

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Rafael Pagliarini Bagagli

**Criação de marcas d'água visíveis mais resistentes a
algoritmos de remoção em imagens digitais**

São Carlos

2019

Rafael Pagliarini Bagagli

**Criação de marcas d'água visíveis mais resistentes a
algoritmos de remoção em imagens digitais**

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Marcelo A. C. Vieira

**São Carlos
2019**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

B144c Bagagli, Rafael Pagliarini
Criação de marcas d'água visíveis mais
resistentes a algoritmos de remoção em imagens digitais
/ Rafael Pagliarini Bagagli; orientador Marcelo Andrade
da Costa Vieira. São Carlos, 2019.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2019.

1. Marca D'água Visível. 2. Processamento de
Imagens. 3. Restauração de Imagens. 4. Qualidade de
Imagens. 5. Visão Computacional. 6. Algoritmo. I.
Título.

FOLHA DE APROVAÇÃO

Nome: Rafael Pagliarini Baqagli

Título: "Implementação de um algoritmo de melhora da robustez de marcas d'água digitais"

Trabalho de Conclusão de Curso defendido e aprovado
em 21 / 11 / 2019,

com NOTA 10,0 (DEZ, ZERO), pela Comissão Julgadora:

**Prof. Associado Marcelo Andrade da Costa Vieira - Orientador -
SEL/EESC/USP**

**Prof. Associado Adilson Gonzaga - SEL/EESC/USP (docente
aposentado)**

**Prof. Associado Evandro Luis Linhari Rodrigues - SEL/EESC/USP
(docente aposentado)**

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

AGRADECIMENTOS

Agradeço primeiramente ao meu professor orientador, Marcelo, e todos os outros professores e funcionários desta universidade, sem os quais nada disto seria possível, assim como agradeço a todos que de alguma forma contribuem para a existência do ensino público superior de qualidade no país.

Agradeço a minha família, que me suportou sempre que necessário para que eu pudesse estudar e atingir meus objetivos.

Agradeço a todos meus amigos e colegas de curso, que estiveram ao meu lado nestes últimos inesquecíveis anos. Agradecimento especial ao meu amigo Ricardo, sem o qual este trabalho não existiria.

Por fim, agradeço a todos que trabalharam ou contribuíram com as ferramentas e materiais utilizados durante o meu estudo e escrita desta monografia.

RESUMO

Bagagli, R. **Criação de marcas d'água visíveis mais resistentes a algoritmos de remoção em imagens digitais.** 2019. 81p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

Este trabalho apresenta uma proposta de método de adição de marcas d'água visíveis em imagens digitais com a finalidade de defesa de direitos autorais de fotografias, como é realizado em grandes escalas para imagens comerciais do tipo “*Stock Photos*”, dificultando reproduções não autorizadas dessas imagens. Neste cenário de aplicação e utilização de marcas d'água, foi utilizado um algoritmo de remoção de marcas d'água da divisão de pesquisa da Google para validar a proposta apresentada. Esse algoritmo busca remover as marcas d'água baseando-se no fato que estas são identicamente adicionadas à grandes quantidades de fotos facilmente obtidas na internet, permitindo que esse problema seja resolvido a partir da resolução de um sistema de equações cujo objetivo é minimizar o erro na imagem reconstruída final. O funcionamento do algoritmo de remoção de marcas d'água visíveis foi verificado com a sua aplicação em imagens comercialmente marcadas, mostrando ótimo desempenho quanto à remoção das marcas e qualidade dos resultados reconstruídos. A partir disso, foram elaboradas técnicas de aumento da robustez de marcas d'água frente a este algoritmo de ataque, visando que as marcas aprimoradas não fossem tão facilmente removidas, comparativamente às marcas tradicionais. Os métodos de melhora abordados baseiam-se na aplicação de distorções e de “*ditherização*” às marcas. Estes métodos foram analisados quantitativamente por meio de métricas de qualidade de imagens, obtendo-se resultados que indicam um aumento real na robustez destas marcas que passaram pelo processamento adicional. Conclui-se, por fim, que as marcas d'água tradicionais, amplamente utilizadas por grandes empresas de “*Stock Photography*”, são pouco robustas frente a um ataque de remoção baseado na abundância de imagens identicamente marcadas, porém, o método proposto nesse trabalho é capaz de aumentar significativamente a robustez destas marcas, indicando que é possível, e desejável, o uso de marcas mais avançadas para melhor garantir a proteção dos direitos autorais de imagens e impedir seu uso não autorizado.

Palavras-chave: Marca D'água Visível; Processamento de Imagens; Restauração de Imagens; Qualidade de Imagens; Algoritmo; Visão Computacional

ABSTRACT

Bagagli, R. **Creation of visible watermarks more resistant to removal algorithms in digital images.** 2019. 81p. Undergraduate Final Project, Sao Carlos School of Engineering, University of Sao Paulo, Sao Carlos, Brazil, 2019

This paper proposes a method of adding visible watermarks to digital images with the goal of protecting intellectual rights, as is commonly done in large volumes for “Stock Photos”, defending against unauthorized uses of these images. Within this context of application and use of watermarks, a watermark removal algorithm from Google Research was implemented to validate the proposed method. This algorithm seeks to remove watermarks through the fact that those marks are identically applied to a large quantity of images freely available on the internet, allowing this problem to be solved through the solution of a system of equations which results in a final reconstructed image with minimal errors. The performance of this visible watermark removal algorithm was validated through its application on a dataset of commercially watermarked images, resulting in high quality reconstructed images without highly visible watermarks or visual artifacts. Following this, techniques for improving watermark robustness were explored, so that these improved watermarks would not be so easily removed from images, when compared to traditional watermarks. The methods explored are based on applying distortions and ditherization to the watermark. Quantitatively, these results were analyzed using image quality metrics, achieving results that indicate a real improvement of the watermark’s robustness when they go through this additional processing. In conclusion, traditional watermarks, widely in use by large Stock Photography companies, were shown not to be robust when faced with a removal algorithm that employs the abundance of identically watermarked images, however, the proposed method in this paper is capable of significantly improving their robustness, showing that the use of more advanced watermarks would be a desirable change in order to better protect an image against unlicensed use and copyright infringement.

Keywords: Visible Digital Watermarks; Digital Image Processing; Image Restoration; Image Quality; Algorithm; Computer Vision

LISTA DE FIGURAS

Figura 1 – Marca d’água tradicional em uma cédula de 20 euros	18
Figura 2 – Exemplo de marca d’água digital (“Wikipedia”)	18
Figura 3 – Exemplificação do modelo de composição de uma imagem marcada . .	22
Figura 4 – PSNR de várias imagens com diferentes quantidades de ruído	31
Figura 5 – Amostra de imagens limpas adquiridas (Fonte: pexels.com)	33
Figura 6 – Marca D’água e seu Canal Alpha gerados	34
Figura 7 – Exemplo de Imagem Marcada Sinteticamente	34
Figura 8 – Exemplo de imagem marcada comercialmente - “Adobe Stock” (Fonte: stock.adobe.com)	35
Figura 9 – Outras imagens marcadas adquiridas (Fonte: stock.adobe.com)	35
Figura 10 – Processo de Obtenção da Estimativa Inicial da Marca D’água	36
Figura 11 – Exemplo de Imagens utilizadas para localização da Marca	37
Figura 12 – Exemplo de marcas localizadas pelo método de filtragem, denotadas pelo X vermelho	38
Figura 13 – Exemplo do funcionamento do algoritmo de Levin et al.	39
Figura 14 – Imagem α da Marca D’água, obtida através do algoritmo de Levin et al. (Com ajuste de contraste para melhor visualização)	39
Figura 15 – Esquema de aplicação de distorção	41
Figura 16 – Exemplo de marca distorcida aplicada a uma imagem	42
Figura 17 – Exemplo da redução do efeito de bandas com o efeito <i>dither</i> , na imagem central (Fonte: Wikipedia)	42
Figura 18 – Exemplo do efeito <i>Dither</i> em imagens (Fonte: Wikipedia)	43
Figura 19 – Funcionamento do efeito de transparência “Porta de Tela”	43
Figura 20 – Exemplo de Marca <i>Ditherizada</i> Ordenadamente, Aplicada a uma Imagem (50% dos Pixels)	44
Figura 21 – Exemplo de Marca <i>Ditherizada</i> Aleatoriamente, Aplicada a uma Imagem (50% dos Pixels)	44
Figura 22 – Efeito da variação da constante de mesclagem	45
Figura 23 – Gradientes medianos obtidos	46
Figura 24 – Intensidades reconstruídas dos canais R, G e B	46
Figura 25 – Reconstrução da marca d’água obtida através do processamento de 48 imagens	47
Figura 26 – Estimativas da marca d’água para diferentes números de imagens processadas	47
Figura 27 – Marca <i>Adobe Stock</i> reconstruída	48
Figura 28 – Amostra do resultado da remoção nas imagens <i>Adobe Stock</i>	49

LISTA DE TABELAS

Tabela 1 – Variáveis	26
Tabela 2 – Funções	26
Tabela 3 – Métricas para Marca Simples $c = 0,25$	53
Tabela 4 – Métricas para Marca Simples $c = 0,5$	53
Tabela 5 – Métricas para Marca Simples $c = 0,75$	53
Tabela 6 – Métricas para Marca Distorcida $c = 0,5$	55
Tabela 7 – Métricas para Marca <i>Ditherizada</i> $c = 0,5$	57
Tabela 8 – Métricas para Marca <i>Ditherizada</i> $c = 0,75$	57
Tabela 9 – Métricas para Marca <i>Ditherizada</i> $c = 1$	57
Tabela 10 – Métricas para Marca <i>Ditherizada Aleatoriamente</i> $c = 0,5$	58
Tabela 11 – Métricas para Marca <i>Ditherizada Aleatoriamente</i> $c = 0,75$	58
Tabela 12 – Métricas para Marca <i>Ditherizada Aleatoriamente</i> $c = 1$	59
Tabela 13 – Métricas para Marca Sólida $c = 1$	61
Tabela 14 – Métricas para Marca Distorcida e <i>Ditherizada</i> ($c = 0,75$)	62

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Contextualização	17
1.2	Motivação e Justificativa	18
1.3	Objetivo	19
1.4	Estrutura do Trabalho	20
2	TEORIA	21
2.1	Adição de uma Marca D'água	21
2.1.1	Modelo de Composição Tradicional	21
2.2	Remoção de uma Marca D'água	22
2.2.1	Estimação e Reconstrução da Marca D'água	23
2.2.2	Localização da Marca D'água	24
2.2.3	Decomposição de uma Imagem Marcada	24
2.2.3.1	Detalhamento da Resolução do Sistema	27
2.3	Avaliação Da Qualidade de Imagens	30
3	MATERIAIS E MÉTODOS	33
3.1	Criação do Banco de Imagens Utilizado	33
3.1.1	Marca D'água Sintética	34
3.1.2	Marca D'água Comercial (<i>Adobe Stock</i>)	35
3.2	Remoção das Marcas D'água	35
3.3	Criação e Aplicação de Marcas Melhoradas	40
3.3.1	Distorções	40
3.3.2	Efeito <i>Dithering</i> ("Porta de Tela")	42
4	RESULTADOS	45
4.1	Adição de Marca D'Água	45
4.2	Estimativa e Reconstrução da Marca D'Água	46
4.3	Remoção da Marca D'água	49
4.3.1	Resultados para Marca Comercial ("Adobe Stock")	49
4.3.2	Resultados para Marca Simples	51
4.3.3	Resultados para Marca Distorcida ($c = 0,5$)	53
4.3.4	Resultados para Marca <i>Ditherizada</i>	56
4.3.4.1	<i>Dither regular</i>	56
4.3.4.2	<i>Dither aleatório</i>	58
4.3.5	Outros Resultados	59

4.3.5.1	Marca Sólida	59
4.3.5.2	Distorções e <i>Ditherização</i> ($c = 0,75$)	61
4.3.5.3	Rotações Aleatórias	63
4.3.6	Comparação Gráfica	64
5	CONCLUSÃO	69
	Referências	71

1 INTRODUÇÃO

1.1 Contextualização

Com o advento de imagens digitais, distribuídas em quantidades enormes através da internet e entre diversos modos de armazenamento de dados virtuais, surge a necessidade da criação de métodos novos para a proteção dos direitos autorais, o “*copyright*”, destas imagens. Para isso, surgiu a ideia da criação de “marcas d’água digitais”, do inglês “*Digital Watermarking*” ([Minerva M. Yeung, 1997](#)). Tradicionalmente, as marcas d’água são usadas em documentos de papel oficiais e cédulas monetárias como defesa a sua falsificação por entidades não autorizadas. Para este uso, estas marcas são comumente geradas através de variações na densidade e espessura do papel utilizado, gerando diferentes características ópticas na imagem impressa dependendo da iluminação ou ângulo de visão. Dada a dificuldade na reprodução destas marcas, sua presença é evidência da originalidade do documento ou impressão ([Biermann, 1996](#)). Um exemplo deste tipo de marca está na figura 1.

No mundo digital, marcas d’água também são sobrepostas a imagens as quais se deseja proteger, como em fotografia profissional do tipo “*Stock Photos*”, que são fotos profissionalmente capturadas cujos direitos de uso são comercialmente vendidos, para aplicação em campanhas de *marketing*, embalagens de produtos, propagandas, entre outros objetivos diversos ([Frosh, 2001](#)), necessitando então da aquisição de uma licença para seu uso de forma legal. Sem esta licença, somente é fornecida ao público uma versão da foto que contém uma marca d’água altamente visível sobre a imagem, denotando quem detém seus direitos intelectuais e desincentivando seu uso de forma gratuita por entidades não autorizadas. Devido a grande facilidade de compartilhamento e reprodução de imagens pela internet, este tipo de proteção é muito demandado. Estas marcas d’água comumente possuem opacidade reduzida, mas que após sua aplicação degradam a qualidade da imagem ou foto original, de forma que a remoção desta marca não é um processo trivial, sendo estes um dos objetivos fundamentais desejados com o uso de marcas d’água visíveis ([Fred Mintzer, 1997](#)). Um exemplo de marca d’água visível aplicada em uma imagem digital está na figura 2.



Figura 1: Marca d’água tradicional em uma cédula de 20 euros

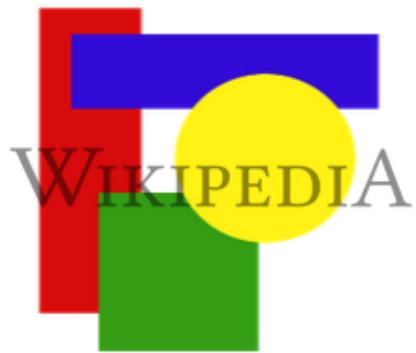


Figura 2: Exemplo de marca d’água digital (“Wikipedia”)

Além de marcas d’água de alta visibilidade, existem também as de baixa visibilidade, que não alteram visivelmente a imagem mas adicionam informações a esta de modo que seja possível a certificação da origem da imagem, através da análise dos bits específicos que a formam. Esta técnica está relacionada aos conceito de impressão digital, em inglês “*digital fingerprinting*”, e ao conceito de esteganografia, que se refere à ocultação de uma mensagem no interior de outra (Potdar et al., 2005). Estas marcas d’água invisíveis, apesar de servirem um objetivo similar às marcas visíveis, não substituem inteiramente seu uso, visto que há diversas situações em que deseja-se prontamente alertar o visualizador da imagem quem é seu detentor de “*copyright*” (Kankanhalli et al., 1999), por exemplo. Neste trabalho serão abordadas apenas as marcas do tipo visível, cujo característica principal é ser prontamente visível quando aplicada.

