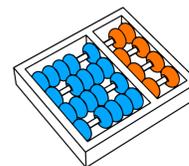


Ewerton Almeida Silva

**“Identificação de Manipulações de Cópia-Colagem
em Imagens Digitais.”**

CAMPINAS
2012



Universidade Estadual de Campinas
Instituto de Computação

Ewerton Almeida Silva

“Identificação de Manipulações de Cópia-Colagem em Imagens Digitais.”

Orientador(a): **Prof. Dr. Anderson de Rezende Rocha**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Ciência da Computação.

ESTE EXEMPLAR CORRESPONDE À
VERSÃO FINAL DA DISSERTAÇÃO DEFEN-
DIDA POR EWERTON ALMEIDA SILVA, SOB
ORIENTAÇÃO DE PROF. DR. ANDERSON
DE REZENDE ROCHA.

Assinatura do Orientador(a)

CAMPINAS

2012

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

Si38i Silva, Ewerton Almeida, 1988-
Identificação de manipulações de cópia-colagem em imagens digitais / Ewerton Almeida Silva. – Campinas, SP : [s.n.], 2012.

Orientador: Anderson de Rezende Rocha.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Análise forense de imagem. 2. Análise de imagem. 3. Visão por computador. 4. Descritor de imagem. I. Rocha, Anderson de Rezende, 1980-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em inglês: Copy-move forgery identification in digital images

Palavras-chave em inglês:

Digital forensic

Image analysis

Computer Vision

Image descriptor

Área de concentração: Ciência da Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Anderson de Rezende Rocha [Orientador]

Hélio Pedrini

Paulo André Vechiatto de Miranda

Data de defesa: 12-07-2012

Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 12 de Julho de 2012, pela
Banca examinadora composta pelos Professores Doutores:

Paulo André Vechiatto de Miranda

Prof. Dr. Paulo André Vechiatto de Miranda
IME / USP

Hélio Pedrini

Prof. Dr. Hélio Pedrini
IC / UNICAMP

Anderson de Rezende Rocha

Prof. Dr. Anderson de Rezende Rocha
IC / UNICAMP

Identificação de Manipulações de Cópia-Colagem em Imagens Digitais.

Ewerton Almeida Silva¹

12 de julho de 2012

Banca Examinadora:

- Prof. Dr. Anderson de Rezende Rocha (Orientador)
- Prof. Dr. Hélio Pedrini
Institute of Computing - UNICAMP, Campinas, SP
- Prof. Dr. Paulo André Vechiatto de Miranda
USP, São Paulo, SP
- Prof. Dr. João Paulo Papa
UNESP, Bauru, SP
- Prof. Dr. Ricardo da Silva Torres
Institute of Computing - UNICAMP, Campinas, SP

¹Financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) de Setembro de 2010 a Fevereiro de 2011 e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo número 2010/12196-9 de Março de 2011 a Junho de 2012.

Abstract

In this work, we investigate two approaches toward Copy-move Forgery detection in digital images. The first approach relies on the Generalized PatchMatch algorithm [4], which aims at finding patch correspondences in one or more images. Our approach consists in applying the Generalized PatchMatch algorithm in a certain image in order to obtain, for each of its patches, a set of similar patches based on their histogram distances. Next, we check the correspondences of each patch to decide whether or not they are portions of a duplicated region. Our second approach is based on a Voting and Multiscale Analysis process of an image. Given a suspicious image, we extract its interest points robust to scale and rotation transformations and we find possible correspondences among them. Next, we group the correspondent points into regions considering some geometric constraints, such as physical distance and inclination of the line between points of interest. After that, we construct a multiscale pyramid to represent the image scale-space. In each image, we examine the created groups using a descriptor robust to rotation, scaling and compression. This process decreases the search space of duplicated regions and yields a detection map. The final decision depends on a voting among all the detected maps, in which a pixel is considered as part of a manipulation if it is marked as so in the majority of the pyramid scales. We validate both methods using a dataset we have built comprising 108 original and realistic clonings. We compare the proposed methods to others from the state-of-the-art using such cloning dataset.

Resumo

Neste trabalho, nós investigamos duas abordagens para detecção de manipulações de Cópia-colagem (*Copy-move Forgery*) em imagens digitais. A primeira abordagem é baseada no algoritmo *PatchMatch Generalizado* [4], cuja proposta é encontrar correspondências de *patches* (blocos de *pixels* de tamanho definido) em uma ou mais imagens. A nossa abordagem consiste na aplicação do *PatchMatch Generalizado* em uma dada imagem com o propósito de encontrar, para cada *patch* desta, um conjunto de *patches* similares com base nas distâncias de seus histogramas. Em seguida, nós verificamos as correspondências de cada *patch* para decidir se eles são segmentos de uma região duplicada. A segunda abordagem, que consiste em nossa principal contribuição, é baseada em um processo de Votação e Análise Multiescala da imagem. Dada uma imagem suspeita, extraímos pontos de interesse robustos a operações de escala e rotação, encontramos correspondências entre eles e os agrupamos em regiões com base em certas restrições geométricas, tais como a distância física e a inclinação da reta que os liga. Após a aplicação das restrições geométricas, criamos uma pirâmide multiescala que representará o espaço de escalas da imagem. Nós examinamos, em cada imagem, os grupos criados usando um descritor robusto a rotações, redimensionamentos e compressões. Este processo diminui o domínio de busca de regiões duplicadas e gera um mapa de detecção para cada escala. A decisão final é dada a partir de uma votação entre todos os mapas, na qual um segmento é considerado duplicado se este assim o é na maioria das escalas. Nós validamos ambos os métodos em uma base de imagens que construímos. A base é composta por 108 clonagens originais e com elevado grau de realismo. Comparamos os métodos propostos com outros do estado da arte nessa mesma base de imagens.

Acknowledgements

Dois anos e meio após minha decisão de encarar o desafio do mestrado em outra cidade (bem longe de minha família), eu ganho um espacinho no meu trabalho final para agradecer a todos que foram importantes nesta empreitada. Este é, para mim, o capítulo mais importante deste trabalho.

Minha mãe (Ruth) e minha tia (Layce) são pessoas que fazem a minha vida valer a pena todos os dias. Obrigado a vocês duas! De alguma forma, acho que vocês me mandaram energia durante todo esse período do mestrado. Não posso esquecer também do meu velho pai (Riba), que me apoiou todo esse tempo não apenas financeiramente, mas confiando e tendo orgulho de mim. E à minha irmã de nome estranho (Hivigne), eu digo: Valeu por estar sempre por perto, mana!

No final de 2010, conheci um amigo sensacional. Nunca ri tanto quanto nas conversas com esse cara (sem exageros). Os chats no MSN, Caduzito, foram uma terapia perfeita, principalmente nos momentos de desestímulo e frustração com a pesquisa. Aprendi muito com você sobre mim. Muito obrigado pelo apoio que nunca me foi negado. Este trabalho também deveria levar seu nome, caro amigo.

Meus amigos do RECOD, também sou grato a todos vocês. Tiago, Priscila, Gordinho (Fábio), Tripodi, Sansão (Felipe), Otávio, Panaggio, Guilherme, Filipe, Jefersson, Sheila, Greice, Ramon... cada um de vocês me deu um pouco de motivação e alegria para concluir esta pesquisa. Trabalhar ao lado de vocês foi um prazer (quando eu dava as caras no lab, pelo menos, ha ha ha)!

Finalmente, desejo agradecer a você, Anderson, pela oportunidade que me concedeu de estudar numa universidade como a Unicamp e por ter sido paciente comigo nos momentos complicados da pesquisa. Foi um prazer escrever trabalhos sob a sua supervisão. Você me conhece: um texto, para mim, é um desafio de superação.

“O medo sempre me guiou para o que eu quero. E porque eu quero, temo. Muitas vezes foi o medo que me tomou pela mão e me levou. O medo me leva ao perigo. E tudo o que eu amo é arriscado.”

Clarice Lispector

Sumário

Abstract	ix
Resumo	xi
Acknowledgements	xiii
Epigraph	xv
1 Introdução	1
2 Revisão Bibliográfica	4
2.1 Cópia-Colagem	4
2.2 Estado da Arte	5
2.2.1 Abordagens de Fridrich et al. [12]	6
2.2.2 Abordagem de Popescu e Farid [25]	8
2.2.3 Abordagem de Luo et al. [19]	8
2.2.4 Abordagem de Zhang et al. [37]	8
2.2.5 Abordagem de Li et al. [15]	9
2.2.6 Abordagem de Myna et al. [22]	9
2.2.7 Abordagem de Bayram et al. [6]	9
2.2.8 Abordagem de Mahdian et al. [20]	10
2.2.9 Abordagem de Lin et al. [17]	10
2.2.10 Abordagem de Ardizzone e Mazzola [2]	10
2.2.11 Abordagem de Wang et al. [34]	11
2.2.12 Abordagem de Huang et al. [13]	11
2.2.13 Abordagem de Pan e Lyu [23]	12
2.2.14 Abordagem de Xu et al. [36]	12
2.2.15 Abordagem de Barnes et al. [4]	12
2.3 Considerações Finais	13

3	PatchMatch	14
3.1	PatchMatch Original	14
3.1.1	Aspectos Importantes	19
3.2	PatchMatch Generalizado	20
3.2.1	Aspectos Importantes	23
3.3	Deteção de cópia-colagem usando o PatchMatch Generalizado	25
3.4	Considerações Finais	28
4	Deteção de cópia-colagem baseada em Votação e Análise Multiescala	29
4.1	Abordagens Inspiradoras	29
4.2	Votação e Análise Multiescala de regiões de pontos de interesse	30
4.2.1	Etapa 1: Pré-processamento da imagem	32
4.2.2	Etapa 2: Deteção de pontos de interesse	33
4.2.3	Etapa 3: Pareamento de pontos de interesse	34
4.2.4	Etapa 4: Agrupamento	35
4.2.5	Etapa 5: Decomposição Piramidal	37
4.2.6	Etapa 6: Análise Multiescala	38
4.2.7	Etapa 7: Votação	39
4.3	Considerações Finais	40
5	Experimentos e Validação	42
5.1	Descrição do Dataset	42
5.2	Métricas de Avaliação	43
5.3	Detalhamento dos Testes	44
5.3.1	Deteção de cópia-colagem utilizando o PatchMatch Generalizado	44
5.3.2	Votação e Análise Multiescala de regiões de pontos de interesse . . .	45
5.4	Avaliação Qualitativa	46
5.4.1	Deteção de cópia-colagem utilizando o PatchMatch Generalizado	46
5.4.2	Votação e Análise Multiescala de regiões de pontos de interesse . . .	52
5.5	Avaliação Quantitativa	57
5.6	Considerações Finais	61
6	Conclusões	62
	Referências Bibliográficas	64

Lista de Tabelas

5.1	Resultados dos experimentos com a nossa abordagem.	58
5.2	Resultados dos experimentos com a abordagem de Popescu e Farid [25]. . .	58
5.3	Resultados dos experimentos com a abordagem de Li et al. [15].	58
5.4	Resultados dos experimentos com a abordagem de Wang et al. [34].	59

Lista de Figuras

2.1	Exemplos de cópia-colagem. (a) e (c) são as imagens originais e (b) e (d) são as imagens manipuladas. A imagem (a) é de autoria de J. A. Finnis.	5
2.2	Aplicação das operações de Rotação, Escala, Espelhamento (horizontal) e Suavização em um segmento duplicado.	6
2.3	Exemplo de blocos e da matriz de blocos antes da ordenação lexicográfica para uma imagem 7×7	7
3.1	Exemplos de correspondências entre <i>patches</i>	15
3.2	Representação do NNF para uma imagem. A localização do <i>patch</i> é dada pelas coordenadas de seu ponto central. Na figura, o <i>patch</i> (4,6) de tamanho 3×3 (em vermelho) da imagem original tem como correspondência o <i>patch</i> (6,5) (em azul), de mesmo tamanho. Logo, a posição (4,6) do NNF armazena os valores de $x = 6$, $y = 5$ e d (distância entre os <i>patches</i> vermelho e azul), aqui exemplificada como sendo igual a 11.	15
3.3	Cálculo da distância SSD entre dois <i>patches</i> correspondentes. No exemplo, a correspondência do <i>patch</i> vermelho é o <i>patch</i> de cor azul.	16
3.4	(a) Imagem de exemplo e (b) reconstrução desta usando o NNF aleatório inicial.	17
3.5	Esquemática da propagação para o <i>patch</i> de cor roxa. A imagem da esquerda apresenta os mapeamentos inspecionados (blocos à esquerda e acima). A imagem central evidencia as comparações efetuadas, visando descobrir se alguma das duas correspondências (vermelha ou azul) melhora o mapeamento para o <i>patch</i> roxo no NNF. A imagem da direita apresenta o resultado da propagação, na qual houve melhoria do mapeamento para o <i>patch</i> roxo utilizando a boa correspondência de seu vizinho à esquerda (vermelho) movida de 1 <i>pixel</i> à direita.	18
3.6	Busca Aleatória ao redor da nova correspondência encontrada, na propagação (Figura 3.5), para o <i>patch</i> roxo.	19
3.7	Resultados do <i>PatchMatch</i> na reconstrução da imagem (a) utilizando o NNF obtido após (b) 1, (c) 5 e (d) 10 iterações.	20

3.8	Nova estrutura do NNF, que passa a agregar um max-heap e uma lista encadeada para cada <i>patch</i> (posição) da matriz. Os elementos dessas estruturas são registros que armazenam três informações: as coordenadas x e y e a distância (d) entre o <i>patch</i> atual (quadrado azul do NNF) e sua correspondência. O valor da distância é a chave (em vermelho) usada para comparação e organização dos elementos do max-heap. Esta organização não se faz necessária dentro da lista encadeada.	22
3.9	Resultados do <i>PatchMatch Generalizado</i> na reconstrução da imagem (a) utilizando o NNF obtido após (b) 2 iterações com $K = 3$ e após (c) 2 e (d) 5 iterações com $K = 5$	24
3.10	Esquematização da nossa proposta para detecção de clonagens baseada no <i>PatchMatch Generalizado</i> . O método examina todos os conjuntos de correspondências do NNF em busca daquelas que se encontram em uma vizinhança similar à vizinhança de um <i>patch</i> (x, y). Na Figura, a vizinhança do <i>patch</i> de cor roxa sendo avaliado é similar à vizinhança do <i>patch</i> de cor vermelha (duplicação).	27
4.1	Esquematização completa da nossa metodologia.	31
4.2	Representação tridimensional do espaço de cor HSV (extraída de [1]). . . .	32
4.3	Ilustração das etapas 2 e 3. (a) pontos de interesse de uma imagem após aplicação do detector SURF; (b) correspondências entre pontos de interesse.	36
4.4	Exemplo de agrupamentos. Considere que os ângulos formados pelas retas pontilhadas 1 e 2 com o eixo horizontal da imagem pertencem ao mesmo intervalo. Dado que há proximidade física entre os pontos ligados por elas (correspondências), podem ser formados um grupo de origem e um de destino, ilustrados na figura pela cor vermelha. Em outra situação, os pontos ligados pela reta 3 (azul) não podem fazer parte do grupo vermelho (apesar da proximidade física com este), pois a orientação da reta não integra o intervalo de orientações requerido para a participação nesse grupo. Finalmente, a orientação da reta 4 (marrom) poderia permitir a entrada dos pontos correspondentes para o grupo verde, mas isso não ocorre devido à falta de proximidade física para os pontos deste.	37
4.5	Decomposição Piramidal de uma imagem.	38
4.6	Janela deslizante circular e Descritor utilizados em nossa abordagem. Os valores de média em cada círculo concêntrico são representados por m_1 , m_2 , m_3 e m_4	39

4.7	Esquematisação da análise de grupos a partir da localização de regiões candidatas na imagem. Em uma dada escala (nível da pirâmide), o grupo vermelho é analisado (origem e destino), e os descritores de seus <i>patches</i> são armazenados numa matriz específica que é ordenada lexicograficamente. Em seguida, buscamos por linhas parecidas nessa matriz e produzimos o mapa de detecção parcial (para esse grupo). A etapa prossegue com a verificação dos demais grupos e a atualização do mapa. Encerrados os grupos, o mapa de detecção final é aprimorado. Este procedimento é repetido para todas as imagens da pirâmide multiescala.	40
5.1	Exemplos de imagens do nosso <i>dataset</i> . (a), (d) e (g) São as imagens originais; (b), (e) e (h) são exemplos de clonagens e (c), (f) e (i) são seus respectivos mapas de referência.	43
5.2	Resultados visuais de nossa primeira abordagem para clonagens simples. (a), (e) e (i) São as clonagens; (b), (f) e (j) são os mapas de referência; (c), (g) e (k) são mapas de detecção obtidos para imagens sem compressão e (d), (h) e (l) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80 e 70, respectivamente.	48
5.3	Resultados visuais de nossa primeira abordagem para clonagens com espelhamento. (a), (e) e (i) São clonagens com espelhamento horizontal, vertical e ambos, respectivamente; (b), (f) e (j) são os mapas de referência; (c), (g) e (k) são os mapas de detecção obtidos para imagens sem compressão e (d), (h) e (l) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80 e 70, respectivamente.	49
5.4	Resultados visuais de nossa primeira abordagem para clonagens com rotações. (a), (e), (i), (m) e (q) São clonagens contendo rotações de 90, 180, 270, 10 e -5 graus, respectivamente; (b), (f), (j), (n) e (r) são os mapas de referência; (c), (g), (k), (o) e (s) são os mapas de detecção obtidos para imagens sem compressão e (d), (h), (l), (p) e (t) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80, 70, 90 e 80, respectivamente.	50
5.5	Resultados visuais de nossa primeira abordagem para clonagens com operações de escala. (a) e (e) São clonagens com ampliação de 7% e diminuição de 10% dos segmentos duplicados, respectivamente; (b) e (f) são os mapas de referência; (c) e (g) são os mapas de detecção obtidos para imagens sem compressão e (d) e (h) são os mapas de detecção obtidos para imagens com compressão JPEG nos fatores 90 e 80, respectivamente.	51

5.6	Resultados visuais de nossa segunda abordagem para clonagens simples. (a), (e) e (i) São as clonagens; (b), (f) e (j) são os mapas de referência; (c), (g) e (k) são os mapas de detecção obtidos para imagens sem compressão e (d), (h) e (l) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80 e 70, respectivamente.	53
5.7	Resultados visuais de nossa segunda abordagem para clonagens com rotação. (a), (e) e (i) São as clonagens contendo rotações dos segmentos duplicados (28, -30 e 7 graus, respectivamente); (b), (f) e (j) são os mapas de referência; (c), (g) e (k) são os mapas de detecção obtidos para imagens sem compressão e (d), (h) e (l) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80 e 70, respectivamente.	54
5.8	Resultados visuais de nossa segunda abordagem para clonagens com operações de escala. (a), (e), (i), (m) e (q) São as clonagens contendo redimensionamento dos segmentos duplicados (ampliações de 19%, 5% e 18%, e reduções de 7% e 10%, respectivamente); (b), (f), (j), (n) e (r) são os mapas de referência; (c), (g), (k), (o) e (s) são os mapas de detecção obtidos para imagens sem compressão e (d), (h), (l), (p) e (t) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80, 70, 90 e 70, respectivamente.	55
5.9	Resultados visuais de nossa segunda abordagem para clonagens com operações de escala e rotação combinadas. (a), (e), (i), (m) e (q) São as clonagens (-10 graus e ampliação de 8%; -90 graus e ampliação de 1%; 2 graus e ampliação de 7%; 3 graus e redução de 16%; -18 graus e redução de 10%, respectivamente); (b), (f), (j), (n) e (r) são os mapas de referência; (c), (g), (k), (o) e (s) são os mapas de detecção obtidos para imagens sem compressão e (d), (h), (l), (p) e (t) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80, 70, 90 e 80, respectivamente.	56
5.10	<i>Ranking</i> das imagens para a nossa abordagem, considerando-se as três métricas propostas.	59
5.11	<i>Ranking</i> das imagens para a abordagem Popescu e Farid [25], considerando-se as três métricas propostas.	59
5.12	<i>Ranking</i> das imagens para a abordagem de Li et al. [15], considerando-se as três métricas propostas.	60
5.13	<i>Ranking</i> das imagens para a abordagem de Wang et al. [34], considerando-se as três métricas propostas.	60
5.14	<i>Ranking</i> Geral das abordagens examinadas, considerando-se as três métricas propostas.	61

Capítulo 1

Introdução

O professor Woo-Suk-Hwang, da *Seoul National University*, ganhou notoriedade em 2004 quando publicou, com sua equipe, trabalhos que mostravam avanços significativos nos estudos referentes a células-tronco. Os trabalhos haviam sido publicados na *Science*, uma das mais renomadas revistas científicas do mundo. Meses depois, entretanto, o professor tornou-se o protagonista de um dos mais complicados casos de fraude científica já noticiados [28]. Investigações revelaram evidências de fraude em imagens que ilustravam resultados importantes da pesquisa publicada. Tais imagens haviam sido alteradas digitalmente. O escândalo e a pressão da comunidade científica levaram à retratação dos artigos, pela *Science*, em 2006 [27, 10, 33].

Em outro caso, a edição do dia 5 de abril de 2009 de um periódico brasileiro retratava, na primeira página, a imagem de uma suposta ficha criminal da então ministra brasileira da Casa Civil Dilma Rousseff. A ficha relatava a participação da ministra em assaltos e planejamentos de sequestros durante o regime militar brasileiro. Um laudo técnico, entretanto, negou a autenticidade da ficha ao revelar que o documento se tratava de uma fabricação [26].

Os fatos acima mencionados apontam para um problema pelo qual a sociedade tem passado nos últimos anos, principalmente, em razão dos avanços tecnológicos. Tendo em mãos dispositivos de captura de imagens e ferramentas de *software* aprimoradas para edição destas, o usuário comum tem participado ativamente na manipulação desse tipo de conteúdo digital, executando desde simples ajustes de iluminação de cena, até a confecção de montagens criminosas.

Dentre as diversas formas de manipulação de imagens, uma manipulação que pode ser efetuada sem muito esforço é a Cópia-Colagem (*Copy-Move Forgery*), também denominada Clonagem. Ela consiste em copiar regiões de uma imagem e colá-las sobre outras áreas, que se deseja mascarar, da mesma imagem [6]. A cópia-colagem pode, por exemplo, ser utilizada para fazer uma pessoa “desaparecer” de uma fotografia, a partir

da sobreposição de segmentos da própria imagem aos *pixels* da região contendo a pessoa. Ademais, pode-se utilizar este tipo de manipulação para multiplicar ou mesmo auxiliar o reposicionamento de elementos em uma imagem.

O desafio imposto pela cópia-colagem reside na facilidade e nas diversas formas de executá-la. Fridrich et al. [12] argumentam que adicionar ou eliminar elementos a uma imagem digital são tarefas simples de serem feitas por meio de aplicativos de *software* como GIMP e Adobe Photoshop. Tais aplicativos incluem ferramentas que favorecem a execução de manipulações em imagem de maneiras variadas. Pode-se, por exemplo, aprimorar o resultado de uma clonagem a partir do emprego de operações de compressão JPEG [32], rotação, redimensionamento, espelhamento e suavização nas regiões duplicadas, o que propicia a eliminação de vestígios visuais da adulteração e a adição de um desafio extra à identificação desta.

