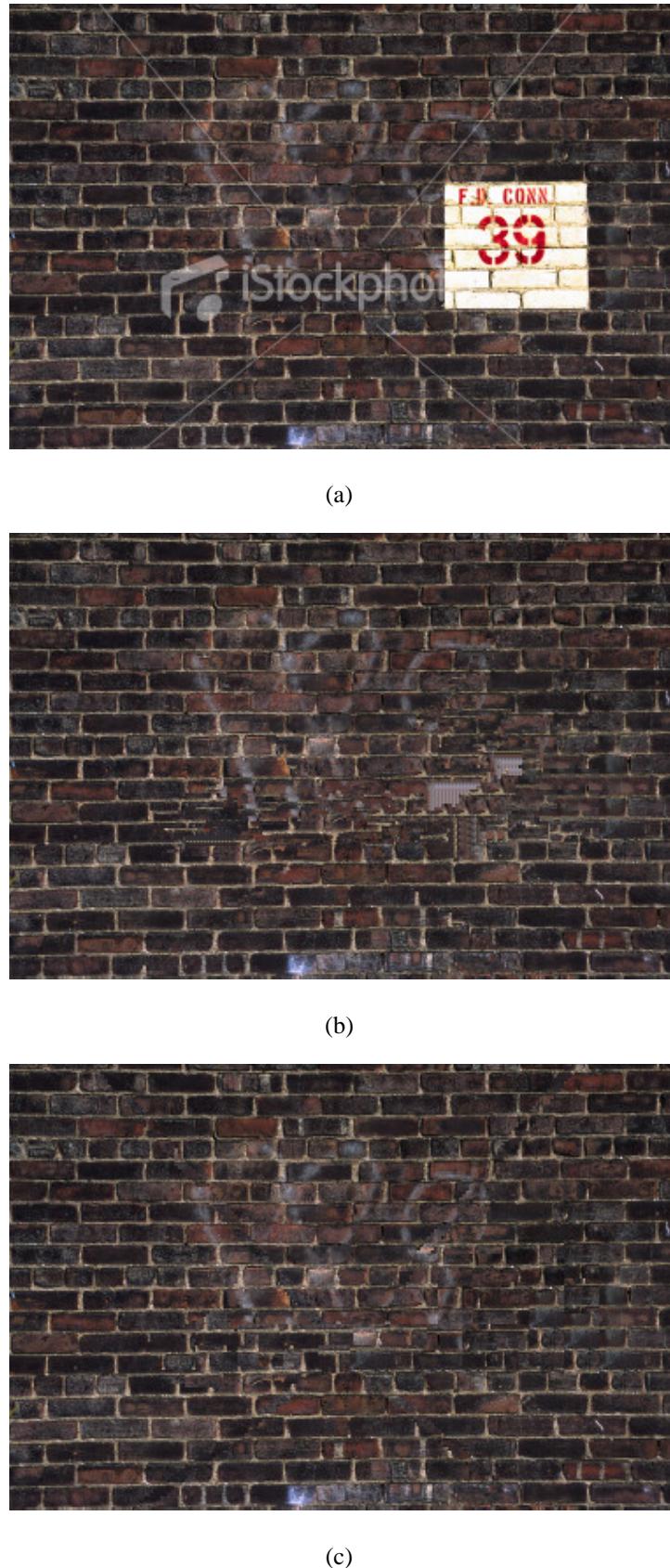


Figura 4.20: Resultados do preenchimento com tamanho de bloco fixo e variável. O inpainting é realizado figura (a), e o resultado da figura (b) foi gerado com um valor fixo de  $M = 3$ . Já o melhor resultado em (c) utiliza valores de  $M$  iniciando em 3 e crescendo em uma unidade até o valor 16.

# Capítulo 5

## Resultados Experimentais e Discussão

Este capítulo apresenta os resultados obtidos com a implementação do modelo criado neste trabalho, descrevendo maiores detalhes sobre resultados obtidos e discutindo as vantagens e oportunidades de melhoria do modelo. Também são apresentadas algumas comparações entre a técnica proposta e outros algoritmos competitivos para realização do inpainting digital disponíveis na literatura.

### 5.1 Implementação do Modelo de Inpainting Digital no Domínio de Wavelets

Um sistema protótipo foi construído, implementando o modelo descrito no capítulo 4, especializando o algoritmo apresentado, dividido em duas partes com dois objetivos distintos: inpainting digital de pequenas áreas (seção 4.2) e inpainting por blocos de pixels (seção 4.3). Este sistema foi implementado utilizando a ferramenta **MATLAB** (produzido pela companhia *The MathWorks*) e o *toolkit* de **Processamento de Sinais** foi utilizado apenas nas operações de transformada wavelet 2D e da transformada wavelet 2D inversa [GON 2003, MCA 2004]. Os resultados gerados pela implementação corroboram com sucesso os aspectos teóricos do modelo apresentado anteriormente.

As seções seguintes analisam os resultados obtidos nas implementações de inpainting digital em pequenas áreas e por blocos.

## 5.2 Resultados Preliminares do Inpainting em Pequenas Áreas

Os resultados apresentados na seção 4.2 demonstram a viabilidade da técnica, gerando bons resultados, mas podendo falhar em algumas situações. O motivo pelo qual a falha ocorre caracteriza uma oportunidade de melhoria para esta técnica.

Como é possível perceber na figura 4.10, a região de inpainting cruza em sentido diagonal a parte do telhado da casa que apresenta o problema, ilustrado no detalhe 4.11. Neste caso, o algoritmo não conseguiu propagar corretamente a estrutura naquela parte da imagem em virtude da escolha da máscara direcional, necessária para definir os pixels (coeficientes) que preencheram a área afetada (determinação da ordem de preenchimento na equação (4.2)). Neste caso, o problema está no fato de que a máscara direcional diagonal  $\gamma^d$  toma como válido os coeficientes em duas diagonais, como pode ser visto na matriz abaixo.

$$\gamma^d = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Desta forma, a máscara direcional  $\gamma^d$  considera duas vezes mais coeficientes do que as máscaras  $\gamma^h$  e  $\gamma^v$ . O algoritmo acaba sendo influenciado por este fator, e, além de escolher incorretamente a máscara direcional, pode utilizar pixels de uma diagonal para completar a informação que foi destacada na estrutura da outra diagonal, fazendo com que a cor interpolada não seja adequada. O formato da máscara direcional  $\gamma^d$  proposto está de acordo com o comportamento da transformada wavelet, que captura as informações das altas frequências da imagem em duas diagonais (coeficientes de detalhe diagonal). Porém, melhores resultados poderiam ser obtidos artificialmente separando os coeficientes de wavelet de detalhe diagonal, em dois conjuntos de detalhe, um para cada diagonal capturada. A utilização de duas máscaras direcionais diagonais  $\gamma_1^d$  e  $\gamma_2^d$ , uma para cada diagonal corrige a desigualdade entre o número de pixels válidos para o caso da energia, e também permitindo que a interpolação para o preenchimento utilize apenas os pixels da diagonal que realmente destaca a estrutura capturada nos coeficientes wavelet.

Ainda sobre o método de inpainting pixel-a-pixel, podemos observar que a reconstrução de grandes áreas não é uma aplicação que produza excelentes resultados. O algoritmo consegue reproduzir as estruturas faltantes com sucesso na maioria dos casos, mas não tem um bom desempenho na reconstrução de texturas, como pode ser visto na figura 5.1.



Figura 5.1: Reprodução de áreas com texturas. A região de inpainting (área escura da figura à esquerda) é reconstruída (figura à direita) pelo método pixel-a-pixel. Contudo, a textura não é sintetizada satisfatoriamente, exagerando as linhas da estrutura que se destaca na vizinhança da região de inpainting.