1.2 Motivação e Justificativa

Utilizando técnicas tradicionais de edição de imagens, como o software “*Adobe Photoshop*”, a remoção de marcas d’água do tipo visível é um processo árduo e demorado, além de exigir experiência por parte do editor, que deve manualmente reconstruir as partes afetadas pela marca d’água na imagem. Contudo, através do avanço em técnicas de processamento de imagens e aprendizado de máquina, há a possibilidade da implementação de um algoritmo que estima a marca d’água e sua máscara de opacidade, aplicadas em conjunto na foto alvo. Tendo a marca corretamente estimada, deseja-se que o algoritmo remova a marca da imagem, restaurando seu estado natural. Tal procedimento baseia-se no pressuposto que a mesma marca d’água é aplicada de forma consistente a um grande número de fotos, o que é verdade para as “*Stock Photos*”, permitindo que a marca d’água seja melhor estimada (Dekel et al., 2017). A viabilidade de um algoritmo deste tipo foi estudada e demonstrada no trabalho publicado pela *Google Research*, intitulado “*On the Effectiveness of Visible Watermarks*”, (Dekel et al., 2017), que servirá de base para a elaboração deste trabalho. Por ser um trabalho recente, ainda não há na literatura exemplos

de marcas d'água visíveis feitas explicitamente com o objetivo de serem resistentes a esse algoritmo de remoção.

Devido então à possibilidade do desenvolvimento de um programa capaz de remover estas marcas d'água de alta visibilidade de forma sistematizada e automatizada, existe o interesse em melhor compreender o funcionamento deste ataque e a proposição de técnicas capazes de aprimorar a robustez de marcas d'água tradicionais, por meio de transformações de tamanho, formato ou cor, adição de ruídos aleatórios, ou técnicas de sobreposição da marca de maior complexidade, como a proposta por [Mohanty et al.](#) (2000), que desenvolve um modelo matemático para a sobreposição de uma marca d'água com base no domínio DCT (“*Discrete Cosine Transform*”).

1.3 Objetivo

Este projeto propõe a implementação de um ataque a marcas d'água de alta visibilidade com base na análise de um banco de dados de imagens identicamente marcadas, e em seguida a avaliação de possíveis métodos de proteção a este procedimento, de forma que o mesmo algoritmo de remoção não seja capaz de remover por completo as marcas de robustez ampliada. Serão estudados três métodos de forma detalhada: Distorções aleatórias, aplicação de *dithering*, e rotações aleatórias. Estes métodos serão analisados e comparados utilizando métricas numéricas de qualidade.

1.4 Estrutura do Trabalho

Este documento está dividido nas seguintes seções, brevemente descritas abaixo:

- **Introdução:** Contextualização, motivação e objetivos do trabalho.
- **Teoria:** Descrição teórica dos conceitos necessários para a implementação dos algoritmos.
- **Materiais e Métodos:** Detalhamento dos processos realizados para a implementação de cada passo necessário dos algoritmos.
- **Resultados:** Demonstração e discussão dos resultados obtidos e dos dados quantitativos de qualidade calculados.
- **Conclusão:** Condensação dos resultados e discussão desta monografia.
- **Apêndice:** Exemplos adicionais do desempenho do algoritmo de remoção de marcas d'água.

2 TEORIA

Esta seção aborda a teoria necessária para a compreensão dos procedimentos relacionados a adição de uma marca d'água a uma imagem, bem como o equacionamento teórico relacionado ao algoritmo de ataque às marcas d'água adicionadas de forma idêntica a grandes bancos de imagens.

2.1 Adição de uma Marca D'água

Para adicionar uma marca d'água a uma imagem, além da marca e da própria imagem, também é necessária a informação de como a marca será sobreposta à imagem, isto é, qual será sua intensidade, ou transparência, na imagem final. A informação necessária para isto é conhecida como a opacidade da marca, que é representada por meio de uma máscara “*alpha*”, que é o terceiro sinal a ser considerado, além da imagem e marca.

Apesar de existirem modelos complexos de composição de imagens sobrepostas por marcas d'água, que utilizam transformadas “*Wavelet*” ou no domínio “*DCT*”, a maioria das imagens marcadas encontradas em uso são compostas através de um modelo padrão aditivo para a adição da marca d'água em imagens (Dekel et al., 2017), descrito na subseção seguinte.

2.1.1 Modelo de Composição Tradicional

Consideramos então que a imagem original é representada por $I(p)$, a marca d'água por $W(p)$, e sua máscara alpha por $\alpha(p)$, temos então que a imagem marcada final, $J(p)$, é obtida através da composição aditiva denotada na equação 2.1:

$$J(p) = \alpha(p)W(p) + (1 - \alpha(p))I(p) \quad (2.1)$$

Em que “*p*” representa a posição “*x, y*” de cada pixel da imagem, marca e máscara alpha.

Tipicamente, deseja-se que tanto a imagem original como a marca d'água sejam parcialmente visíveis, e portanto para isto considera-se que $\alpha(p) = c \times \alpha_n(p)$, em que $c < 1$ é uma constante de mesclagem entre imagem e marca, e $\alpha_n(p) \in [0, 1]$ é a máscara normalizada para cada um dos pixels. Este processo de composição é exemplificado na figura 3, utilizando o fator de mesclagem $c = 0,5$ e uma marca d'água sintética criada para os testes:

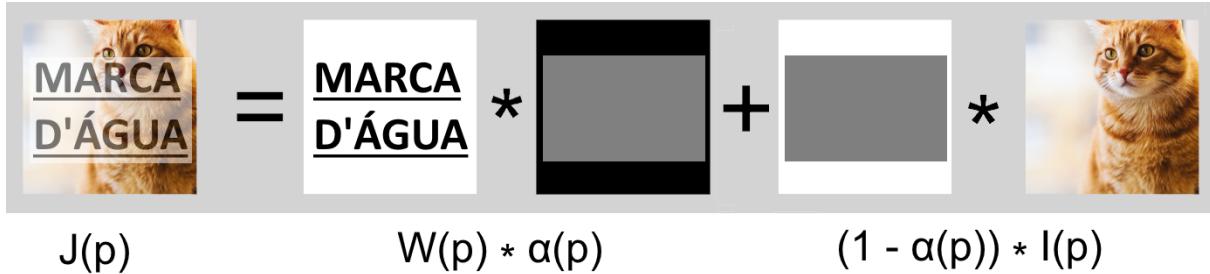


Figura 3: Exemplificação do modelo de composição de uma imagem marcada

Enquanto que uma constante de mesclagem de 0,5 permite a visualização simultânea da marca d'água e da imagem original, valores menores para α tornam a marca mais sutil, enquanto que maiores dão prevalência à marca na imagem final, com o valor limite 1, em que os pixels da imagem original são totalmente substituídos por pixels da marca d'água, de maneira irrecuperável.

2.2 Remoção de uma Marca D'água

Dada a equação 2.1, de adição de uma marca d'água, o processo inverso de remoção pode ser explícito na equação 2.2:

$$I(p) = \frac{J(p) - \alpha(p)W(p)}{1 - \alpha(p)} \quad (2.2)$$

Contudo, a remoção na prática de uma marca d'água por meio desta equação é inviável, uma vez que para cada pixel existem 3 incógnitas: W , α , e I , e apenas uma equação de formação 2.1. É devido a esta subdeterminação do sistema, portanto, que surge a necessidade de se utilizar uma coleção de imagens marcadas consistentemente pela mesma marca W e opacidade α , permitindo que este problema seja solucionado com boa precisão (Dekel et al., 2017).

A coleção de K imagens marcadas, com o índice “(p)” omitido para brevidade, pode ser representada como:

$$J_k = \alpha W + (1 - \alpha)I_k, \quad k = 1, \dots, K \quad (2.3)$$

A remoção de marcas d'água consistentemente aplicadas a uma coleção de imagens, portanto, baseia-se em um algoritmo de alguns passos:

- Estimar e Reconstruir a Marca D'água
- Encontrar Posição das Marcas na Coleção de Imagens
- Decompor Imagens Marcadas e Aprimorar Estimativas Iterativamente

2.2.1 Estimação e Reconstrução da Marca D'água

O passo fundamental para a remoção de uma marca é a estimação desta. Para este objetivo, é calculado o gradiente de K imagens da coleção nas direções X e Y, e então é obtida a mediana entre todos estes gradientes, de forma que apenas os gradientes consistentes a todas as imagens permaneçam, formando uma estimativa do gradiente da marca d'água utilizada na coleção de imagens. Esta estimativa de partida é equacionada a seguir:

$$\nabla \hat{W}_m(p) = \text{mediana}_k(\nabla J_k(p)) \quad (2.4)$$

Como demonstrado em [Dekel et al. \(2017\)](#), quanto maior o número K de imagens processadas, mais a estimativa do gradiente $\nabla \hat{W}_m(p)$ se aproxima dos gradientes reais da marca d'água ($W_m = \alpha W$) com a presença de um desvio, como demonstrado a seguir, a partir da equação de formação [2.1](#) e do pressuposto que I_k (Imagen original) e W_k (Marca d'água original) são sinais aleatórios, em que $E[X]$ representa seu valor estatístico esperado.

$$E[\nabla J_k] = E[\nabla W_m] + E[\nabla((1 - \alpha)I_k)] = E[\nabla W_m] + E[\nabla I_k] - E[\nabla(\alpha I_k)] \quad (2.5)$$

Utilizando propriedades de multiplicação da derivada (regra do produto), $E[\nabla(\alpha I_k)]$ pode ser expandido:

$$E[\nabla J_k] = \nabla W_m + E[\nabla I_k] - E[\nabla(\alpha)I_k + \alpha \nabla(I_k)] = \nabla W_m + E[\nabla I_k] - \nabla \alpha E[I_k] - \alpha E[\nabla I_k] \quad (2.6)$$

Por fim, tendo que a probabilidade de muitas imagens terem gradientes fortes em uma única posição é pequena, $E[I_k] \approx 0$,

$$E[\nabla J_k] = \nabla W_m - \nabla \alpha E[I_k] \quad (2.7)$$

Ou seja, o valor esperado para a mediana dos gradientes das imagens da coleção coincide com os gradientes da marca real diferindo apenas nos pixels em que o gradiente de alpha, $\nabla \alpha$, é diferente de zero, por um desvio de $\nabla \alpha E[I_k]$. A princípio este desvio é aceito para a formação da estimativa inicial de W_m , mas ele será minimizado e corrigido na estimativa final.

Tendo então a estimativa $\nabla \hat{W}_m(p)$ calculada, é realizada a reconstrução da marca d'água através da integração dos gradientes na direção X e Y, para cada um dos canais R, G, e B da marca.

2.2.2 Localização da Marca D'água

Comumente marcas d'água são aplicadas no centro da imagem ou em alguma posição pré-definida, para toda imagem da coleção. Neste casos, é apenas necessário a identificação prévia deste local, sem grandes dificuldades então para se encontrar a região marcada em todas as imagens. Contudo, também é frequente, como uma proteção básica contra métodos de remoção automática de marcas d'água, a randomização das coordenadas na imagem em que a marca é aplicada. Para que qualquer algoritmo de remoção seja aplicado, é antes necessário o conhecimento destas coordenadas.

Para que a posição da marca seja encontrada em todas as imagens da coleção, deseja-se utilizar técnicas de reconhecimento de padrões em imagens a partir de uma marca d'água pré-estimada, algo que pode ser realizado através da seleção prévia manual da região em que a marca d'água está em apenas uma das imagens, através da qual é realizada a pré-estimação de seu gradiente, assim como definido na seção anterior. Essa estimativa será refinada iterativamente assim que mais marcas forem detectadas na coleção, resultando por fim em uma estimativa mais robusta e capaz de detectar todas ou quase todas as marcas desejadas.

2.2.3 Decomposição de uma Imagem Marcada

Tendo em mãos uma aproximação inicial da marca d'água $\widehat{W}_m(p)$, e sua localização em cada uma das imagens da coleção, deseja-se decompor as imagens marcadas $J_k(p)$ em suas componentes $W(p)$, $\alpha(p)$ e $I_k(p)$, respectivamente, a marca d'água, sua máscara de opacidade, e imagem original. Contudo, não há informações suficientes nas imagens marcadas para o encontro de uma solução exata, mesmo com o uso de uma coleção grande de imagens. Assim, este problema é redefinido como a busca de uma solução estimada que minimize uma função de erro adequada, ou seja, é resolvido por meio de um sistema de otimização iterada. São então formuladas as seguintes equações que estimam o erro da solução do problema (Dekel et al., 2017):

$$E_{\text{data}}(I_k, W, \alpha) = \sum_p \Psi(|\alpha W + (1 - \alpha)I_k - J_k|^2) \quad (2.8)$$

$$\Psi(s^2) = \sqrt{s^2 + \epsilon^2}, \epsilon = 0.001 \quad (2.9)$$

Em que $E_{\text{data}}(I_k, W, \alpha)$ é a função de erro que penaliza diferenças entre a imagem formada pela equação 2.1 de formação feita com os resultados estimados da imagem decomposta. A função $\Psi(s^2)$ é apenas uma função simples de regularização dos dados.

$$E_{\text{reg}}(\nabla I) = \sum_p \Psi(|\alpha_x| I_x^2 + |\alpha_y| I_y^2) \quad (2.10)$$

$$E_{\text{reg}}(\nabla W) = \sum_p \Psi(|\alpha_x| W_x^2 + |\alpha_y| W_y^2) \quad (2.11)$$

$$E_{\text{reg}}(\nabla \alpha) = \sum_p \Psi(\alpha_x^2 + \alpha_y^2) \quad (2.12)$$

$E_{\text{reg}}(\nabla I)$, $E_{\text{reg}}(\nabla W)$ e $E_{\text{reg}}(\nabla \alpha)$ são as funções de erro de regularização que penalizam mudanças bruscas de intensidade na marca d'água e imagem reconstruídas nas regiões em que o gradiente da máscara de opacidade é mais forte, ou seja, encoraja o resultado final a ser suave nestas regiões de borda. I_x , W_x e α_x são as derivadas na direção X destes sinais. Similarmente, I_y , W_y e α_y , são as derivadas na direção y.

$$E_f(\nabla W_m) = \sum_p \Psi(\|\nabla W_m - \nabla \widehat{W}_m\|^2) \quad (2.13)$$

$E_f(\nabla W_m)$ é a função de erro que estimula o resultado a apresentar gradientes similares aos obtidos na estimativa inicial \widehat{W}_m . Por fim, na equação 2.14, com o uso destas funções de erro é escrita a função que se deseja minimizar, para encontrar uma solução otimizada ao problema de decomposição de uma coleção de K imagens marcadas:

$$\begin{aligned} \arg \min_{W, \alpha, \{I_k\}} & \sum_k (E_{\text{data}}(W, \alpha, I_k) + \lambda_I E_{\text{reg}}(\nabla I_k)) \\ & + \lambda_w E_{\text{reg}}(\nabla W) + \lambda_\alpha E_{\text{reg}}(\nabla \alpha) + \beta E_f(\nabla(\alpha W)) \end{aligned} \quad (2.14)$$

Em que os parâmetros λ e β são constantes que controlam a influência de cada termo. Encontrar os argumentos mínimos desta equação não é uma tarefa fácil, porém Dekel et al. (2017) sugerem a resolução deste problema pelo método dos mínimos quadrados iterativamente ponderados (“*Iteratively-Reweighted-Least-Square (IRLS)*”) (Burrus, 2012).

Para otimizar a resolução deste problema não linear e de muitas variáveis, são introduzidas variáveis auxiliares W_k , que representam uma estimativa individual da marca d'água para cada uma das imagens marcadas presentes no banco de dados, de forma que cada W_k deve ser próxima a estimativa geral da marca, W , de acordo com a função de erro E_{aux} , definida na equação 2.15, de forma similar às funções de erro anteriores: (Dekel et al., 2017)

$$E_{\text{aux}}(W, W_k) = \sum_p |W - W_k| \quad (2.15)$$

Com a adição de uma constante γ relacionada à função E_{aux} , o sistema final a ser resolvido, a fim de se encontrar os argumentos mínimos, é então o seguinte, dado pela

equação 2.16:

$$\arg \min \sum_k (E_{\text{data}}(I_k, W_k, \alpha) + \lambda_I E_{\text{reg}}(\nabla I_k) + \lambda_w E_{\text{reg}}(\nabla W_k) + \lambda_\alpha E_{\text{reg}}(\nabla \alpha) + \beta E_f(\nabla(\alpha W_k)) + \gamma \sum_k E_{\text{aux}}(W, W_k)) \quad (2.16)$$

A resolução deste sistema pode ser interpretada como as variáveis “ I_k ” (Imagens com marcas d’água retiradas), “ W_k ” (Aproximações da marca d’água aplicada a cada imagem), e “ α ” (Máscara de transparência da marca d’água) que produzem os menores valores de erros nas funções “ E_{data} ” (Garante que os resultados obtidos, quando recombinados, produzem imagens semelhantes às utilizadas na entrada), “ E_{reg} ” (Garante que os resultados tenham aparência relativamente suave), “ E_f ” (Estimula que a marca d’água estimada final seja similar à estimativa inicial), e “ E_{aux} ” (Aproxima estimativas individuais de cada marca à estimativa geral). As variáveis estão resumidas e organizadas na tabela 1, e as funções de erro na tabela 2.