Em virtude da facilidade atual em se adulterar imagens digitais com propósitos excusos, a elaboração de métodos para a autenticação destas tem obtido um papel de destaque na Computação Forense¹. Segundo Farid [11], em muitas situações, o exame a olho nu nem sempre é suficiente para apontar a presença de alterações em imagens. Nesses casos, a verificação de autenticidade pode ser feita com a ajuda de métodos computacionais específicos. O termo “específico” indica que, por exemplo, um método para identificação de uma clonagem na qual o segmento duplicado também foi rotacionado pode não ser eficaz na detecção de uma clonagem que envolveu compressão JPEG.

Neste trabalho, nós investigamos duas abordagens para detecção de manipulações de cópia-colagem em imagens digitais. A primeira abordagem é baseada no *PatchMatch Generalizado* [4], cuja proposta é encontrar correspondências de *patches* (blocos de *pixels* de tamanho definido) em uma ou mais imagens. A nossa abordagem consiste na aplicação do *PatchMatch Generalizado* em uma dada imagem com o propósito de encontrar, para cada *patch* desta, um conjunto de *patches* similares com base nas distâncias de seus histogramas. Em seguida, nós verificamos as correspondências de cada *patch* para decidir se eles são, ou não, segmentos de uma região duplicada.

A segunda abordagem é baseada em um processo de Votação e Análise Multiescala da imagem. Dada uma imagem suspeita, extraímos pontos de interesse robustos a operações de escala e rotação, encontramos correspondências entre eles e os agrupamos em regiões com base em certas restrições geométricas, tais como a distância física e a inclinação da reta que os liga. Após a aplicação das restrições geométricas, criamos uma pirâmide multiescala que representará o espaço de escalas da imagem. Nós examinamos, em cada

¹Área de pesquisa que se destina a determinar os aspectos dinâmicos, materiais e de auditoria referentes a eventos ilícitos praticados na área de informática, e que tem como uma de suas questões centrais o emprego de métodos técnico-científicos para coletar, identificar, preservar e processar evidências digitais, conferindo-lhes legitimidade em juízo probatório [8, 9].

imagem, os grupos criados usando um descritor robusto a rotações, redimensionamentos e compressões. Este processo diminui o domínio de busca de regiões duplicadas e gera um mapa de detecção para cada escala. A decisão final é dada a partir de uma votação entre todos os mapas, na qual um segmento é considerado duplicado se este assim o é na maioria das escalas.

Com ambas as metodologias, nós fornecemos contribuições que enriquecem a literatura por meio da fusão de diversas ideias originais e já existentes e que proporcionam ao perito uma ferramenta de apoio à investigação de imagens digitais suspeitas de adulteração. O auxílio diz respeito à análise do conteúdo da cena visando apontar regiões candidatas à cópia-colagem. A partir de um exame visual posterior das regiões detectadas e da vizinhança destas, o perito deve tomar uma decisão quanto à autenticidade da imagem, agilizando e aumentando a precisão da tarefa de investigação.

Nós validamos ambos os métodos em uma base de imagens que construímos. A base é composta por 108 clonagens originais e com elevado grau de realismo. Nós avaliamos os métodos de maneira qualitativa (visual) e quantitativa, e comparamos a segunda abordagem com outros trabalhos da literatura que implementamos [24, 15, 34].

Organizamos este texto em seis capítulos. No Capítulo 2, apresentamos uma revisão bibliográfica sobre cópia-colagem, que inclui uma visão geral sobre este tipo de manipulação, os desafios dela provenientes e as principais abordagens da literatura para sua detecção. No Capítulo 3, descrevemos os métodos *PatchMatch Original* e *Generalizado* e introduzimos nossa primeira metodologia para identificação de clonagens. No Capítulo 4, apresentamos nossa segunda, e principal, contribuição para detecção de cópia-colagem e, no Capítulo 5, mostramos os experimentos efetuados e validações qualitativas e quantitativas de ambos os métodos propostos e de métodos da literatura. Finalmente, apresentamos as conclusões, possibilidades de trabalhos futuros e de extensões desta pesquisa no Capítulo 6.

Capítulo 2

Revisão Bibliográfica

Neste capítulo, nós apresentamos uma visão geral da cópia-colagem, os desafios comumente enfrentados para a sua detecção e os principais métodos existentes na literatura para lidar com tais desafios. Nós fornecemos uma descrição resumida de cada abordagem, apresentando as estratégias adotadas, os resultados obtidos e os pontos positivos e negativos que, em nossa opinião, são determinantes para avaliar o potencial do método.

2.1 Cópia-Colagem

Uma manipulação de cópia-colagem consiste em se clonar segmentos dentro de uma imagem. Por meio dela, pode-se ocultar ou multiplicar elementos presentes em uma cena, tais como folhagem, objetos e pessoas. Para isso, copia-se um segmento da imagem e, caso necessário, aplicam-se operações adicionais a este. Em seguida, o segmento modificado é posicionado na área que se deseja mascarar. Finalmente, uma operação global pode ser aplicada na imagem final. O potencial desta manipulação é retratado na Figura 2.1.

Dentre as operações usualmente efetuadas sobre os segmentos clonados, destacam-se a *Rotação*, o *Redimensionamento* (ou *Escala*), o *Espelhamento* (horizontal ou vertical) e a *Suavização de Bordas*. Adicionalmente, o *Casamento de Iluminação* poderia ser aplicado. Por outro lado, as operações globais mais comuns são a *Adição de Ruídos Gaussianos* e a *Compressão JPEG* da imagem. Essas operações, quando empregadas, podem dificultar o trabalho de identificação da cópia-colagem, pois modificam os valores de *pixel* dos segmentos duplicados (rotações, suavizações, compressões e ruídos), alteram a disposição destes com relação aos segmentos originais (rotações e espelhamentos) ou, ainda, inserem/eliminam informações dos *pixels* da região duplicada (redimensionamentos). A Figura 2.2 ilustra a aplicação das operações locais em um segmento clonado.

O maior desafio referente à detecção de cópia-colagem, portanto, reside na identificação de segmentos similares dentro de uma imagem digital. A similaridade, contudo, pode

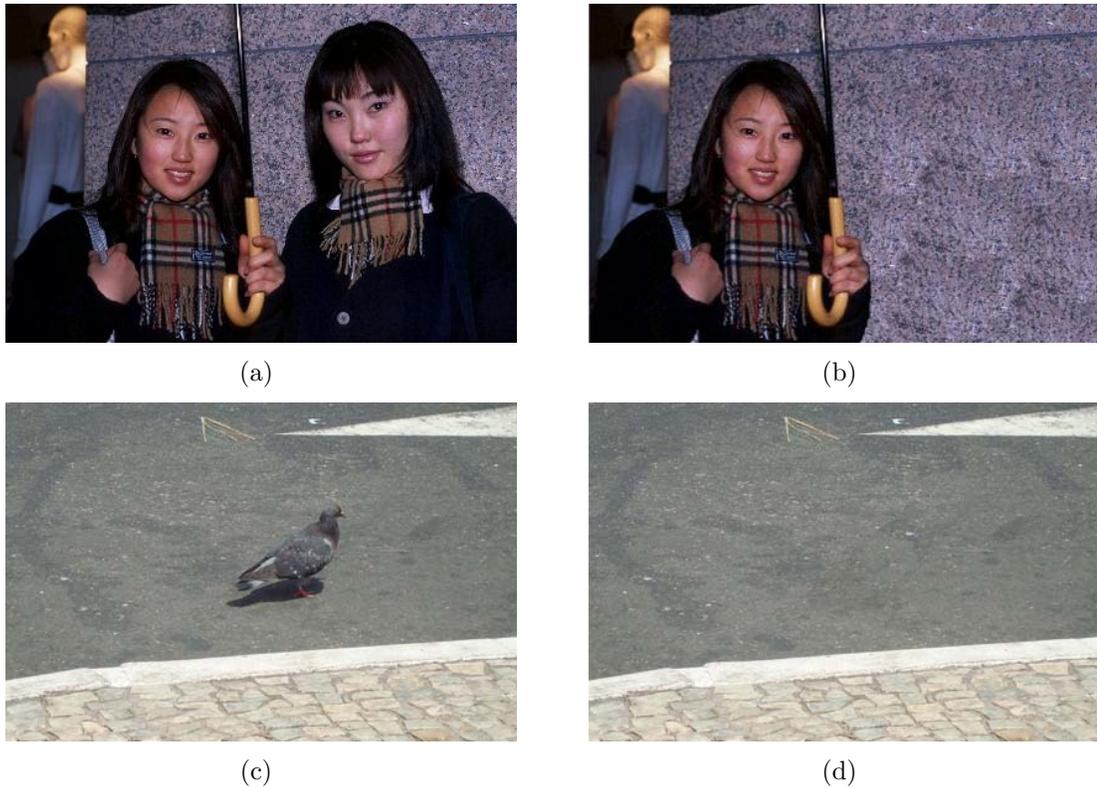


Figura 2.1: Exemplos de cópia-colagem. (a) e (c) são as imagens originais e (b) e (d) são as imagens manipuladas. A imagem (a) é de autoria de J. A. Finnis.

decorrer tanto da aplicação de operações aos segmentos e à imagem como, também, da existência de regiões naturalmente semelhantes na cena, tais como céu azul, asfalto e partes de objetos sob forte iluminação. A seguir, apresentamos abordagens da literatura que se propõem a encarar este desafio.

2.2 Estado da Arte

A literatura apresenta inúmeros algoritmos para detecção de manipulações de cópia-colagem. Uma abordagem direta consiste na comparação de cada par de *pixels* da imagem. A complexidade deste método é, entretanto, exponencial no número de *pixels*, o que o torna impraticável. Além disso, o número de falsos-positivos (regiões incorretamente identificadas como duplicadas) pode ser elevado [11]. A seguir, apresentamos algumas abordagens da literatura que tratam deste problema de maneira mais apropriada.



Figura 2.2: Aplicação das operações de Rotação, Escala, Espelhamento (horizontal) e Suavização em um segmento duplicado.

2.2.1 Abordagens de Fridrich et al. [12]

Fridrich et al. [12] propõem duas abordagens nas quais a busca por regiões duplicadas é realizada bloco a bloco (e não mais *pixel a pixel*). A seguir, apresentamos o funcionamento do primeiro método, denominado Casamento Exato (*Exact Match*):

- Um bloco, geralmente quadrado, de $b \times b$ *pixels* é definido. Este bloco pode ser considerado uma máscara ou janela que, ao ser posicionado sobre um *pixel* da imagem, cobre uma determinada área de tamanho b^2 ;
- O bloco é deslizado por cada *pixel* da imagem de dimensão $M \times N$ (da esquerda para a direita e de cima para baixo) e os valores das colunas de cada posição em que o bloco se encontra são extraídos e armazenados nas linhas de uma matriz com b^2 colunas e $(M - b + 1)(N - b + 1)$ linhas;
- Duas linhas idênticas correspondem a segmentos idênticos. Para encontrar tais segmentos, ordena-se a matriz lexicograficamente no tempo $O(MN \lg MN)$. Esta ordenação é análoga à alfabética, na qual os caracteres mais à esquerda são dispostos em ordem crescente primeiro;

- Finalmente, percorre-se a matriz ordenada buscando linhas consecutivas idênticas que correspondem às regiões duplicadas.

A Figura 2.3 ilustra blocos de tamanho 3×3 (apresentados em diferentes cores) para uma imagem 7×7 e a matriz na qual eles são armazenados. No exemplo, a matriz ainda não se encontra ordenada lexicograficamente.

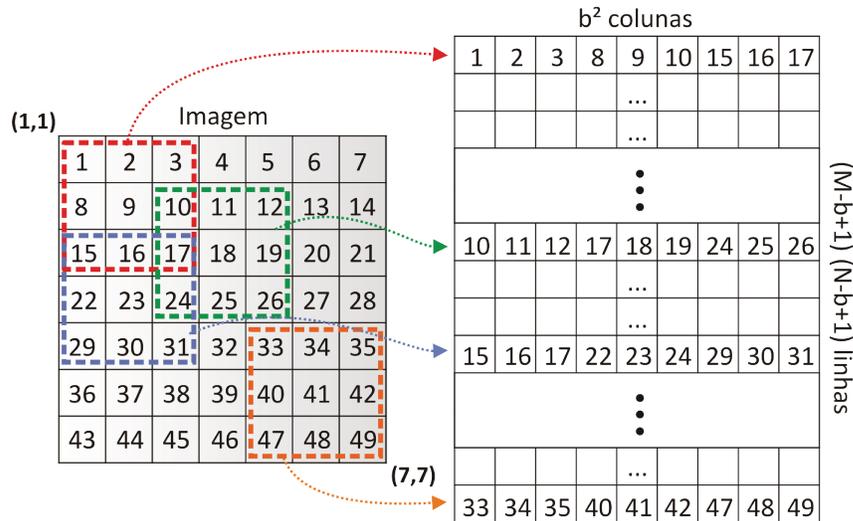


Figura 2.3: Exemplo de blocos e da matriz de blocos antes da ordenação lexicográfica para uma imagem 7×7 .

O Casamento Exato promove flexibilidade na análise da correspondência entre pequenos segmentos conectados da imagem, além de proporcionar a identificação das regiões clonadas em um tempo razoável. No entanto, há uma desvantagem expressiva nesta abordagem, justificada pela reduzida eficácia no caso de o segmento duplicado (ou a imagem como um todo) ter sofrido manipulações adicionais. Em uma tentativa de contornar esta limitação, os autores propõem a utilização da Transformada Discreta do Cosseno (*Discrete Cosine Transform – DCT*) para quantizar os coeficientes que representam cada bloco. A quantização faz parte do método de Casamento Aproximado (*Robust Match*) e gera resultados de detecção melhores para o caso de a imagem adulterada ter sido comprimida em JPEG. Entretanto, não há tratamento para quaisquer outras operações, tais como rotações e escala.

A estratégia utilizada no método de Casamento Exato é empregada por grande parte dos métodos da literatura, os quais realizam extensões e modificações visando torná-lo efetivo na identificação das regiões clonadas.

2.2.2 Abordagem de Popescu e Farid [25]

Nesse método, os autores utilizam Análise de Componentes Principais (*Principal Component Analysis* – PCA) para reduzir a dimensionalidade de cada bloco criando, assim, uma representação destes robusta à ocorrência de pequenas variações na imagem, tais como aquelas geradas por ruídos aditivos e pela compressão JPEG. Ao utilizar PCA, o método captura as informações mais representativas de cada bloco (suas componentes principais) e analisa apenas este conjunto reduzido de dados. Os resultados reportados pelos autores mostraram que o método é eficaz na detecção de cópia-colagem em imagens comprimidas em JPEG e corrompidas com Ruído Gaussiano em diferentes graus. Entretanto, para compressões em fatores pequenos (o que diminui a qualidade da imagem) e ruídos em quantidade excessiva, o método se mostra pouco eficaz.

2.2.3 Abordagem de Luo et al. [19]

Luo et al. [19] extraem um conjunto de sete informações específicas de cada bloco e realizam a comparação destes com base em tais características. Os autores afirmam que o método supera aqueles propostos em [24, 12] em termos de complexidade computacional e efetividade na detecção das regiões em cenários de suavização, ruídos, compressão JPEG e combinações destas operações de pós-processamento. Os resultados reportados mostraram que o algoritmo obtém melhores resultados quando blocos de tamanhos maiores do que 32×32 *pixels* são utilizados, já que a chance de se encontrar blocos similares é maior quando há mais informação em cada bloco de comparação. O método, porém, não trata os casos nos quais regiões duplicadas são previamente rotacionadas e nem quando possuem transformações de escala.

2.2.4 Abordagem de Zhang et al. [37]

Zhang et al. [37] propõem um método baseado na análise da sub-banda recursiva da imagem usando a Transformada *Wavelet* Discreta (*Discrete Wavelet Transform* – DWT) e cálculos de correlação de fase. A abordagem consiste no exame da imagem em sua versão reduzida em 1/4 do tamanho original. Apenas a sub-banda recursiva é analisada. Esta, então, é particionada em quatro regiões sem sobreposição sobre as quais a Correlação de Fase (*Phase Correlation*) é calculada. Este cálculo envolve o cômputo da Transformada de Fourier (*Fourier Transform* – FT). Os resultados mostraram que a técnica é robusta para certos fatores de compressão JPEG e na presença de suavização dos segmentos duplicados. Não há, todavia, tratamento para outras operações de pós-processamento.

2.2.5 Abordagem de Li et al. [15]

Li et al. [15] apresentam um método baseado na DWT e na Decomposição em Valores Singulares (*Singular Value Decomposition* – SVD), objetivando reduzir a quantidade de informação a ser processada na imagem e a dimensão dos blocos. Após o emprego de DWT para redução do tamanho da imagem, o método prossegue calculando os valores singulares para cada matriz (bloco de *pixels*). Um dado bloco passa, então, a ser representado por meio de seus valores singulares e não mais pelos seus valores de *pixels*. O método é eficaz, principalmente, para casos em que a imagem é comprimida com fator de qualidade até 70%; para compressões maiores, a baixa qualidade afeta a detecção. No caso de imagens coloridas, a detecção pode proceder sobre cada canal RGB (*Red*, *Green* e *Blue*) ou sobre a imagem convertida para níveis de cinza.

2.2.6 Abordagem de Myna et al. [22]

Myna et al. [22] apresentam um método que também faz uso da DWT para gerar uma representação da imagem com dimensionalidade menor. A imagem é decomposta (suas dimensões são diminuídas) um certo número de vezes e depois examinada partindo-se da menor até a sua maior resolução (a imagem original). Neste processo, o tamanho do bloco deslizante sobre a imagem vai dobrando com a duplicação do tamanho desta, e blocos similares são armazenados a fim de serem comparados novamente nos níveis posteriores. Cada bloco é mapeado para coordenadas log-polares como uma estratégia de representá-los de maneira robusta a operações de escala e rotação. A correlação de fase também é calculada, sendo esta a métrica de similaridade entre blocos adotada. Apesar disso, o método aparenta funcionar somente quando as regiões clonadas são idênticas ou quando elas estão rotacionadas em 90, 180 e 270 graus.

2.2.7 Abordagem de Bayram et al. [6]

No método proposto por Bayram et al. [6], cada bloco tem suas características principais extraídas a partir da Transformada de Fourier-Melin (*Fourier-Melin Transform* – FMT). O tempo de detecção de blocos com características similares também é reduzido com o uso de uma abordagem diferenciada de ordenação denominada *Counting Bloom Filters* (CBF), que computa o valor hash do *pixel* de cada bloco e torna a tarefa de se encontrar regiões similares mais rápida. Apesar do ganho em eficiência, a robustez do método usando CBF é menor, já que trata apenas a ocorrência de blocos muito parecidos. O método alcança bons resultados para imagens comprimidas em JPEG com fator de, no máximo, 70%. Também são apresentados resultados para os casos de redimensionamentos (até 5%) ou rotações (até 10 graus) das regiões duplicadas.

2.2.8 Abordagem de Mahdian et al. [20]

Mahdian et al. [20] utilizam uma estratégia baseada em Momentos Invariantes a Borramento (*Blur Moment Invariants*). O objetivo é definir um algoritmo que descreva os blocos por meio de características invariantes à presença de degradações causadas pela suavização. Adicionalmente, a quantidade de informação que representa cada bloco é reduzida por meio de PCA. Na análise de similaridade entre duas regiões, cada bloco pertencente à vizinhança das duas regiões também é comparado visando à localização das regiões efetivamente copiadas e coladas. O método funciona bem nos cenários de compressão JPEG com qualidade até 50%, suavização com máscaras de tamanho máximo 3×3 e ruído Gaussiano aditivo. Porém, o método apresenta desvantagens críticas, como o alto número de falsos-positivos (em regiões uniformes, como céu-azul) e o elevado tempo de execução (aproximadamente 40 minutos para uma imagem de tamanho 640×480 *pixels* em uma máquina com processador de 2.1 GHz e 512 MB de memória RAM).

2.2.9 Abordagem de Lin et al. [17]

Lin et al. [17] extraem nove características de cada bloco e adotam o algoritmo *Radix-Sort*, que exige tempo linear, para ordenação lexicográfica e comparação daqueles. A fim de lidar com rotações, os autores consideraram três diferentes orientações da imagem original (90, 180 e 270 graus), comparando-a com uma combinação das imagens rotacionadas nestas três direções.

2.2.10 Abordagem de Ardizzone e Mazzola [2]

Ardizzone e Mazzola [2] propõem efetuar a detecção de clonagem em imagens a partir da análise da imagem decomposta em seus planos de bits (*Bit-Plane Analysis*). O processo é similar às demais abordagens de comparação de blocos de *pixels* mas, neste caso, as comparações são efetuadas em cada plano de bit da imagem (uma imagem de n bits de profundidade de cor, e.g., 24 bpp, possui n planos, um para cada bit). Partindo do plano respectivo ao bit menos significativo, o método codifica cada bloco em código ASCII e efetua as comparações. No plano de bit seguinte, apenas os blocos similares encontrados na etapa anterior são comparados e assim sucessivamente até o plano de bit mais significativo. Esta abordagem, entretanto, é efetiva apenas para o caso de cópia-colagem sem operações de pós-processamento adicionais.

2.2.11 Abordagem de Wang et al. [34]

Wang et al. [34] estabelecem uma janela deslizante circular com o objetivo de criar uma solução robusta para os casos em que as regiões copiadas são rotacionadas em graus diversos antes da colagem. Segundo os autores, a maioria das abordagens da literatura têm dificuldade em lidar com estes casos pelo fato de considerarem blocos quadrados, que não são totalmente invariantes a rotações. O esquema proposto é baseado na extração de quatro características representadas pelas médias dos círculos concêntricos de cada bloco e da Decomposição Piramidal Gaussiana, que reduz a dimensão da imagem para 25% do tamanho original. Os vetores de características são comparados por meio da distância Euclidiana.

Os testes abrangeram regiões rotacionadas em graus não considerados nas demais abordagens da literatura, tais como -20 e 12 graus. A detecção foi efetiva para estas orientações. Experimentos com *blur* Gaussiano, compressão JPEG (máxima de 40%) e ruídos Gaussianos (de 15db a 40db) mostraram que o método apresenta acerto médio superior a 90% para tais casos. Os autores também reportaram um teste com a operação de Espelhamento Horizontal (*Horizontal Flipping*), obtendo sucesso na detecção da região clonada. O método, contudo, não contempla operações de escala eventualmente efetuadas nos blocos duplicados.

As abordagens anteriores consideram tamanhos pequenos fixos de bloco, levando em conta que as áreas duplicadas são maiores que estes. Dessa forma, a qualidade da detecção tem melhorias, ao custo de um aumento no número de computações efetuadas (já que mais comparações são realizadas com blocos pequenos). Quando blocos grandes são usados, o tempo de execução dos métodos pode ser diminuído, mas também há chance de elevar a taxa de falsos-negativos, isto é, regiões duplicadas podem não ser identificadas devido ao bloco ser maior do que elas. As abordagens a seguir adotam um enfoque para lidar com a detecção de cópia-colagem que não se baseia na análise de blocos.