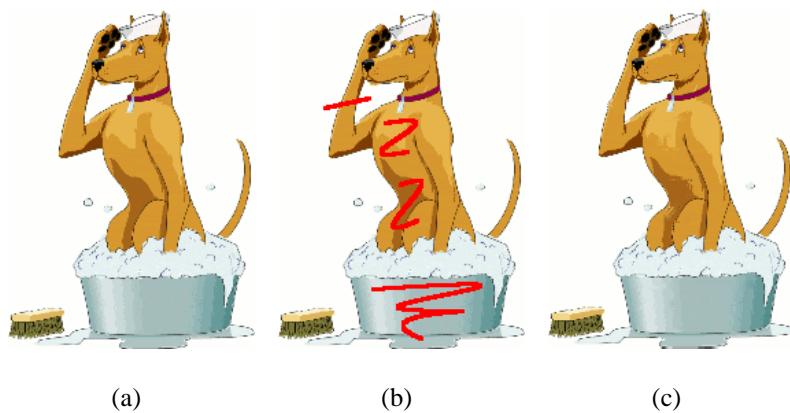


Figura 5.2: Aplicação da técnica de inpainting por pixel em uma imagem colorida. A figura (a) contém a imagem original, que artificialmente foi danificada, gerando uma máscara de inpainting (em vermelho na figura (b)), e o resultado recuperado pelo processamento do algoritmo em (c).

Embora os problemas decretos acima possam ocorrer, a técnica proposta de preenchimento pixel-a-pixel produz geralmente bons resultados quando a região de inpainting é pequena. Resultados de sucesso dessa técnica são ilustrados nas Figuras 5.2 e 5.3. Embora não seja o escopo do algoritmo pixel-a-pixel, ele pode ser aplicado em áreas de inpaitning um pouco maiores, desde que a imagem não apresente texturas mais complexas. Um exemplo disso é ilustrado na figura 5.4(b).

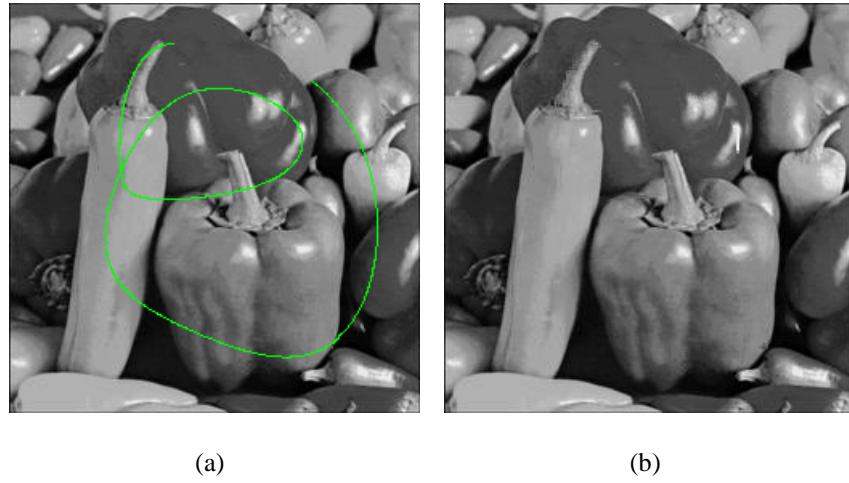


Figura 5.3: Restauração de uma imagem com o método pixel-a-pixel. A figura (a) destaca a região danificada em verde, que é restaurada e apresentada em (b).

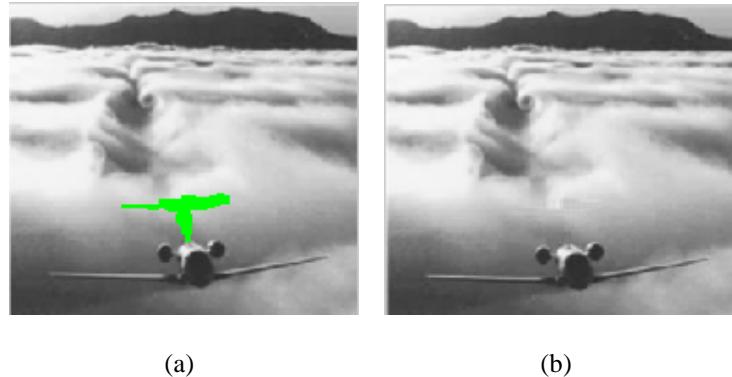


Figura 5.4: Desocclusão pelo método pixel-a-pixel. A região a ser removida é representada na figura (a) em verde, e a figura em (b) após processada, com a região removida.

Um exemplo da evolução temporal da propagação de estruturas pode ser visto na figura 5.5. Neste exemplo, a região de alta freqüência - neste caso causada pela mudança de intensidade (ou seja, borda) - gera coeficientes de wavelet com altas energias, sendo preenchida com maior prioridade.

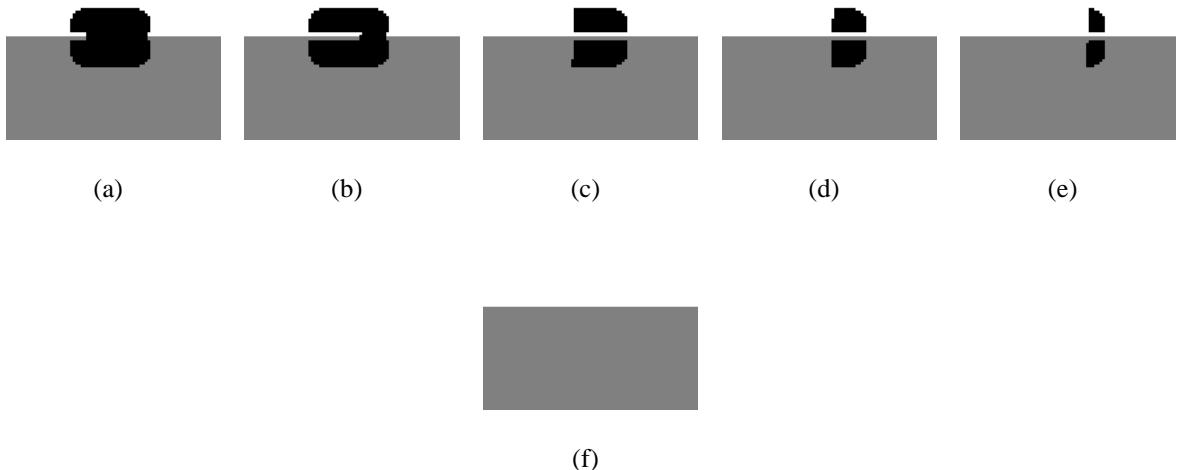


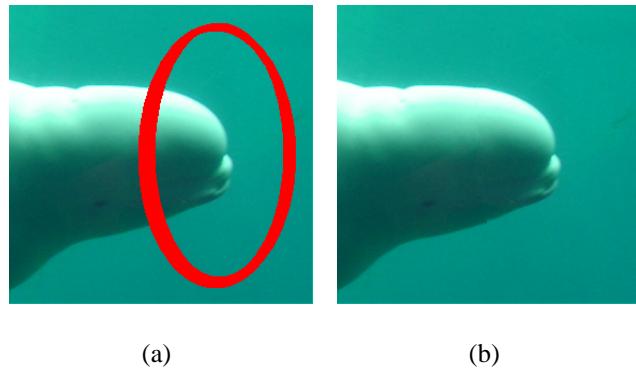
Figura 5.5: As figuras de (a) até (e) ilustram as etapas do inpainting por pixel e a imagem final reconstruída em (f). A área onde os coeficientes de wavelet possuem energias mais altas são preenchidos com maior prioridade.

### 5.3 Resultados Preliminares do Inpainting por Blocos

O objetivo principal da técnica de inpainting por blocos é conseguir preencher a região de inpainting sintetizando no seu interior as texturas vizinhas de forma mais eficiente, e que gere resultados visuais com poucos artefatos. O modelo descrito na seção 4.3 foi implementado, utilizando o método misto para determinação de prioridade, e síntese de texturas no domínio de wavelets, descritos nas seções 4.3.1 e 4.3.2, respectivamente.