Tabela 1: Variáveis

Variável	Descrição
I_k	Imagens Limpas
W_k	Estimativa Individual da Marca
α	Máscara de Transparência

Tabela 2: Funções

Função	Descrição
E_{data}	Imagen decomposta gera imagem original quando recomposta
E_{reg}	Suavidade na Aparência
E_f	Similaridade com Estimativa Inicial
E_{aux}	Similaridade de Estimativas Individuais com a Geral

Dekel et al. (2017) resolve este sistema em duas partes, primeiramente fixando “ α ” e “ W ”, a partir de estimativas iniciais, e através do sistema simplificado a seguir, encontra “ W_k ” e “ I_k ”:

$$\arg \min_{W_k, I_k} E_{\text{data}}(I_k, W_k) + \lambda_I E_{\text{reg}}(\nabla I_k) + \lambda_w E_{\text{reg}}(\nabla W_k) + \beta E_f(\nabla(\alpha W_k)) + \gamma E_{\text{aux}}(W, W_k) \quad (2.17)$$

Após a resolução deste sistema, que será detalhada posteriormente, a estimativa geral para a marca d’água “ W ” é atualizada de maneira simples utilizando o conjunto de

estimativas individuais “ W_k ”, por meio do cálculo da mediana:

$$W = \text{median}_k W_k \quad (2.18)$$

Por fim, a variável “ α ” é atualizada, mantendo as anteriores fixas, de maneira análoga a como “ W_k ” e “ I_k ” foram obtidas, mas desta vez minimizando o erro para obter o valor de α .

$$\sum_k E_{\text{data}}(\alpha, I_k, W) + \lambda_\alpha E_{\text{reg}}(\nabla \alpha) + \beta E_f(\nabla(\alpha W)) \quad (2.19)$$

Este processo é então iterado quantas vezes forem necessárias para a convergência do resultado, cuja velocidade de convergência depende das constantes $\lambda_I, \lambda_w, \lambda_\alpha, \beta$ e γ , que agem como pesos para cada uma das funções de erro que levam aos valores ideais de “ I_k ”, “ W_k ” e “ α ”.

2.2.3.1 Detalhamento da Resolução do Sistema

A resolução do sistema dado pela equação 2.16, como detalhado no material suplementar por Dekel et al. (2017), se dá pela transferência do domínio discreto (somatória) para o domínio contínuo (integral), com o uso de equações de Euler-Lagrange. Especificamente, através de um sistema de várias funções de várias variáveis (Courant and Hilbert, 1989), cujo formato está denotado nas equações 2.20, 2.21 e 2.22.

$$I[f_1, f_2, \dots, f_m] = \int_{\Omega} \mathcal{L}(x_1, \dots, x_n, f_1, \dots, f_m, f_{1,1}, \dots, f_{1,n}, \dots, f_{m,1}, \dots, f_{m,n}) \, dx \quad (2.20)$$

$$f_{i,j} := \frac{\partial f_i}{\partial x_j} \quad (2.21)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial f_1} - \sum_{j=1}^n \frac{\partial}{\partial x_j} \left(\frac{\partial \mathcal{L}}{\partial f_{1,j}} \right) &= 0_1 \\ \frac{\partial \mathcal{L}}{\partial f_2} - \sum_{j=1}^n \frac{\partial}{\partial x_j} \left(\frac{\partial \mathcal{L}}{\partial f_{2,j}} \right) &= 0_2 \\ \frac{\partial \mathcal{L}}{\partial f_m} - \sum_{j=1}^n \frac{\partial}{\partial x_j} \left(\frac{\partial \mathcal{L}}{\partial f_{m,j}} \right) &= 0_m \end{aligned} \quad (2.22)$$

Quando deseja-se resolver o sistema de equações 2.17, as funções de várias variáveis f_1, f_2, \dots, f_m representam as funções I_k e W_k , cujas variáveis são as coordenadas x e y. I_k e W_k são o resultado desejado na resolução deste problema. A somatória em k no sistema

[2.17](#) é substituída pela integral genericamente definida em Ω , e o termo L representa o termo interior a somatória, isto é:

$$\begin{aligned} L = & L_{\text{data}}(I_k, W_k, \alpha) + \lambda_I L_{\text{reg}}(\nabla I_k) + \lambda_w L_{\text{reg}}(\nabla W_k) + \\ & \lambda_\alpha L_{\text{reg}}(\nabla \alpha) + \beta L_f(\nabla(\alpha W_k)) + \gamma L_{\text{aux}}(W, W_k) \end{aligned} \quad (2.23)$$

Com as funções L_x sendo análogas as funções E_x anteriores, no domínio discreto. Com estas definições, as equações de Euler-Lagrange para o sistema, como exemplificadas genericamente na equação [2.22](#), são as seguintes:

$$\begin{aligned} \frac{\partial L}{\partial I^k(\mathbf{p})} - \frac{\partial}{\partial x} \frac{\partial L}{\partial(I_x^k(\mathbf{p}))} - \frac{\partial}{\partial y} \frac{\partial L}{\partial(I_y^k(\mathbf{p}))} \\ \frac{\partial L}{\partial W^k(\mathbf{p})} - \frac{\partial}{\partial x} \frac{\partial L}{\partial(W_x^k(\mathbf{p}))} - \frac{\partial}{\partial y} \frac{\partial L}{\partial(W_y^k(\mathbf{p}))} \end{aligned} \quad (2.24)$$

E substituindo os termos relevantes,

$$\frac{\partial L_{\text{data}}}{\partial I^k(\mathbf{p})} - \lambda_I \frac{\partial L_{\text{reg}}(\nabla I_k)}{\partial(I_x^k(\mathbf{p}))} - \lambda_I \frac{\partial L_{\text{reg}}(\nabla I_k)}{\partial(I_y^k(\mathbf{p}))} = 0 \quad (2.25)$$

$$\frac{\partial(L_{\text{data}} + \gamma L_{\text{aux}})}{\partial W^k(\mathbf{p})} - \frac{\partial}{\partial x} \frac{\partial(\beta L_f + \lambda_w L_{\text{reg}})}{\partial(W_x^k(\mathbf{p}))} - \frac{\partial}{\partial y} \frac{\partial(\beta L_f + \lambda_w L_{\text{reg}})}{\partial(W_y^k(\mathbf{p}))} = 0 \quad (2.26)$$

Com a finalidade de reescrever este sistema de forma matricial e resolvê-lo, [Dekel et al. \(2017\)](#) define as seguintes notações que serão utilizadas também para a implementação deste algoritmo computacionalmente:

$$\boldsymbol{\alpha} = \text{diag}(\alpha) \quad \bar{\boldsymbol{\alpha}} = \text{diag}(1 - \alpha) \quad (2.27)$$

Em que “diag” representa uma matriz diagonal com os elementos de “ α ”.

$$\begin{aligned} \Psi'_{\text{data}} &= \text{diag}\left(\Psi'\left(\left(\alpha W^k + (1 - \alpha) I_k - J_k\right)^2\right)\right) \\ \Psi'_{\text{w}} &= \text{diag}\left(\Psi'\left(\left(|\alpha_x| W_x^k + |\alpha_y| W_y^k\right)^2\right)\right) \\ \Psi'_{\text{I}} &= \text{diag}\left(\Psi'\left(\left(|\alpha_x| I_x^k + |\alpha_y| I_y^k\right)^2\right)\right) \\ \Psi'_{\text{f}} &= \text{diag}\left(\Psi'\left(\|\nabla(\alpha W^k) - \nabla W_m\|^2\right)\right) \\ \Psi'_{\text{aux}} &= \text{diag}\left(\Psi'\left(\left(W^k - W\right)^2\right)\right) \\ \Psi'_{\text{rI}} &= \text{diag}\left(\Psi'\left(\left(|\alpha_x| I_x^2 + |\alpha_y| I_y^2\right)\right)\right) \\ \Psi'_{\text{rw}} &= \text{diag}\left(\Psi'\left(\left(|\alpha_x| W_x^2 + |\alpha_y| W_y^2\right)\right)\right) \end{aligned} \quad (2.28)$$

$$\begin{aligned} \mathbf{L}_I &= \mathbf{D}_x^T \mathbf{c}_x \Psi'_{\text{rI}} \mathbf{D}_x + \mathbf{D}_y^T \mathbf{c}_y \Psi'_{\text{rI}} \mathbf{D}_y \\ \mathbf{L}_w &= \mathbf{D}_x^T \mathbf{c}_x \Psi'_{\text{rw}} \mathbf{D}_x + \mathbf{D}_y^T \mathbf{c}_y \Psi'_{\text{rw}} \mathbf{D}_y \\ \mathbf{A}_f &= \boldsymbol{\alpha}^T \underbrace{(\mathbf{D}_x^T \Psi'_f \mathbf{D}_x + \mathbf{D}_y^T \Psi'_f \mathbf{D}_y)}_{L_f} \boldsymbol{\alpha} + \gamma \Psi'_{\text{aux}} \end{aligned} \quad (2.29)$$

$$\mathbf{c}_x = \text{diag}(|\alpha_x|) \quad \mathbf{c}_y = \text{diag}(|\alpha_y|) \quad (2.30)$$

$$\begin{aligned} b_w &= \boldsymbol{\alpha}^T \Psi'_{\text{data}} J^k + \beta \mathbf{L}_f W_m + \gamma \Psi'_{\text{aux}} W \\ b_I &= \bar{\boldsymbol{\alpha}}^T \Psi'_{\text{data}} J^k \end{aligned} \quad (2.31)$$

Em que $\Psi(s)'$ é a derivada da função $\Psi(s)$, (equação 2.9), de regularização definida anteriormente, e D_x , D_y representam operadores de derivada horizontal e vertical, respectivamente.

Com esta notação, é então composto o seguinte sistema matricial, que representa as equações 2.25 e 2.26 (Dekel et al., 2017):

$$\left[\begin{array}{c|c} \boldsymbol{\alpha}^2 \Psi' \Psi'_{\text{data}} + \lambda_w \mathbf{L}_w + \beta \mathbf{A}_f & \boldsymbol{\alpha} \bar{\boldsymbol{\alpha}} \Psi'_{\text{data}} \\ \boldsymbol{\alpha} \bar{\boldsymbol{\alpha}} \Psi'_{\text{data}} & \bar{\boldsymbol{\alpha}}^2 \Psi'_{\text{data}} + \lambda_I \mathbf{L}_I \end{array} \right] \left[\begin{array}{c} W^k \\ I_k \end{array} \right] = \left[\begin{array}{c} b_w \\ b_I \end{array} \right] \quad (2.32)$$

A implementação computacional desta solução envolve, portanto, a composição desta matriz através do cálculos dos termos não lineares Ψ' , através da estimativa atual de α e W , e então resolução do sistema de equações dado em 2.32, tendo como resultado estimativas atualizadas de W_k e I_k .

Por fim, com base nestas estimativas atualizadas, o valor de α é recalculado, por um método análogo ao anterior, porém com o sistema de equações dado em 2.19, que mantém as variáveis anteriores fixas, para encontrar α . O sistema, na forma “ $A\alpha = b$ ”, e as definições necessárias, são as seguintes:

$$\left(\sum_k \Psi'_k + \lambda_\alpha L_\alpha + \beta \tilde{A}_f \right) \alpha = \sum_k \mathbf{A}_k (J - I_k) + \beta \mathbf{W}^T \mathbf{L}_f W_m \quad (2.33)$$

$$\begin{aligned} \mathbf{W} &= \text{diag}(W) \\ \Psi'_k &= \text{diag} \left(\Psi' \left((\alpha W + (1 - \alpha) I^k - J^k)^2 (W - I_k) \right) \right) \\ \mathbf{L}_\alpha &= \mathbf{D}_x^T \Psi'_\alpha \mathbf{D}_x + \mathbf{D}_y^T \Psi'_\alpha \mathbf{D}_y \\ \tilde{A}_f &= \mathbf{W}^T \mathbf{L}_f \mathbf{W} \\ \Psi'_\alpha &= \text{diag} \left(\Psi' \left(\|\nabla \alpha\|^2 \right) \right) \end{aligned} \quad (2.34)$$

As imagens I_k e W_k são então obtidas de forma intercalada com α , até que o resultado seja satisfatório.

2.3 Avaliação Da Qualidade de Imagens

Para a avaliação da qualidade de imagens são frequentemente utilizados métodos como o “Erro Quadrático Médio” (MSE - *Mean Squared Error*), “Pico da Relação Sinal Ruído” (PSNR - *Peak Signal to Noise Ratio*), “Relação Sinal Ruído” (SNR - *Signal to Noise Ratio*) e o Índice de Similaridade Estrutural (SSIM - *Structural Similarity Index*) ([Zhou Wang et al., 2004](#)).

O método mais simples e direto, MSE, é obtido através da média das diferenças das intensidades elevadas ao quadrado de duas imagens, ou seja:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (2.35)$$

Em que I, K representam as duas imagens sendo comparadas, e m, n , suas dimensões. K é a imagem de referência, não degradada por ruídos, ou no caso deste trabalho, por marcas d’água, e I é a imagem que sofreu alguma degradação. O valor desta métrica pode então ser calculado uma vez comparando a imagem original e a imagem com marca d’água antes do processo de remoção, e uma segunda vez com a imagem marcada após o processo de remoção, obtendo-se dois valores cuja diferença indicará a melhora ou piora na qualidade da imagem, em relação à imagem original não modificada.

Contudo, o método MSE não é muito robusto e depende muito dos possíveis valores de intensidade da imagem, portanto é mais frequentemente aplicada a métrica PSNR (com unidades em dB), que continua o conceito da métrica MSE da seguinte forma:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10} (MAX_I) - 10 \cdot \log_{10} (MSE) \end{aligned} \quad (2.36)$$

Em que MAX_I representa o valor máximo de intensidade da imagem, por exemplo, 255 em uma imagem de 8 bits. Desta forma a métrica PSNR não é mais dependente da amplitude dos dados da imagem, e por ser uma métrica logarítmica, funciona bem mesmo para grandes variações. De forma similar à métrica PSNR, a métrica SNR é definida como a relação entre a potência do sinal puro e do sinal de ruído em uma imagem, diferindo da PSNR que é a relação entre o valor máximo alcançável para o tipo da imagem e a potência do ruído, de forma que $PSNR \geq SNR$, porém com ambas sendo métricas comparáveis e similares.

Um exemplo do comportamento da métrica PSNR em uma imagem de teste com diferentes níveis de ruído pode ser visto na figura 4.