2.2.12 Abordagem de Huang et al. [13]

Huang et al. [13] propõem um método baseado na comparação de descritores SIFT (*Scale-Invariant Feature Transform*) [18], os quais são razoavelmente invariantes a ruídos, rotações, transformações de escala e mudanças na iluminação. O método realiza comparações entre descritores SIFT com 128 características, o que pode apontar regiões similares dentro da imagem. Segundo os autores, a detecção é efetiva nos casos de compressão JPEG com qualidade de até 40% e na presença de ruído aditivo Gaussiano (até 20db). Apesar de mencionar invariância à rotação, apenas as orientações menos problemáticas foram testadas (90, 180 e 270 graus). Também não são mostrados testes em imagens contendo suavização. Finalmente, o método possui dificuldades na determinação das regiões

de cópia-colagem quando estas são muito pequenas.

2.2.13 Abordagem de Pan e Lyu [23]

O método proposto por Pan e Lyu [23] também é baseado na extração de descritores SIFT da imagem. Primeiramente, um conjunto de pontos SIFT (e suas respectivas descrições) são encontrados na imagem e esta é dividida em blocos de inspeção. Tais blocos correspondem a regiões contendo pontos característicos (*keypoints*) da imagem. A partir da comparação dos descritores de cada *keypoint* (vetores de 128 dimensões) pertencentes a dois blocos de inspeção diferentes, torna-se possível estimar a transformação efetuada entre duas regiões da imagem. O método é robusto a operações de rotação, escala, compressão JPEG e adição de ruídos. Entretanto, os resultados são escassos e as regiões duplicadas possuem tamanhos significativos.

2.2.14 Abordagem de Xu et al. [36]

Xu et al. [36] utilizam o algoritmo SURF (*Speeded-Up Robust Features*) [5], uma alternativa ao SIFT para localizar regiões de cópia-colagem. A estratégia consiste na comparação de descritores SURF da imagem. A comparação envolve a divisão do conjunto total de descritores em dois grupos aleatórios de tal forma que, em cada grupo, os descritores similares são encontrados. Em seguida, os grupos são subdivididos novamente em dois. Esse procedimento é aplicado sucessivamente até que todos os grupos contenham apenas um elemento. Finalmente, as correspondências encontradas são apresentadas na imagem.

Os resultados experimentais apresentados demonstraram o potencial do método na localização de regiões clonadas rotacionadas em graus variados, redimensionadas, suavizadas e nos casos em que ruídos são adicionados à imagem. Um ponto interessante do trabalho se deve à detecção eficaz mesmo quando a região clonada é rotacionada e logo depois redimensionada (composição de operações). Entretanto, os testes realizados não levaram em conta a cópia-colagem de regiões homogêneas, o que poderia reduzir a acurácia do método, uma vez que tais áreas poderiam não conter *keypoints* suficientes para comparação dos descritores.

2.2.15 Abordagem de Barnes et al. [4]

O trabalho de Barnes et al. [4], denominado *PatchMatch Generalizado* (*Generalized PatchMatch*), apesar de não se apresentar como uma estratégia voltada diretamente à solução do problema da cópia-colagem, é um esforço que pode ser empregado nos desafios aqui discutidos. Esse algoritmo é uma expansão do algoritmo *PatchMatch*, proposto em [3], que objetiva possibilitar a localização de correspondências de blocos (*patches*) dentro de uma

imagem de maneira rápida. Em [4], a aplicação do *PatchMatch Generalizado* na detecção de clonagens simples é mencionada sem maiores detalhes. Neste trabalho, nós implementamos esses métodos, investigamos a proposta do autor para detecção de clonagens e desenvolvemos nossa própria contribuição para a solução deste problema no Capítulo 3.

2.3 Considerações Finais

Neste capítulo, nós apresentamos a manipulação de cópia-colagem em maiores detalhes, assim como os desafios que sua detecção agrega. Descrevemos, ainda, as principais abordagens da literatura existentes para a solução desses desafios. No Capítulo 3 a seguir, detalharemos o algoritmo *PatchMatch Original* e sua versão estendida, o *PatchMatch Generalizado*, cujas ideias foram utilizadas por nós na elaboração de uma metodologia original para detecção de clonagens utilizando um mecanismo de busca aleatório de *patches* duplicados em imagens. Essa metodologia será abordada no final do mesmo capítulo. Posteriormente, no Capítulo 4, nós descrevemos a nossa principal abordagem destinada à identificação de cópia-colagem em imagens digitais, e que é baseada em um procedimento de Votação e Análise Multiescala destas.

Capítulo 3

PatchMatch

Um dos algoritmos que acreditamos ter elevado potencial na detecção de clonagens em imagens digitais é uma abordagem inovadora proposta por Barnes et al. [4]. Denominada *PatchMatch Generalizado* (*Generalized PatchMatch*), a técnica é uma extensão daquela apresentada em Barnes et al. [3] com o nome de *PatchMatch*. O algoritmo foi desenvolvido em colaboração com a Adobe Systems e integra algumas ferramentas do Photoshop CS5, tais como o *Content-Aware Scale*, o *Content-Aware Fill* e o *Spot Healing Brush*. Neste capítulo, nós apresentamos aspectos relativos aos algoritmos *PatchMatch Original* e *Generalizado*. Ao final, nós mostramos a nossa primeira contribuição que utiliza ideias deste último para detecção de clonagens simples.

3.1 PatchMatch Original

O *PatchMatch* é um algoritmo randomizado que se propõe a encontrar correspondências aproximadas de *patches* (blocos de *pixels* de tamanho definido, e.g., 7×7) em uma ou mais imagens. Em comparação com outros métodos da literatura de Amostragem baseada em Patches (*Patch-based Sampling* [16]), o *PatchMatch* alcança resultados favoráveis em tempos de execução menores e utilizando menos espaço em memória. A principal novidade do algoritmo é que a busca por um *patch b* similar ao *patch a* é mais rápida ao ser realizada por um mecanismo aleatório. Ao invés de se percorrer a imagem inteira na busca por um *patch* aproximado, ou que sejam usadas árvores (e.g., *VP-trees* [14]) e técnicas para redução da dimensionalidade (para diminuir o tempo de busca), o *PatchMatch* adota uma estratégia baseada em Propagação (*Propagation*) de boas correspondências para *patches* vizinhos e Busca Aleatória (*Random Search*) na imagem visando melhorar as correspondências encontradas.

O conceito de correspondência pode ser melhor compreendido por meio da Figura 3.1. Nela, estão ilustradas algumas correspondências entre *patches* de uma mesma imagem.

Nota-se que algumas delas são exatas (blocos idênticos), enquanto outras são apenas similares.

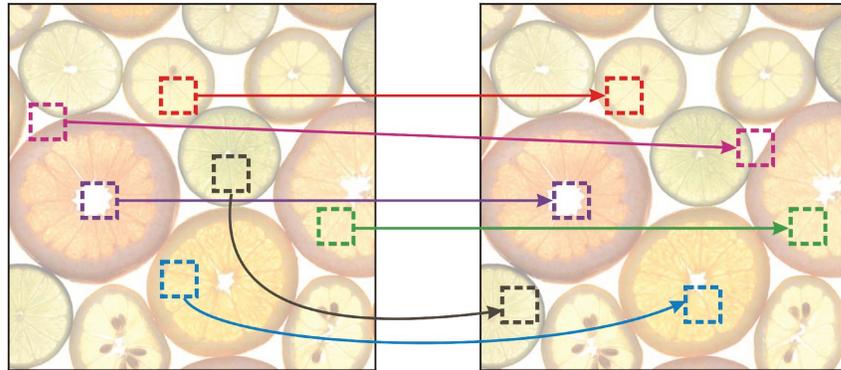


Figura 3.1: Exemplos de correspondências entre *patches*.

O resultado final de uma execução do *PatchMatch* é um conjunto de correspondências, uma para cada *patch* possível em uma dada imagem. Esse conjunto é denominado Espaço de Vizinhos mais Próximos (*Nearest Neighbor Field* – NNF), e pode ser entendido como uma matriz N , de dimensões idênticas à da imagem, em que cada elemento n_{ij} (exceto aqueles das bordas) é tomado como o centro de um *patch* da imagem em questão, e armazena a posição (x, y) relativa ao centro do *patch* que é a sua correspondência atual, além da distância entre eles. A Figura 3.2 ilustra o NNF para uma imagem.

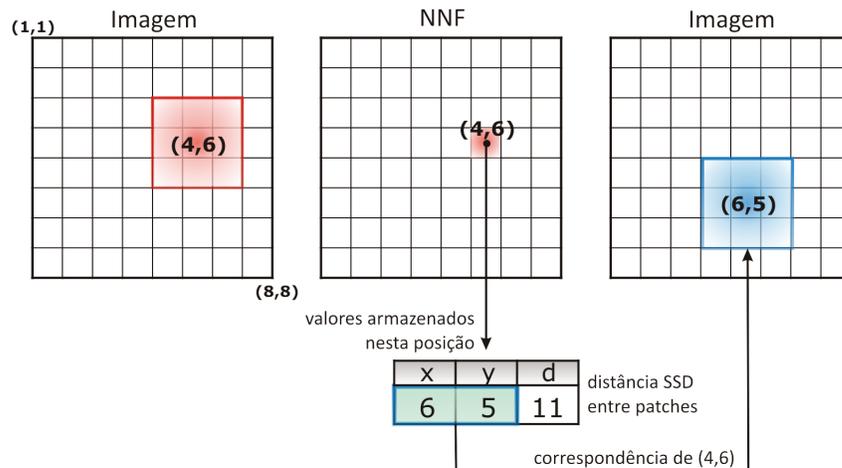


Figura 3.2: Representação do NNF para uma imagem. A localização do *patch* é dada pelas coordenadas de seu ponto central. Na figura, o *patch* (4,6) de tamanho 3×3 (em vermelho) da imagem original tem como correspondência o *patch* (6,5) (em azul), de mesmo tamanho. Logo, a posição (4,6) do NNF armazena os valores de $x = 6$, $y = 5$ e d (distância entre os *patches* vermelho e azul), aqui exemplificada como sendo igual a 11.

A métrica de distância utilizada no *PatchMatch* é a Soma dos Quadrados das Diferenças (*Sum of Squared Differences* – SSD). O cálculo é efetuado percorrendo-se os dois *patches* e calculando-se o quadrado da diferença entre os valores dos *pixels* correspondentes de ambos. A soma das diferenças quadradas entre todos os pares de *pixels* representa a SSD. Quanto mais próxima de zero, maior a similaridade entre os *patches*. A Equação 3.1 representa o cálculo da SSD em que W representa o domínio do *patch* P_1 , enquanto x e y são os deslocamentos (*offsets*) nas respectivas direções entre dois *patches* na imagem. A Figura 3.3 ilustra o cálculo da SSD para dois *patches* quaisquer.

$$\sum_{(i,j) \in W} (P_1(i, j) - P_2(x + i, y + j))^2. \quad (3.1)$$

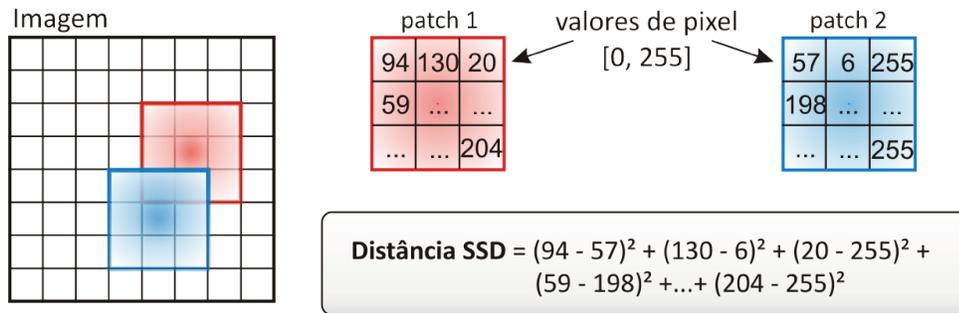


Figura 3.3: Cálculo da distância SSD entre dois *patches* correspondentes. No exemplo, a correspondência do *patch* vermelho é o *patch* de cor azul.

As etapas do algoritmo PatchMatch são descritas detalhadamente a seguir.

1. **Inicialização:** Nesta etapa, associamos uma correspondência a cada *patch* da imagem. Os autores mencionam que uma inicialização do NNF com *patches* aleatórios é suficiente para que o algoritmo gere resultados satisfatórios, porém, caso alguma informação sobre boas correspondências estiver disponível, ela pode ser utilizada. A Figura 3.4 ilustra a reconstrução de uma imagem a partir de um NNF aleatório. Para reconstruir uma imagem, percorremos o NNF em *scan order* e dispomos os *patches* indicados por ele (com sobreposições) numa imagem "vazia".
2. **Iteração:** Após a inicialização, o algoritmo executa etapas sucessivas para aprimoramento do NNF. A cada iteração, o método inspeciona o NNF e a imagem, percorrendo os *patches* de cima para baixo e da esquerda para a direita (*scan order*). Em seguida, cada *patch* é alvo de dois mecanismos de melhoria: propagação e busca aleatória.

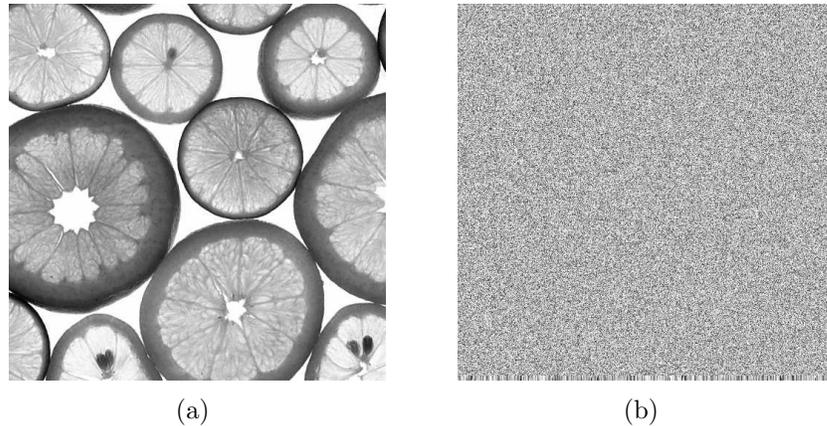


Figura 3.4: (a) Imagem de exemplo e (b) reconstrução desta usando o NMF aleatório inicial.

- (a) **Propagação:** Esta etapa objetiva aprimorar o NMF por meio da propagação de boas correspondências. Ela consiste na melhoria da correspondência encontrada para um *patch* (x, y) , a partir da avaliação dos mapeamentos encontrados para os *patches* localizados à esquerda $(x - 1, y)$ e acima $(x, y - 1)$ daquele (em iterações pares, o algoritmo verifica os *patches* à direita e abaixo). Por exemplo, se o *patch* da esquerda possui uma boa correspondência, associamos ao *patch* (x, y) tal correspondência movida de um *pixel* à direita. Este procedimento permite disseminar boas correspondências a regiões vizinhas e ajuda a manter a coerência natural da imagem, uma vez que as estruturas dos objetos da cena são consideradas. Caso seja encontrado um mapeamento satisfatório para um *patch* dentro de uma região R , os demais mapeamentos desta região são gradativamente melhorados tendo por base a boa correspondência. Esta etapa é esquematizada na Figura 3.5.
- (b) **Busca Aleatória:** Em algumas situações, faz-se necessária uma busca ao redor das correspondências encontradas, a fim de melhorá-las ou de evitar que elas permaneçam retidas em mínimos locais (correspondências muito próximas que se devem a regiões também muito parecidas da imagem). Isso pode ser efetuado testando-se uma sequência de *patches* aleatórios (na imagem) a uma distância exponencial decrescente da correspondência do *patch* (x, y) . Tal distância é, inicialmente, igual à maior distância do centro do *patch* em análise a uma das bordas da imagem. Caso um *patch* (escolhido aleatoriamente) melhore o mapeamento para (x, y) , ele passa a ser a nova correspondência deste. Esta etapa é feita para cada *patch* e prossegue até que o raio da região de busca seja menor do que 1 *pixel*. A Figura 3.6 ilustra a execução desta etapa.

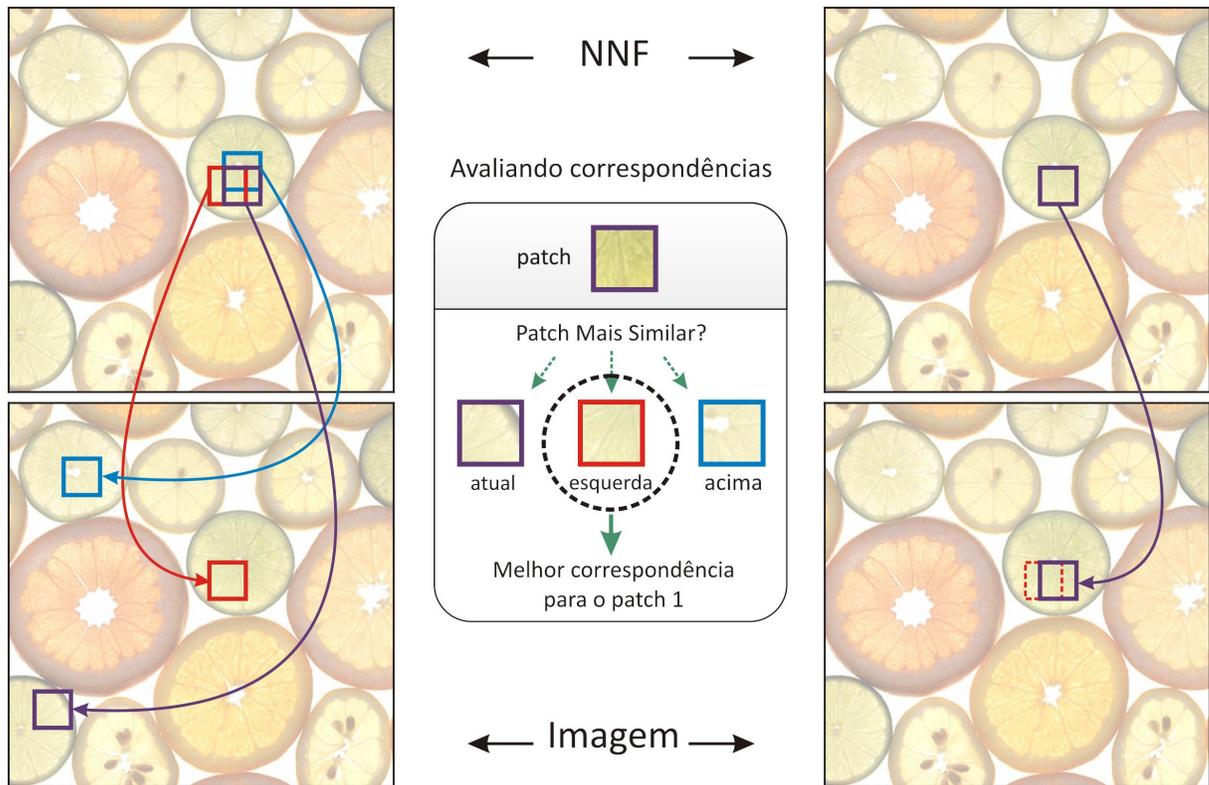


Figura 3.5: Esquemática da propagação para o *patch* de cor roxa. A imagem da esquerda apresenta os mapeamentos inspecionados (blocos à esquerda e acima). A imagem central evidencia as comparações efetuadas, visando descobrir se alguma das duas correspondências (vermelha ou azul) melhora o mapeamento para o *patch* roxo no NNF. A imagem da direita apresenta o resultado da propagação, na qual houve melhoria do mapeamento para o *patch* roxo utilizando a boa correspondência de seu vizinho à esquerda (vermelho) movida de 1 *pixel* à direita.

Entende-se por Convergência o momento em que o NNF alcança um mapeamento satisfatório para a maioria de seus *patches*. Em termos visuais, se reconstruíssemos a imagem com base no NNF final, teríamos uma imagem muito parecida com a original. Segundo Barnes et al. [3], o *PatchMatch* converge com alta probabilidade em, aproximadamente, cinco iterações. A Figura 3.7 apresenta alguns dos resultados obtidos. Nota-se que apesar de as reconstruções não serem perfeitas, o algoritmo consegue, a partir de uma inicialização completamente aleatória, encontrar correspondências próximas para cada um dos *patches* da imagem.

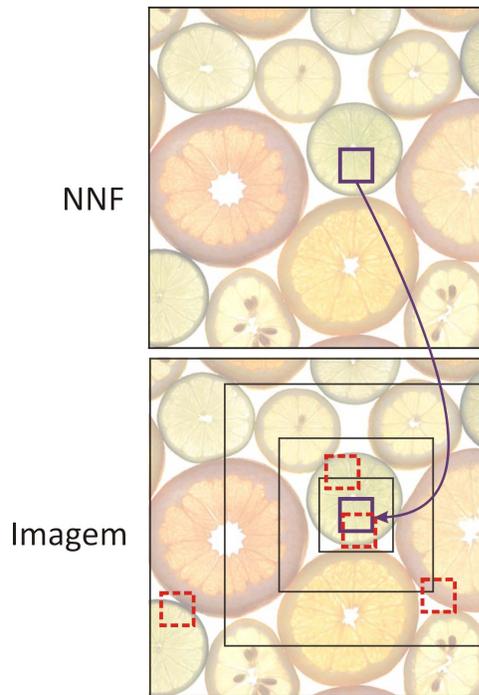


Figura 3.6: Busca Aleatória ao redor da nova correspondência encontrada, na propagação (Figura 3.5), para o *patch* roxo.

3.1.1 Aspectos Importantes

Em nossa implementação, um ponto importante a ser mencionado diz respeito à convergência obtida por nós, que foi inferior àquela relatada pelos autores em [3]. Acreditamos que o método de inicialização empregado no artigo pode ter influenciado diretamente neste resultado. Efetuamos uma tentativa de inicialização em multiescala, na qual o método foi executado para a imagem original redimensionada em 25% do seu tamanho original. Em seguida, utilizamos o NNF final encontrado para inicializar o NNF da imagem do nível acima na pirâmide (original redimensionada em 50%). Esse último foi utilizado na etapa de inicialização (para a imagem original). O resultado, no entanto, foi inferior ao obtido com a inicialização aleatória do NNF.

Outro ponto relevante diz respeito à busca aleatória, que gera resultados insatisfatórios (imagens reconstruídas ruidosas) quando aplicada na primeira iteração. Acreditamos que esta etapa atrapalha a etapa de propagação no início do processo, pois estabelece correspondências imprecisas para *patches* em um momento em que a similaridade entre estes ainda é baixa. Em resumo, se usada na primeira iteração a busca aleatória pode destruir o primeiro mapeamento efetivo encontrado para cada *patch* da imagem. A fim de contornar esta dificuldade, adaptamos o método para que essa etapa fosse executada somente nas demais iterações. Este fenômeno não foi relatado pelos autores do trabalho.