Este método funciona adequadamente para preencher tanto imagens monocromáticas quanto coloridas. O destaque deste mecanismo é boa qualidade visual das regiões de inpainting que necessitam ser preenchidas com texturas complexas (figura 5.7), mas também funciona muito bem quando aplicado a texturas mais simples (figura 5.6). Todos os resultados apresentados a seguir utilizam a estratégia de bloco com dimensão  $M$  variável, como descrito na seção 4.3.2

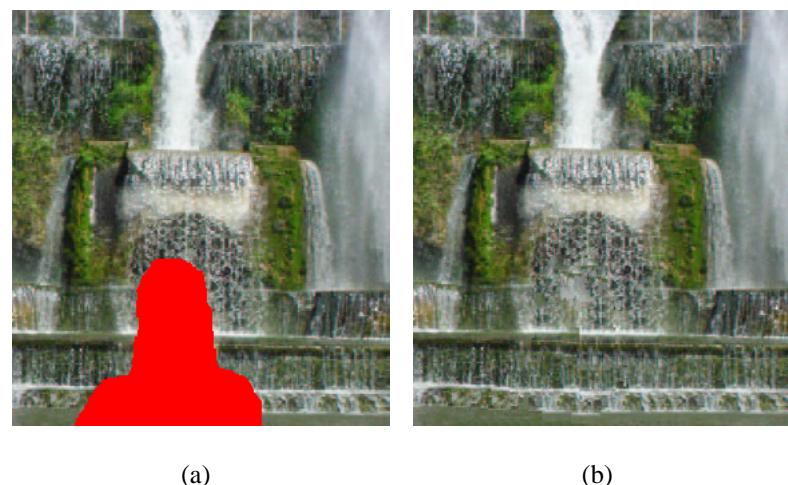
O inpainting digital pode ser aplicado com diversos objetivos, e entre os mais comuns estão a recuperação de danos em imagens e desoclução de áreas (ou remoção de objetos). Exemplos de remoção de objetos utilizando a técnica proposta são ilustrados na figuras 5.3, 5.9 e 5.10. Nesses resultados, pode-se ver o funcionamento do modelo proposto, preenchendo tanto pequenas como largas áreas de uma imagem, seja ela colorida ou monocromática. A qualidade visual das imagens geradas mostra que é possível remover objetos pequenos e grandes utilizando a



(a)

(b)

Figura 5.6: Inpainting por blocos sintetizando texturas simples. A figura (a) destaca a máscara de inpainting em vermelho, que é preenchida, gerando o resultado apresentado em (b). A textura ao redor da região de inpainting não é muito complexa, possui uma cor predominante e quase nenhuma estrutura.



(a)

(b)

Figura 5.7: Inpainting por blocos sintetizando texturas mais complexas, onde o preenchimento da região de inpainting (em vermelho na figura (a)) sintetiza a textura da vizinhança em seu interior (b).

mesma implementação do método de inpainting por blocos. Em particular, a figura 5.8(f) mostra como é importante ter uma boa síntese de textura para preencher largas áreas, preservando simultaneamente a estrutura da imagem. Um exemplo clássico de desocclusão aparece na figura 5.11, onde um texto que foi inserido artificialmente em uma fotografia digital foi removido pelo algoritmo desenvolvido.

Outra aplicação importante do inpainting digital é a capacidade de corrigir falhas em imagens. O modelo desenvolvido neste trabalho é capaz tanto de remover objetos quanto corrigir falhas em imagens, sem a necessidade de algoritmos diferenciados ou configuração de parâmetros específicos. A correção de falhas pode ser aplicada quando o que foi danificado foi no nível digital (pixels perdidos e/ou corrompidos como a figura 5.12), quando o dano está em uma fotografia em papel (foto rasgada como na figura 5.14) que posteriormente foi digitalizada ou ainda quando uma foto foi tirada de algo que foi danificado, como ilustrado na figura 5.13.

A manipulação da imagem na figura 5.13 simula a remoção das manchas sobre uma superfície real. Naturalmente, o processamento é feito sobre os pixels de uma fotografia no formato digital. Desta forma, erros ou partes indesejadas em fotografias podem ser corrigidos automaticamente após já terem sido digitalizadas. Neste caso específico, uma situação indesejada do cenário real (manchas de gotas sobre uma superfície) foi corrigida.

Como citado no capítulo anterior, o tamanho do bloco a ser preenchido, de dimensões  $M \times M$  pode ser variável à cada iteração do algoritmo de inpainting, o que proporciona melhores resultados em suas saídas. Submeteu-se a imagem com falha na figura 5.14 - de resolução  $283 \times 292$  ao algoritmo de inpainting, porém, utilizando tamanho de  $M$  fixo durante todo o processo de preenchimento. Na figura 5.15(a) utilizou-se  $M = 3$ , um valor pequeno. É possível perceber que a síntese das texturas fica comprometida, gerando diversos artefatos. Blocos de tamanho  $3 \times 3$  neste caso, não são suficientemente grandes para capturar a informação de textura necessária para um bom preenchimento. Já na figura 5.15(b) utilizou-se  $M = 16$  um valor grande para a figura. Foi possível perceber que, apesar da textura da imagem ter sido melhor reproduzida do que utilizando um valor de  $M$  pequeno, a estrutura em curva da imagem foi prejudicada. Os grandes blocos não preenchem corretamente as partes curvas da imagem, por não conseguir capturar seus pequenos detalhes. O melhor resultado foi adquirido utilizando valores de  $M$  que variaram entre 3 e 16, com incrementos de 1 unidade a cada iteração. Desta forma, o próprio algoritmo conseguiu escolher os menores blocos que preencheriam melhor as regiões curvilíneas, e os blocos mais largos para o preenchimento das complexas texturas, como