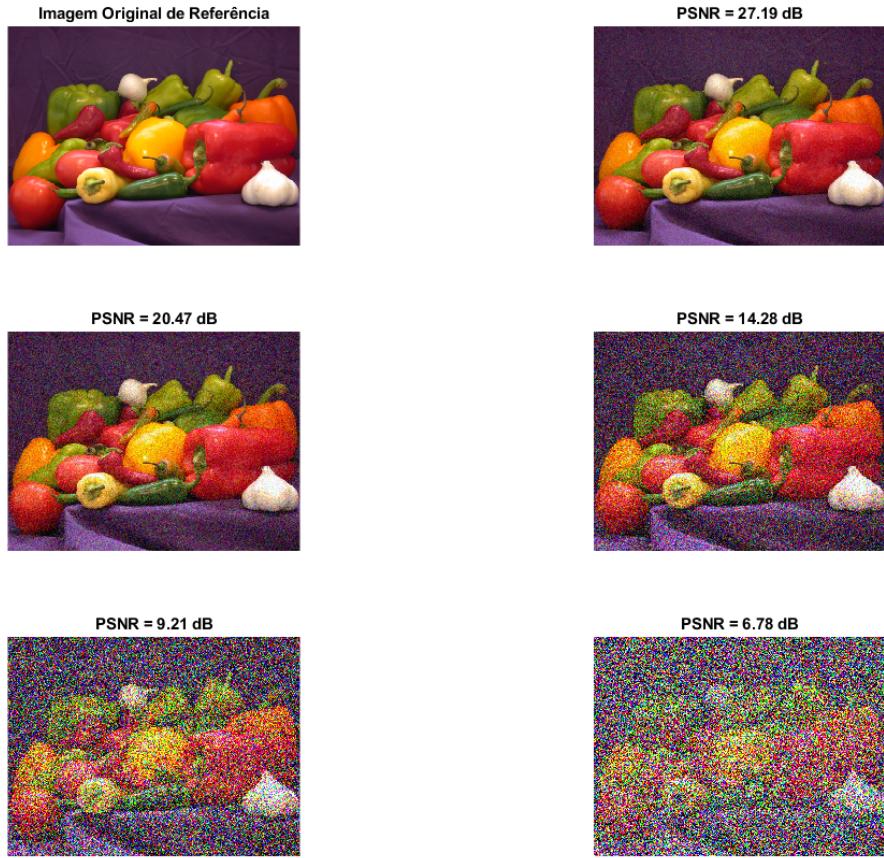


Figura 4: PSNR de várias imagens com diferentes quantidades de ruído

Apesar de muito úteis no estudo de ruídos, as métricas SNR e PSNR nem sempre são adequadas para a análise da qualidade visual de uma imagem, isto é, é possível que uma imagem de menor PSNR possua, visualmente, uma qualidade superior a uma imagem de maior PSNR, devido à forma com que interpretamos imagens, em que certos tipos de ruídos ou erros são mais visíveis que outros, algo não levado em consideração por estas métricas. Foi, portanto, desenvolvida a métrica SSIM, com o intuito de resolver estes problemas na avaliação de qualidade de imagens. ([Zhou Wang et al., 2004](#))

O índice de similaridade, SSIM, também entre uma imagem de referência e uma degradada, seja por ruído, marcas, ou compressão, denotadas por x e y , é calculado da seguinte forma:

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (2.37)$$

Que é uma combinação dos termos “ l ”, de luminância, “ c ”, de contraste, e “ s ”, de

estrutura (*structure*), com pesos α , β e γ . Estes termos são definidos a seguir:

$$\begin{aligned} l(x, y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(x, y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(x, y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{aligned} \quad (2.38)$$

$$C_1 = (K_1 L)^2 \quad (2.39)$$

$$C_2 = (K_2 L)^2 \quad (2.40)$$

$$C_3 = C_2/2 \quad (2.41)$$

Em que, μ_x , μ_y representam a média das imagens, σ_x , σ_y , a variância, e σ_{xy} , a covariância. K_1 e K_2 são constantes pequenas para estabilidade da divisão, por padrão definidas como $K_1 = 0.01$ e $K_2 = 0.03$. Os pesos α , β e γ , quando todos iguais a 1, permitem a simplificação das fórmulas anteriores ao resultado explícito na equação 2.42:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.42)$$

O cálculo desse índice entre duas imagens retorna um valor entre 0 e 1, em que 1 representa duas imagens idênticas, e 0 representa similaridade nula. Assim como as métricas SNR e PSNR, este índice será usado para comparar a similaridade da imagem marcada com a imagem original, antes e depois de seu processamento, com o objetivo de se observar um aumento na similaridade entre as imagens após a tentativa de remoção da marca d'água adicionada à imagem.

3 MATERIAIS E MÉTODOS

Esta seção abordará os passos tomados a fim da implementação computacional do algoritmo para remoção de marcas d'água detalhado na seção teórica deste trabalho, assim como as tarefas necessárias para validar e quantificar seu funcionamento. Por fim, também são abordadas as tarefas relacionadas aos métodos sugeridos de melhoria da robustez de marcas d'água, com a finalidade que este algoritmo de remoção abordado apresente eficiência reduzida.

O computador utilizado para a elaboração e execução final dos programas relevantes possui um processador “*Intel Core i7 4790k*”, 12GB de memória RAM, placa de vídeo “*GeForce GTX 970*” e sistema operacional Windows 8.1. Devido a implementação não otimizada por paralelização dos algoritmos, o fator principal que limita sua velocidade de execução é a velocidade do processador, em apenas uma *thread* lógica. Com exceção do algoritmo de remoção, que será detalhado separadamente no trabalho, foi utilizado o ambiente de programação *Matlab* para a manipulação das imagens e marcas, devido a grande disponibilidade de funções e algoritmos relacionados ao processamento de imagens já fornecidos pela linguagem.

3.1 Criação do Banco de Imagens Utilizado

Para a elaboração e teste do algoritmo, foram primeiramente adquiridas fotos suficientes para simular um grande banco de dados de “*Stock Photos*” identicamente marcadas por marcas d'água, como os disponíveis na internet para uso comercial, por exemplo, “*Adobe Stock*” ou “*Getty Images*”, entre muitos outros. Para que o algoritmo possa ser avaliado de forma objetiva, foram requisitadas fotos originalmente limpas, que não apresentam marcas d'água, de modo que uma marca sintética possa ser aplicada a estas e que o resultado final da remoção possa ser comparado com imagens limpas. Assim, o provedor de imagens limpas utilizadas foi o website “*pexels.com*”, no qual todas as fotos disponíveis estão em alta resolução, sem marcas d'água, e possuem licença que permite seu uso e reprodução de maneira livre.

Foram escolhidas 48 imagens de temática e aparência variadas, a fim de representar a grande variabilidade de imagens possíveis. Uma amostra destas imagens está na figura 5.

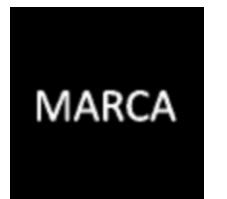


Figura 5: Amostra de imagens limpas adquiridas (Fonte: pexels.com)

Para facilitar o processamento das imagens, todas elas tiveram suas dimensões regularizadas para 500x500 pixels, através das funções “*imresize*” e “*padarray*” do *Matlab*, utilizado para o tratamento inicial do banco de imagens. Esta dimensão foi escolhida por ser suficientemente alta para que os detalhes das imagens e marcas sejam facilmente visíveis e claros, porém também buscando minimizar o tamanho das imagens para facilitar o seu processamento, tanto em quantidade de memória necessária e tempo de processamento.

3.1.1 Marca D’água Sintética

A marca d’água sintética, que foi aplicada nas imagens limpas do banco de dados, foi então gerada de forma simples e direta com o auxílio do programa gratuito de edição de imagens “*Paint.NET*”, bem como seu canal *alpha*, necessário para a aplicação das marcas, conforme demonstrado na figura 3, na seção de teoria do trabalho. A marca e seu canal *alpha* gerados estão na figura 6.



(a) Marca D’água (“ $W(p)$ ”)



(b) Canal Alpha (“ $\alpha(p)$ ”)

Figura 6: Marca D’água e seu Canal Alpha gerados

Estas marcas d’água foram geradas com a resolução de 200x200 pixels, para serem aplicadas no centro das imagens do banco de dados.

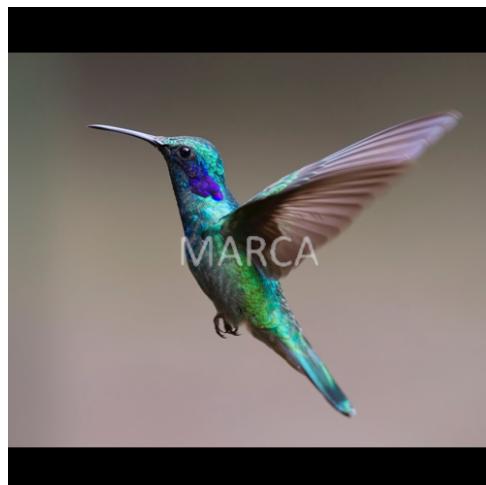


Figura 7: Exemplo de Imagem Marcada Sinteticamente

3.1.2 Marca D'água Comercial (*Adobe Stock*)

Além do banco de imagens limpas, também foram adquiridas imagens previamente marcadas por motivos comerciais, a fim de validar o funcionamento do algoritmo com exemplos reais e demonstrar a fragilidade destas marcas cujo objetivo é a identificação do detentor dos direitos autorais da foto. Para tal, foram obtidas imagens da provedora “*Adobe Stock*”, que marca todas suas imagens diretamente disponíveis em seu *website*, de maneira a incentivar a compra dos direitos de uso das imagens em sua resolução máxima e não marcada. Ao total, foram obtidas 60 imagens com a marca “*Adobe Stock*”, amostradas nas figuras 8 e 9.



Figura 8: Exemplo de imagem marcada comercialmente - “*Adobe Stock*” (Fonte: stock.adobe.com)



Figura 9: Outras imagens marcadas adquiridas (Fonte: stock.adobe.com)

Assim como as imagens anteriores, estas também são processadas para que tenham o mesmo formato 500x500 pixels, através de recortes em imagens maiores e do *padding* em imagens menores.

3.2 Remoção das Marcas D'água

Para a remoção das marcas d'água o algoritmo foi implementado na linguagem de programação *Python*, para ambas as versões 2.7, cujo suporte oficial será encerrado em breve, e 3.7, versão mais recente da linguagem *Python*. Essa linguagem foi escolhida por ter utilização muito ampla e bem documentada na internet, e por possuir sintaxe intuitiva, sendo uma boa escolha para a manipulação de imagens e outros dados.

As bibliotecas principais utilizadas foram “*OpenCV*”, uma biblioteca de código aberto com mais de 2000 funções relacionadas a visão computacional, “*numpy*”, “*scipy*”, bibliotecas da linguagem *Python* para a manipulação numérica de matrizes e resolução do sistema linear final obtido através do algoritmo, e “*matplotlib*”, também uma biblioteca *Python*, para a exibição das imagens e resultados durante a execução do código. O fluxo do algoritmo implementado é o seguinte:

1. Estimar Marca D’água Inicial
2. Detectar Localização da Marca em Todas as Imagens e recortá-las
3. Estimar Máscara α Inicial
4. Estimar Fator de Mesclagem c Inicial
5. Para cada uma das imagens de entrada:
 - a) Obter I_k e W_k mantendo α e W fixos
 - b) Atualizar Valor de W mantendo os demais fixos
 - c) Atualizar Valor de α mantendo os demais fixos
6. Repetir o passo acima um número pré-definido de iterações

O primeiro passo, estimar a marca inicial, se dá através do cálculo do gradiente de cada umas fotos do conjunto, seguido pela mediana entre todos estes gradientes, obtendo-se então um gradiente médio. A partir deste gradiente médio são utilizadas técnicas de *thresholding* (Gonzalez and Woods, 2006) a fim de que apenas os gradientes provenientes da marca d’água estejam presentes no resultado processado, obtendo-se neste passo o formato e tamanho que será utilizado de recorte para as marcas das imagens. Os resultados desse passo encontram-se na figura 10.



(a) Gradiente da Marca Não Processado



(b) Gradiente da Marca Processado



(c) Marca Recortada e Reconstruída através do Gradiente

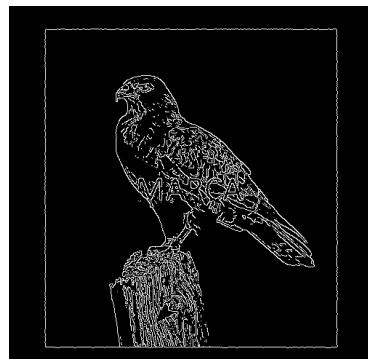
Figura 10: Processo de Obtenção da Estimativa Inicial da Marca D’água

A reconstrução da marca d'água a partir de seus gradientes, na direção X e Y, se dá através da reconstrução de Poisson (Pérez et al., 2003) Shen et al. (2005), com o cálculo do Laplaciano das imagens, isto é, o divergente do gradiente previamente calculado. O código implementado para esta reconstrução foi adaptado do disponível em: <https://web.media.mit.edu/~raskar/photo/code.pdf> (acessado em 22/11/2019).

O segundo passo, localizar a marca d'água em todas as imagens da coleção e recortá-las, pode ser simplificado quando todas as marcas são adicionadas às imagens na mesma posição, como é comum nas marcas comerciais, no entanto, o método de localização da marca implementado baseia-se na comparação entre as bordas detectadas em uma imagem marcada e as bordas de uma estimativa inicial da marca d'água, que pode até mesmo ter sido obtida através da reconstrução de poucas imagens cuja localização da marca foi manualmente denotada. A comparação, que resultará na posição da marca d'água, é feita através da filtragem no domínio do espaço (Gonzalez and Woods, 2006) da imagem marcada com suas bordas detectadas através do algoritmo de Canny (Canny, 1986) utilizando as bordas da estimativa da marca como o *kernel* do filtro. Isto é, é calculada a correlação entre estas duas imagens. Muitas vezes este processo de filtragem é referido como convolução entre imagens, contudo, a convolução inverte o *kernel*, enquanto que a correlação mantém sua orientação. As imagens relevantes a este passo estão na figura 11.



(a) Bordas Da Marca a ser Detectada



(b) Bordas Da Imagem Marcada



(c) Ampliação da região da marca na imagem, onde suas bordas serão detectadas

Figura 11: Exemplo de Imagens utilizadas para localização da Marca

Após o cálculo da correlação entre estas duas imagens, o resultado não terá

semelhança visual a nenhuma delas, porém, se a detecção for bem sucedida, o pixel de maior valor no resultado corresponde ao ponto na correlação em que o *kernel* do filtro, quando multiplicado e somado com os pixels correspondentes na imagem alvo, melhor coincidiu sobre as bordas desta imagem. Ou seja, onde houve maior sobreposição entre as bordas da marca estimada e das bordas da imagem alvo, denotando assim a posição esperada da marca na imagem. Na figura 12 a seguir, encontram-se exemplos dessa localização.

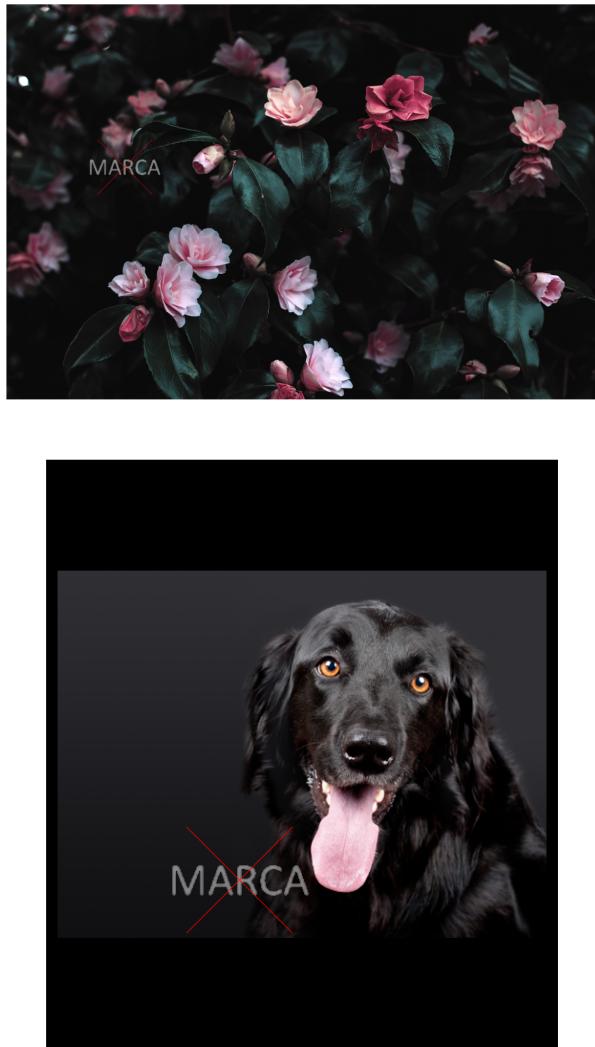


Figura 12: Exemplo de marcas localizadas pelo método de filtragem, denotadas pelo X vermelho

Para a simplificação dos testes, contudo, as marcas utilizadas para os testes a seguir foram todas adicionadas ao centro de todas as imagens, posição comum em “*stock photos*” comerciais, como nas imagens “*Adobe Stock*” obtidas, de modo que para a localização e recorte de todas as marcas foi apenas necessária a definição manual de uma janela em torno de uma das marcas, que será aplicada para todo o banco de imagens. Além do

método de filtragem descrito anteriormente para encontrar marcas, outras técnicas de localização de padrões em imagens também poderiam ser utilizadas com desempenhos similares ou superiores ao método da correlação, como por exemplo, as funções de detecção e extração de características de imagens, disponíveis no *toolbox* de visão computacional do *Matlab*, através da função “*matchFeatures*”.