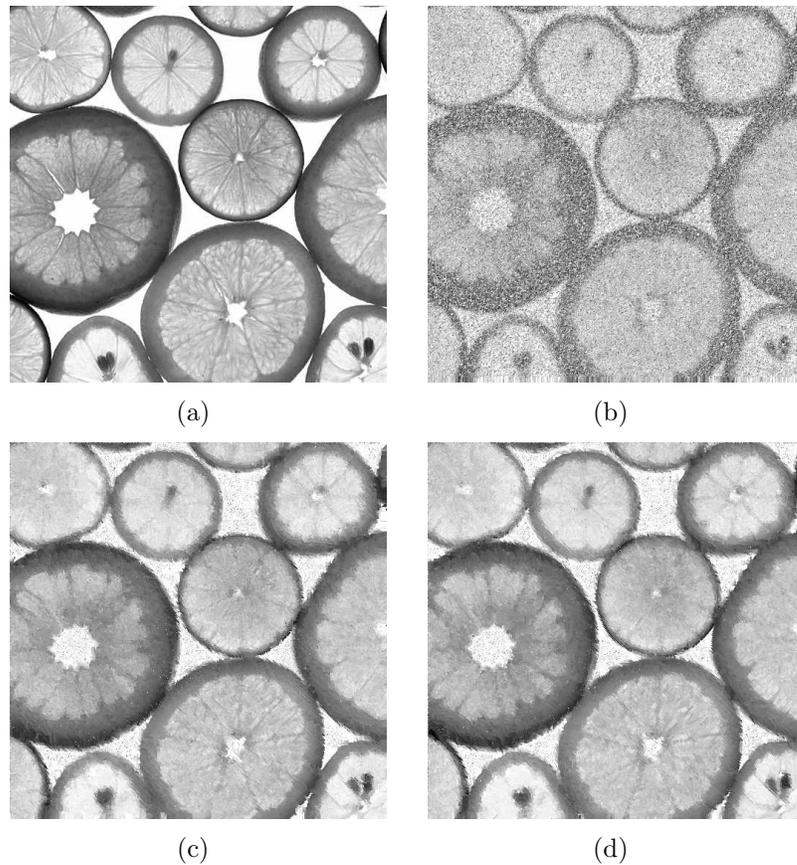


Figura 3.7: Resultados do *PatchMatch* na reconstrução da imagem (a) utilizando o NNF obtido após (b) 1, (c) 5 e (d) 10 iterações.

Finalmente, salientamos que em nossos experimentos utilizamos *patches* de tamanho 7×7 *pixels*, cinco iterações e imagens em tons de cinza de tamanho 512×512 *pixels*. Neste cenário, o tempo médio de execução foi inferior a dois minutos.

O *PatchMatch* foi implementado na linguagem C e faz uso de bibliotecas do OpenCV. O algoritmo não foi utilizado por nós na detecção de cópia-colagem dado que este não é seu objetivo e, sim, apenas como forma de cimentar o caminho para a implementação do algoritmo *PatchMatch Generalizado*, que mostraremos adiante.

3.2 PatchMatch Generalizado

A versão generalizada do *PatchMatch* recebeu esta denominação por se diferenciar do algoritmo original em alguns pontos. O primeiro e mais crucial se deve ao fato de o método encontrar não mais apenas uma correspondência para cada *patch* da imagem, e

sim um número pre-estabelecido K de correspondências. O *PatchMatch* efetua a busca do vizinho mais próximo (*Nearest Neighbor* — NN) para cada *patch*, isto é 1-NN. Em oposição, a versão mais geral busca por qualquer quantidade K de vizinhos mais próximos (K -NN). O segundo ponto, é referente à busca por *patches* similares em todas as escalas e orientações. Para isso, os autores adaptaram o método a fim de permitir a comparação de descritores SIFT, ao invés dos *pixels* de cada *patch*, apenas. Há, ainda, duas novas etapas: Enriquecimento Direto (*Forward Enrichment*) e Enriquecimento Inverso (*Inverse Enrichment*), que melhoram, em muito, as correspondências para os *patches*.

Dentre as principais diferenças internas entre os dois métodos, estão a utilização de novas estruturas de dados. São elas:

1. **Max-Heap:** Trata-se de uma estrutura para armazenamento de dados baseada no conceito de árvore binária e que possibilita recuperação e inserção eficiente de elementos. Em um Max-Heap, cada nó armazena os valores de x , y e da *distância*, e possui valor (*chave*) maior do que seus filhos à esquerda e à direita. O elemento com maior valor (de distância) no heap está localizado na raiz deste e pode ser recuperado em tempo $O(1)$, e extraído em tempo $O(\log n)$. Para inserir um elemento qualquer à estrutura gasta-se tempo $O(\log n)$. Assim, há um max-heap de K elementos para cada *patch* da imagem. Isso significa que o NNF transforma-se numa matriz de heaps.
2. **Hash:** Um Hash é utilizado para diminuir o tempo de busca por um determinado elemento dentro do max-heap. O objetivo é evitar que o heap contenha dois ou mais elementos idênticos (correspondências iguais). Por exemplo, se $(3, 5)$ é uma das K correspondências de $(7, 1)$, ela deve ser única dentro do heap de $(7, 1)$ e não deve ser adicionada novamente à esta estrutura. Em nossa implementação, entretanto, não fazemos uso de hash, pois acreditamos que a busca pode ser efetuada de forma rápida a partir de uma busca direta no max-heap, que não possui mais do que $K = 8$ elementos.
3. **Lista Encadeada:** Este tipo de estrutura é utilizado na etapa de Enriquecimento Inverso. Trata-se de uma lista encadeada simples, na qual cada elemento guarda os valores de x , y e da *distância* referentes a outro *patch*, e há uma lista encadeada para cada um destes.

A Figura 3.8 retrata a nova estrutura do NNF que contém, em cada posição, um max-heap e uma lista encadeada.

As etapas do método são detalhadas abaixo.

1. **Inicialização:** Esta etapa é similar àquela executada no *PatchMatch*, com a

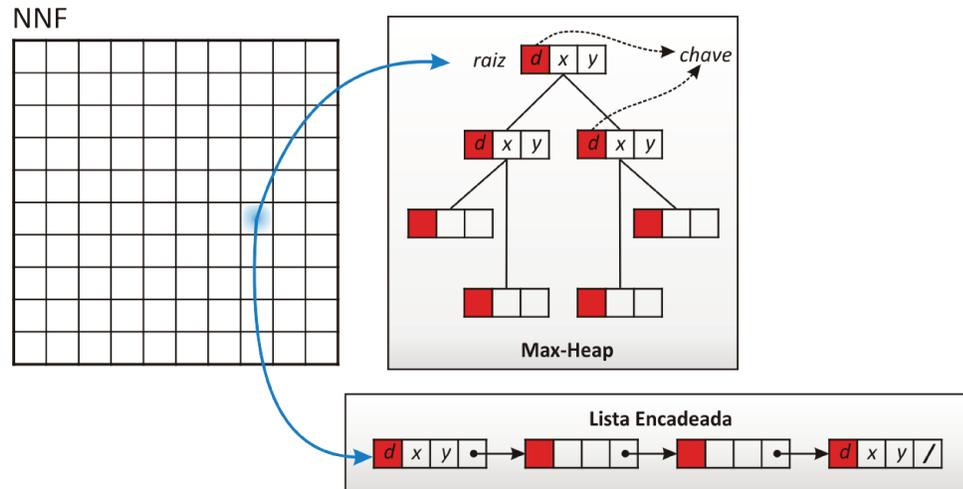


Figura 3.8: Nova estrutura do NNF, que passa a agregar um max-heap e uma lista encadeada para cada *patch* (posição) da matriz. Os elementos dessas estruturas são registros que armazenam três informações: as coordenadas x e y e a distância (d) entre o *patch* atual (quadrado azul do NNF) e sua correspondência. O valor da distância é a chave (em vermelho) usada para comparação e organização dos elementos do max-heap. Esta organização não se faz necessária dentro da lista encadeada.

diferença de que, para cada *patch* são associadas K correspondências aleatórias na imagem. As K correspondências de um dado *patch* devem ser distintas.

2. **Propagação:** Na propagação, a comparação entre as correspondências se dá entre o *patch* sendo examinado e os elementos dos max-heaps que possuem maior distância e que, portanto, estão localizados na raiz destes. Caso seja possível melhorar o mapeamento para um *patch*, extraímos a raiz do heap deste e inserimos a nova correspondência em tempo $O(\log n)$.
3. **Busca Aleatória:** Esta etapa é análoga à do *PatchMatch*, com a exceção de que as buscas são efetuadas ao redor de cada uma das K correspondências dos *patches*.
4. **Enriquecimento:** Consiste em uma otimização para o *PatchMatch* que visa propagar boas correspondências de um *patch* para o espaço de correspondências desse mesmo *patch* (em oposição à etapa de propagação, que efetua tal difusão através das dimensões espaciais da imagem). Logo, o método passa a considerar um conjunto mais rico de boas correspondências que podem melhorar o NNF. O enriquecimento é subdividido em direto e inverso.
 - **Enriquecimento Direto:** Examina as K correspondências de cada correspondência de um *patch* obtendo, portanto, um conjunto de K^2

possíveis candidatos para melhoria do respectivo mapeamento. A melhor correspondência entre as K^2 é inserida no heap do *patch* em questão, e a pior (raiz) é extraída.

- **Enriquecimento Inverso:** Produz um conjunto de potenciais boas correspondências a partir da inspeção, para um determinado *patch* A , de todos os *patches* que apontam para A . Se algum desses *patches* puder melhorar o mapeamento atual, isto é, se ele for melhor que a correspondência localizada na raiz do heap de A , ele será inserido neste heap (após a extração da raiz). De acordo com os autores, os melhores resultados são obtidos combinando-se os enriquecimentos inverso e direto, nesta ordem. Comprovamos a eficácia desta nova otimização, que aprimorou o NNF de maneira acentuada. Além disso, a ordem estipulada para a execução desta etapa (enriquecimento inverso e depois direto) influenciou positivamente nos resultados, que se apresentaram inferiores quando a ordem contrária foi considerada.

O *PatchMatch Generalizado* pode acomodar diversas métricas de similaridade (distância) entre *patches*, bem como é facilmente adaptável para o emprego de descritores de imagens. Em [4], Barnes et al. utilizam as métricas SSD, L^2 , além de descritores SIFT na detecção de clonagens, detecção de simetrias e de objetos específicos em imagens, respectivamente. Nestas duas últimas tarefas, um descritor SIFT [18] é amostrado em cada *pixel* da imagem e o algoritmo gera bons resultados na identificação de objetos em diferentes orientações e escalas.

3.2.1 Aspectos Importantes

Alguns aspectos pertinentes quanto a este algoritmo merecem ser salientados. São eles:

1. As etapas de Enriquecimento são cruciais para a obtenção de bons resultados. Apesar de custosas computacionalmente, são imprescindíveis no aprimoramento do NNF;
2. Como esperado, o algoritmo demora mais tempo para execução do que sua versão 1-NN. O tempo médio gasto para uma imagem de tamanho 512×512 *pixels* e três iterações é de aproximadamente cinco minutos;
3. A reconstrução da imagem com base no NNF é quase perfeita, como mostra a Figura 3.9 mas, para isso, é necessário armazenar a posição (dentro do max-heap) da melhor correspondência encontrada para cada *patch*, uma vez que o max-heap provê meios de recuperação em tempo constante apenas da correspondência com maior distância ao *patch* (situada na raiz).

4. O tamanho do *patch* considerado nos experimentos variou de 7×7 a 13×13 *pixels*, entretanto, o resultado final não sofre mudanças significativas.
5. O algoritmo converge de forma satisfatória em até três iterações para K igual a 5.

Implementamos este algoritmo na linguagem C e utilizamos bibliotecas do OpenCV para manuseio de imagens e extração de descritores (mostrada mais à frente). Os testes foram efetuados em imagens em tons de cinza e coloridas. A Figura 3.9 ilustra alguns dos resultados obtidos com o método na reconstrução de uma imagem.

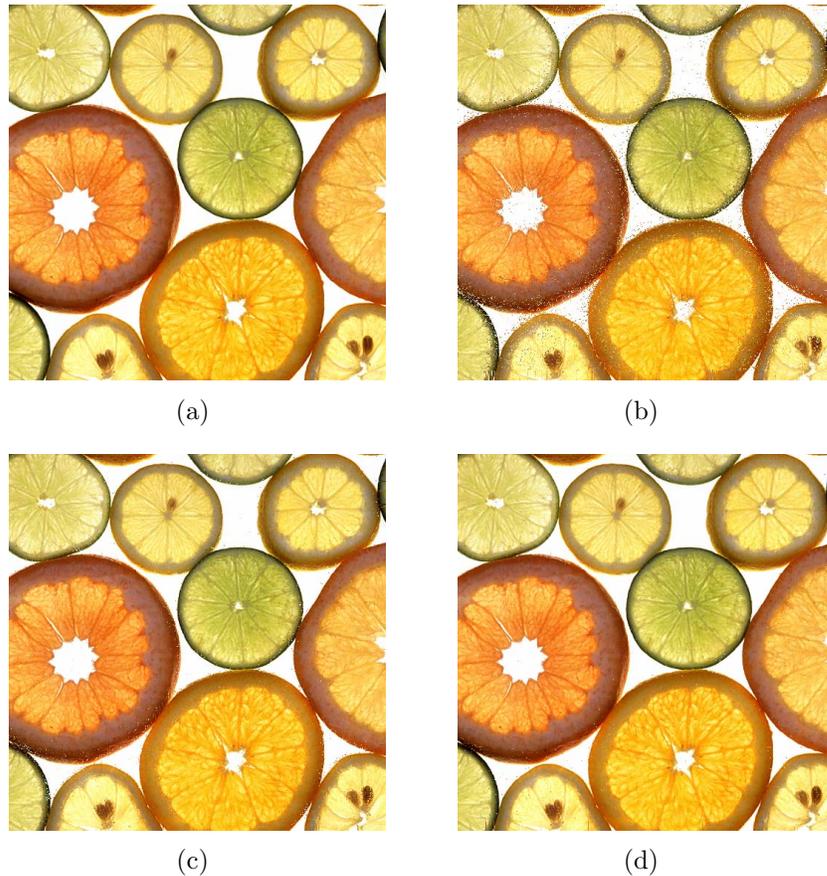


Figura 3.9: Resultados do *PatchMatch Generalizado* na reconstrução da imagem (a) utilizando o NNF obtido após (b) 2 iterações com $K = 3$ e após (c) 2 e (d) 5 iterações com $K = 5$.

Na Seção 3.3 a seguir, detalhamos a nossa primeira contribuição referente a um método para detecção de clonagens que faz emprego do algoritmo *PatchMatch Generalizado*.

3.3 Detecção de cópia-colagem usando o PatchMatch Generalizado

Em [4], os autores propuseram aplicar o *PatchMatch Generalizado* na tarefa de revelar a existência de manipulações de cópia-colagem em imagens em razão de o método ser capaz de encontrar, para cada *patch*, um conjunto de K boas correspondências. Com isso, é possível apontar os blocos de *pixels* duplicados na cena, revelando a manipulação. O método examina o NNF adquirido com a execução do *PatchMatch Generalizado*, percorrendo-o da esquerda para a direita e de cima para baixo. A imagem original também passa a ser vista como um grafo, em que cada posição (x, y) é um nó deste. Em seguida, compara-se o conjunto de correspondências de cada *patch* com as correspondências de seus vizinhos (*patches* abaixo e à direita no NNF). Caso os conjuntos de correspondências sejam similares, uma aresta entre os dois *patches* é traçada no grafo. Dois conjuntos A e B de correspondências são similares se, para qualquer par de *patches* $(ax, ay) \in A$ e $(bx, by) \in B$, estes se encontram a uma distância física máxima T um do outro, e ambos possuem uma distância de *patch* (SSD) menor do que um limiar D máximo.

Os autores efetuaram experimentos com clonagens envolvendo compressões JPEG, ruídos e suavizações, mas não detalharam os testes realizados. Por isso, não é possível saber exatamente em quais cenários o algoritmo funciona com eficácia. A única informação mencionada é que esta estratégia não é robusta a tais tipos de manipulação. Nas imagens de exemplo apresentadas, a região duplicada foi encontrada quase que totalmente, mas não sabemos se a imagem havia sido comprimida em JPEG, por exemplo. Finalmente, os autores não fazem alusão aos valores dos limiares estabelecidos.

Implementamos esta estratégia e a testamos para valores de limiares diversificados, porém, não foi possível reproduzir com precisão os resultados dos autores. A detecção frequentemente apresentou elevada quantidade de falsos-positivos e falsos-negativos. Analisando o método e suas possíveis implicações nos resultados finais, concluímos que ou há algo incorreto na descrição da estratégia ou faltam informações. Assim, acreditamos que a abordagem, como descrita no artigo, não é suficiente para se detectar corretamente regiões duplicadas em imagens.

Apesar dos resultados insatisfatórios alcançados com aquela abordagem, acreditávamos que o *PatchMatch Generalizado* tinha potencial para aplicação em nossa pesquisa. Desenvolvemos, então, um método de detecção de clonagens a partir de um raciocínio inovador. Em resumo, o NNF obtido com a execução do *PatchMatch Generalizado* é visto como detentor de um conjunto de potenciais candidatos à duplicação para cada *patch* da imagem em análise. O funcionamento do método é descrito a seguir e esquematizado na Figura 3.10.

Detecção de cópia-colagem usando o PatchMatch Generalizado

1. Empregamos o *PatchMatch Generalizado* na busca pelas K correspondências para cada *patch* da imagem. A inicialização do NNF é aleatória;
2. Percorremos o NNF final obtido da esquerda para a direita e de cima para baixo;
3. Visitamos o max-heap de correspondências de cada *patch* (x, y) ;
4. Caso uma correspondência (x_i, y_i) , sendo $i \leq K$, de (x, y) esteja a uma distância física deste inferior a um limiar T , não prosseguimos com a análise e passamos para a correspondência seguinte. A justificativa para isso é que, após a execução do *PatchMatch Generalizado*, o conjunto de correspondências candidatas para (x, y) poderá incluir ela própria e outras localizadas na vizinhança desta (devido à etapa de propagação). Tais regiões candidatas não são de nosso interesse. Eventualmente, entretanto, o algoritmo encontrará correspondências similares a (x, y) na região pertencente à clonagem;
5. Comparamos a região ao redor do *patch* (x, y) com a região ao redor de cada uma de suas correspondências (x_i, y_i) . Esta região de comparação abrange o *patch* mais um acréscimo de dois *pixels* em todas as direções. Caso uma das regiões não possa crescer em quaisquer direções (borda da imagem), a outra região a ser comparada também não poderá crescer naquelas direções. Por exemplo, se (x, y) não puder ser ampliado à direita, a mesma restrição se aplicará a (x_i, y_i) e vice-versa;
6. Se as duas regiões (ao redor de (x, y) e (x_i, y_i)) forem similares, isto é, se a distância de similaridade entre elas for menor do que um limiar D , marcamos as duas regiões como duplicadas no mapa de detecção;
7. Se uma região já foi marcada, ela não é examinada novamente.

Inicialmente, a métrica de similaridade adotada foi a SSD. Porém, esta medida não é robusta a pequenas alterações nas regiões duplicadas. Assim, decidimos empregar uma métrica baseada em comparação de histogramas, a qual aprimorou os resultados de detecção. O cálculo da distância, de acordo com esta métrica, é efetuado de maneira diferente para imagens em tons de cinza e coloridas. No primeiro caso, o histograma de níveis de cinza para cada *patch* é computado. Tais histogramas variam de 0 a 255 valores de *pixels*. Considere A e B como sendo os histogramas do primeiro e segundo *patches*, respectivamente. Calculamos o módulo da subtração entre o valor da posição $x_a \in A$ e o valor da posição $x_b \in B$, e somamos todos esses valores. Este cálculo é denominado Soma das Diferenças Absolutas (*Sum of Absolute Differences* – SAD). O valor desta soma é a métrica de similaridade para os *patches* em tons de cinza.

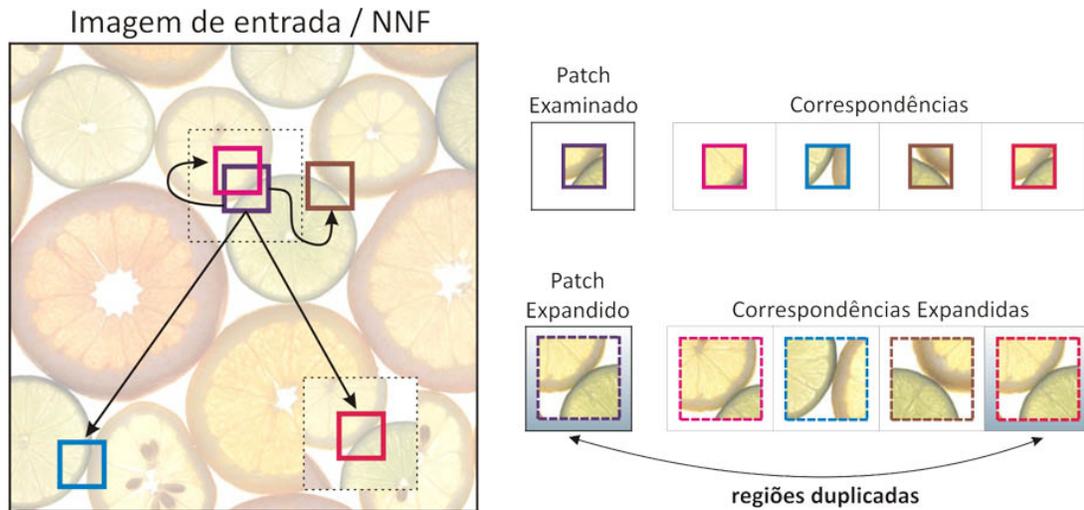


Figura 3.10: Esquematização da nossa proposta para detecção de clonagens baseada no *PatchMatch Generalizado*. O método examina todos os conjuntos de correspondências do NNF em busca daquelas que se encontram em uma vizinhança similar à vizinhança de um *patch* (x, y) . Na Figura, a vizinhança do *patch* de cor roxa sendo avaliado é similar à vizinhança do *patch* de cor vermelha (duplicação).

Caso a imagem possua mais de um canal de cor, como é o caso do espaço RGB (*Red*, *Green* e *Blue*), calculamos um histograma para cada canal, e cada *patch* passa a ser descrito por três histogramas. Em seguida, efetuamos um procedimento análogo ao realizado para os *patches* em tons de cinza, calculando três valores de SAD entre os histogramas referentes ao mesmo canal de cor. Finalmente, obtemos um único valor aplicando cada soma obtida na Equação 3.2 a seguir, usada na conversão de valores RGB para tons de cinza. Os valores de R, G e B são os obtidos com o cálculo da SAD para os histogramas vermelho, verde e azul, respectivamente.

$$I = 0.299R + 0.587G + 0.114B. \quad (3.2)$$

A abordagem para detecção de clonagens utilizando o *PatchMatch Generalizado* acima descrita, apesar de pouco eficaz em casos de operações que alteram os valores dos *pixels* duplicados, funciona para os cenários de rotações de 90, 180 e 270 graus e espelhamentos horizontais e verticais, em decorrência da utilização de histogramas de cor (que são invariantes a tais operações) para representação dos blocos inspecionados. No Capítulo 5, apresentamos mais discussões a respeito desta abordagem, tais como os valores de limiares estabelecidos, limitações e resultados.