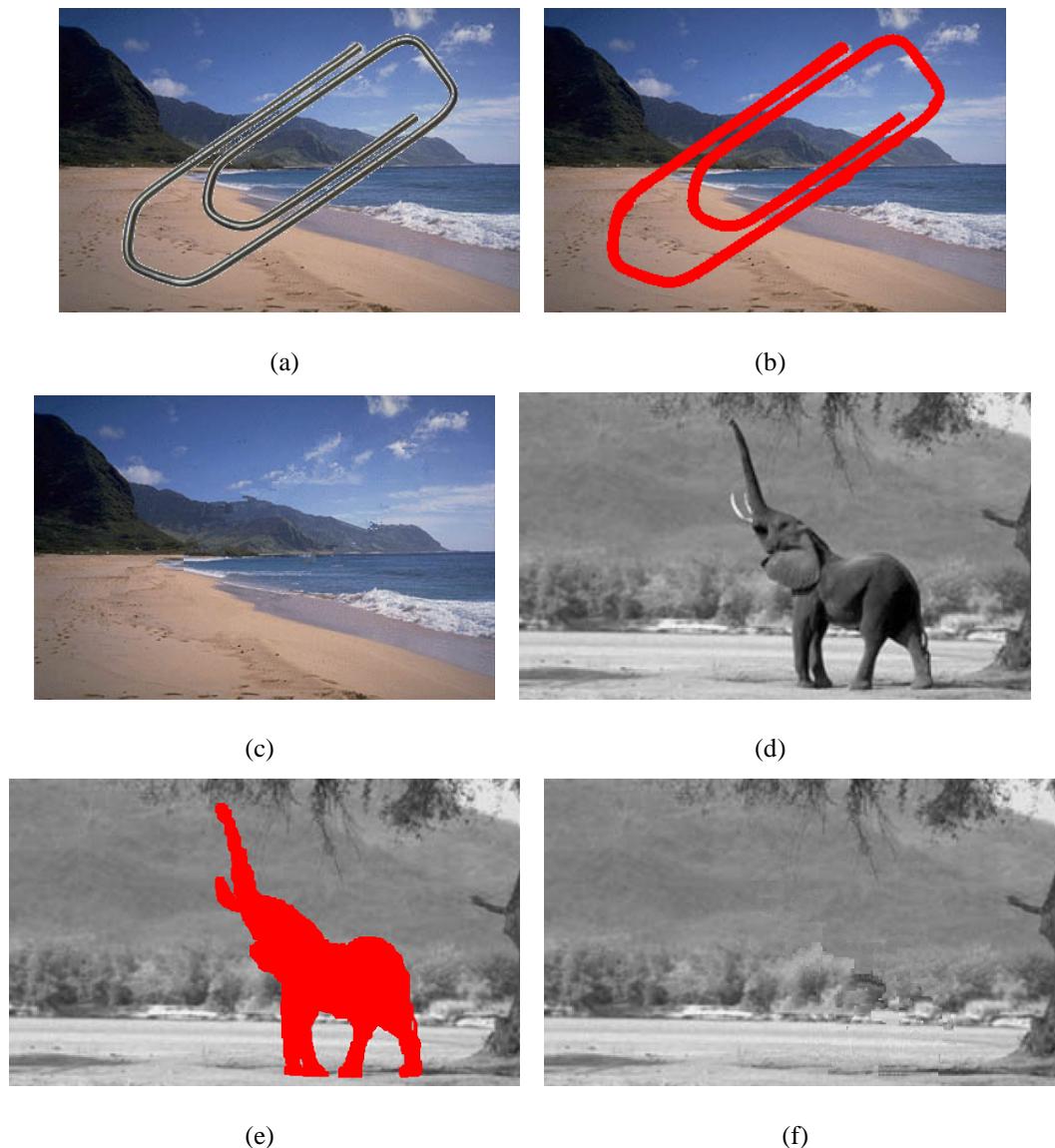


Figura 5.8: Desoclução utilizando o modelo proposto. O algoritmo implementado funciona bem tanto para imagens coloridas (figuras a,b,c) quanto para monocromáticas (d,e,f). Pequenas regiões de inpainting (b) (c) podem ser completadas satisfatoriamente, e o mecanismo de síntese de texturas garante que o preenchimento de largas áreas (e) (f) seja feito de maneira correta.



Figura 5.9: Remoção de objetos uma imagem colorida, com largas regiões de inpainting. A placa na figura da esquerda é removida e o algoritmo sintetiza blocos de imagem, reconstruindo a região que está atrás da placa.



Figura 5.10: A remoção de objetos em imagens funciona muito bem também para regiões de inpainting pequenas e rodeadas por texturas simples.



Figura 5.11: Remoção de texto em imagens digitais, onde o texto inserido artificialmente na figura da esquerda, foi removido pelo algoritmo, e o resultado apresentado na imagem da direita.

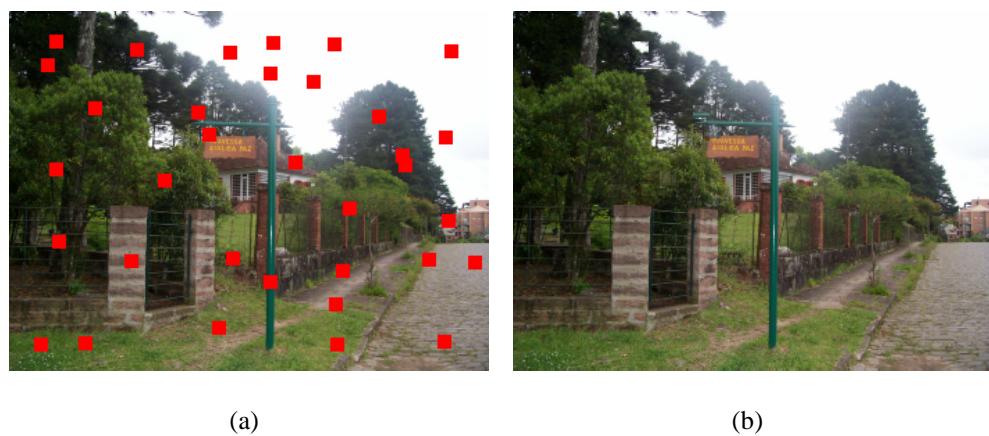


Figura 5.12: A imagem (a) ilustra uma fotografia onde blocos de pixels foram corrompidos de alguma forma (em vermelho), posteriormente corrigidos (b) pelo algoritmo proposto neste trabalho.

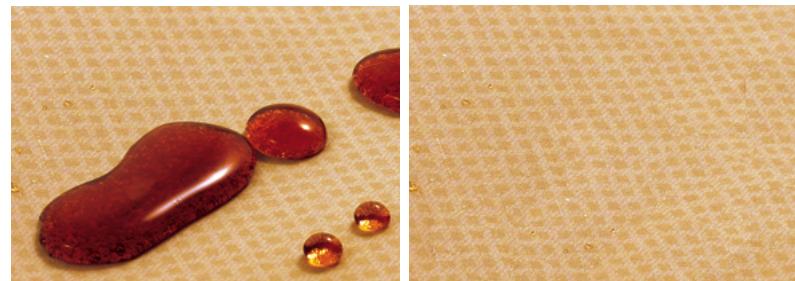


Figura 5.13: Correção de falhas indesejadas em uma cena fotografada

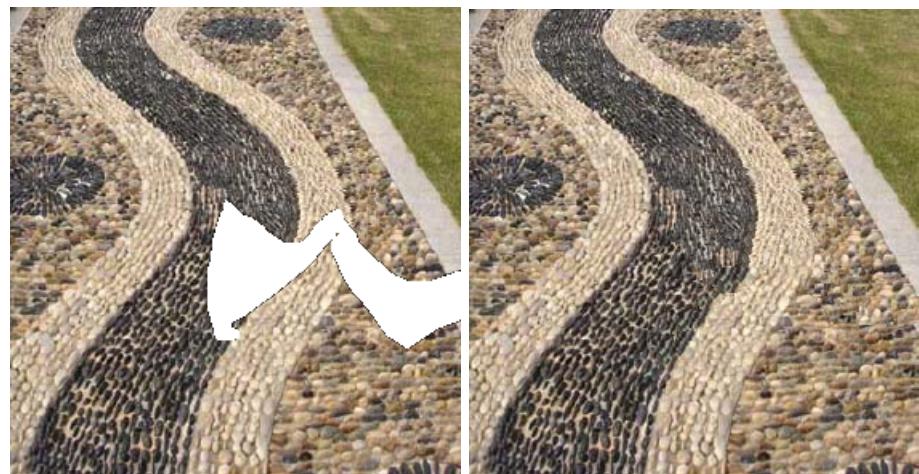


Figura 5.14: Correção das falhas físicas de uma fotografia posteriormente digitalizada

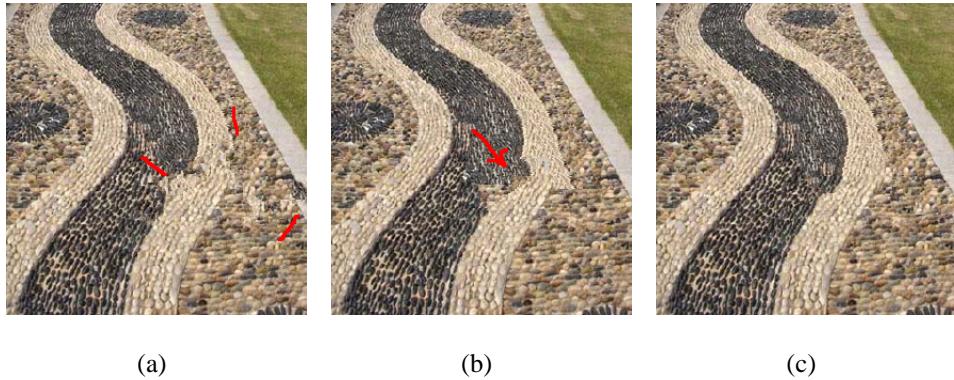


Figura 5.15: A imagem (a) foi gerada com bloco de tamanho fixo  $M = 3$ , criando artefatos (indicados em vermelho) nas texturas reproduzidas. Em (b) o valor de  $M$  que multiplica  $M$  é alto (bloco de tamanho 16), e as estruturas curvilíneas da imagem não foram bem reconstruídas. O resultado melhorado da figura (c) utiliza valores de  $M$  que sequencialmente geram blocos com dimensão variando de 3 a 16.

pode ser visto na figura 5.15(c).