O terceiro e quarto passos, responsáveis pela inicialização das estimativas da máscara de transparência, α , e da constante de mesclagem, c , são completados a partir da implementação do algoritmo de separação de plano de fundo e plano frontal de Levin et al. (2008), cujos resultados estão demonstrados nas figuras 13 e 14



Figura 13: Exemplo do funcionamento do algoritmo de Levin et al.



Figura 14: Imagem α da Marca D'água, obtida através do algoritmo de Levin et al. (Com ajuste de contraste para melhor visualização)

Como entrada deste passo são utilizados os recortes das imagens em suas regiões marcadas e a estimativa inicial de W adquirida anteriormente pela reconstrução Poisson, e a saída final é obtida pela mediana de todas as imagens “ α ” resultantes do algoritmo de separação de planos.

Por fim, o quinto e sexto passos são a implementação do algoritmo detalhado na seção de teoria, “Decomposição de uma Imagem Marcada”, também implementado em *Python*, cujo programa baseia-se na declaração das variáveis presentes nas equações de 2.27 a 2.34, na forma de matrizes diagonais esparsas, com o auxílio da biblioteca “*scipy*”, para o

sistema matricial principal [2.32](#) ser montado e solucionado através da função “`spsolve`”. As constantes utilizadas para a resolução do problema, em todos os exemplos deste trabalho, foram as seguintes: $\lambda_I = 1$; $\lambda_W = 0,005$; $\lambda_\alpha = 0,01$; $\beta = 1$ e $\gamma = 1$. Estes valores controlam a influência relativa de cada uma das funções avaliadoras de erro ([Tabela 2](#)), e foram escolhidos de modo que a influência de cada uma destas funções fosse perceptível, porém não excessiva em relação as demais.

O fim da execução do algoritmo ocorre quando se completa o número de iterações definidas no início da execução do programa. A quantidade de repetições é definida observando-se a qualidade da saída após cada iteração, de modo qualitativo, quando a marca já não é mais prominente na imagem, ou quantitativamente, quando a relação sinal-ruído da imagem ultrapassa um limiar desejado. Menos iterações são necessárias quando a estimativa inicial já é de boa qualidade.

3.3 Criação e Aplicação de Marcas Melhoradas

Tendo o algoritmo de remoção completo, foram abordadas técnicas cujo objetivo é o aumento da robustez de uma marca d’água frente a este algoritmo de remoção. Essas técnicas são a aplicação de distorções geométricas no formato da marca de forma aleatória, para que todas as imagens não compartilhem de exatamente a mesma marca, e da aplicação da “*ditherização*” nas marcas, que tem o objetivo de minimizar a regularidade e suavidade da marca, fator que é pressuposto para o bom funcionamento do algoritmo de remoção.

O detalhamento e implementação destes métodos está descrito nas seções seguintes deste trabalho.

3.3.1 Distorções

O fundamento do algoritmo de remoção baseia-se no fato que existe apenas uma marca que é aplicada identicamente a muitas imagens, portanto uma ideia natural para a melhora da robustez das marcas é a alteração destas de modo que não sejam todas idênticas. Para isto, é possível apenas a adição de ruídos simples aditivos à marca antes de sua aplicação, contudo, alterações sutis apenas nos valores dos pixels da marca são muito pouco eficientes frente ao algoritmo de remoção ([Dekel et al., 2017](#)), uma vez que seu funcionamento utiliza a mediana de muitas imagens, limitando naturalmente o efeito de ruídos aditivos aplicados.

Foram então testadas alterações geométricas aleatórias na marca. Para tal também foram utilizados ruídos aleatórios gerados proceduralmente ([Lagae et al., 2010](#)), porém com a finalidade de distorcer a imagem, e não como ruído aditivo.

O tipo de ruído escolhido para a execução das distorções aleatórias foi o ruído “*Perlin*” ([Perlin, 1985](#)), um tipo de ruído, normalmente bidimensional, mas extensível a

maiores dimensões, muito comumente aplicado quando deseja-se um ruído suave, isto é, que não possui variações abruptas de um ponto a outro (Lagae et al., 2010).

A distorção é obtida utilizando o ruído Perlin 2D (Perlin, 2002) gerado como mapas de deslocamento (“*Displacement Fields*”) para a imagem da marca e de seu canal α de opacidade, antes de serem aplicados na imagem alvo. O esquema pelo qual o ruído é adicionado como distorções na marca está representado na figura 15.

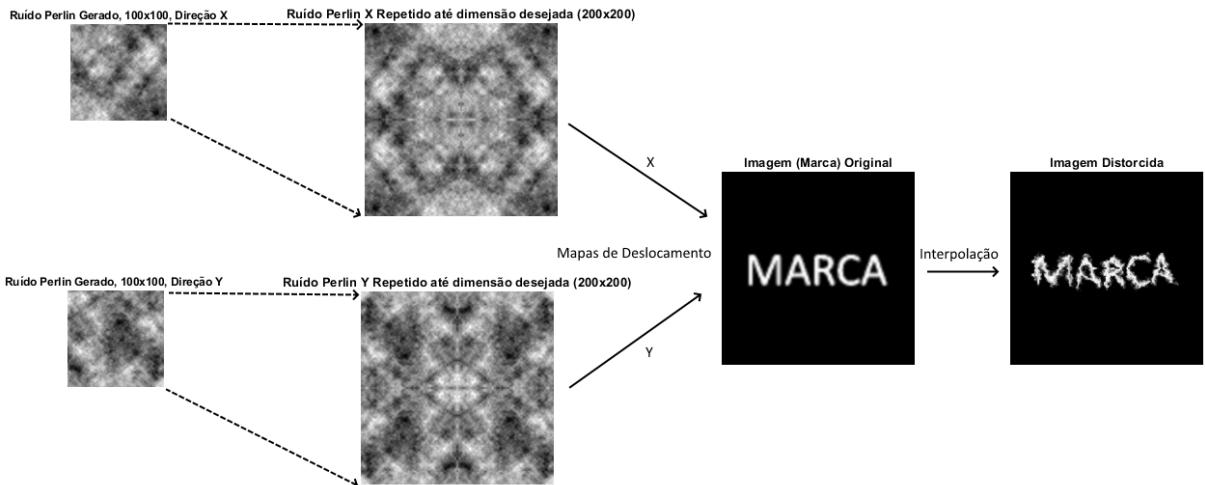


Figura 15: Esquema de aplicação de distorção

São inicialmente geradas duas imagens de ruído, uma representando o deslocamento que os pixels da imagem alvo sofrerão na direção X, e outra na direção Y. A amplitude deste deslocamento é definida de acordo com o valor do pixel na imagem de ruído, multiplicado por uma constante de intensidade de distorção. Como o ruído Perlin é contínuo, pixels próximos sofrem deslocamentos próximos, mantendo até certo ponto a integridade da imagem. A amplitude de distorção utilizada para os testes foi de 15 pixels, e o resultado final obtido pela função “*interp2*” do Matlab.

Essas imagens de ruído são geradas em resoluções menores que a final e espelhadas até que tenham o mesmo tamanho da imagem da marca d’água. Este processo é realizado pois quanto menor as dimensões da imagem de ruído inicial gerado, mais vezes esta terá que ser espelhada e portanto aumentará a frequência do ruído de distorção até o nível desejado. Este espelhamento também gera periodicidade na distorção aplicada, contudo esta é praticamente imperceptível, salvo em casos que o ruído inicial gerado é muito menor que a imagem alvo.

Com esta distorção cada uma das imagens recebe, portanto, uma marca estruturalmente única, a fim de melhorar sua robustez e dificultar sua remoção. Um exemplo encontra-se na figura 16.



Figura 16: Exemplo de marca distorcida aplicada a uma imagem

3.3.2 Efeito *Dithering* (“Porta de Tela”)

A segunda estratégia para o aumento da robustez das marcas está relacionada ao fato do algoritmo de remoção pressupor que as marcas são contínuas e suaves, o que é em geral verdade para as marcas simples utilizadas comercialmente. Para quebrar a continuidade das imagens foi utilizado o efeito de “*Dithering*” nas marcas.

“*Dithering*” refere-se a adição intencional de ruído a uma imagem de modo seu ruído de quantização seja aleatorizado, reduzindo, portanto, o efeito indesejado de “bandas” visíveis na imagem, como na figura 17 de exemplo.

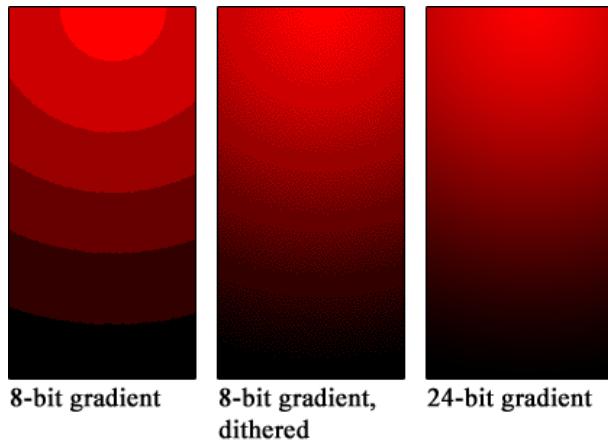


Figura 17: Exemplo da redução do efeito de bandas com o efeito *dither*, na imagem central (Fonte: Wikipedia)

No entanto, a relevância deste método está ligada ao uso do efeito *dither* para a geração de imagens “halftone” ([Veryovka and Buchanan, 1999](#)) ([Sindhu, 2013](#)). Nesta aplicação, a quantidade de pixels *ditherizados* escuros em uma imagem é proporcional à intensidade do tom de cinza na imagem de origem, gerando a impressão da existência de muitos tons de cinza na imagem *ditherizada*, mesmo quando esta é apenas binária. Um exemplo do uso deste efeito para a representação de tons de cinza está na figura 18.

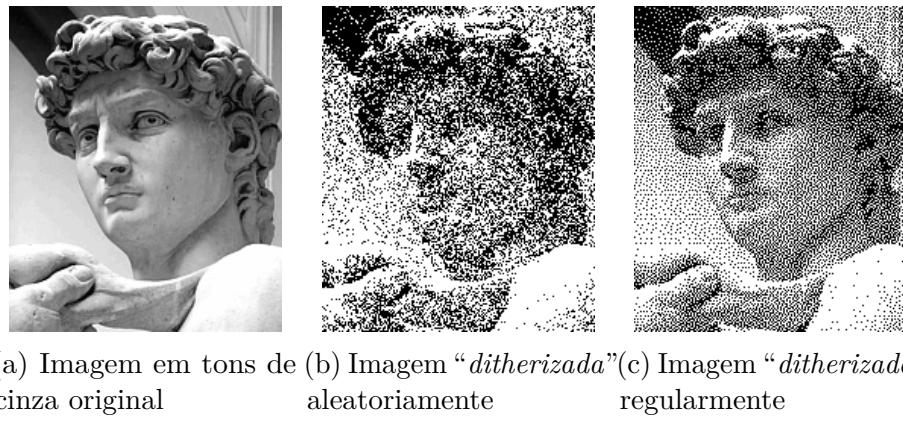


Figura 18: Exemplo do efeito *Dither* em imagens (Fonte: Wikipedia)

O método implementado para a melhora das marcas d’água funciona através do efeito de transparência obtido quando uma imagem é “*ditherizada*”, como no exemplo da figura 18. O objetivo é implementar o efeito de transparência “Porta de Tela” (*Screen-door Transparency*) (Sen et al., 2003), que é alcançando intercalando na imagem final os pixels visíveis entre a imagem de fundo e a imagem da marca, *ditherizada*.

Para tal, foram criadas máscaras de *ditherização* através da função “*dither*” do Matlab, aplicada a imagens de tons de cinza com a intensidade desejada para a transparência final da imagem. O efeito e aplicação destas marcas pode ser observado no esquema da figura 19:

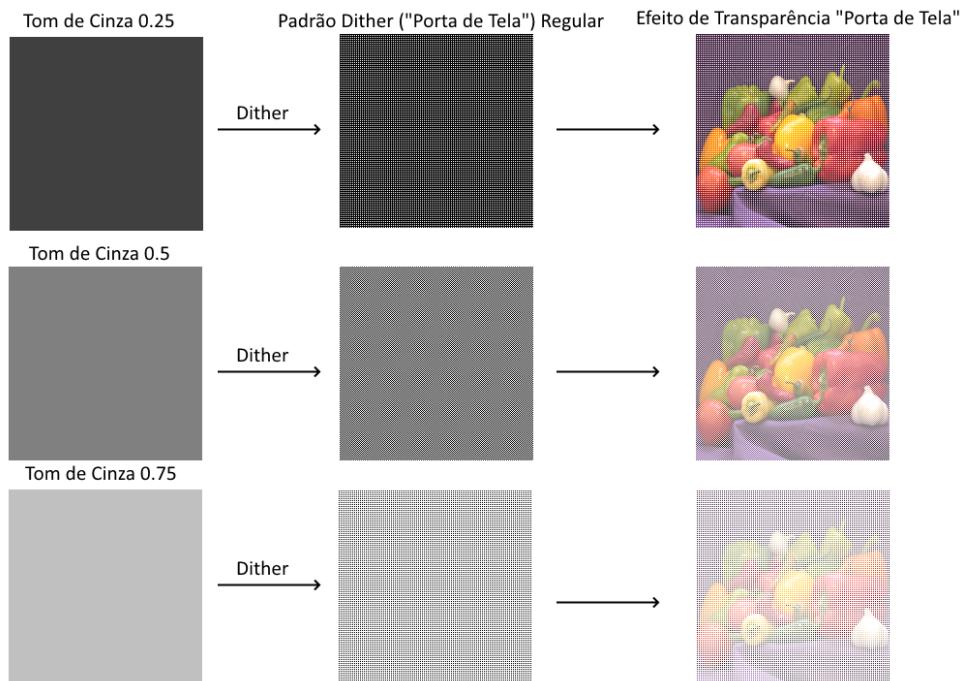


Figura 19: Funcionamento do efeito de transparência “Porta de Tela”

Na figura 19 fica visível que quanto menos pixels da imagem final estiverem presentes, maior é o efeito de transparência obtido, apesar de que na realidade cada pixel individual da imagem final ainda possui opacidade igual a 1, isto é, total. Um exemplo da aplicação deste método encontra-se na figura 20.



Figura 20: Exemplo de Marca *Ditherizada* Ordenadamente, Aplicada a uma Imagem (50% dos Pixels)

Também foram geradas marcas *ditherizadas* de forma aleatória, exemplificadas pela figura 21. Ao invés de pixels intercalados regularmente, os pixels que estarão presentes na marca são escolhidos aleatoriamente, na proporção desejada. Devido à aleatoriedade da marca, este método garante também que todas as marcas são diferentes das outras, apesar de que com um banco de imagens suficientemente grande a mediana das marcas tenderá rapidamente à marca original, intacta. Ambos os métodos, regular ou aleatório, garantem que marca possui muitas descontinuidades, dificultando sua remoção pelo algoritmo apresentado anteriormente.