3.4 Considerações Finais

Neste capítulo, nós abordamos os algoritmos *PatchMatch Original* e *Generalizado* e discutimos pontos relevantes advindos da análise das nossas implementações desses algoritmos. Apresentamos, também, nossa primeira contribuição à detecção de clonagens baseada nas ideias desses métodos. No Capítulo 4, descrevemos uma nova abordagem para identificação de cópia-colagem baseada em processos de votação e análise multiescala de regiões correspondentes na imagem. Essa proposta é robusta a rotações, a redimensionamentos e à compressão JPEG.

Capítulo 4

Detecção de cópia-colagem baseada em Votação e Análise Multiescala

Neste capítulo, nós descrevemos nossa contribuição mais importante para detecção de cópia-colagem em imagens digitais. Nossa abordagem explora e agrega características de determinados trabalhos que focam o problema da detecção nos cenários de rotações, redimensionamentos e compressões JPEG. Apresentamos uma análise mais aprofundada sobre as abordagens verificadas nesses trabalhos. A inovação da nossa metodologia parte do princípio de que tais trabalhos podem unir forças e se complementar em busca de uma detecção de cópia-colagem mais precisa. Para isso, utilizamos uma estratégia eficaz envolvendo Vizinhos mais Próximos pela Razão da Distância (*Nearest Neighbor Distance Ratio* – NNDR), na busca por pontos de interesse correspondentes, votação e análise multiescala de regiões ao redor desses pontos.

4.1 Abordagens Inspiradoras

A utilização de descritores de imagem para detecção de cópia-colagem é uma estratégia que gera bons resultados. Os trabalhos de Huang et al. [13], Pan e Lyu [23] e Xu et al. [36] mostrados no Capítulo 2 fornecem evidências disso. As metodologias apresentadas pelos autores consistem na extração e comparação de pontos SIFT [18] ou SURF [5] na imagem, visando àqueles que possuem descrições similares. A descrição de um ponto é armazenada em um vetor de características de, usualmente, 128 dimensões. Mesmo este valor sendo maior do que a dimensão de um bloco de 8×8 *pixels* linearizado, por exemplo, o número de pontos a serem comparados é muito menor que o total de blocos com sobreposição que uma imagem pode ter. Além desta vantagem, pontos SIFT e SURF são robustos a transformações de escala, rotação entre outras, permitindo a identificação de objetos numa imagem e, no caso de clonagens, a identificação de regiões semelhantes. Contudo, a não

geração de mapas de detecção, o que permitiria a comparação visual e quantitativa com outras abordagens, é um ponto fraco desses métodos. Regiões duplicadas parcialmente homogêneas (e.g., céu azul) também podem ser de difícil localização, em razão de não conterem pontos SIFT ou SURF em quantidades suficientes. Finalmente, os trabalhos [13, 23, 36] não realizam experimentos detalhados com as operações de rotação e escala, não sendo possível, portanto, obter uma estimativa das taxas de falsos-positivos e de acurácia que os métodos propostos resultam nestas condições.

Outra ideia promissora é aquela descrita em Wang et al. [34]. O estabelecimento de uma janela deslizante em formato circular faz muito sentido na busca por segmentos duplicados e rotacionados, pois os *pixels* próximos ao vértices das convencionais janelas quadradas influenciam negativamente na comparação de blocos. Mesmo assim, o uso de tal janela não é suficiente para a detecção daqueles segmentos. Os autores adotam uma descrição dos blocos circulares com base nas médias dos quatro círculos concêntricos que podem ser extraídos desses. A janela deslizante possui raio igual a quatro e o menor círculo concêntrico possui raio igual a um. Comparações diretas entre os valores de *pixel* dos blocos não são efetuadas. Há também uma redução no tamanho da imagem em 25% que, segundo os autores, é crucial para o processo de detecção. Esta abordagem reduz o número de comparações efetuadas e melhora os resultados de detecção nos cenários supracitados.

A seção a seguir apresenta a nossa metodologia em detalhes.

4.2 Votação e Análise Multiescala de regiões de pontos de interesse

Neste trabalho, nós introduzimos uma nova metodologia para detecção de cópia-colagem que consiste na análise multiescala da imagem suspeita juntamente com um procedimento de votação para decidir se uma dada região da imagem faz parte de uma área adulterada. Na prática, a ideia do método é proporcionar uma análise computacional de uma imagem suspeita aos indivíduos envolvidos na averiguação da genuinidade desta. O nosso método é capaz de apontar regiões fortemente candidatas à clonagem dentro de uma imagem, o que será de grande valia para diminuir, agilizar ou assegurar o trabalho realizado por peritos.

O processo de detecção que apresentamos segue por sete etapas. São elas: (i) Pré-processamento da imagem; (ii) Detecção de pontos de interesse; (iii) Pareamento de pontos de interesse; (iv) Agrupamento; (v) Decomposição Piramidal; (vi) Análise Multiescala e (vii) Votação. A Figura 4.1 apresenta uma esquematização geral com todas as etapas de nossa metodologia. Nós desdobramos cada uma dessas etapas a seguir.

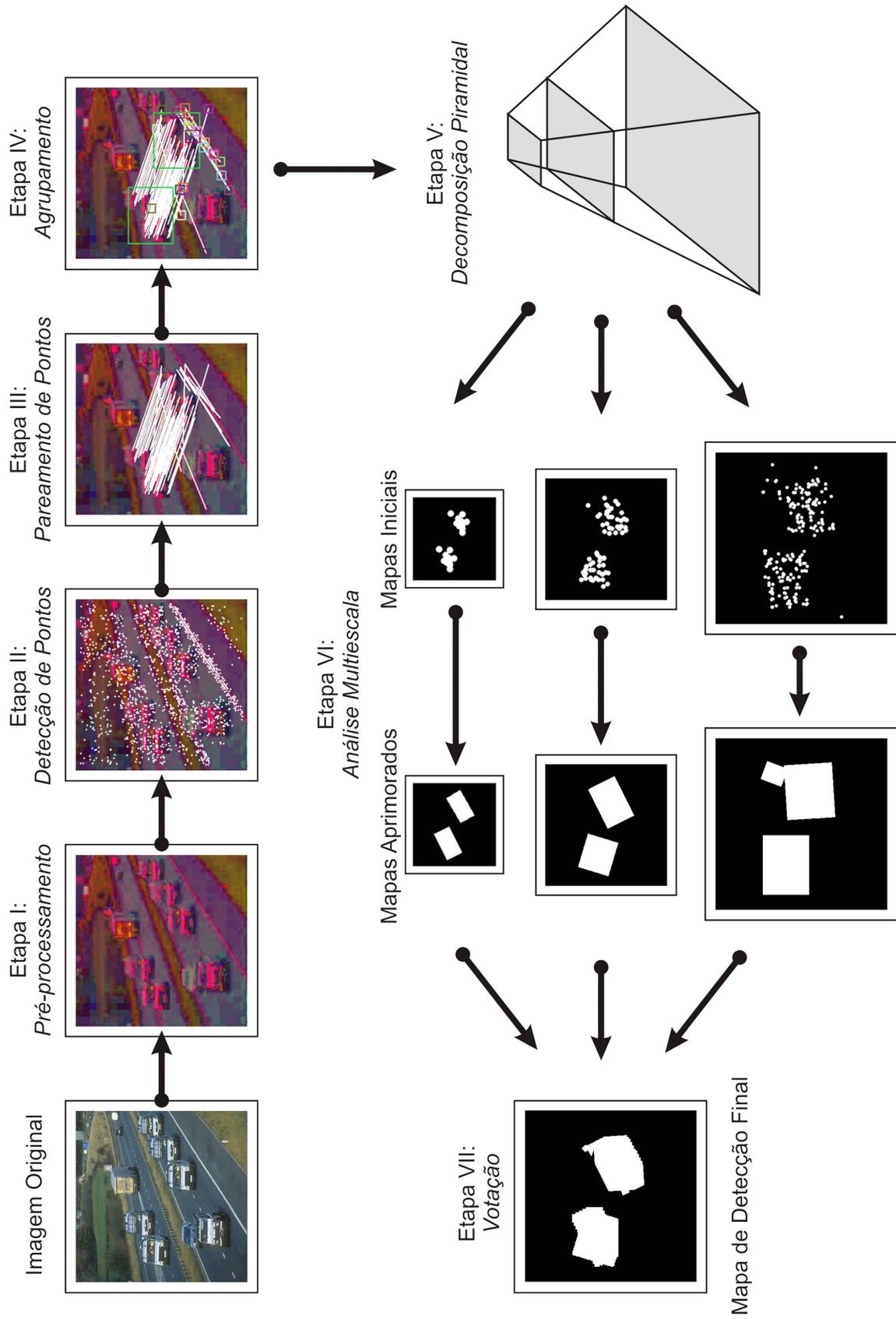


Figura 4.1: Esquemática completa da nossa metodologia.

4.2.1 Etapa 1: Pré-processamento da imagem

A primeira transformação pela qual a imagem em análise é submetida é a conversão entre espaços de cores RGB e HSV. O espaço de cor HSV constituído pelas componentes Matiz (*Hue* – H), cuja função é representar todas as cores tradicionais do espectro (vermelho, laranja, amarelo, verde e azul) mais as suas variações de tonalidade; Saturação (*Saturation* – S), que mede a pureza da cor (valores baixos tornam a cor próxima do cinza); e Brilho (*Value* – V), que indica a intensidade da cor. O espaço HSV pode ser representado no espaço tridimensional por um cone, como mostra a Figura 4.2. Nele, o eixo vertical sinaliza as variações de intensidade. Em uma visualização bidimensional da base deste cone encontramos os valores de matiz, que é o ângulo entre o eixo central do círculo e uma das cores do espectro localizadas na borda, e de saturação, que é entendida como a distância radial até a borda [30].

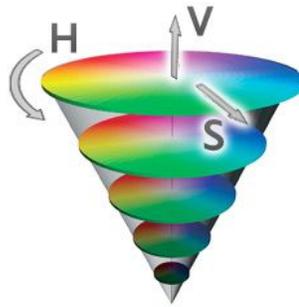


Figura 4.2: Representação tridimensional do espaço de cor HSV (extraída de [1]).

Esta transformação teve como objetivo reduzir o número de falsos-positivos do nosso método localizados, principalmente, em regiões com características de homogeneidade (e.g., céu azul, paredes sem textura, asfalto, regiões sob forte iluminação etc.). Acreditamos que a avaliação das clonagens no espaço HSV favorece a diferenciação entre regiões duplicadas e aquelas naturalmente similares. A explicação para isso pode estar relacionada à informação de intensidade (V), que no espaço HSV encontra-se separada da informação de cor (HS). Isso indica, por exemplo, que duas regiões consideradas erroneamente como duplicadas no espaço RGB podem não o ser em um espaço de cor que trata a intensidade de seus *pixels* de forma independente. Apesar de as regiões conterem semelhanças visuais, quando avaliamos suas informações de cor e intensidade separadamente as diferenças entre elas tornam-se mais perceptíveis.

Para conversão entre os espaços de cores RGB e HSV, nós utilizamos a biblioteca OpenCV, assim como no restante da implementação. O método de conversão disponibilizado na biblioteca, contudo, gera valores de matiz compreendidos entre 0 e 179, ao passo que os demais canais apresentam *dynamic ranges* (DR) variando de 0 a 255. Tal fato aponta para uma possível tendência no favorecimento dos canais S e V em detrimento do

canal H. Dado que as comparações de distâncias nos três canais devem pesar igualmente no cálculo final, nós normalizamos os valores de H, S e V com base no maior e no menor valor de todas as componentes. O procedimento é executado internamente aos blocos circulares de inspeção.

A normalização segue o exemplo mostrado na Equação 4.1. Nela, p_H e p'_H representam os valores do canal H de um *pixel* p antes e após a normalização, respectivamente. As variáveis $\min(HSV)$ e $\max(HSV)$ representam o menor e o maior valor dentre todas as componentes dos *pixels* referentes ao bloco circular sendo investigado.

$$p'_H = \frac{p_H - \min(HSV)}{\max(HSV) - \min(HSV)}. \quad (4.1)$$

4.2.2 Etapa 2: Detecção de pontos de interesse

Nesta etapa, objetivamos coletar pontos característicos (*keypoints*) que apresentam invariância a diversas operações, tais como escala e rotação, na imagem. Utilizamos a técnica *Speeded-Up Robust Features* (SURF) [5], que possui um detector e um descritor de pontos com essa finalidade.

Para detecção de pontos, o SURF emprega uma aproximação da matriz de Hessian, em conjunto com filtros caixa (*box filters*), para o cálculo de derivadas Gaussianas de segunda ordem. As convoluções da imagem com esses filtros podem ser avaliadas por meio de Imagens Integrais (*Integral Images*) a um custo computacional muito baixo. A ideia é que, ao invés de convoluir o filtro com a imagem em suas diferentes escalas, este é ampliado gradativamente e convoluído apenas com a imagem em sua escala original. Assim, é possível aplicar filtros caixa de qualquer tamanho na imagem a velocidades idênticas. Finalmente, os pontos de interesse são localizados no espaço de escalas da imagem, que é comumente implementado usando-se uma pirâmide de imagens. Para isso, aplica-se uma supressão não-maximal em uma vizinhança de tamanho $3 \times 3 \times 3$. Os pontos de interesse candidatos são selecionados encontrando-se os máximos locais do determinante de Hessian nas várias escalas [5, 36].

A descrição consiste na construção de um vetor de características para cada ponto de interesse encontrado. Tal descrição deve indicar a invariância do ponto a operações de translação, escala e rotação. No último caso, o algoritmo estima a orientação dominante dos pontos da seguinte forma:

- Calcula-se o valor das respostas das *wavelets* de Haar nas direções x e y em uma região circular definida ao redor e centrada no ponto inspecionado;
- Os valores das respostas são ponderados usando uma Gaussiana;

- Uma janela de tamanho $\frac{\pi}{3}$ é deslizada sobre as respostas Haar, e os valores nas direções x e y que estiverem contidos nessa janela são somados separadamente;
- O resultado das somas é usado na determinação de um vetor, que representará a orientação local.

A fase final do processo envolve a estipulação de uma região quadrada ao redor e centrada em um dado ponto. A orientação de tal região deve estar em conformidade com a orientação previamente determinada. O próximo passo é dividir a região em quatro partes menores, o que auxilia a preservação de informações espaciais. Em cada região, calcula-se novamente as respostas da *wavelet* de Haar em pontos regularmente espaçados. As respostas nas direções vertical e horizontal (conforme a orientação local) são ponderadas por uma Gaussiana, e seus valores são somados separadamente dentro de cada subregião definida. Finalmente, o descritor é construído extraíndo-se informações de cada sub-região com base nas somas calculadas [5, 36].

A extração dos pontos é efetuada sobre a imagem convertida primeiramente para o domínio HSV e depois para escala de cinza, pois o SURF considera apenas informações de intensidade. As etapas posteriores, no entanto, trabalham sobre a imagem original em HSV (a informação de cor, se existir, não é descartada). Finalmente, nesta etapa encontramos todos os pontos possíveis com o detector SURF que utilizamos. A Figura 4.3 (a) apresenta os pontos característicos SURF extraídos em uma imagem. Nós mostramos os pontos na imagem original (colorida), mas a detecção é realizada no espaço de intensidades (escala de cinza) da imagem.

4.2.3 Etapa 3: Pareamento de pontos de interesse

Tendo em mãos os pontos de interesse e suas respectivas descrições, nós encontramos seus pareamentos, se existirem. Idealmente, um ponto deve conter apenas uma correspondência na imagem. Nossa metodologia contempla a utilização do algoritmo de busca pelo vizinho mais próximo pela razão da distância (*Nearest Neighbor Distance Ratio* – NNDR) [21].

Tradicionalmente, dado um ponto A , o Vizinho mais Próximo (*Nearest Neighbor* – NN) deste pode ser encontrado comparando-o com todos os demais pontos. Um pareamento é efetivo quando a distância entre os descritores de A e de outro ponto B é inferior a um limiar de similaridade. Na ocorrência de múltiplos pareamentos, apenas o melhor é considerado. Um problema deste método é que o limiar (usado na diminuição dos falsos-positivos) pode ser difícil de ser estabelecido, em virtude das variadas transformações que a imagem (e os segmentos duplicados) pode sofrer [35, 31]. Dessa forma, a técnica pode prover resultados pouco efetivos.

A NNDR é uma alternativa favorável à superação do problema apresentado pela abordagem de vizinho mais próximo. Na heurística NNDR, o limiar é aplicado à razão entre as

distâncias de A para seu vizinho mais próximo e para seu segundo vizinho mais próximo. Isto indica que o ponto A terá como correspondência seu vizinho mais próximo B , somente se a distância entre eles for inferior à distância entre A e C (o segundo mais similar) multiplicada por uma constante [35]. A ideia é que as correspondências devem ser muito próximas em comparação com a correspondência incorreta mais próxima, isto é, pontos que possuem similaridades elevadas com vários outros são penalizados [21]. A Equação 4.2 mostra o cálculo da NNDR, no qual D_{sim} é a medida de distância usada no cálculo da similaridade entre os pontos A e B (vizinhos mais próximos) e A e C (segunda melhor correspondência).

$$Score = \frac{D_{sim}(A, B)}{D_{sim}(A, C)}. \quad (4.2)$$

A métrica de distância de similaridade que utilizamos foi a Euclidiana. Adicionalmente, para evitar o estabelecimento de correspondências entre pontos muito próximos fisicamente na imagem, o que pode elevar o número de falsos-positivos no problema de detecção de clonagem, aplicamos um limiar a esta medida. A Figura 4.3 (b) apresenta os pareamentos encontrados para os pontos de interesse detectados em uma imagem.

4.2.4 Etapa 4: Agrupamento

A finalidade desta etapa é diminuir o espaço de busca por regiões duplicadas por meio da reunião de pontos em grupos com base em suas correspondências. Além disso, uma análise posterior da imagem (Etapa 6) será efetuada sobre regiões correspondentes fortemente candidatas a possuir uma cópia-colagem.

Um agrupamento gera duas regiões que se correspondem: origem e destino. Nós analisamos as seguintes variáveis para realização deste procedimento:

- A proximidade (física) entre os pontos de um mesmo grupo. Para pertencer ao um grupo, um ponto deve estar a uma distância física do ponto mais próximo (referente ao grupo em questão) menor do que a estabelecida por um limiar. É importante lembrar que a correspondência entre dois pontos é mútua e, portanto, faz-se necessário analisar ambos os pontos que a integram.
- O ângulo formado entre a reta que liga os pontos correspondentes e o eixo horizontal da imagem. As regiões devem possuir correspondências com ângulos dentro de intervalos de orientações predefinidos (e.g., com 4 partições, teríamos os intervalos $[0^\circ, 89^\circ]$, $[90^\circ, 179^\circ]$, $[180^\circ, 269^\circ]$ e $[270^\circ, 359^\circ]$). Isso sugere que pontos próximos ao grupo, mas com ângulo de correspondência fora do intervalo estabelecido, devem fazer parte de outro grupo. Esta variável é um mecanismo que confere robustez ao método para o caso da existência de segmentos rotacionados.



(a)



(b)

Figura 4.3: Ilustração das etapas 2 e 3. (a) pontos de interesse de uma imagem após aplicação do detector SURF; (b) correspondências entre pontos de interesse.

O tamanho dos grupos de origem e destino depende da distribuição espacial de seus pontos. As regiões devem englobar todos os pontos que integram seu grupo. Isso é alcançado delimitando-se os grupos (origem e destino) em uma janela retangular. A largura desta janela é igual à maior distância horizontal entre dois pontos pertencentes ao grupo; o mesmo é válido para a altura (maior distância vertical). Em seguida, amplia-se esta janela em todas as direções de um valor predeterminado. O objetivo é agregar à região delimitada segmentos que porventura façam parte da duplicação e nos quais o detector não assinalou pontos de interesse. A Figura 4.4 ilustra o funcionamento desta etapa.

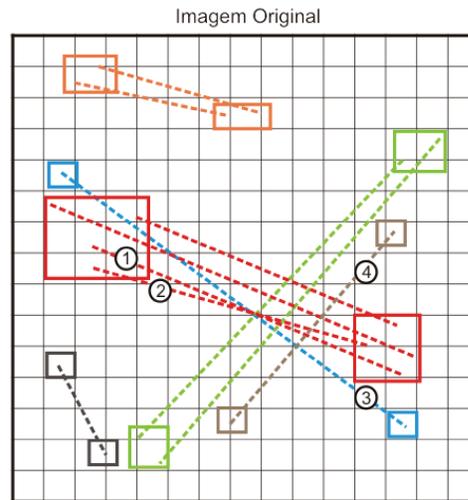


Figura 4.4: Exemplo de agrupamentos. Considere que os ângulos formados pelas retas pontilhadas 1 e 2 com o eixo horizontal da imagem pertencem ao mesmo intervalo. Dado que há proximidade física entre os pontos ligados por elas (correspondências), podem ser formados um grupo de origem e um de destino, ilustrados na figura pela cor vermelha. Em outra situação, os pontos ligados pela reta 3 (azul) não podem fazer parte do grupo vermelho (apesar da proximidade física com este), pois a orientação da reta não integra o intervalo de orientações requerido para a participação nesse grupo. Finalmente, a orientação da reta 4 (marrom) poderia permitir a entrada dos pontos correspondentes para o grupo verde, mas isso não ocorre devido à falta de proximidade física para os pontos deste.

4.2.5 Etapa 5: Decomposição Piramidal

A análise multiescala (Etapa 6) proposta por nós está diretamente relacionada à Decomposição Piramidal da imagem de entrada. Esta técnica é amplamente usada em aplicações de Visão Computacional. Nela, a imagem é suavizada e redimensionada sucessivamente até o nível desejado. O resultado é uma coleção de imagens que, se empilhadas da maior para a menor, formam uma espécie de pirâmide, como retrata a Figura 4.5.

O processo para reduzir uma imagem é feito por meio da Decomposição Piramidal Gaussiana (*Gaussian Pyramid Decomposition* – GPD). Inicialmente, um filtro Gaussiano é utilizado para suavizar a imagem. Em seguida, as linhas e colunas pares desta são removidas. A imagem decomposta possui densidade de *pixels* igual a 25% da imagem original. A próxima decomposição é efetuada sobre esta nova representação e assim sucessivamente para os níveis posteriores [7]. Todas essas imagens são examinadas separadamente na Etapa 6, a seguir.

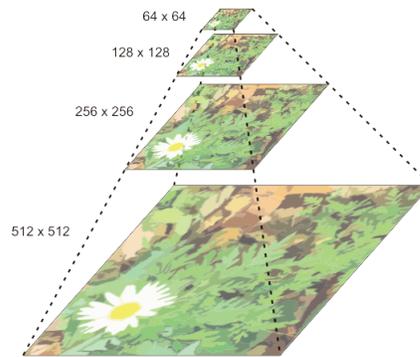


Figura 4.5: Decomposição Piramidal de uma imagem.