Como o conteúdo da região de inpainting não existe, o algoritmo que o preenche apenas utiliza a informação ao redor dela para completar o conteúdo. Embora o preenchimento objetive a maior coerência visual possível (manutenção de estrutura e textura), o resultado produzido pelo algoritmo pode não ser realizado da mesma forma que uma pessoa faria manualmente. Por exemplo, o resultado mostrado na figura 5.16 ilustra a remoção do elefante na imagem. O detalhe na figura 5.16(c) mostra uma região com uma reentrância na margem do rio, que embora pareça natural ao olho humano, provavelmente não seria inserida em um inpainting manual.

## 5.4 Avaliação Quantitativa

A análise dos resultados das figuras geradas por inpainting digital, onde há remoção de objetos, geralmente são feitas apenas por inspeção visual. No entanto, é possível medir quantitativamente o erro cometido no completamento da imagem em situações controladas, desde que se conheça explicitamente a imagem original. Para medir o erro, utilizou-se neste trabalho a razão-sinal-ruído-de-pico (PSNR, ou *Peak Signal-to-Noise Ratio*) [HAY 2002], bastante comum para avaliar técnicas de remoção de ruído e compressão de imagens. Partindo do mesmo conceito, foram criados imagens teste com a intenção de medir o PSNR. O procedimento executado mantém a imagem original intacta, cria manualmente falhas em uma cópia desta imagem e

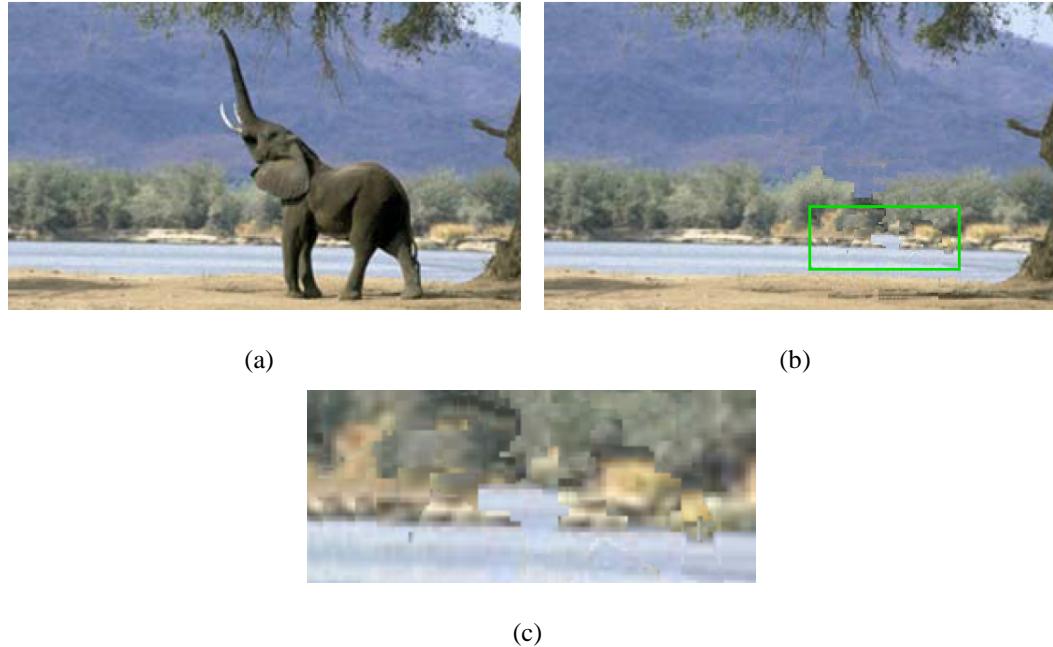


Figura 5.16: Detalhe do preenchimento por blocos, a imagem original (a) submetida a remoção de objetos pelo algoritmo de inpainting desenvolvido neste trabalho (b). A figura (c) demonstra o detalhe da borda do rio, gerado automaticamente pelo processo de preenchimento com síntese de texturas.

aplica o inpainting. A imagem resultante  $B$  e a imagem original  $A$  formam sinais 2D cujo erro médio quadrado (MSE, ou *Mean Squared Error*) é medido da seguinte forma:

$$\text{MSE} = \frac{1}{LC} \sum_{x=0}^{L-1} \sum_{y=0}^{C-1} (A(x, y) - B(x, y))^2$$

onde  $L$  e  $C$  são as dimensões de  $A$  e  $B$  (altura e largura devem ser iguais para ambas). O PSNR entre  $A$  e  $B$  é calculado utilizando o MSE entre ambas, através da equação:

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right),$$

onde 255 é o valor máximo utilizado para cada canal de cor, ou pixel em escala de cinza, e o valor do PSNR é dado em decibéis. Quanto maior o valor do PSNR obtido, mais a imagem gerada (neste caso  $B$ ) se aproxima da original ( $A$ ). Se ambas forem idênticas, o valor do PSNR será infinito.

Comparou-se então o resultado do inpainting em três imagens, cada uma submetida a três técnicas de inpainting: a implementação realizada neste trabalho, uma implementação do algoritmo desenvolvido por Criminisi *et al* [CRI 2004] e a técnica de Tschumperle e Deriche

[TSC 2005]. Os resultados quantitativos podem ser vistos na tabela 5.1, e uma análise visual pode ser feita nas imagens da figura 5.17.

Figura/Método	Wavelets (dB)	Criminisi (dB)	Tschumperle (dB)
Orca	51,5	51,3	43,3
Zebra	39,7	38,1	34,9
Calçada	45,5	43,7	24,1
Lagoa	45,8	24,1	48,8

Tabela 5.1: Comparação do valor do PSNR (em dB) aplicado sobre resultados de várias técnicas. A coluna **Wavelets** mostra os valores do PSNR para as imagens processadas com a técnica desenvolvida neste trabalho. As colunas **Criminisi** e **Tschumperle** mostram os valores de PSNR com imagens processadas utilizando as técnicas apresentadas em [CRI 2004] e [TSC 2005], respectivamente.

É possível observar na tabela acima que para as imagens testadas, o valor do PSNR calculado para os resultados gerados com o algoritmo aqui proposto é maior do que os concorrentes na maioria das ocasiões. Algumas vezes este valor é apenas ligeiramente superior aos comparados, como por exemplo, os resultados medidos sobre a figura denominada *Zebra* submetida a técnica de inpainting no domínio de wavelets e também à técnica de Criminisi *et al.*. Contudo é possível observar nas figuras em 5.17 que, apesar da diferença do PSNR medido ser pequena, a imagem *Zebra* na coluna **Wavelets** é visivelmente bem mais semelhante à original do que a imagem na coluna **Criminisi**.

O fato do PSNR nem sempre expressar a melhor qualidade visual de uma imagem em comparação com outras é bastante conhecido, e é um problema em aberto. Um exemplo que destaca esta situação pode ser visto em detalhe na figura 5.4, onde percebe-se que, apesar de seu PSNR ser melhor (figura **Lagoa** na tabela 5.1), a imagem 5.18(d) - modelo proposto - possui maior nitidez na região reconstruída, sendo mais aparentemente semelhante à mesma na imagem original 5.18(a) (ainda há um excesso de *aliasing*). Já esta região no resultado da imagem 5.18(c) - técnica em [TSC 2005] é de uma suavização acentuada, gerando borrões na imagem reconstruída.