Figura 21: Exemplo de Marca *Ditherizada* Aleatoriamente, Aplicada a uma Imagem (50% dos Pixels)

4 RESULTADOS

4.1 Adição de Marca D'Água

Aplicando uma marca d'água de teste a uma imagem para várias constantes de mesclagem, obteve-se o seguinte comportamento, ilustrado na figura 22:



(a) Marca com $c = 0,25$



(b) Marca com $c = 0,50$



(c) Marca com $c = 0,75$



(d) Marca com $c = 1,00$

Figura 22: Efeito da variação da constante de mesclagem

Ou seja, quanto menor o valor da constante “ c ”, menos visível é a marca d’água, algo que pode ser ideal caso o desejado seja que a marca influencie o mínimo possível na aparência da imagem, enquanto que valores mais altos de “ c ” a marca obstrui muito mais a imagem, situação que também pode ser desejável, caso o objetivo da marca seja, além de identificar a imagem, desincentivar ao máximo seu uso não licenciado. Contudo, o valor limite de “ $c = 1$ ”, apesar de evitar a reconstrução da imagem original, em geral

não é desejado pois toda informação visual original da imagem é perdida onde a marca é aplicada.

O valor “ $c = 0,5$ ” mostrou-se um bom ponto intermediário para a aplicação da marca d’água, uma vez que mantém componentes da marca e imagem original em mesma proporção.

4.2 Estimativa e Reconstrução da Marca D’Água

Utilizando uma coleção de 48 imagens identicamente marcadas em seu centro, foi então calculado o gradiente nas direções X e Y de todas as imagens, e em seguida a mediana de todos esses gradientes, para cada um dos canais RGB, obtendo-se os resultados dispostos na figura 23:



(a) Gradientes X e Y medianos - Canal R (b) Gradientes X e Y medianos - Canal G (c) Gradientes X e Y medianos - Canal B

Figura 23: Gradientes medianos obtidos

Através destes gradientes os canais RGB foram então reconstruídos individualmente, para em seguida serem compostos na estimativa inicial da marca d’água.



(a) Intensidade do canal vermelho reconstruído (b) Intensidade do canal verde reconstruído (c) Intensidade do canal azul reconstruído

Figura 24: Intensidades reconstruídas dos canais R, G e B

Devido a marca d’água de teste utilizada ser branca, os três canais obtidos, apresentados na figura 24, são muito semelhantes uns aos outros.

A composição destes canais em uma imagem colorida RGB, obtida através do processamento de 48 imagens marcadas, resulta na estimativa da figura 25, de qualidade suficiente para uma estimativa inicial que ainda será aprimorada iterativamente:



Figura 25: Reconstrução da marca d'água obtida através do processamento de 48 imagens

Observa-se que existem imperfeições na reconstrução ao redor da marca, efeito que é amenizado com o uso de mais imagens nesta estimação inicial, e intensificado caso sejam usadas menos imagens, como é visível nos seguintes resultados, para reconstruções realizadas com um número N de imagens entre apenas 2 até 48, na figura 26.



Figura 26: Estimativas da marca d'água para diferentes números de imagens processadas

Observa-se, portanto, que quanto menor o número de imagens utilizadas na estima-

ção maior é a influência destas imagens na reconstrução final, visto que para o caso $N = 2$, na sub-figura (a), elementos das imagens utilizadas são facilmente notados na reconstrução, enquanto que com o uso de uma coleção maior de imagens este efeito indesejado é cada vez menor e menos impactante no resultado.

O mesmo processo, utilizando as 60 imagens com a marca d'água “*Adobe Stock*”, resultou na seguinte reconstrução da marca, na figura 27:



Figura 27: Marca *Adobe Stock* reconstruída

4.3 Remoção da Marca D'água

4.3.1 Resultados para Marca Comercial (“Adobe Stock”)

Após o processamento do banco de 60 imagens com a marca “Adobe Stock”, com o uso de 5 iterações do algoritmo, foram obtidos os seguintes resultados, apresentados nas figuras 28 e 29, utilizando duas imagens como exemplo da efetividade de remoção para esta marca d’água comercial:



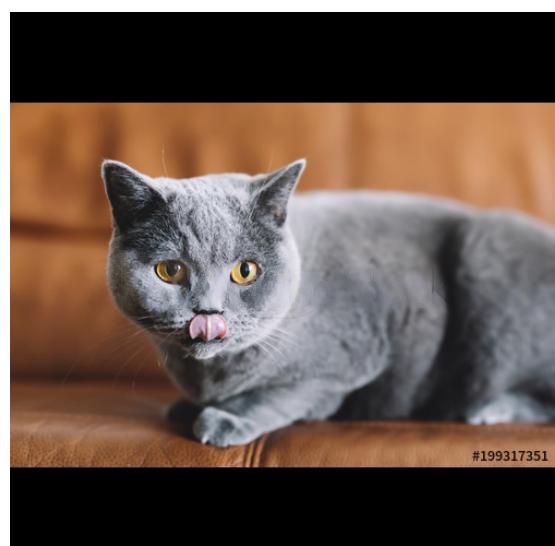
(a) Imagem Marcada Original



(b) Imagem Processada



(c) Imagem Marcada Original



(d) Imagem Processada

Figura 28: Amostra do resultado da remoção nas imagens *Adobe Stock*



(a) Imagem Marcada Original (Ampliação)



(b) Imagem Processada (Ampliação)



(c) Imagem Marcada Original (Ampliação)



(d) Imagem Processada (Ampliação)

Figura 29: Ampliação do resultado da remoção nas imagens *Adobe Stock*

O resultado, portanto, sendo analisado qualitativamente, atingiu um nível satisfatório, de forma que para as imagens não ampliadas é muito difícil a percepção dos traços residuais da marca. Ampliando a seção das imagens onde a remoção foi efetuada é possível notar estes resíduos da marca como pequenas distorções de aparência sombreada e borradada, especialmente em volta das bordas da marca. Mais exemplos dos resultados obtidos para esta marca estão no final deste trabalho, como apêndice.

É possível que com aprimoramentos ao algoritmo de remoção que estes defeitos se tornem ainda mais imperceptíveis, contudo, este resultado já mostra que uma implementação simples do algoritmo é capaz de remover uma marca d'água comercial ao ponto que esta não é mais facilmente reconhecível de maneira visual nas fotos. Assim, as

fotos processadas poderiam ser utilizadas de forma não autorizada sem que o detentor de seus direitos autorais (*Adobe Stock*) esteja visível na foto, destacando a existência da vulnerabilidade destas marcas d'água.

4.3.2 Resultados para Marca Simples

Para o banco de imagens marcadas sinteticamente, foram testados os efeitos de diferentes constantes de mesclagem no resultado final da remoção, com $c = 0,25$; $c = 0,5$ e $c = 0,75$:



(a) Marca Simples $c = 0,25$



(b) Marca Simples $c = 0,25$ pós Remoção



(c) Marca Simples $c = 0,5$



(d) Marca Simples $c = 0,5$ pós Remoção



(e) Marca Simples $c = 0,75$



(f) Marca Simples $c = 0,75$ pós Remoção

Figura 30: Antes e depois da remoção para imagens marcadas de forma simples

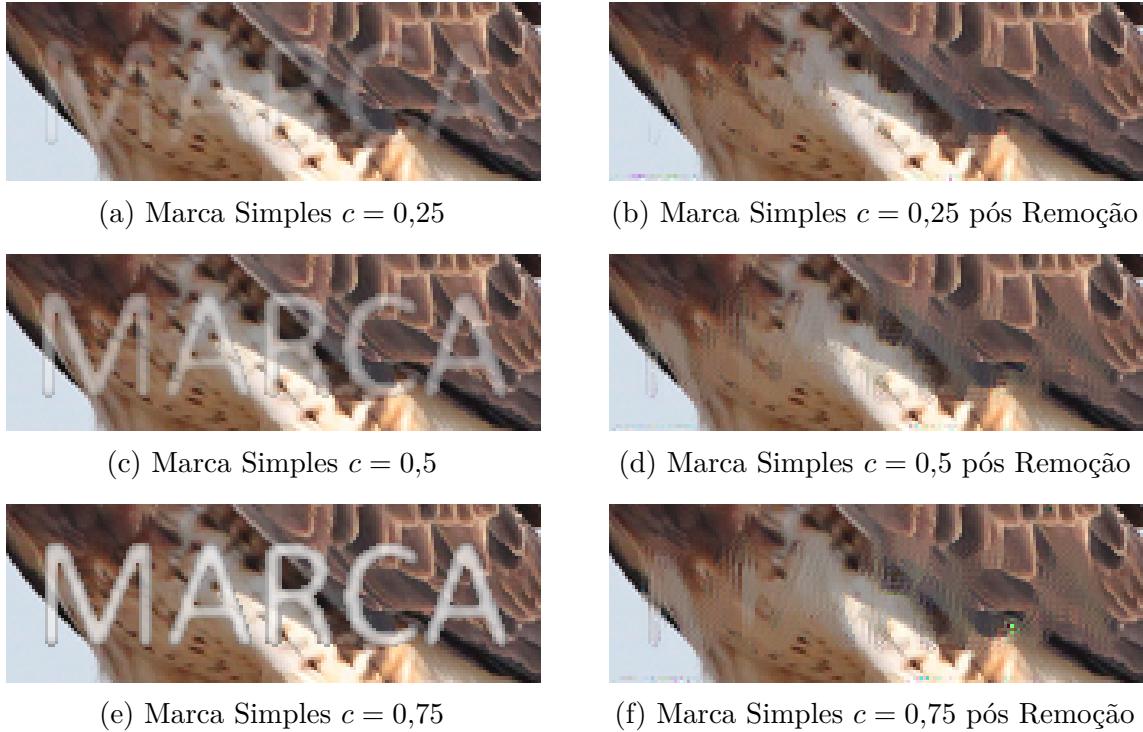


Figura 31: Antes e depois da remoção para imagens marcadas de forma simples (Ampliado)

Novamente pode-se observar, nas figuras 30 e 31, que o resultado da remoção é satisfatório para todas as constantes de mesclagem testadas, de modo que a marca d’água é quase imperceptível em imagens não ampliadas. Contudo, como pode ser observado nas seções ampliadas, quanto maior a constante de mesclagem mais intenso são os defeitos visíveis após a remoção, algo esperado, uma vez que estes casos representam imagens cuja marca tem maior impacto na imagem original. Assim, uma possível ação para a melhora da robustez das marcas é simplesmente torná-las menos transparentes, em situações em que a diminuição da visibilidade da imagem original é um compromisso aceitável.

Analizando estes resultados de forma quantitativa, foram calculadas as seguintes métricas médias entre todas as 60 imagens do banco de dados, antes e depois do processamento, em uma região central em torno da marca, em relação à imagem original não marcada: SNR (*Signal to Noise Ratio*), PSNR (*Peak Signal to Noise Ratio*) e SSIM (*Structural Similarity Index*).

O tamanho da região escolhida ao redor da marca tem influência nos valores absolutos obtidos das métricas PSNR, SNR e SSIM, porém não influenciam a variação encontrada entre as imagens antes e depois do processamento, contanto que a região englobe a marca inteira. Assim, o valor principal que deve ser analisado para comparação dos diferentes métodos é a diferença entre os valores da métrica, que representará uma melhora ou deterioração, caso variação negativa, na imagem.

Entende-se, portanto, que a marca d’água age como um ruído na imagem limpa

original, e o aumento dessas métricas representa a redução do efeito da marca na imagem.

Tabela 3: Métricas para Marca Simples $c = 0,25$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	26,2357 dB	32,2495 dB	6,0138 dB
SNR	20,1728 dB	26,1867 dB	6,0139 dB
SSIM	0,8814	0,9485	0,0671

Tabela 4: Métricas para Marca Simples $c = 0,5$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	20,2147 dB	27,1058 dB	6,8911 dB
SNR	14,1518 dB	21,0429 dB	6,8911 dB
SSIM	0,7734	0,8838	0,1104

Tabela 5: Métricas para Marca Simples $c = 0,75$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	16,71 dB	24,0032 dB	7,2932 dB
SNR	10,6472 dB	17,9403 dB	7,2931 dB
SSIM	0,6935	0,8339	0,1404

Através destes dados apresentados nas tabelas 3, 4 e 5, é então observado que quanto maior a constante de mesclagem, maior é a melhora relativa na imagem, porém quando os resultados são comparados entre si, quanto menor esta constante, melhor é a qualidade absoluta do resultado, visto que para o caso $c = 0,25$ o índice de similaridade final obtido foi de 0,9485, indicando que a imagem processada é muito similar à imagem original que nunca foi marcada.

Também fica claro por meio destes dados que as métricas PSNR e SNR diferem sempre pelo mesmo valor, sendo então redundantes. Para simplificação dos resultados apresentados seguintes serão apenas utilizadas as métricas PSNR e SSIM.

4.3.3 Resultados para Marca Distorcida ($c = 0,5$)

Repetindo a aplicação do algoritmo de remoção para as imagens cujas marcas foram distorcidas aleatoriamente, são obtidos os seguintes resultados, nas figuras 32 e 33:



(a) Imagem Com Marca Distorcida



(b) Imagem Processada

Figura 32: Antes e depois da remoção para marca distorcida



(a) Imagem Com Marca Distorcida (Ampliada)



(b) Imagem Processada (Ampliada)

Figura 33: Ampliação do resultado da remoção na imagem com marca distorcida

Diferentemente do obtido anteriormente para as marcas simples não distorcidas, desta vez o algoritmo é inefetivo na remoção da marca distorcida, de forma que esta é quase tão visível na imagem processada quanto na imagem marcada original. Isto ocorre devido

a impossibilidade do algoritmo estimar corretamente a marca aplicada a cada imagem, uma vez que cada uma delas difere significativamente devido a suas distorções aleatórias, que mostraram-se efetivas no aumento de robustez da marca frente a este algoritmo de remoção. Quantitativamente, as seguintes métricas foram levantadas, na tabela 6:

Tabela 6: Métricas para Marca Distorcida $c = 0,5$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	20,4639 dB	21,8468 dB	1,3829 dB
SSIM	0,7722	0,7919	0,0197

Ou seja, numericamente houve uma melhora menor, de apenas 1,3829 dB, após o processamento das imagens, quando comparado ao comportamento do algoritmo para as marcas regulares não distorcidas, que tiveram melhorias numericamente maiores e mais significativas.

4.3.4 Resultados para Marca *Ditherizada*

Para as marcas *ditherizadas* foram repetidos os testes alterando a transparência dos pixels presentes da marca, entre $c = 0,5$ e $c = 1$. É importante notar que como são aplicados apenas 50% dos pixels da marca a “transparência média”, que a marca aparenta ter quando a imagem é vista como um todo é, de fato, metade do valor da constante c . Ou seja, o efeito de transparência visual obtido para a situação $c = 1$ é similar ao visto na marca regular com $c = 0,5$.

4.3.4.1 *Dither* regular



(a) *Ditherizada* Regular $c = 0,5$



(b) *Ditherizada* Regular $c = 0,5$ Pós Remoção



(c) *Ditherizada* Regular $c = 0,75$



(d) *Ditherizada* Regular $c = 0,75$ Pós Remoção



(e) *Ditherizada* Regular $c = 1$



(f) *Ditherizada* Regular $c = 1$ Pós Remoção

Figura 34: Antes e depois da remoção para imagens com marcas *ditherizadas* regularmente (Ampliadas)

Nestes testes, apresentados na figura 34, é observado, assim como nos realizados para a marca comum, que marcas com constante de mesclagem maiores produzem defeitos de remoção mais visíveis, contudo, para as marcas *ditherizadas* estes defeitos são muito mais pronunciados ao longo de toda a marca, e não apenas nas bordas, como anteriormente. Mesmo para a marca bastante sutil, com $c = 0,5$, estes defeitos geram um efeito quadriculado bastante visível na imagem processada, devido a incompatibilidade do efeito *dither* com o algoritmo de remoção. Estas observações são afirmadas pelas métricas calculadas, a seguir, nas tabelas 7, 8 e 9:

Tabela 7: Métricas para Marca *Ditherizada* $c = 0,5$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	23,2336 dB	26,297 dB	3,0634 dB
SSIM	0,8197	0,8509	0,0312

Tabela 8: Métricas para Marca *Ditherizada* $c = 0,75$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	19,7286 dB	23,5306 dB	3,802 dB
SSIM	0,7551	0,8016	0,0465

Tabela 9: Métricas para Marca *Ditherizada* $c = 1$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	17,2334 dB	21,3458 dB	4,1124 dB
SSIM	0,7052	0,7598	0,0546

Este comportamento assemelha-se ao obtido para as marcas comuns, porém com melhorias inferiores em todas as métricas quando comparadas com as melhorias obtidas para as marcas comuns de mesma constante de mesclagem, sem modificações, demonstrando que a aplicação do efeito *Dither* gerou aumento na robustez das marcas como desejado.