4.2.6 Etapa 6: Análise Multiescala

Aqui está uma de nossas mais importantes contribuições. A Análise Multiescala é efetuada para cada imagem da pirâmide gerada na etapa anterior. Esta análise é análoga àquela empregada nos métodos tradicionais da literatura para identificação de cópia-colagem, tal como o Casamento Aproximado [12]: deslizamos uma janela de tamanho fixo pela imagem, extraímos blocos de *pixels*, efetuamos alguma transformação nestes, ordenamos lexicograficamente a matriz de blocos e a percorremos em busca por linhas duplicadas. Nosso método, entretanto, possui algumas diferenças:

- Não deslizamos a janela sobre a imagem por completo, mas apenas sobre as regiões correspondentes encontradas na etapa de agrupamento. Tal estratégia reduz o tempo de detecção.
- Para as imagens nos níveis superiores da pirâmide, nós tomamos os mesmos grupos encontrados para a imagem no nível inicial. Os grupos são redimensionados proporcionalmente ao fator de redução da imagem sendo examinada (e.g., $\frac{1}{2}$, $\frac{1}{4}$).
- Duas regiões correspondentes (candidatas) são analisadas de maneira isolada. Dessa forma, segmentos que não fazem parte de nenhuma das regiões não interferem no processo de detecção. Isso é particularmente interessante, pois evita o aparecimento de falsos-positivos e eleva a precisão do método.
- O bloco é, antes de tudo, normalizado conforme mostrado na Etapa 1.
- Após a ordenação lexicográfica, só é possível marcar dois segmentos como duplicados (no mapa de detecção parcial) se estes pertecerem a regiões correspondentes distintas (origem e destino). Além disso, os blocos devem estar situados a uma distância física mínima, dada por um limiar preestabelecido. O valor deste limiar é decrementado de maneira proporcional à redução no tamanho da imagem (na pirâmide multiescala)

em análise. Por exemplo, para um limiar inicial igual a 32 *pixels*, a imagem no nível seguinte da pirâmide é examinada com um novo limiar igual à $\frac{32}{2} = 16$ *pixels*.

- O mapa de detecção final é analisado e modificado. Neste caso, regiões com poucas detecções são eliminadas, e não farão parte da votação (Etapa 7). Em oposição, regiões fortemente candidatas à cópia-colagem (com um número de detecções significativo) passam por um processo de aprimoramento. O processo consiste em unificar os segmentos detectados em uma região única, a partir de uma delimitação encontrada por um algoritmo de Retângulo de Limite Mínimo (*Minimum Bounding Rectangle*). A área retangular encontrada é considerada, em sua totalidade, como parte integrante de uma possível clonagem.

Nossa metodologia também se caracteriza pela utilização de uma janela deslizante em formato circular e um descritor idênticos àqueles adotados em [34]. O descritor coleta quatro valores de média em cada círculo concêntrico do bloco em análise. Dado que a informação de cor é levada em conta (espaço HSV), a média dos círculos é extraída por canal, o que resulta em um descritor de 12 dimensões. A Figura 4.6 ilustra o descritor empregado. Porque trabalha com mais informações, o que favorece uma distinção maior entre segmentos similares, tal descritor aprimora o resultado de detecção.

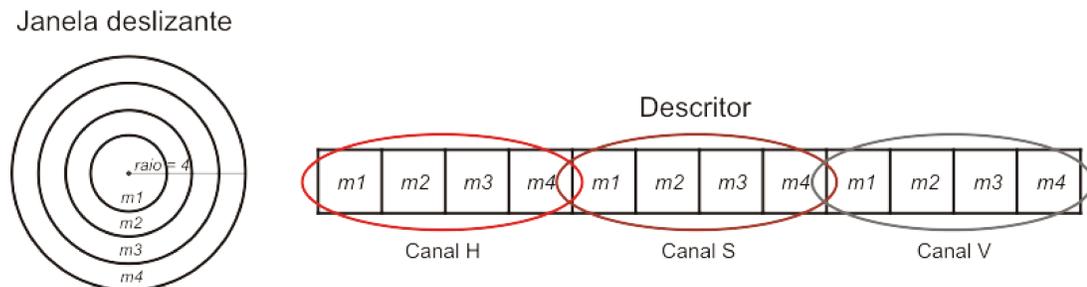


Figura 4.6: Janela deslizante circular e Descritor utilizados em nossa abordagem. Os valores de média em cada círculo concêntrico são representados por m_1 , m_2 , m_3 e m_4 .

Ilustramos a execução completa desta etapa na Figura 4.7.

4.2.7 Etapa 7: Votação

Ao se explorar uma imagem em busca de regiões candidatas à adulteração, esbarramos no seguinte problema: se ajustarmos o limiar de distância de similaridade visando que regiões duplicadas similares sejam identificadas, também poderemos aumentar o número de detecções incorretas. Nossa proposta para contornar tal dificuldade é efetuar uma análise da imagem em diferentes níveis (escalas), tal como mostrado na Etapa 6, e submeter o resultado da detecção de cada uma a um processo de Votação.

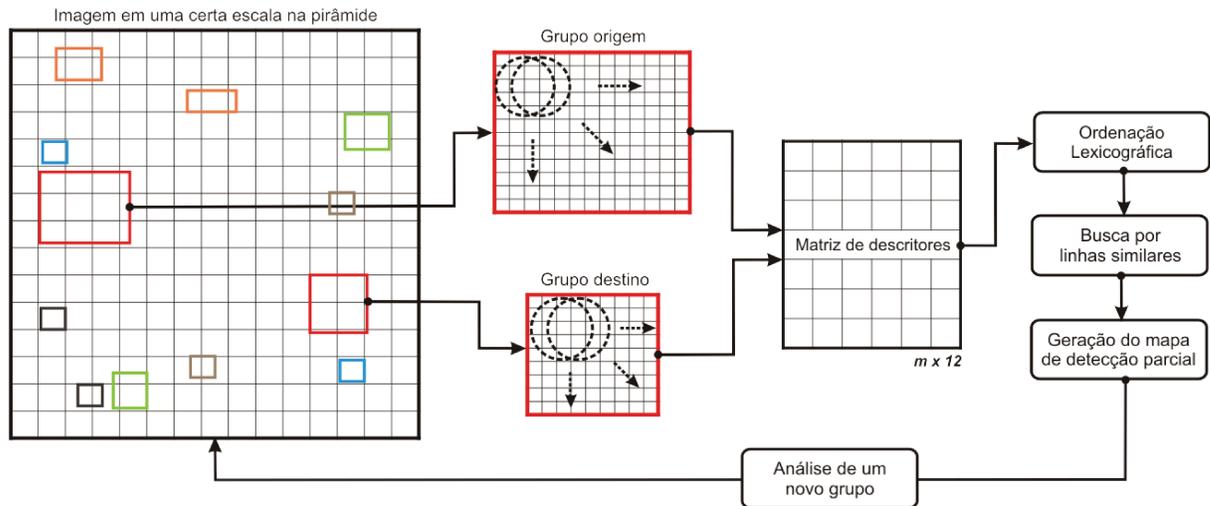


Figura 4.7: Esquematisação da análise de grupos a partir da localização de regiões candidatas na imagem. Em uma dada escala (nível da pirâmide), o grupo vermelho é analisado (origem e destino), e os descritores de seus *patches* são armazenados numa matriz específica que é ordenada lexicograficamente. Em seguida, buscamos por linhas parecidas nessa matriz e produzimos o mapa de detecção parcial (para esse grupo). A etapa prossegue com a verificação dos demais grupos e a atualização do mapa. Encerrados os grupos, o mapa de detecção final é aprimorado. Este procedimento é repetido para todas as imagens da pirâmide multiescala.

O julgamento final considera como *pixels* clonados apenas aqueles que assim o são em, pelo menos, um número predefinido de vezes nos diversos níveis examinados. Dado que os mapas de detecção gerados possuem tamanhos distintos, para a votação nós redimensionamos os mapas menores para as dimensões da imagem na escala original. O procedimento favorece a eliminação de alguns falsos-positivos presentes, principalmente, no mapa da imagem original e, ao mesmo tempo, pode tornar a região verdadeiramente clonada mais visível e lapidada.

4.3 Considerações Finais

Neste capítulo, nós descrevemos uma metodologia inovadora para detecção de cópia-colagem em imagens digitais. Nós apresentamos as abordagens inspiradoras que foram essenciais para a criação do nosso método. Detalhamos, também, cada uma das sete etapas pela qual uma imagem de exame é submetida até o resultado final, que consiste em um mapa de detecção indicando possíveis regiões de cópia-colagem. Tais regiões são encontradas por meio da análise e agrupamento de correspondências entre pontos de interesse, análise multiescala da imagem e votação das regiões suspeitas.

No Capítulo 5, a seguir, nós apresentamos informações sobre os experimentos e as validações realizadas com as duas metodologias propostas neste trabalho, bem como a comparação das abordagens com o estado da arte.

Capítulo 5

Experimentos e Validação

Neste capítulo, nós descrevemos os experimentos e a validação de nossas propostas para detecção de clonagens. Descrevemos, primeiramente, o banco de clonagens produzido e as métricas de avaliação selecionadas. Fornecemos, também, detalhes relativos aos testes com as duas abordagens. Em seguida, apresentamos uma avaliação qualitativa das abordagens, na qual mostramos resultados visuais para clonagens em cenários diversificados e as limitações dos métodos. Finalmente, descrevemos os resultados obtidos com a avaliação quantitativa apenas com a abordagem de votação e análise multiescala (mais efetiva e eficiente), e os comparamos com os resultados de três abordagens da literatura [15, 25, 34] que implementamos.

5.1 Descrição do Dataset

Para avaliar o nosso método de detecção de cópia-colagem e compará-lo com as técnicas da literatura, nós desenvolvemos um *dataset* contendo 108 clonagens originais (uma clonagem por imagem) e de elevado grau de realismo. Armazenamos tais imagens no formato PNG [29]. A menor resolução de imagem é 845×634 *pixels*, e a maior é 1296×972 *pixels*. Cada cópia-colagem possui, também, seu próprio mapa de referência binário que indica as regiões original e duplicada (*pixels* de cor branca). As manipulações criadas contemplam as operações a seguir:

- Clonagens sem modificação da área duplicada (23 imagens);
- Clonagens com rotação da área duplicada (25 imagens);
- Clonagens com redimensionamento da área duplicada (25 imagens);
- Clonagens envolvendo rotação e redimensionamento (ao mesmo tempo) da área clonada (35 imagens);

Os graus de rotação aplicados nos segmentos duplicados variaram de -90 a 180 graus. Para o redimensionamento, utilizamos fatores de escala variando de -20% (*downsampling*) a 54% (*upsampling*). Adicionalmente, nós comprimimos todas as imagens do nosso *dataset* no formato JPEG em 3 fatores de qualidade (70, 80 e 90). Consideramos cada fator sobre 36 imagens. A Figura 5.1 mostra algumas das clonagens mais desafiadoras do nosso *dataset*. A base de imagens será disponibilizada gratuitamente na Internet após a publicação dos resultados deste trabalho.

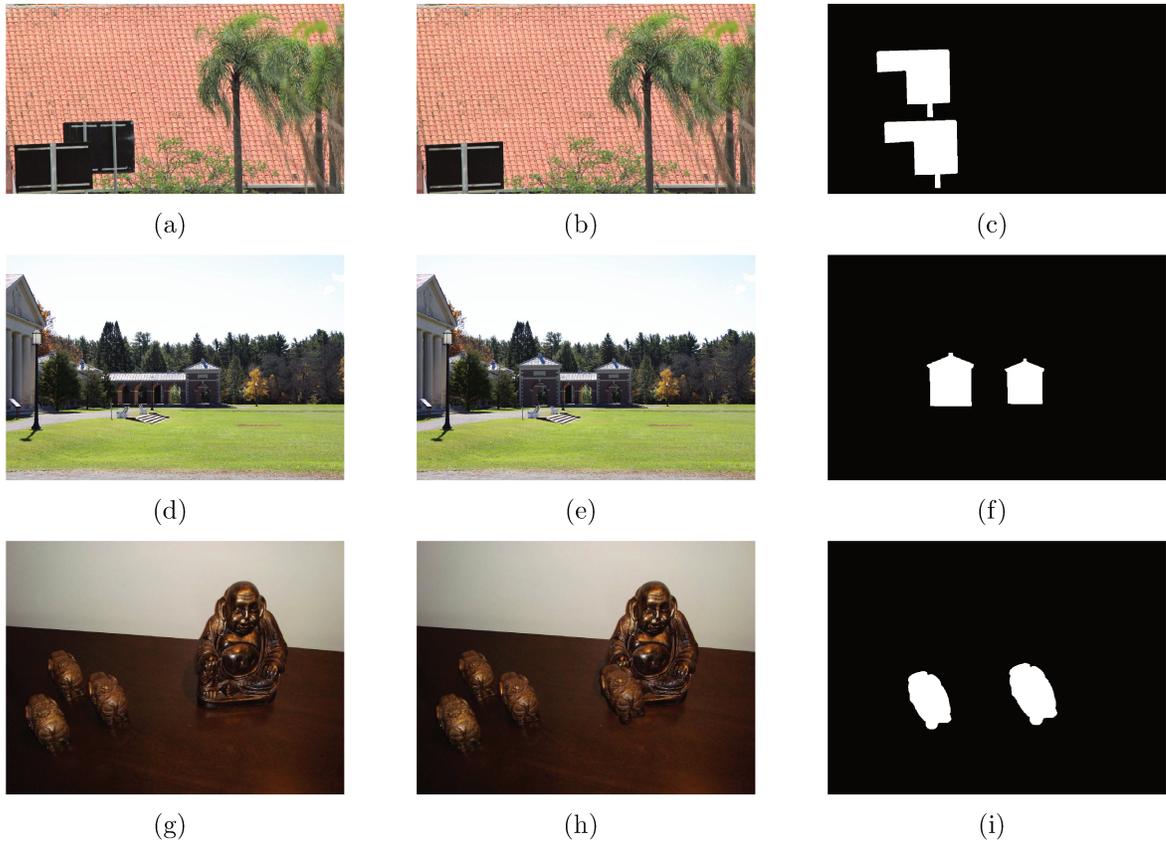


Figura 5.1: Exemplos de imagens do nosso *dataset*. (a), (d) e (g) São as imagens originais; (b), (e) e (h) são exemplos de clonagens e (c), (f) e (i) são seus respectivos mapas de referência.

5.2 Métricas de Avaliação

Nós escolhemos as três métricas para avaliação dos mapas de detecção de clonagens a seguir.

- **Taxa de Verdadeiros-Positivos** (*True Positive Rate* – TPR): Indica a porcentagem de acerto do método com relação apenas às regiões duplicadas (marcadas com a cor branca nos mapas de referência e de detecção). Um *pixel* é um verdadeiro-positivo (*true positive* – TP) se ele faz parte da clonagem e foi corretamente detectado como tal. Esta métrica é calculada como mostra a Equação 5.1, na qual $|TP|$ indica o número de verdadeiros-positivos do mapa de detecção e $|R_{clone}|$ representa o número real de *pixels* clonados (no mapa de referência).

$$TPR = \frac{|TP|}{|R_{clone}|}. \quad (5.1)$$

- **Taxa de Falsos-Positivos** (*False Positive Rate* – FPR): Esta métrica é análoga à TPR, com exceção de que ela mede a quantidade de falsos-positivos (*false positive* – FP), que são *pixels* incorretamente apontados como clonados no mapa de detecção. A Equação 5.2 apresenta o cálculo desta métrica. Nela, $|FP|$ indica o número de falsos-positivos do mapa de detecção e $|R_{normal}|$ representa o número de *pixels* que não integram a clonagem (*pixels* de cor preta no mapa de referência).

$$FPR = \frac{|FP|}{|R_{normal}|}. \quad (5.2)$$

- **Acurácia** (*Accuracy* – ACC): Mede a qualidade da detecção em termos das taxas de verdadeiros-positivos e verdadeiros-negativos (*True Negative Rate* - TNR), que indica a taxa de *pixels* corretamente detectados como não duplicados. A TNR pode ser obtida diretamente da sua relação inversa com a FPR. Equação 5.3 representa o cálculo desta métrica.

$$ACC = \frac{TPR + (1 - FPR)}{2}. \quad (5.3)$$

5.3 Detalhamento dos Testes

Nesta seção, nós detalhamos os parâmetros utilizados na execução dos testes com as duas metodologias propostas.

5.3.1 Detecção de cópia-colagem utilizando o PatchMatch Generalizado

A seguinte configuração de parâmetros foi utilizada na execução dos testes:

- tamanho da janela deslizante: 9×9 *pixels*;
- número de iterações: 5;
- número de correspondências de cada *patch* (K): 5;
- distância de similaridade mínima: 5;
- distância física (Euclidiana) mínima: 30 *pixels*;
- acréscimo ao tamanho do bloco: 2 *pixels* em todas as direções, se possível.

O algoritmo SURF foi executado com Limiar de Hessian (*Hessian Threshold*) igual a zero, visando a extração de todos os pontos característicos possíveis dentro da imagem e suas respectivas descrições. O número de características coletadas para cada ponto foi igual a 128.

5.3.2 Votação e Análise Multiescala de regiões de pontos de interesse

A execução dos testes se deu com a configuração de parâmetros a seguir. Considere a distância física como a distância Euclidiana entre dois pontos na imagem.

- **Agrupamento**

- distância física máxima entre pontos de interesse de um mesmo grupo: 50 *pixels*;
- distância física mínima entre pontos de interesse correspondentes: 50 *pixels*;
- score: 0.6;
- acréscimo ao tamanho do grupo: 10 *pixels* em todas as direções, se possível;
- número de intervalos de orientação: 20 ($[0^\circ, 17^\circ]$, $[18^\circ, 35^\circ]$ etc.);

- **Análise Multiescala**

- tamanho da janela deslizante: 9×9 *pixels* (raio igual a 4);
- número de níveis adicionais na decomposição da imagem: 2;
- número mínimo de detecções dentro de um grupo: 3 pareamentos;
- distância de similaridade mínima entre blocos: 0.05;
- distância física mínima entre blocos: 30 *pixels*;
- número de comparações com linhas vizinhas na matriz de blocos ordenada: 5.

A seleção dos parâmetros acima se deu por meio de uma avaliação visual com imagens diferentes daquelas usadas nos nossos experimentos. Nós criamos um pequeno *dataset* englobando clonagens sem operações adicionais e envolvendo operações de rotação, escala, compressão JPEG e combinações destas. Tais imagens não fazem parte do *dataset* descrito na Seção 5.1 e, portanto, não são usadas nos experimentos que mostramos adiante.

Os parâmetros cujos valores produzem maiores impactos na identificação da clonagem são o score, o número de intervalos de orientação e a distância de similaridade mínima entre blocos. No primeiro caso, valores de score abaixo de 0.6 resultam em uma etapa de pareamento mais “suja”, na qual diversas correspondências entre pontos sem relação são estabelecidas. Por outro lado, valores muito altos podem suprimir pares importantes para a análise. Acerca do número de intervalos de orientação, valores muito baixos podem gerar agrupamentos entre regiões sem relação de similaridade, ao passo que o oposto resulta no estabelecimento de grupos em quantidade elevada, o que pode tornar a detecção menos eficiente. Finalmente, a distância de similaridade é de real importância no processo de localização de duplicações, uma vez que essa medida é a responsável por definir se dois blocos serão (ou não) enquadrados como clonados. O valor mínimo definido nos testes foi de 0.05. Tal valor produziu os resultados visuais de detecção mais equilibrados, levando em conta os falsos e verdadeiros-positivos produzidos.

5.4 Avaliação Qualitativa

Nesta seção, nós avaliamos a qualidade (subjetiva) da detecção das duas abordagens propostas. Apresentamos resultados visuais de clonagens em cenários de rotações, redimensionamentos, compressão JPEG e clonagens simples. Descrevemos, também, as limitações de cada método.

5.4.1 Detecção de cópia-colagem utilizando o PatchMatch Generalizado

Apenas para esta avaliação, criamos um pequeno *dataset* adicional contendo clonagens de tamanho 512×512 *pixels*. Utilizamos imagens deste tamanho, em razão da limitação de tempo que o método impõe. Tal restrição, que inclui o tempo (aproximado) de cinco minutos para analisar imagens desse tamanho, ocorre porque não guardamos os histogramas dos *patches* verificados. Assim, o recálculo dos histogramas eleva o tempo total de análise. A fim de reduzir esse tempo, calculamos e armazenamos previamente todos os histogramas da imagem. Tal procedimento favorece a redução do tempo de execução para dois minutos, aproximadamente. Contudo, ele não é aplicável a imagens grandes, pois exige um custo de armazenamento em memória elevado.

Obtivemos bons resultados para detecção de cópia-colagem simples e envolvendo rotações nas três orientações ordinárias (90, 180 e 270 graus). Também é possível revelar duplicações espelhadas verticalmente e horizontalmente. No entanto, o método possui limitações, não sendo robusto a alterações na região duplicada, tais como compressão JPEG, redimensionamentos e rotações em graus diferentes dos mencionados anteriormente (e.g., 25°). As Figuras 5.2, 5.3, 5.4 e 5.5 apresentam alguns dos resultados visuais alcançados com o método.

Acreditamos que a baixa robustez mediante tais transformações acontece, principalmente, devido ao caráter aleatório do algoritmo *PatchMatch Generalizado*, que impede a localização efetiva das regiões clonadas. Assim, um *patch* p duplicado qualquer terá como correspondências mais próximas apenas aquelas situadas em sua vizinhança. Sem uma correspondência próxima à região da duplicação, que neste caso não é idêntica, o algoritmo não propaga a informação que desejamos para os *patches* ao seu redor, tornando a detecção falha. Nestas circunstâncias, mesmo com a flexibilização do limiar de similaridade, o resultado permanece incipiente, gerando quantidades elevadas de falsos-positivos em regiões parcialmente homogêneas da imagem.