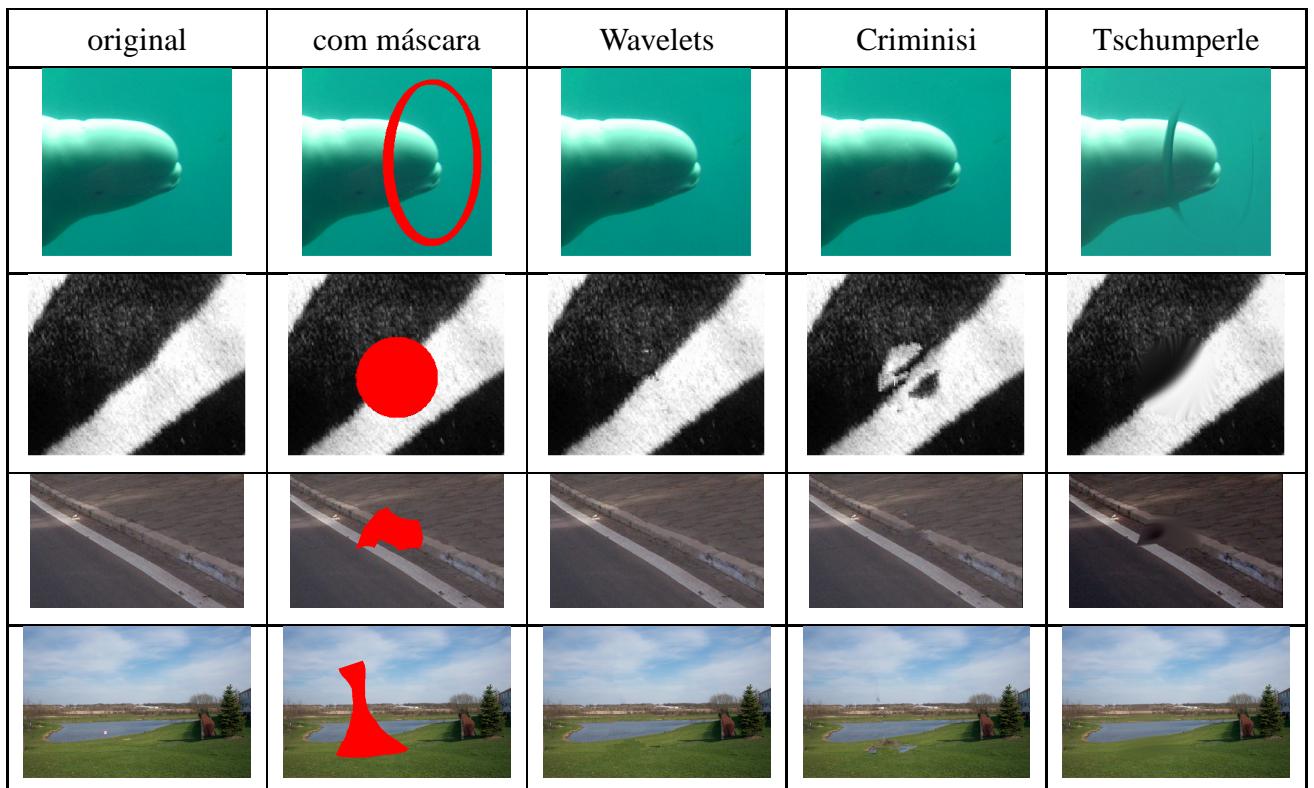


Figura 5.17: Imagens submetidas a diversas técnicas de inpainting digital. As primeiras duas colunas apresentam respectivamente a imagem original e a imagem com a região de inpainting a ser trabalhada. A coluna **Wavelets** mostra as imagens resultantes do inpainting digital desenvolvido neste trabalho. As colunas **Criminisi** e **Tschumperle** mostram as imagens processadas utilizando as técnicas em [CRI 2004] e [TSC 2005], respectivamente.

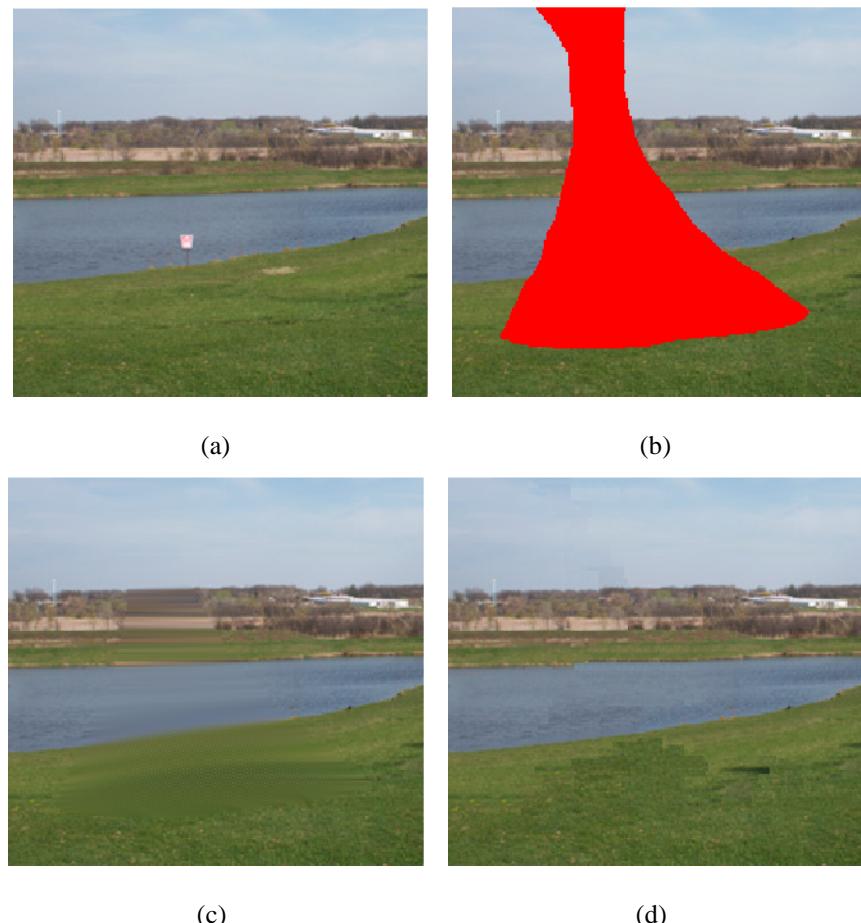


Figura 5.18: Relação entre a imagem gerada e seu PSNR. A figura original (a) foi artificialmente danificada (b) e submetida à técnica de inpainting de [TSC 2005] (c) e a técnica desenvolvida neste trabalho (d). Apesar do PSNR ser maior com o algoritmo de [TSC 2005], a imagem (d) assemelha-se mais com a original (a).

# Capítulo 6

## Conclusões e Trabalhos Futuros

Este trabalho apresentou um modelo para inpainting digital no domínio de wavelets. Foi realizada uma boa cobertura da bibliografia disponível, especialmente na aquisição de informações sobre as (escassas) técnicas de inpainting existentes que atuem no domínio de wavelets, e contribuindo com material bibliográfico para este tipo de pesquisa.

Um modelo genérico foi desenvolvido, destacando os pontos chave para uma boa implementação de inpainting digital. Apesar de ter o foco no uso da transformada wavelet, o modelo básico pode ser utilizado para o desenvolvimento de outras técnicas de inpainting digital. A elucidação das etapas de manutenção da estrutura da imagem e preenchimento coerente da região de inpainting são rotinas importantes no algoritmo de inpaintiting digital, e foi apresentada na seção 4.1.

O modelo básico foi especializado em duas técnicas: a primeira, formulada mais no início do trabalho, realiza inpainting um pixel por iteração; uma segunda técnica mais elaborada, desenvolvida posteriormente, preenche as regiões por blocos, utilizando para tal síntese de texturas. O método de preenchimento pixel-a-pixel mostrou-se adequado para regiões de inpainting pequenas (finas), e utiliza os coeficientes de detalhe da transformada wavelet como indicadores de como o preenchimento da região de inpainting deve ser feito, visando preservar sua estrutura. Já o mecanismo por blocos incorpora informações sobre a geometria da região de inpatining, utilizando o vetor gradiente combinado com a energia dos coeficientes de wavelet para determinar a ordem de preenchimento, visando manter a estrutura da imagem. O preenchimento em si é realizado utilizando uma técnica de síntese de texturas no domínio de wavelet que toma vantagem da informação da estrutura contida nos coeficientes de detalhe, juntamente com as informações de cor fornecidos pela imagem aproximação. Ambos mecanismos geram bons

resultados tanto para imagens em escala de cinza quanto para imagens coloridas RGB.