4.3.4.2 Dither aleatório

Para as marcas *ditherizadas* aleatoriamente, são obtidos resultados visualmente similares aos anteriores, apresentados na figura 35, porém com defeitos de remoção de formato irregular ao invés de quadriculado. Numericamente, os resultados das métricas de qualidade estão nas tabelas 10, 11 e 12.

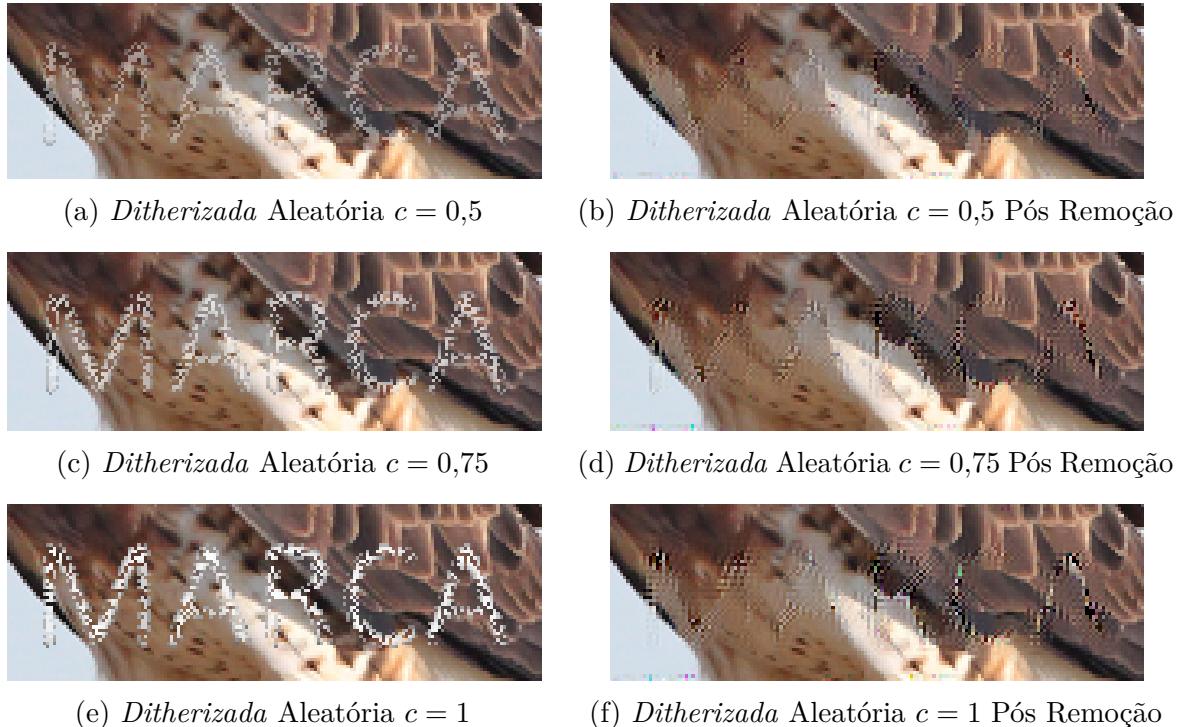


Figura 35: Antes e depois da remoção para imagens com marcas *ditherizadas* aleatoriamente (Ampliadas)

Tabela 10: Métricas para Marca *Ditherizada Aleatoriamente* $c = 0,5$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	23,2008 dB	24,7204 dB	1,5196 dB
SSIM	0,8215	0,8041	-0,0174

Tabela 11: Métricas para Marca *Ditherizada Aleatoriamente* $c = 0,75$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	19,7276 dB	25,3595 dB	5,6319 dB
SSIM	0,7583	0,8542	0,0959

Tabela 12: Métricas para Marca *Ditherizada* Aleatoriamente $c = 1$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	17,2135 dB	23,4382 dB	6,2247 dB
SSIM	0,708	0,8234	0,1154

Analizando os resultados qualitativos e quantitativos para as marcas aleatoriamente *ditherizadas* observa-se que o desempenho do algoritmo foi inferior ao obtido com as marcas simples, tradicionais. No entanto, apenas a situação de $c = 0,5$ mostrou-se mais robusta ao algoritmo quando comparada ao *dither* regular. Uma possível razão para este resultado é que os defeitos adicionados devido à aleatoriedade são capazes de sobrepor a melhora decorrente pela tentativa de remoção da marca $c = 0,5$, visto que, assim como observado nos resultados anteriores, marcas de baixa opacidade fornecem menores oportunidades de melhora, e para as situações $c = 0,75$ e $c = 1$, o efeito de remoção da marca é mais intenso que os defeitos decorrentes da aleatoriedade.

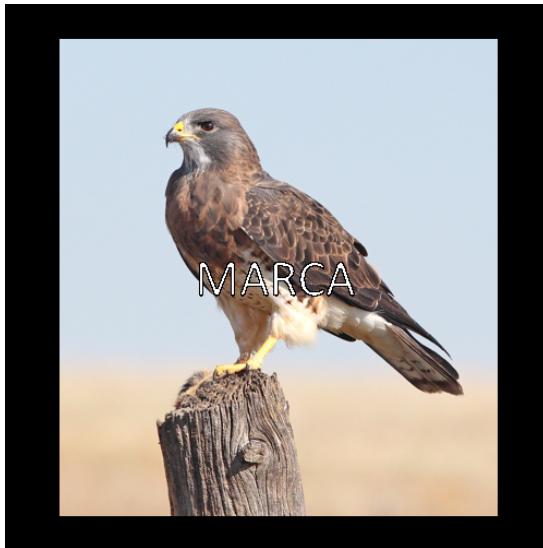
Esta linha de raciocínio também explica a razão do resultado negativo para a métrica SSIM no caso da marca com $c = 0,5$. O valor negativo indica que a qualidade visual da imagem piorou após a execução do algoritmo, mesmo que apenas ligeiramente, com um valor muito próximo a zero. Sua ocorrência para esta situação se deve ao fato das imperfeições adicionadas pelo algoritmo serem de magnitude aproximadamente equivalente ao efeito de melhora na imagem decorrente da remoção da marca. Ou seja, o efeito desejado do algoritmo, remover a marca, está equilibrado com os efeitos indesejados adicionados devido a irregularidade desta marca.

Logo, ambas as técnicas de *dithering* podem ser válidas para o aumento da robustez das marcas d'água, uma vez que a diferença principal entre o resultado destas duas é a aparência final dos defeitos de remoção, que podem ser regulares ou irregulares, e possuem intensidade relacionada com a constante de mesclagem escolhida para a aplicação da marca.

4.3.5 Outros Resultados

4.3.5.1 Marca Sólida

A maneira mais simples de marcar uma imagem é apenas sobrepor uma região desta com a marca desejada, substituindo inteiramente seus pixels, equivalente ao uso de uma constante de mesclagem $c = 1$. Isto não é um modo comum de se marcar imagens uma vez que impacta intensamente a qualidade visual da imagem, contudo, também foi testado o desempenho do algoritmo de remoção para essa situação, obtendo-se os seguintes resultados, nas figuras 36 e 37:



(a) Imagem Com Marca Sólida

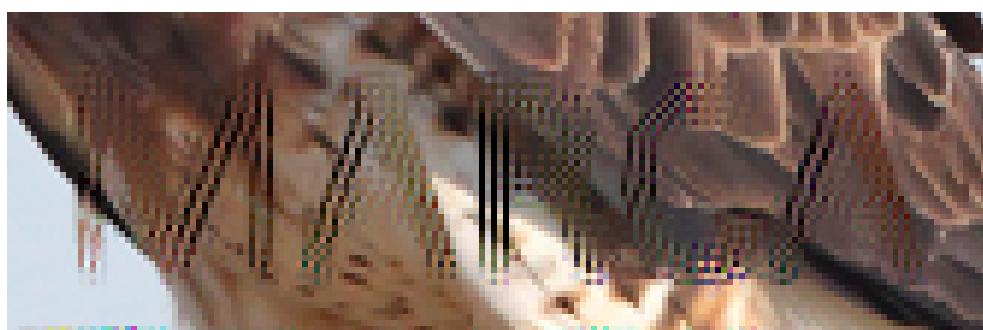


(b) Imagem Processada

Figura 36: Antes e depois da remoção para marca sólida



(a) Imagem Com Marca Sólida (Ampliada)



(b) Imagem Processada (Ampliada)

Figura 37: Ampliação do resultado da remoção na imagem com marca sólida

Ou seja, mesmo para uma imagem que teve alguns de seus pixels totalmente substituídos, o algoritmo tem o comportamento de remover a marca, que é facilmente estimada e reconstruída, e “borrar” as regiões desconhecidas de acordo com os pixels

próximos. Embora o resultado final ainda esteja claramente marcado e apresentando defeitos, mesmo para imagens não ampliadas, a melhora relativa entre a imagem marcada e a processada é maior que para as marcas testadas anteriormente, como demonstrado pelas métricas na tabela 13, a seguir:

Tabela 13: Métricas para Marca Sólida $c = 1$

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	11,2974 dB	19,0507 dB	7,7533 dB
SSIM	0,5642	0,7468	0,1826

Mesmo com a maior melhora relativa dos resultados, o resultado final ainda possui a pior das relações absolutas de sinal-ruído após a execução do algoritmo. Ou seja, a qualidade da imagem resultante é ruim quando comparada a imagens cujas marcas aplicadas possuíam índices de mesclagem menores que 1.

4.3.5.2 Distorções e *Ditherização* ($c = 0,75$)

De maneira a melhorar ainda mais a robustez das marcas d'água foram combinados ambas as técnicas de distorção e *ditherização* com $c = 0,75$, obtendo-se os seguintes resultados apresentados nas figuras 38 e 39, altamente resistentes ao algoritmo de remoção:



(a) Imagem Com Marca Distorcida e *Ditherizada*



(b) Imagem Processada

Figura 38: Antes e depois da remoção para marca distorcida e *ditherizada*

(a) Marca Distorcida e *Ditherizada* (Ampliada)

(b) Imagem Processada (Ampliada)

Figura 39: Ampliação do Resultado da Remoção na Imagem com Marca distorcida e *ditherizada*

Tabela 14: Métricas para Marca Distorcida e *Ditherizada* ($c = 0,75$)

Métrica	Antes de Processar	Após Processar	Diferença
PSNR	19,9683 dB	20,4623 dB	0,4940 dB
SSIM	0,7541	0,7468	-0,0073

A partir das figuras 38, 39, e dos dados quantitativos na tabela 14, é visível que o algoritmo foi inefetivo em remover esta marca que combina duas das técnicas abordadas neste trabalho, tendo a saída do algoritmo quase idêntica a sua entrada. É necessário considerar, no entanto, que a distorção e processamento excessivo das marcas pode torná-las difíceis de serem lidas, logo, deve-se realizar um compromisso entre dificuldade de remoção e clareza na marca. Uma maneira simples de controlar esta clareza é diminuir a intensidade das distorções, que foram aplicadas intensamente nas imagens analisadas.

Para este teste a métrica SSIM teve variação negativa, novamente denotando uma deterioração na qualidade da imagem quando quantificada por esta métrica. Esta diferença, de -0,0073, é muito próxima de zero, e indica que o algoritmo não teve grande impacto na direção de melhorar ou piorar a imagem. A mesma conclusão pode ser tomada a partir da diferença de 0,4940 dB na métrica PSNR.

4.3.5.3 Rotações Aleatórias

Por fim, foram testadas imagens cujas marcas receberam rotações aleatórias entre $+45^\circ$ e -45° , de modo que o algoritmo, como foi implementado, é totalmente incapaz de estimar corretamente as marcas d'água e executar o processo de remoção, uma vez que este pressupõe que todas as marcas estão ao menos na mesma orientação. Ao rotacionar as marcas o algoritmo não é capaz de reconhecê-las como a mesma marca em várias imagens, e portanto falha, estimando como marca apenas uma pequena região ao redor do ponto de rotação, demonstrado na figura 40, e nos resultados inefetivos das figuras 41 e 42.

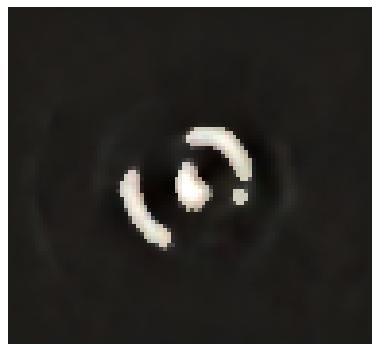


Figura 40: Estimativa Incorreta da Marca Aleatoriamente Rotacionada



(a) Imagem Com Marca Rotacionada

(b) Imagem Processada

Figura 41: Antes e depois da tentativa de remoção de marcas aleatoriamente rotacionadas

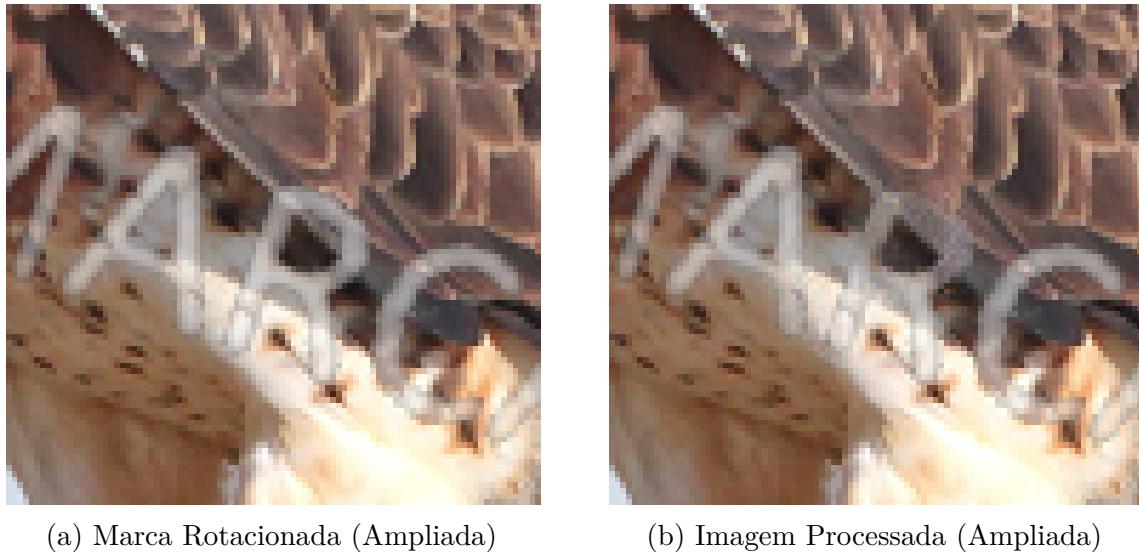


Figura 42: Ampliação do Resultado da Tentativa de Remoção da Marca Rotacionada

Ou seja, uma maneira muito simples de aumentar a robustez de marcas d’água é a realização de rotações aleatórias antes de sua aplicação em cada imagem. Essas rotações também podem ser combinadas com distorções e *ditherização*, se desejado e necessário.

Deve-se notar que existe a possibilidade da melhora do algoritmo de remoção para que este não pressuponha a mesma orientação de todas as marcas, contudo isto aumentaria a complexidade e dificuldade deste algoritmo, que deveria, em algum ponto de seu processo, alinhar todas as marcas após sua identificação por algum método que seja invariante a rotações. Com todas as marcas alinhadas, o algoritmo poderia proceder normalmente.

Apesar de não testado neste trabalho, alterações na escala da marca aplicada devem agir de forma análoga ao método de rotações aleatórias, impedindo que o algoritmo calcule a estimativa inicial sem que seja adicionado um passo adicional que corrija variações também na escala.