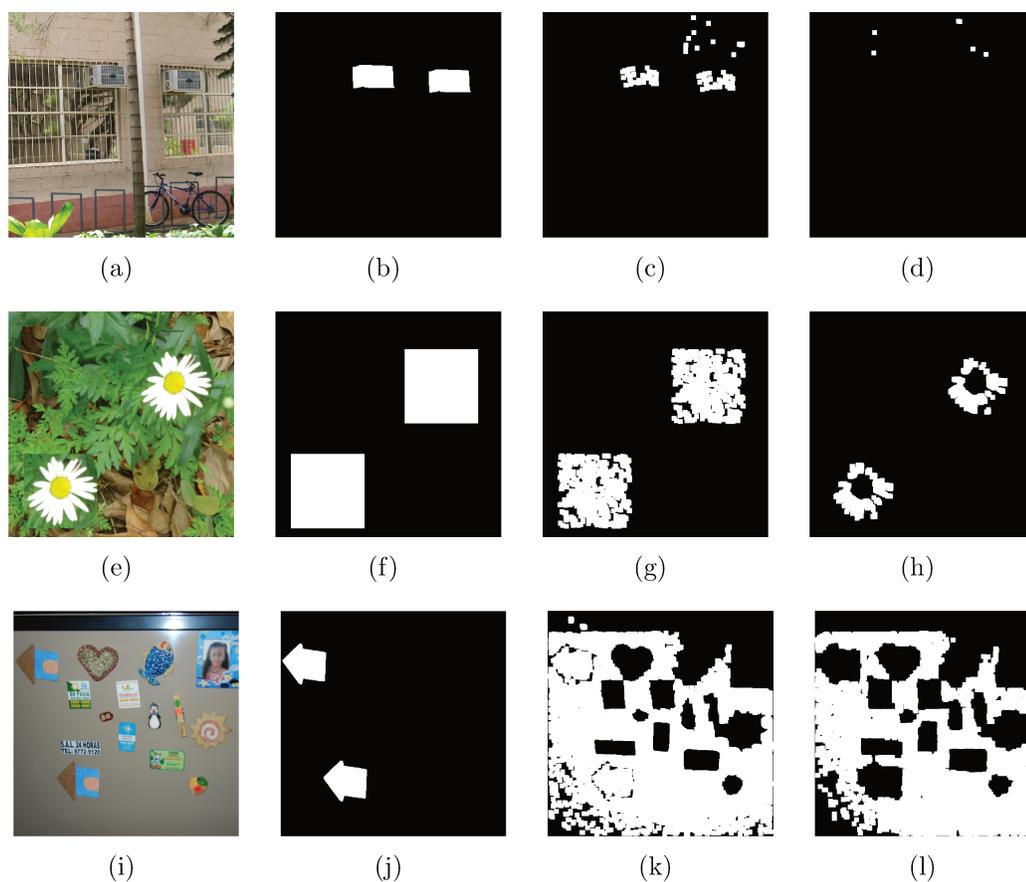


Figura 5.2: Resultados visuais de nossa primeira abordagem para clonagens simples. (a), (e) e (i) São as clonagens; (b), (f) e (j) são os mapas de referência; (c), (g) e (k) são mapas de detecção obtidos para imagens sem compressão e (d), (h) e (l) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80 e 70, respectivamente.

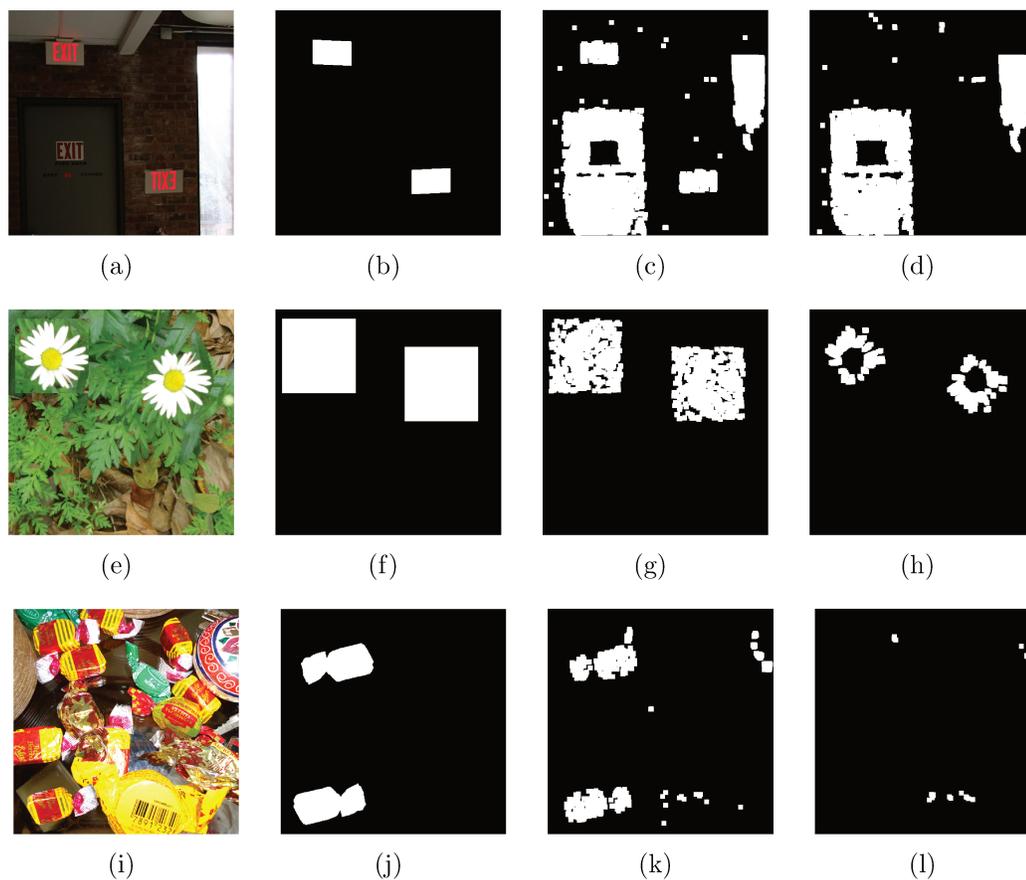


Figura 5.3: Resultados visuais de nossa primeira abordagem para clonagens com espelhamento. (a), (e) e (i) São clonagens com espelhamento horizontal, vertical e ambos, respectivamente; (b), (f) e (j) são os mapas de referência; (c), (g) e (k) são os mapas de detecção obtidos para imagens sem compressão e (d), (h) e (l) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80 e 70, respectivamente.

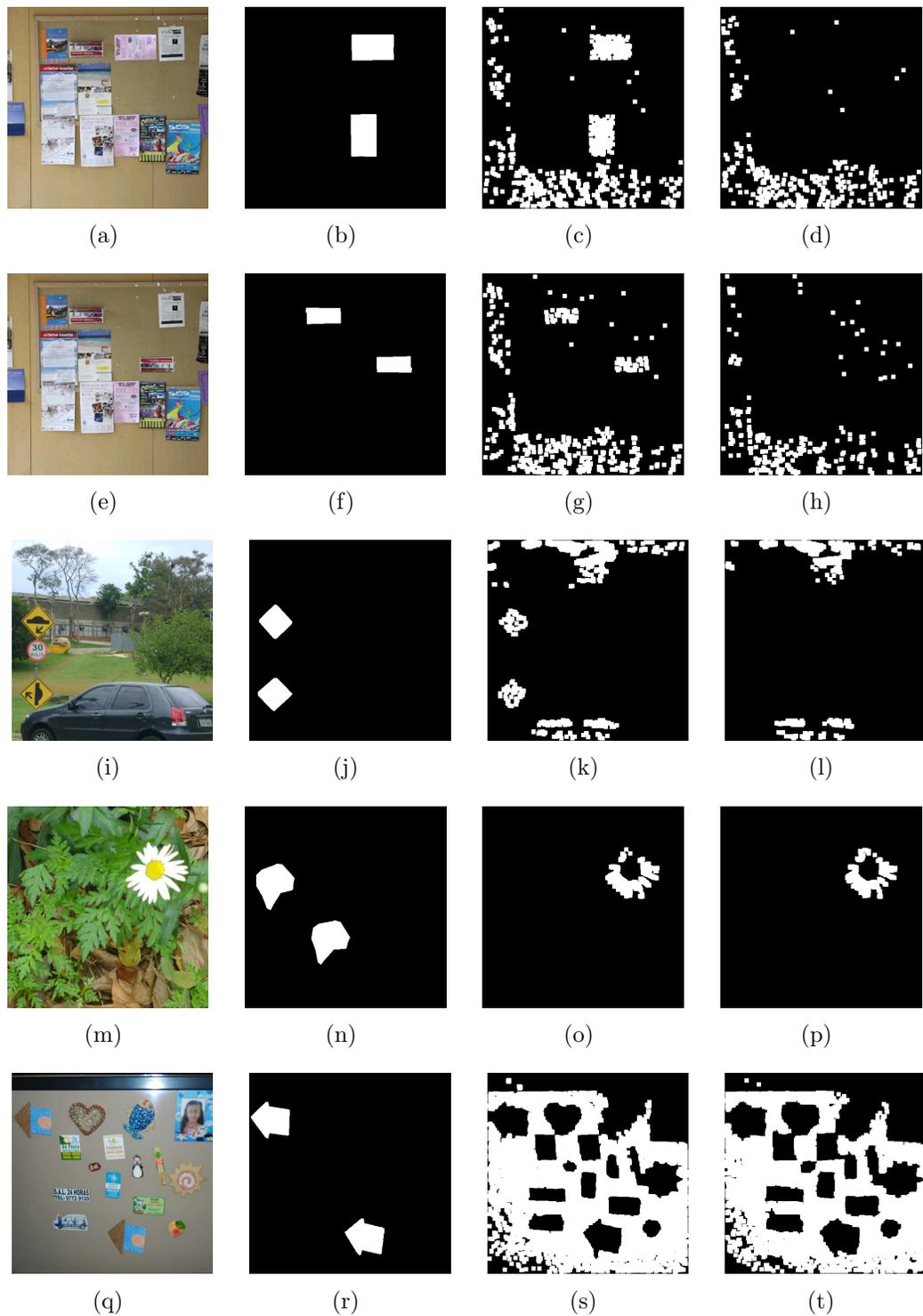


Figura 5.4: Resultados visuais de nossa primeira abordagem para clonagens com rotações. (a), (e), (i), (m) e (q) São clonagens contendo rotações de 90, 180, 270, 10 e -5 graus, respectivamente; (b), (f), (j), (n) e (r) são os mapas de referência; (c), (g), (k), (o) e (s) são os mapas de detecção obtidos para imagens sem compressão e (d), (h), (l), (p) e (t) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80, 70, 90 e 80, respectivamente.

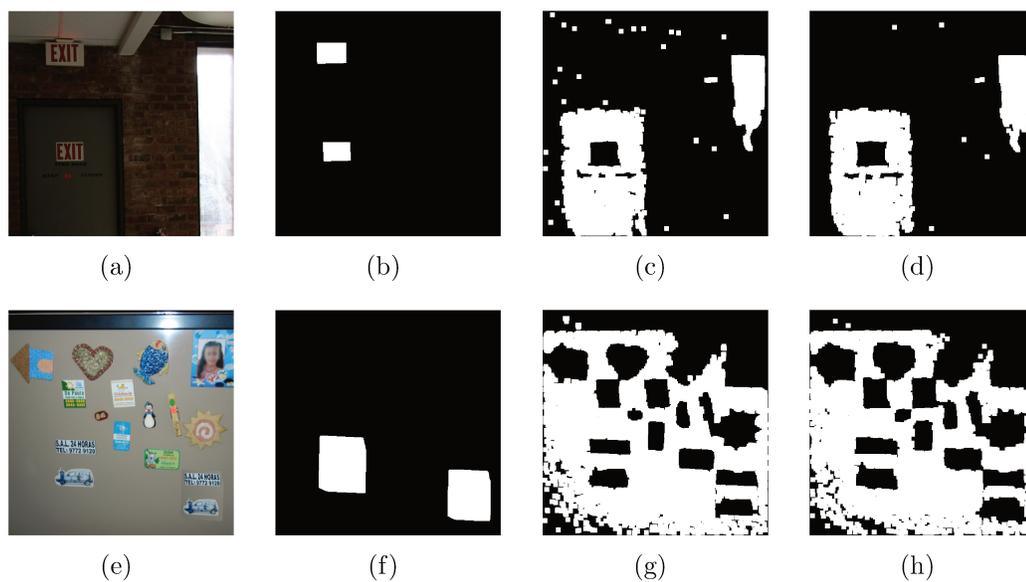


Figura 5.5: Resultados visuais de nossa primeira abordagem para clonagens com operações de escala. (a) e (e) São clonagens com ampliação de 7% e diminuição de 10% dos segmentos duplicados, respectivamente; (b) e (f) são os mapas de referência; (c) e (g) são os mapas de detecção obtidos para imagens sem compressão e (d) e (h) são os mapas de detecção obtidos para imagens com compressão JPEG nos fatores 90 e 80, respectivamente.

5.4.2 Votação e Análise Multiescala de regiões de pontos de interesse

Os resultados visuais obtidos com esta abordagem foram superiores aos do método anterior. Alcançamos maior robustez às operações de escala, rotação e compressão JPEG, e continuamos com resultados satisfatórios na cópia-colagem simples (região duplicada intacta). Além disso, o método se mostrou eficaz mesmo em cenários de composições destas operações. Nós exploramos o potencial da nossa abordagem combinando (i) rotações e redimensionamentos, (ii) rotações e compressão JPEG, (iii) redimensionamentos e compressão JPEG e (iv) todas as operações juntas.

O nosso método, contudo, não está isento de falhas. A limitação mais drástica ocorre quando poucos pontos de interesse são detectados nas regiões duplicadas. Tipicamente, este problema surge quando as regiões clonadas são pequenas demais, ou possuem características de homogeneidade acentuadas. Nestas situações, o detector de pontos presente no SURF [5] pode não encontrar um conjunto de pontos que nos auxilie a agrupar e analisar tais regiões.

Outra limitação ocorre na presença das operações de espelhamento, para as quais o nosso método não é robusto. Tais operações atuam somente na disposição dos *pixels* (seus valores permanecem inalterados) da região clonada em comparação com a original. Uma possível razão para isto é que, ao se espelhar uma região, as retas ligando os pontos de interesse desta e os pontos da região original (correspondências) passarão a ter inclinações completamente distintas. Neste cenário, a etapa de agrupamento pode gerar um número maior de pequenos grupos, levando ao não enquadramento das regiões, em questão, como duplicadas. Mesmo assim, mantivemos tal etapa como parte integrante da metodologia, pois seu papel é relevante no que diz respeito à redução da quantidade de falsos-positivos encontrados.

As Figuras 5.6, 5.7, 5.8 e 5.9 apresentam alguns dos resultados visuais alcançados com o método.

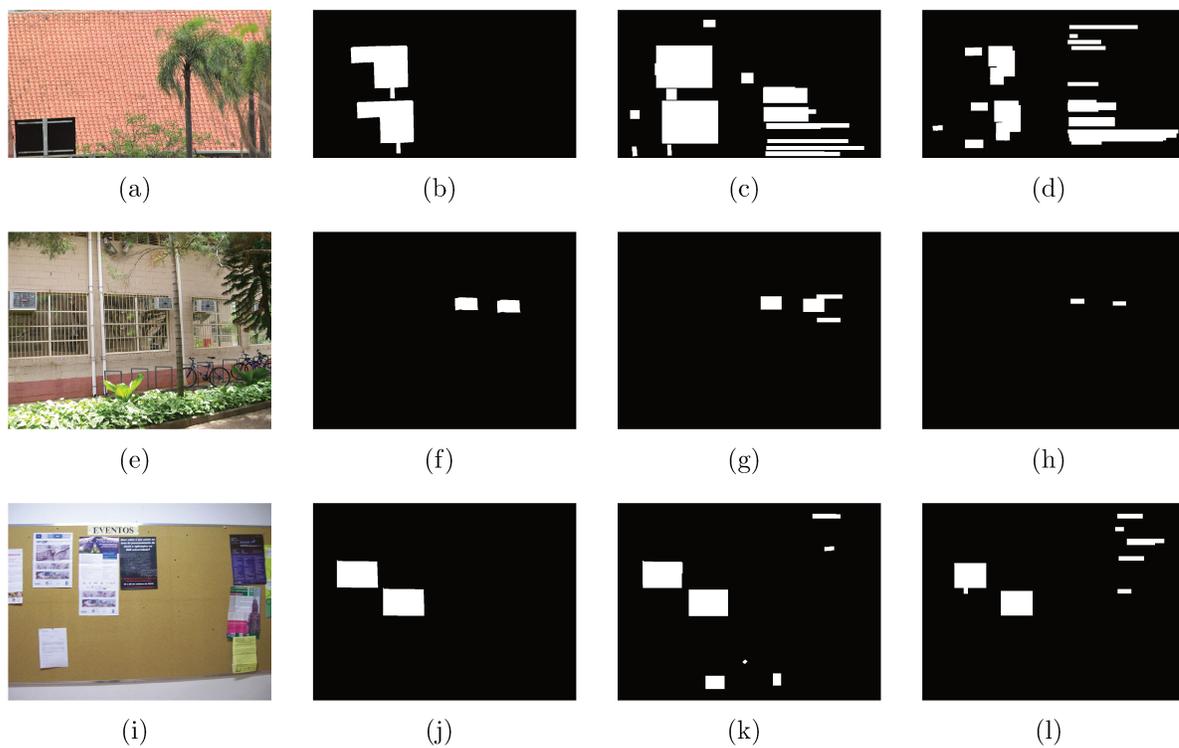


Figura 5.6: Resultados visuais de nossa segunda abordagem para clonagens simples. (a), (e) e (i) São as clonagens; (b), (f) e (j) são os mapas de referência; (c), (g) e (k) são os mapas de detecção obtidos para imagens sem compressão e (d), (h) e (l) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80 e 70, respectivamente.

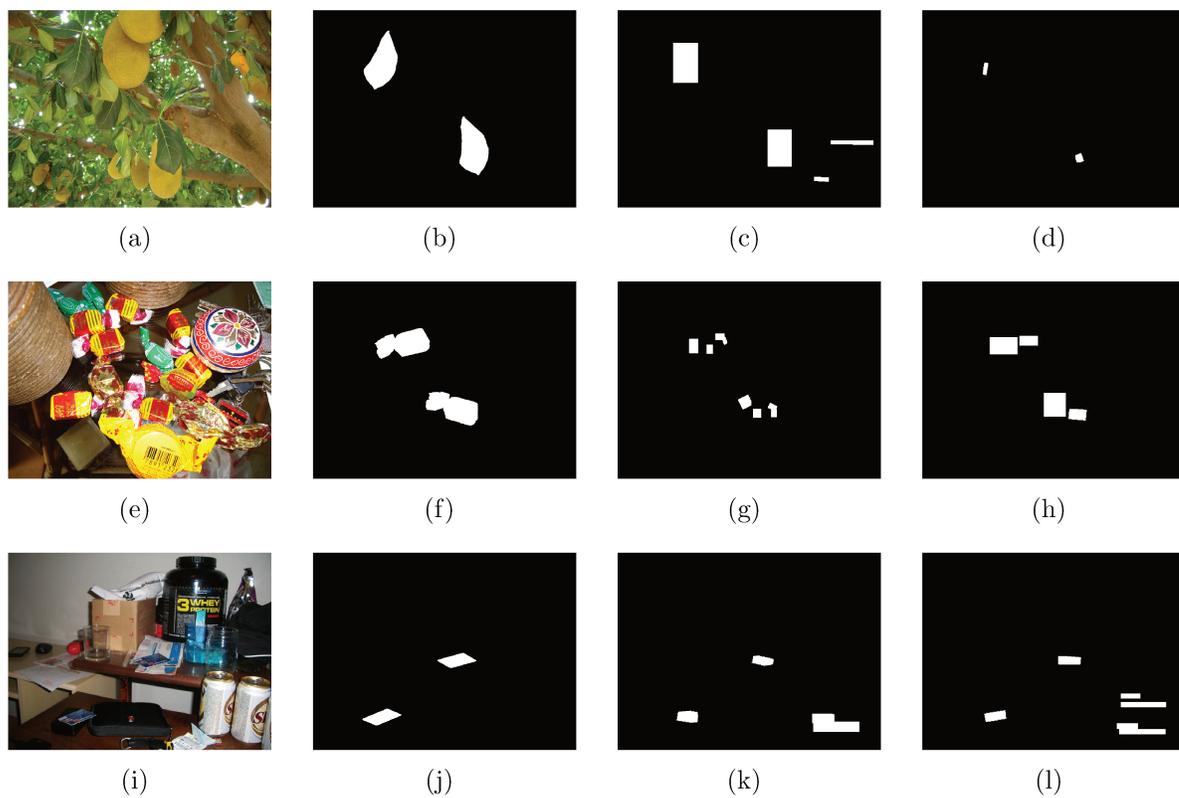


Figura 5.7: Resultados visuais de nossa segunda abordagem para clonagens com rotação. (a), (e) e (i) São as clonagens contendo rotações dos segmentos duplicados (28, -30 e 7 graus, respectivamente); (b), (f) e (j) são os mapas de referência; (c), (g) e (k) são os mapas de detecção obtidos para imagens sem compressão e (d), (h) e (l) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80 e 70, respectivamente.

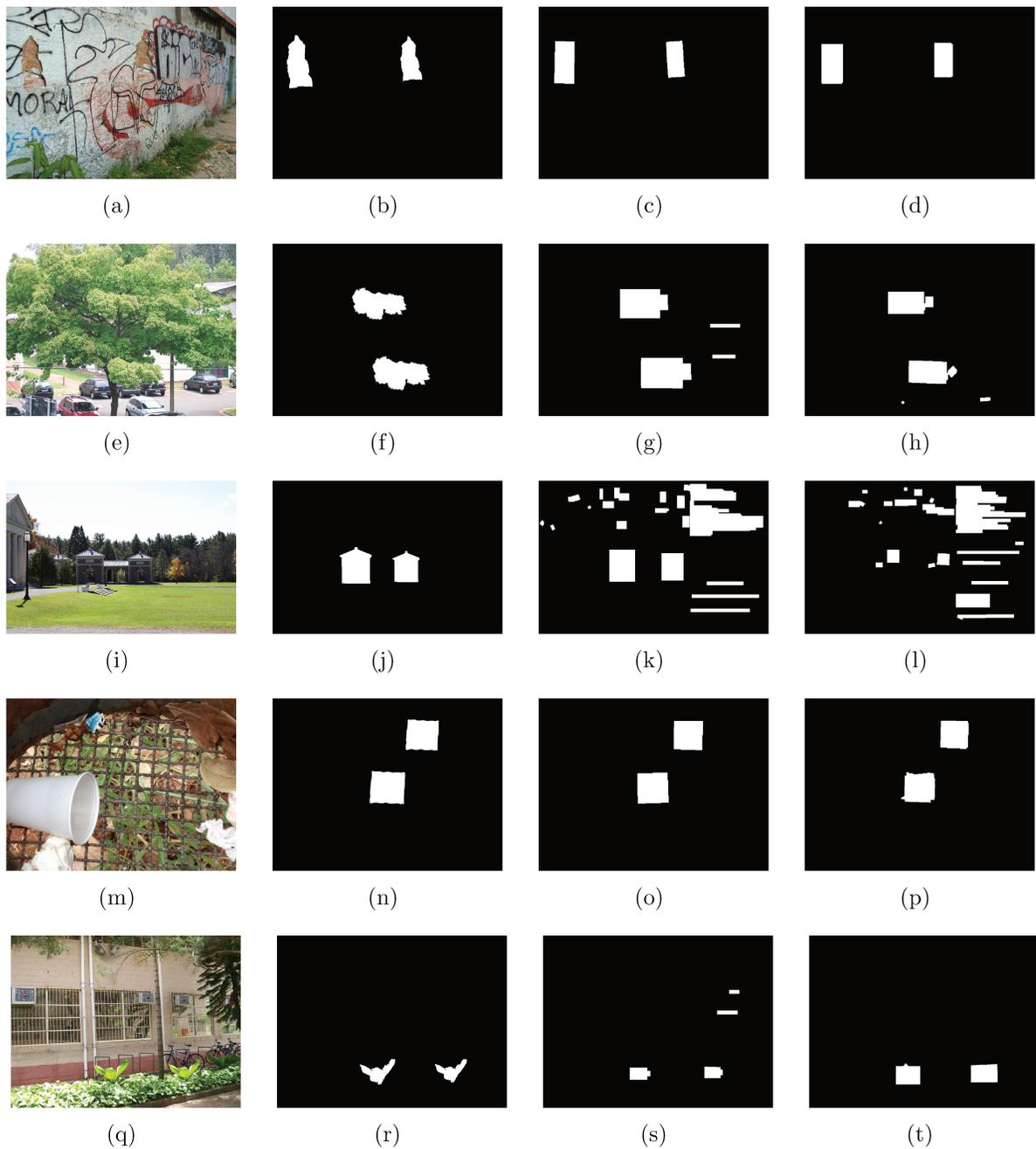


Figura 5.8: Resultados visuais de nossa segunda abordagem para clonagens com operações de escala. (a), (e), (i), (m) e (q) São as clonagens contendo redimensionamento dos segmentos duplicados (ampliações de 19%, 5% e 18%, e reduções de 7% e 10%, respectivamente); (b), (f), (j), (n) e (r) são os mapas de referência; (c), (g), (k), (o) e (s) são os mapas de detecção obtidos para imagens sem compressão e (d), (h), (l), (p) e (t) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80, 70, 90 e 70, respectivamente.