Uma contribuição importante deste trabalho foi a criação de um modelo que realiza inpainting digital totalmente em domínio de wavelets, conseguindo neste mesmo domínio descrever como realizar o preenchimento de largas áreas com síntese de texturas, juntamente com a identificação da correta geometria da imagem. O preenchimento com síntese de texturas proposto foi realizado utilizando uma adaptação de técnicas modernas de síntese de texturas no domínio de wavelet, obtendo resultados comparáveis ou superiores a outras técnicas de inpainting digital consideradas estado da arte. O mecanismo de síntese de texturas com blocos de tamanho variável é importante ponto de contribuição para a própria disciplina de síntese de texturas em domínio de wavelet, e também para a tecnologia de inpainting digital.

Embora os resultados experimentais apresentados fossem considerados bons, houve casos em que a técnica proposta não produziu o resultado esperado. De fato, há perspectivas de melhoria deste trabalho, que serão apresentadas a seguir.

## 6.1 Extensões Futuras para o Modelo

Os resultados apresentados no capítulo anterior demonstram o funcionamento do modelo proposto de inpainting digital no domínio de wavelet, com bons resultados e pontos 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 que melhor preenche uma região no inpainting por blocos.

A utilização da direção do gradiente para a busca do bloco  $\theta$  ideal para o preenchimento em cada iteração pode auxiliar na diminuição do tempo computacional desta. A idéia consiste em restringir a região de busca  $\Theta$  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, pois é o que indica o seu gradiente.

Já outras melhorias podem ser mais elaboradas, como a síntese das texturas no inpainting por blocos. Apesar da síntese de textura por exemplo ser suficiente na maioria dos casos, o futuro do modelo caminha para uma abordagem que utiliza mecanismos de síntese de texturas cada vez mais avançados e complexos. Faz-se necessário prover melhores formas para encaixar

os padrões de texturas (*texels*), bem como fazer a seleção correta do melhor bloco que preencherá o conteúdo da região de inpainting. Técnicas de síntese de texturas que utilizem processamento em GPU [LEF 2005] ou que considerem outros atributos dos texels (como luminância e radiância) [LEF 2006], combinadas com a teoria de wavelets, podem permitir incrementar significantemente a qualidade visual das áreas com texturas complexas sintetizadas.

Um trabalho futuro voltado a melhorar o encaixe entre os blocos de textura pode ser desenvolvido utilizando a técnica do **corte médio** [WU 93]. Nessa técnica, os pixels que estão na região de fronteira dos blocos encaixados têm seus valores ajustados com uma função de minimização, objetivando criar uma maior suavidade visual na intersecção dos blocos, mas sem causar borramento. O corte médio deve ser realizado na etapa de preenchimento dos pixels na região de inpainting, onde o bloco  $\theta$  escolhido preenche o conteúdo do alvo  $\beta$ .

O modelo proposto possui poucas variáveis de ajuste do seu comportamento, pois intende ser o mais automático possível. Contudo, alguns parâmetros que possam ser ajustados pelo usuário podem guiar o algoritmo na solução de problemas particulares. Um ponto interessante para a inserção de parâmetros ajustáveis é na busca do bloco que preencherá a região de inpainting. A equação (4.12) determina o erro entre os blocos nos coeficientes de wavelet, e todos os conjuntos de coeficientes têm peso equivalente. Em outras palavras, os erros cometidos na imagem de aproximação têm o mesmo peso dos erros das imagens de detalhe. Isso pode ser modificado, pela inserção de pesos diferenciados para coeficientes de aproximação e detalhe, gerando um comprometimento entre preservação de intensidades (cores) e a preservação de estruturas (bordas).

A integração entre os mecanismos de inpainting pixel-a-pixel e inpainting por blocos pode construir um mecanismo poderoso, capaz de obter o melhor das duas técnicas. Para isto é preciso determinar qual método será utilizado, baseado na dimensão da região de inpainting a ser trabalhada. Por exemplo, em regiões de inpainting “finas”, utiliza-se o método pixel-a-pixel, e em regiões largas, utiliza-se o método por blocos. O critério de seleção deve ser tema de estudo, mas ferramentas de morfologia matemática [GON 2001] podem ser aplicadas para diferenciar regiões finas de outras mais largas. Esse modelo adaptativo poderá trazer, além de melhores resultados visuais, um desempenho melhor, pois o processamento computacional é direcionado para as situações onde este pode ser ótimo.

Para a preservação eficiente da estrutura, o desenvolvimento de um mecanismo que identifique a estrutura da imagem no domínio de wavelet e que a partir desta crie linhas que guiem o

preenchimento (semelhante ao modelo de esqueleto de imagem em [GUO 2003]) pode ser um recurso valioso. A idéia básica do funcionamento é utilizar o vetor gradiente para reconstruir os coeficientes de detalhe, primeiramente combinando as três direções em um único mapa que serviria como indicador para o processo seguinte, que sintetiza as texturas no interior da região de inpainting. O objetivo é evitar tendências direcionais criadas pelos coeficientes de wavelet, ao mesmo tempo que mantém a estrutura da região coerente com a imagem original.

Essas são apenas algumas das melhorias possíveis, sendo as mais importantes e promissoras as que envolvem mecanismos novos e mais sofisticados de síntese de texturas, combinadas com novas técnicas para a manutenção da estrutura da imagem. Também, a combinação de técnicas mistas, que se adaptam a região de inpainting, podem permitir acelerar bastante o processamento e gerar imagens reconstruídas com grande qualidade visual.

# Bibliografia

- [ASH 2001] ASHIKHHMIN, M. Synthesizing natural textures. In: SI3D '01: PROCEEDINGS OF THE 2001 SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, 2001, New York, NY, USA. **Anais...** ACM Press, 2001. p.217–226.
- [BAR 2006] BARCELOS, C. A. Z.; BATISTA, M. A. Image restoration using digital inpainting and noise removal. In: IMAGE AND VISION COMPUTING, 2006, USA. **Anais...** Elsevier, 2006. p.1–9.
- [BEL 57] BELLMAN, R. E. **Dynamic programming**. Princeton, NJ.: Princeton University Press, 1957.
- [BER 2001] BERTALMIO, M.; BERTOZZI, A.; SAPIRO, G. Navier-stokes, fluid dynamics, and image and video inpainting. In: PROC. IEEE COMPUTER VISION AND PATTERN RECOGNITION, 2001, Hawaii. **Anais...** [S.l.: s.n.], 2001. p.I:355–362.
- [BER 2000] BERTALMIO, M. et al. Image inpainting. In: ACM COMPUTER GRAPHICS PROCEEDINGS (SIGGRAPH), 2000. **Anais...** ACM Press, 2000. p.417–424.
- [BER 2003] BERTALMIO, M. et al. Simultaneous structure and texture image inpainting. In: IEEE TRANSACTIONS ON IMAGE PROCESSING, 2003. **Anais...** [S.l.: s.n.], 2003. v.12, p.882–889.
- [CHA 2001] CHAN, T. F.; SHEN, J.; ZHOU, H.-M. **Total variation wavelet inpainting**. Los Angeles: UCLA Comp. Appl. Math. (CAM) Tech. Report, 2001.
- [CHA 2002] CHAN, T. F.; SHEN, J. Mathematical models for local nontexture inpaintings. **SIAM Journal on Applied Mathematics**, v.62, n.3, p.1019–1043, 2002.

- [CRI 2004] CRIMINISI, A.; PEREZ, P.; TOYAMA, K. Region filling and object removal by exemplar-based image inpainting. **IEEE Transactions on Image Processing**, v.13, n.9, p.1200–1212, September 2004.
- [EFR 2001] EFROS, A. A.; FREEMAN, W. T. Image quilting for texture synthesis and transfer. **Proceedings of SIGGRAPH 2001**, p.341–346, August 2001.
- [EFR 99] EFROS, A. A.; LEUNG, T. K. Texture synthesis by non-parametric sampling. In: IEEE INT. COMPUTER VISION, 1999, Corfu, Greece. **Anais...** [S.l.: s.n.], 1999. p.1033–1038.
- [FOL 2000] FOLEY, J. D. **Computer graphics**: principles and practice. Reading, MA: Addison-Wesley, 2000.
- [GON 2003] GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. L. **Digital image processing using matlab**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.
- [GON 2001] GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.
- [GUO 2003] GUO, C. en; ZHU, S.-C.; WU, Y. N. Towards a mathematical theory of primal sketch and sketchability. In: ICCV '03: PROCEEDINGS OF THE NINTH IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2003, Washington, DC, USA. **Anais...** IEEE Computer Society, 2003. p.1228.
- [HAY 2002] HAYKIN, S.; VAN VEEN, B. **Sinais e sistemas**. Porto Alegre, RS: Bookman, 2002.
- [HER 2001] HERTZMANN, A. et al. Image analogies. In: SIGGRAPH '01: PROCEEDINGS OF THE 28TH ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 2001, New York, NY, USA. **Anais...** ACM Press, 2001. p.327–340.
- [HIR 96] HIRANI, A. N.; TOTSUKA, T. Combining frequency and spatial domain information for fast interactive image noise removal. **Computer Graphics**, v.30, n. Annual Conference Series, p.269–276, 1996.

- [JAI 89] JAIN, A. K. **Fundamentals of digital image processing**. Englewood Cliffs, NJ: Prentice Hall, 1989.
- [KAN 79] KANIZSA, G. **Organization in vision**. New York, USA: Praeger, 1979.
- [KOK 95] KOKARAM, A. et al. Interpolation of missing data in image sequences. In: IEEE TRANSACTIONS ON IMAGE PROCESSING, 1995. **Anais...** [S.l.: s.n.], 1995. v.11, p.1509–1519.
- [KOM 2006] KOMODAKIS, N. Image completion using global optimization. In: CVPR '06: PROCEEDINGS OF THE 2006 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2006, Washington, DC, USA. **Anais...** IEEE Computer Society, 2006. p.442–452.
- [LEF 2005] LEFEBVRE, S.; HOPPE, H. Parallel controllable texture synthesis. In: SIGGRAPH '05: ACM SIGGRAPH 2005 PAPERS, 2005, New York, NY, USA. **Anais...** ACM Press, 2005. p.777–786.
- [LEF 2006] LEFEBVRE, S.; HOPPE, H. Appearance-space texture synthesis. In: SIGGRAPH '06: ACM SIGGRAPH 2006 PAPERS, 2006, New York, NY, USA. **Anais...** ACM Press, 2006. p.541–548.
- [LIA 2001] LIANG, L. et al. Real-time texture synthesis by patch-based sampling. **ACM Transactions on Graphics**, New York, NY, USA, v.20, n.3, p.127–150, 2001.
- [MAL 92] MALLAT, S. G.; ZHONG, S. Characterization of signals from multiscale edges. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.14, n.7, p.710–732, 1992.
- [MAL 89] MALLAT, S. A theory for multiresolution signal decomposition: the wavelet representation. In: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 1989, New York. **Anais...** [S.l.: s.n.], 1989. v.11, p.674–693.
- [MAS 98] MASNOU, S.; MOREL, J. Level lines based disocclusion. In: 5TH IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 1998. **Anais...** [S.l.: s.n.], 1998. p.3:259–263.

- [MCA 2004] MCANDREW, A. **An introduction to digital image processing with matlab.**, United States: Course Technology Press, 2004.
- [MEY 93] MEYER, Y. **Wavelets, algorithms and applications.** Philadelphia: SIAM, 1993.
- [MEY 2001] MEYER, Y. **Oscillating patterns in image processing and nonlinear evolution equations.** Boston, MA, USA: American Mathematical Society, 2001.
- [MöL 2003] MöLLER, T.; HAINES, E. **Real-time rendering.** 2°.ed. Natick, MA: A. K. Peters, 2003.
- [NIT 93] NITZBERG, M.; MUMFORD, D.; SHIOTA, T. **Filtering, segmentation, and depth.** Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1993.
- [NOE 98] NOE, A. P.; L.; THOMPSON, E. . Finding out about filling-in: a guide to perceptual completion for visual science and the philosophy of perception. **Behavioral and Brain Sciences** **6**, p.723–748, 1998.
- [PAT 2003] PATWARDHAN, K.; SAPIRO, G. Projection based image and video inpainting using wavelets. In: IN PROCEEDINGS OF IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2003. **Anais...** [S.l.: s.n.], 2003. p: 857–860.
- [PEA 88] PEARL, J. **Probabilistic reasoning in intelligent systems:** networks of plausible inference. San Mateo, California: Morgan Kaufmann Publishers, 1988.
- [RAN 2002] RANE, S. D.; REMUS, J.; SAPIRO, G. Wavelet-domain reconstruction of lost blocks in wireless image transmission and packet-switched networks. In: IN PROCEEDINGS OF IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2002. **Anais...** [S.l.: s.n.], 2002. p.309–312.
- [RUD 92] RUDIN, L.; OSHER, S.; FATEMI, C. Nonlinear total variation based noise removal algorithms. In: PHYSICA D, 1992, Amsterdam. **Anais...** [S.l.: s.n.], 1992. v.60, p.259–268.

- [RUD 94] RUDIN, L.; OSHER, S. Total variation based image restoration with free local constraints. In: PROC. IEEE INT. CONF. ON IMAGE PROCESSING, 1994. **Anais...** [S.l.: s.n.], 1994. v.1, p.31–35.
- [SAR 93] SARKAR, S.; BOYER, K. L. Integration, inference, and management of spatial information using bayesian networks: perceptual organization. **IEEE Transactions Pattern Analisys and Machine Intelligence**, Washington, DC, USA, v.15, n.3, p.256–274, 1993.
- [SUN 2005] SUN, J. et al. Image completion with structure propagation. **ACM Transactions on Computer Graphics**, New York, NY, USA, v.24, n.3, p.861–868, 2005.
- [TON 2005] TONIETTO, L.; WALTER, M.; JUNG, C. R. Patch-based texture synthesis using wavelets. In: PROCEEDINGS OF SIBGRAPI, 2005, Natal, Brazil. **Anais...** IEEE Computer Society, 2005. p.383–389.
- [TON 2006] TONIETTO, L.; WALTER, M.; JUNG, C. R. A randomized approach for patch-based texture synthesis using wavelets. In: COMPUTER GRAPHICS FORUM, 2006. **Anais...** IEEE Computer Society, 2006. v.25, n.4, p.675–684.
- [TSC 2005] TSCHUMPERLE, D.; DERICHE, R. Vector-valued image regularization with pdes: a common framework for different applications. In: IEEE TRANS. PATTERN ANAL. MACH. INTELL., 2005, Washington, DC, USA. **Anais...** IEEE Computer Society, 2005. v.27, n.4, p.506–517.
- [VES 2003] VESE, L. A.; OSHER, S. J. Modeling textures with total variation minimization and oscillating patterns in image processing. In: J. SCI. COMPUT., 2003, New York, NY, USA. **Anais...** Plenum Press, 2003. v.19, n.1-3, p.553–572.
- [WAN 2005] WANG, J.-F.; HSU, H.-J.; LIAO, S.-C. A novel framework for object removal from digital photograph. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2005. **Anais...** [S.l.: s.n.], 2005. v.2, p.73–76.
- [WIT 91] WITKIN, A.; KASS, M. Reaction-difusion textures. In: IN PROCEEDINGS OF 18TH CONF. GRAPHICS AND INTERACTIVE TECHNIQUES, 1991. **Anais...** [S.l.: s.n.], 1991. p.299–308.

- [WOR 96] WORLEY, S. P. A cellular texture basis function. In: IN PROCEEDINGS OF SIGGRAPH 1996, 1996. *Anais...* [S.l.: s.n.], 1996. v.30, p.291–294.
- [WU 93] WU, Z.; LEAHY, R. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. **IEEE Trans. Pattern Anal. Mach. Intell.**, Washington, DC, USA, v.15, n.11, p.1101–1113, 1993.
- [ZAL 2002] ZALESNY, A. et al. Parallel composite texture synthesis. **Texture 2002 workshop (ECCV 2002)**, Copenhagen, Denmark, p.151–156, June 2002.