Figura 5.9: Resultados visuais de nossa segunda abordagem para clonagens com operações de escala e rotação combinadas. (a), (e), (i), (m) e (q) São as clonagens (-10 graus e ampliação de 8%; -90 graus e ampliação de 1%; 2 graus e ampliação de 7%; 3 graus e redução de 16%; -18 graus e redução de 10%, respectivamente); (b), (f), (j), (n) e (r) são os mapas de referência; (c), (g), (k), (o) e (s) são os mapas de detecção obtidos para imagens sem compressão e (d), (h), (l), (p) e (t) são os mapas de detecção obtidos para compressão JPEG nos fatores 90, 80, 70, 90 e 80, respectivamente.

5.5 Avaliação Quantitativa

Nesta seção, nós avaliamos a precisão da detecção referente à abordagem de Votação e Análise Multiescala de maneira quantitativa. Nós comparamos os resultados do método com os de outras abordagens da literatura implementadas por nós. Não realizamos avaliação quantitativa para nossa abordagem baseada no *PatchMatch Generalizado* em virtude da baixa qualidade de detecção já demonstrada, considerando clonagens mais sofisticadas envolvendo transformações de rotação, escala e compressão JPEG, e do alto tempo de execução necessário para avaliar todas as imagens de nosso *dataset*.

Cada método foi testado com as 108 clonagens sem compressão e com as mesmas 108 clonagens comprimidas em JPEG nos fatores de qualidade 70, 80 e 90. No total, há 36 imagens clonadas comprimidas em cada um desses fatores. Nós calculamos as três métricas para cada mapa de detecção final gerado, comparando-os com seus respectivos mapas de referência. O tempo de execução médio do nosso método para uma imagem da base gira em torno de 3 minutos. É importante lembrar que o tempo é diretamente influenciado pelo tamanho da região duplicada na imagem, que quanto maior mais tempo exige nas comparações de blocos.

As Tabelas 5.1, 5.2, 5.3 e 5.4 mostram os resultados médios obtidos com cada método. Observa-se que a nossa abordagem (Tabela 5.1) obteve resultados significativos de acurácia, variando de 66.94%, no caso da compressão com mais perdas, a 85.44% no cenário sem compressão. As taxas de verdadeiros-positivos (TPR) também foram satisfatórias, partindo de 37.57% e alcançando 74.49%. Essas taxas sugerem que, em média, mais de 35% das regiões clonadas foram localizadas no pior caso de detecção. Nota-se também que, apesar de as demais abordagens produzirem TPRs também elevadas, suas taxas de falsos-positivos (FPR) seguem a mesma tendência. Isto indica que os métodos da literatura possuem dificuldade em distinguir regiões similares próprias da cena e regiões duplicadas intencionalmente por indivíduos. O resultado é que, por vezes, a manipulação continuará escondida no mapa de detecção em meio à gama de regiões incorretamente apontadas como duplicadas. Por outro lado, nosso método gera uma FPR média de 3.68% no pior caso. Tal taxa contribui para facilitar o exame visual posterior da imagem por um perito, ainda que poucos *pixels* da região duplicada tenham sido apontados como tal.

Os gráficos das Figuras 5.10, 5.11, 5.12 e 5.13 apresentam os resultados para cada abordagem em formato de *ranking*. Consideramos o conjunto completo de 216 imagens, sendo metade delas comprimidas em JPEG nos fatores mencionados na Seção 5.1. Nota-se, pelo gráfico da Figura 5.10, que a nossa abordagem obteve a maior acurácia para todas as clonagens testadas, com exceção de uma, para a qual a melhor acurácia foi obtida com método de Wang et al. [34]. A nossa abordagem também obteve o maior número de

imagens com a melhor TPR (imagens na 1^o posição). Além disso, o método apresentado em [34] obteve o maior número de imagens com a pior TPR (imagens na 4^o posição). Finalmente, o maior número de imagens com a **menor FPR** (1^o posição) foi encontrado com a nossa proposta, que também não obteve nenhuma imagem na posição referente às maiores FPRs (4^o posição).

A Figura 5.14 mostra o *ranking* geral, que apresenta como mais bem colocadas as abordagens com maior número de imagens na primeira posição (em caso de empate, compara-se o maior número de imagens na segunda posição e assim por diante).

Tabela 5.1: Resultados dos experimentos com a nossa abordagem.

	Fator de compressão			
Métrica	70	80	90	Nenhum
ACC	66.94%	71.83%	67.19%	85.44%
TPR	37.57%	46.84%	37.22%	74.49%
FPR	3.68%	3.17%	2.83%	3.60%

Tabela 5.2: Resultados dos experimentos com a abordagem de Popescu e Farid [25].

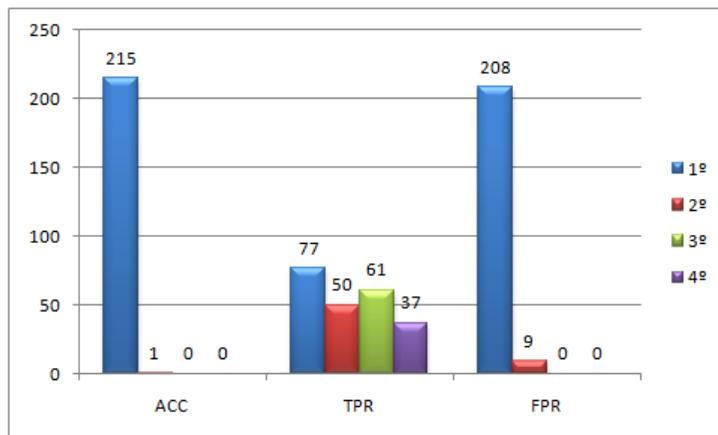
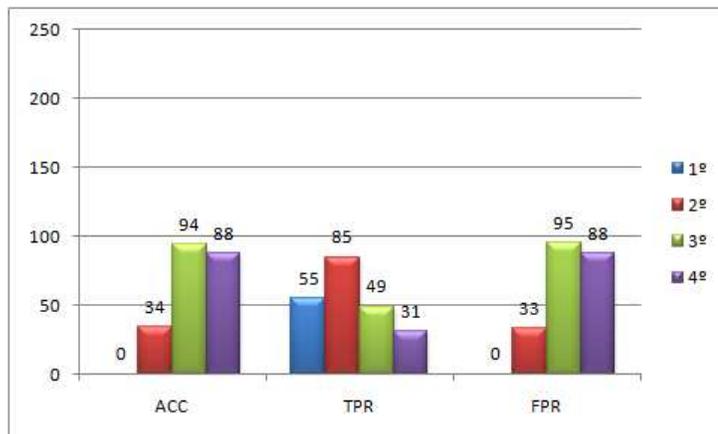
	Fator de compressão			
Métrica	70	80	90	Nenhum
ACC	43.45%	44.27%	53.80%	47.77%
TPR	48.64%	41.06%	61.59%	51.98%
FPR	61.73%	52.52%	53.98%	56.43%

Tabela 5.3: Resultados dos experimentos com a abordagem de Li et al. [15].

	Fator de compressão			
Métrica	70	80	90	Nenhum
ACC	44.02%	44.38%	47.96%	46.62%
TPR	29.09%	20.55%	26.73%	27.03%
FPR	41.05%	31.79%	30.80%	33.79%

Tabela 5.4: Resultados dos experimentos com a abordagem de Wang et al. [34].

	Fator de compressão			
Métrica	70	80	90	Nenhum
ACC	45.28%	48.27%	48.51%	50.29%
TPR	37.46%	39.49%	42.68%	73.87%
FPR	46.90%	42.94%	45.66%	73.28%

Figura 5.10: *Ranking* das imagens para a nossa abordagem, considerando-se as três métricas propostas.Figura 5.11: *Ranking* das imagens para a abordagem Popescu e Farid [25], considerando-se as três métricas propostas.

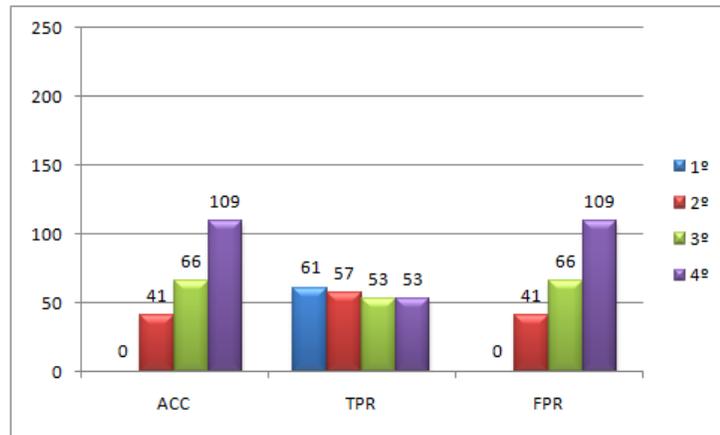


Figura 5.12: *Ranking* das imagens para a abordagem de Li et al. [15], considerando-se as três métricas propostas.

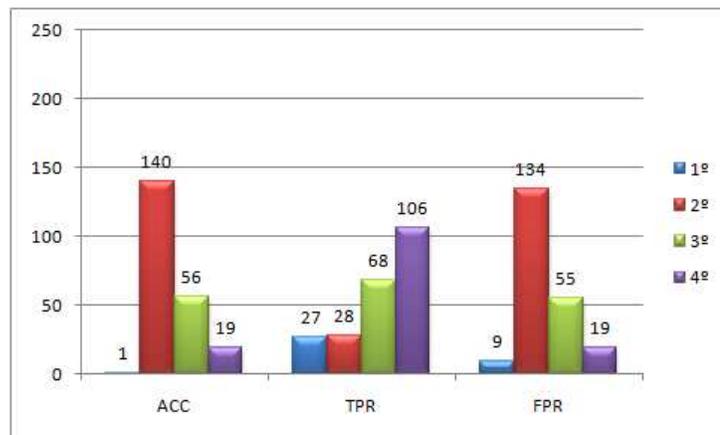


Figura 5.13: *Ranking* das imagens para a abordagem de Wang et al. [34], considerando-se as três métricas propostas.

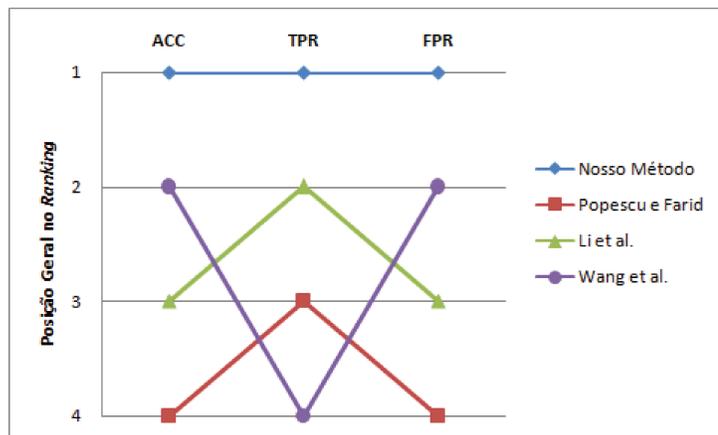


Figura 5.14: *Ranking* Geral das abordagens examinadas, considerando-se as três métricas propostas.

5.6 Considerações Finais

A avaliação visual indica que método baseado no *PatchMatch Generalizado*, apesar de possuir um raciocínio inovador, possui limitações, em razão da dificuldade em lidar com alterações nos valores de *pixels* da região clonada. Em oposição, pelas imagens mostradas, a nossa abordagem de votação e análise multiescala de regiões apresenta boa qualidade de detecção e resultados quantitativos promissores, quando comparado com outras abordagens tradicionais da literatura.

As diferenças mais impactantes entre o nosso método e os demais que testamos são as taxas de acurácia, que no nosso método chegou a 85.44%, e de falsos-positivos, que é consideravelmente mais baixa na nossa proposta (em torno de 3%, em média). As etapas de agrupamento, análise multiescala e votação são as principais responsáveis por esse resultado, pois reduzem o número de regiões a serem comparadas e eliminam aquelas que não persistem nas múltiplas escalas da imagem.

Capítulo 6

Conclusões

Neste trabalho, nós apresentamos duas metodologias para detecção de cópia-colagem. A primeira delas é baseada no algoritmo *PatchMatch* [3], usado na busca por uma correspondência aproximada de cada *patch* de uma imagem dentro dela própria, por meio de uma estratégia baseada na aleatoriedade. Nós implementamos a versão estendida desse algoritmo, denominada *PatchMatch Generalizado* [4] e que visa à busca pelos K vizinhos mais próximos de um ou mais *patches* da imagem.

O nosso método de detecção é inspirado na abordagem sugerida por Barnes et al. [4] para detecção de clonagens usando o *PatchMatch Generalizado*, mas que, segundo nossos testes, falha nesta tarefa. Nós reformulamos o raciocínio dos autores e conseguimos produzir um detector de cópia-colagens que não é tão robusto a rotações, compressões e redimensionamentos nos segmentos duplicados, mas que pode ser visto como uma alternativa promissora ao detectar regiões espelhadas e rotacionadas em ângulos de 90, 180 e 270 graus.

Dado que o desafio principal consiste em reconhecer corretamente duas regiões similares como duplicadas, o mecanismo aleatório e aproximado do *PatchMatch* é uma característica que, em nossa opinião, pode ser empregada efetivamente na tarefa aqui investigada. Destacamos como trabalhos futuros relativos à esta abordagem a investigação aprofundada desse algoritmo, tornando-o menos custoso computacionalmente (tempo e armazenamento) e mais aprimorado na busca por correspondências de *patches* pertencentes às regiões clonadas. Para isso, pode ser necessário adotar um descritor (SURF [5], SIFT [18], entre outros) que capture com maior eficácia as correspondências desejadas, ou mesmo ajustar a ideia central do algoritmo.

A segunda metodologia que mostramos é baseada em Votação e Análise Multiescala de regiões correspondentes na imagem. Esta abordagem reúne características de diversos trabalhos da literatura e introduz ideias inovadoras para a solução do desafio aqui discutido, tal como as etapas de agrupamento e votação. O método mostrou ser um forte concorrente

às abordagens da literatura avaliadas, produzindo resultados visuais de detecção efetivos. O mesmo é válido para os números obtidos com a avaliação quantitativa do método, que apresentou acurácia média de até 85.44%, bem superior aos demais algoritmos testados. A FPR também se mostrou reduzida em todos os cenários experimentados, destoando, em nosso favor, de forma acentuada das demais abordagens avaliadas.

Acerca da segunda metodologia, alguns possíveis trabalhos futuros incluem a extensão do nosso método para contemplar operações de suavização, adição de ruídos, espelhamentos e combinações destas em conjunto com rotações, redimensionamentos e compressões JPEG maiores. Além disso, almejamos descobrir formas diferentes de descrever os blocos, assim como de detectar pontos de interesse na imagem (combinando os resultados de dois ou mais detectores de pontos, por exemplo), visto que estes últimos são cruciais para o funcionamento preciso do nosso método. Outro trabalho plausível envolveria a segmentação da imagem em múltiplas regiões, substituindo ou mesmo agindo em parceria com a etapa de agrupamento. Acreditamos que tal abordagem pode aperfeiçoar a detecção dos segmentos duplicados, elevando a TPR da nossa proposta.

Finalmente, salientamos que um dos desafios que a pesquisa em identificação de cópia-colagem terá de enfrentar, em breve, advém da crescente aplicação de algoritmos de preenchimento automático de regiões em imagens. Uma ferramenta que proporciona esta tarefa é o *Content-Aware Fill*, que faz uso do algoritmo *PatchMatch* e integra o conjunto de funcionalidades do Adobe Photoshop CS5 e CS6. Por meio dela, é possível eliminar um elemento presente na cena sem que, para isso, seja necessário realizar a cópia e a colagem de segmentos manualmente, a fim de sobrepor os ao elemento indesejado. A ferramenta se encarrega de completar a área selecionada (e previamente excluída) com segmentos do restante da imagem de forma que o resultado visual seja satisfatório, o que pode ser visto como um processo de clonagem mais sofisticado. Este é o objeto de outro possível trabalho futuro, em que se objetiva identificar duplicações executadas de forma automática em uma imagem e cujos segmentos podem ter sido provenientes de múltiplas regiões diferentes desta.

Referências Bibliográficas

- [1] 21st century color theory. 21st century color theory - hsv. Online at <http://learn.colorotate.org/color-models.html>, May 2012.
- [2] Edoardo Ardizzone and Giuseppe Mazzola. Detection of duplicated regions in tampered digital images by bit-plane analysis. In *Intl. Conference on Image Analysis and Processing (ICIAP)*, pages 893–901, 2009.
- [3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, pages 24:1–24:11, July 2009.
- [4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision (ECCV)*, pages 29–43, 2010.
- [5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.
- [6] Sevinc Bayram, Husrev Taha Sencar, and Nasir Memon. An efficient and robust method for detecting copy-move forgery. In *IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1053–1056, 2009.
- [7] Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O’Reilly Media, first edition, 2008.
- [8] Pedro Monteiro da Silva Eleutério and Marcio Pereira Machado. *Desvendando a Computação Forense*. Novatec Editora, first edition, 2011.
- [9] Edward J. Delp, Nasir Memon, and Min Wu. Digital forensics. *IEEE Signal Processing Magazine*, 26(3):14–15, March 2009.
- [10] Hany Farid. Exposing digital forgeries in scientific images. In *ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006.

- [11] Hany Farid. Digital image forensics. *Scientific American (SCIAM)*, pages 66–71, June 2008.
- [12] Jessica Fridrich, David Soukal, and Jan Lukas. Detection of copy-move forgery in digital images. In *Digital Forensic Research Workshop (DFRWS)*, Cleveland, USA, 2003.
- [13] Hailing Huang, Weiqiang Guo, and Yu Zhang. Detection of copy-move forgery in digital images using sift algorithm. In *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA)*, pages 272–276. IEEE Computer Society, 2008.
- [14] Neeraj Kumar, Li Zhang, and Shree Nayar. What is a good nearest neighbors algorithm for finding similar patches. In *European Conference on Computer Vision (ECCV)*, pages 364–378, 2008.
- [15] Guohui Li, Qiong Wu, Dan Tu, and Shaojie Sun. A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD. In *IEEE Intl. Conference on Multimedia and Expo.*, pages 1750–1753, 2007.
- [16] Lin Liang, Ce Liu, Yingqing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. Technical report, Microsoft Research, Redmond, WA, USA, 2001.
- [17] Hwei-Jen Lin, Chun-Wei Wang, and Yang-Ta Kao. Fast copy-move forgery detection. *WSEAS Transactions on Signal Processing (WSEAS-TSP)*, pages 188–197, May 2009.
- [18] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, page 1150, 1999.
- [19] Weiqi Luo, Jiwu Huang, and Guoping Qiu. Robust detection of region-duplication forgery in digital image. In *Intl. Conference on Pattern Recognition (ICPR)*, pages 746–749, 2006.
- [20] Babak Mahdian and Stanislav Saic. Detection of copy-move forgery using a method based on blur moment invariants. *Forensic Science Intl.*, pages 180–189, September 2006.
- [21] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

- [22] A. N. Myna, M. G. Venkateshmurthy, and C. G. Patil. Detection of region duplication forgery in digital images using wavelets and log-polar mapping. In *Intl. Conference on Computational Intelligence and Multimedia Applications (ICCIMA)*, pages 371–377, 2007.
- [23] Xunyu Pan and Siwei Lyu. Detecting image region duplication using sift features. In *IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1706–1709, 2010.
- [24] Alin C. Popescu. *Statistical Tools for Digital Image Forensics*. PhD thesis, Dept. of Computer Science – Dartmouth College, Hanover, USA, December 2004.
- [25] Alin C. Popescu and Hany Farid. Exposing digital forgeries by detecting duplicated image regions. Technical Report TR 2004-515, Dept. of Computer Science – Dartmouth College, Hanover, USA, 2004.
- [26] Anderson Rocha and Siome Goldenstein. *Atualizações em Informática (2010)*, chapter CSI: Análise Forense de Documentos Digitais, pages 263–317. Sociedade Brasileira de Computação (SBC), Belo Horizonte, Brazil, 1 edition, July 2010.
- [27] Anderson Rocha, Walter Scheirer, Terrance E. Boult, and Siome Goldenstein. Vision of the unseen: Current trends and challenges in digital image and video forensics. *ACM CSUR*, 4(43):1–42, 2011.
- [28] Ewerton Silva and Anderson Rocha. Análise forense de documentos digitais: além da visão humana. *Saúde, Ética & Justiça*, 16(1):9–17, 2011.
- [29] Specification of PNG image format. Specification of png image format. Online at <http://www.w3.org/TR/PNG/>, May 2012.
- [30] Shamik Sural, Gang Qian, and Sakti Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *Intl. Conference on Image Processing (ICIP)*, pages 589–592, 2002.
- [31] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, first edition, 2011.
- [32] The Joint Photograph Expert Group. JPEG file interchange format. Online at <http://www.jpeg.org/public/jfif.pdf>, July 2010.
- [33] Veja Online. Este homem é um farsante. Online at http://veja.abril.com.br/110106/p_082.html, July 2006.

- [34] Junwen Wang, Guangjie Liu, Hongyuan Li, Yuewei Dai, and Zhiquan Wang. Detection of image region duplication forgery using model with circle block. In *IEEE Intl. Conference on Multimedia Information Networking and Security (MINES)*, pages 25–29, 2009.
- [35] Yuk-Man Wong. Invariant local feature for image matching. Online at <http://www.cse.cuhk.edu.hk/lyu/student/mphil/wyman/term3paper.pdf>, December 2006.
- [36] Bo Xu, Junwen Wang, Guangjie Liu, Hongyuan Li, and Yuewei Dai. Image copy-move forgery detection based on surf. In *IEEE Intl. Conference on Multimedia Information Networking and Security (MINES)*, pages 889–892, 2010.
- [37] Jing Zhang, Zhanlei Feng, and Yuting Su. A new approach for detecting copy-move forgeries in digital images. In *IEEE Intl. Conference on Communication Systems (ICCS)*, pages 362–366, 2010.