4.3.6 Comparação Gráfica

Para a visualização destes dados foram gerados os seguintes gráficos, nas figuras 43 e 44, para as métricas PSNR e SSIM dos resultados anteriores. Gráficos para a métrica SNR se comportam da mesma forma que os gráficos de PSNR, com um desvio de $6,06dB$ igual a todas as amostras, portanto não foram repetidos.

As barras em azul, a esquerda, representam os valores calculados antes do processamento das imagens, as barras em laranja, a direita, representam os resultados obtidos após o processamento. A ordem dos testes representados nos gráficos é, da esquerda para direita: Marca Sólida $c = 1$; Regular $c = 0,75$; Regular $c = 0,5$; Regular $c = 0,25$; *Dither* Aleatório $c = 1$; *Dither* Aleatório $c = 0,75$; *Dither* Aleatório $c = 0,5$; *Dither* Regular $c = 1$;

Dither Regular c = 0,75; Dither Regular c = 0,5; Distorções Aleatórias c = 0,5; Distorções mais Dither Regular c = 0,75; e por fim, Rotações Aleatórias.

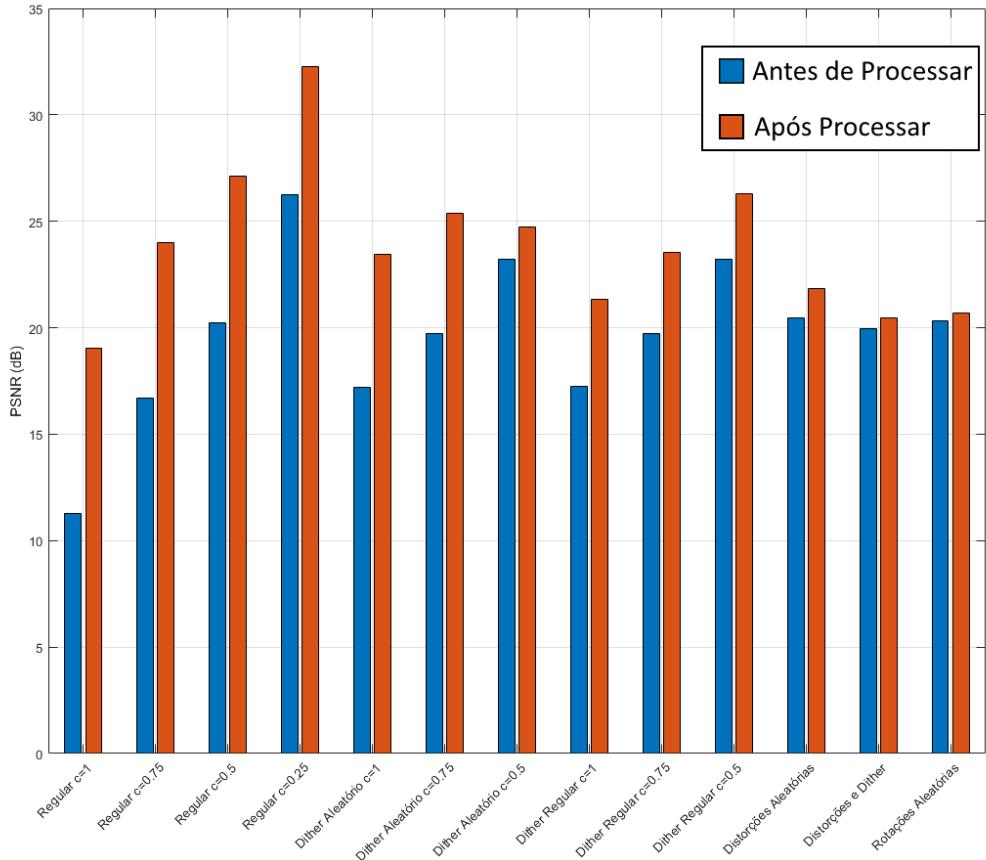


Figura 43: Comparação de valores de PSNR para todos os testes

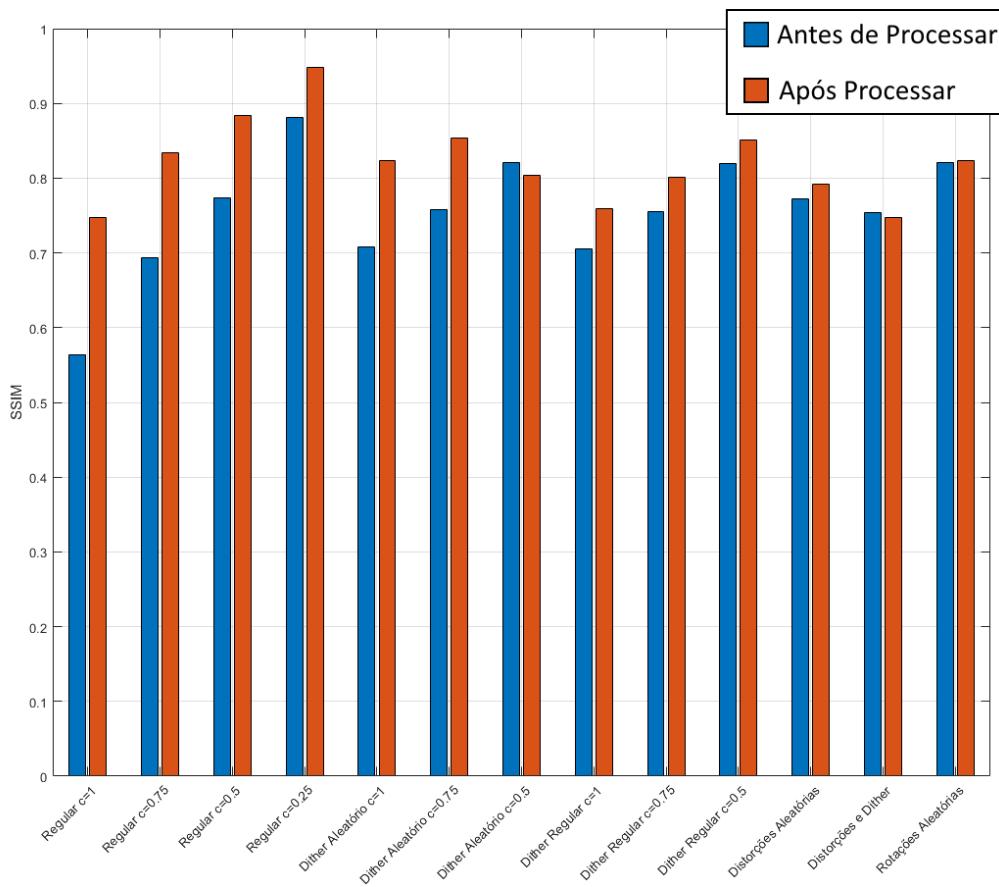


Figura 44: Comparação de valores de SSIM para todos os testes

Observa-se, então, que a melhora dos índices PSNR e SSIM é menor para as marcas que passaram pelos processos de aumento de robustez (distorção, *ditherização*), que para as marcas tradicionalmente adicionadas a imagens, que são os exemplos em que ocorreram as maiores variações destes índices. Quase não ocorreram variações para os métodos mais avançados de distorção e *ditherização* combinados, ou para o método de rotações.

Como o desempenho do algoritmo está ligado à diferença entre os valores obtidos antes e depois do processamento, foram gerados gráficos desta diferença, apresentados nas figuras 45 e 46, que confirmam o comportamento descrito para os diferentes métodos e destacam a vulnerabilidade das marcas adicionadas de forma simples e tradicional, como as imagens “*Adobe Stock*”, utilizadas de exemplo real e que cujas marcas foram removidas de maneira muito bem sucedida.

Para o gráfico de variação dos valores da métrica SSIM, na figura 46, destaca-se a presença de valores negativos e próximos de zero, que como discutido anteriormente, representam uma deterioração da qualidade visual destas imagens após seu processamento, quando quantificada pela métrica SSIM. Esse fenômeno ocorre quando as imperfeições adicionadas pelo algoritmo se sobressaem em relação a eventuais melhorias decorrentes da

tentativa de remoção da marca, e só foi presente nas situações em que a marca possuía baixa opacidade e alta dificuldade de remoção.

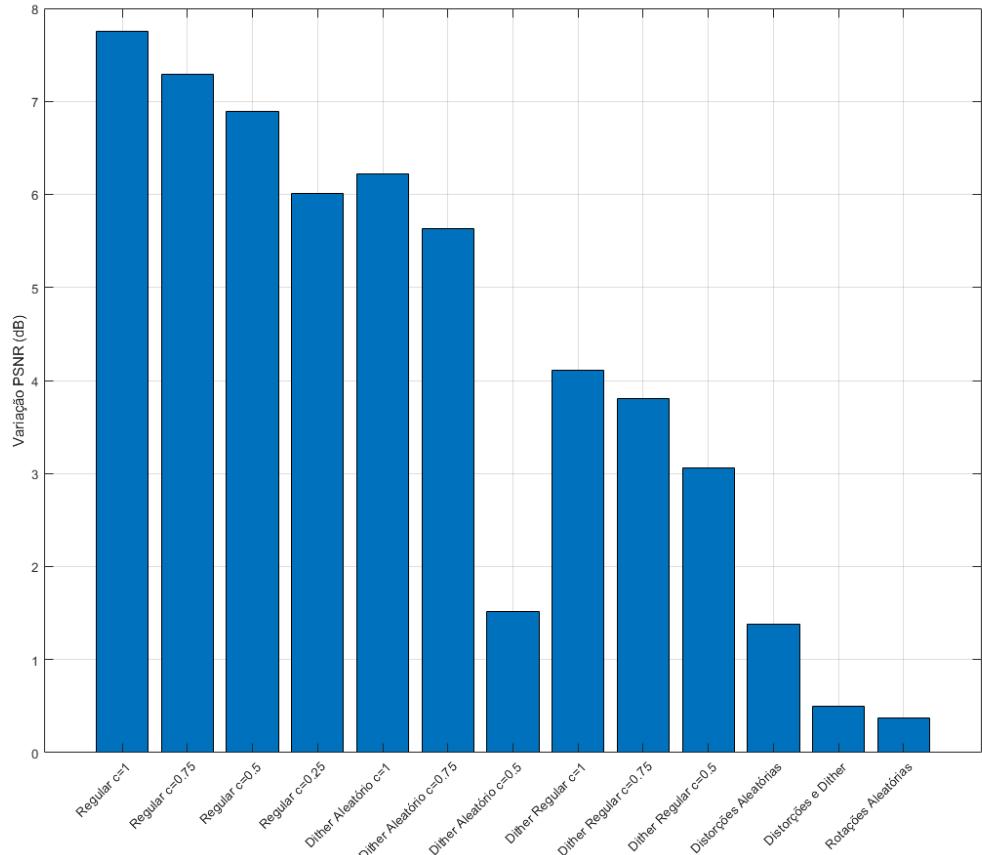


Figura 45: Comparação da variação dos valores de PSNR para todos os testes

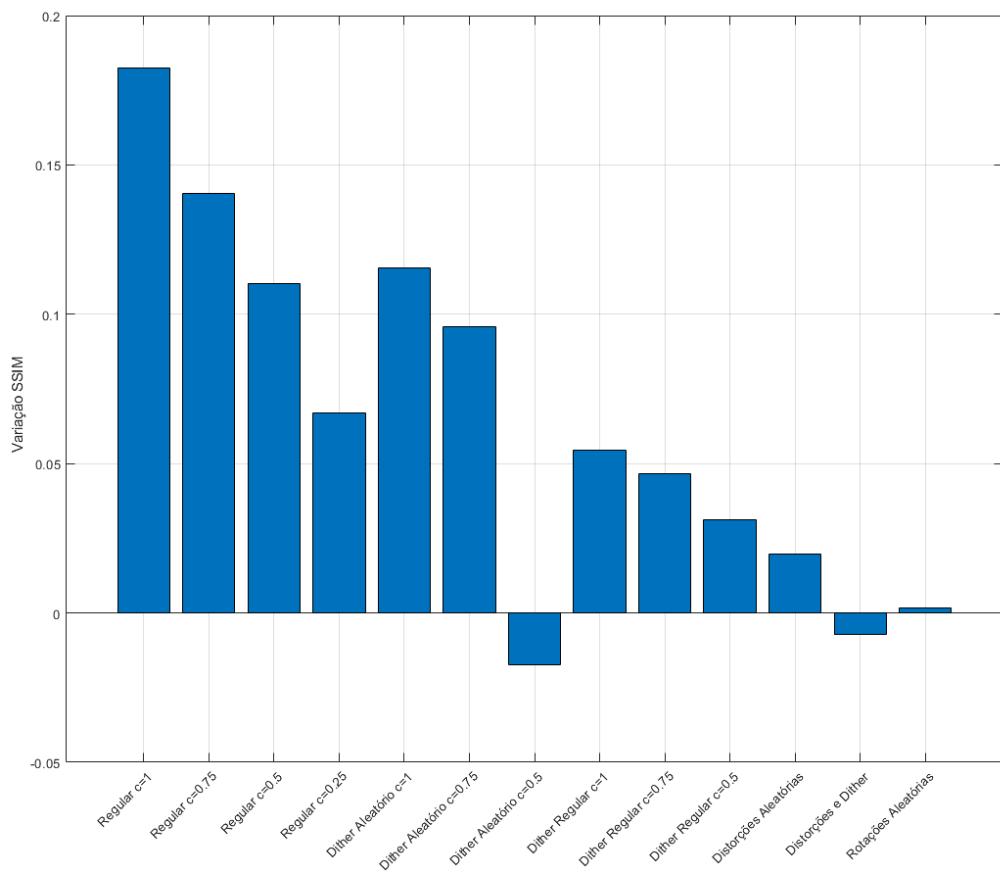


Figura 46: Comparação da variação dos valores de SSIM para todos os testes

5 CONCLUSÃO

Através da implementação do algoritmo de remoção de marcas d'água, foi demonstrado que as marcas d'água comumente utilizadas para a proteção de direitos autorais de imagens e fotografias comerciais, por grandes empresas como a *Adobe Stock*, são de fato pouco robustas frente a um ataque de remoção que se aproveita da disponibilidade de um imenso número de imagens identicamente marcadas na internet, por meio das quais foi possível a decomposição das imagens marcadas em imagens reconstruídas limpas e em sua marca d'água estimada, de modo que o resultado final, para estas marcas tradicionais, possui defeitos dificilmente perceptíveis, mesmo para marcas que foram aplicadas com diferentes níveis de opacidade, nas quais o algoritmo foi capaz de melhorar significativamente suas qualidades, obtendo ganhos entre 6 e 7 dB nas métricas de qualidade SNR e PSNR. Ou seja, este trabalho alcançou o objetivo de demonstrar esta vulnerabilidade.

Os métodos analisados de aumento da robustez das marcas d'água, por sua vez, demonstraram que mesmo processos relativamente simples, como a *ditherização* em padrão quadriculado regular, e a aplicação de distorções aleatórias foram capazes de melhorar as marcas como também era o objetivo deste trabalho. Após a tentativa de remoção destas marcas melhoradas, resíduos e imperfeições são imediatamente visíveis na imagem reconstruída pelo algoritmo, algo que não era presente em tamanha intensidade para as marcas tradicionalmente aplicadas. Além da robustez ser avaliada de forma visual e qualitativa, esta melhora também foi validada objetivamente pelos resultados das métricas de qualidade calculadas. Imagens que foram marcadas pela marca d'água que foi tanto distorcida como *ditherizada* apresentaram um ganho na métrica SNR de apenas 0,49 dB após a aplicação do algoritmo de remoção, uma melhora quase imperceptível, indicando que a aplicação destas técnicas em conjunto foi capaz de inviabilizar o uso deste algoritmo para a remoção destas marcas, e portanto melhor garante a proteção dos direitos autorais destas imagens. Na literatura atual da área não foram encontrados bons exemplos de marcas melhoradas com o objetivo de resistirem ataques como o descrito por [Dekel et al. \(2017\)](#), em “*On the Effectiveness of Visible Watermarks*”, dificultando a comparação dos resultados obtidos neste trabalho com outras marcas d'água propostas na literatura.

Deve-se notar que sempre existirá a possibilidade do aperfeiçoamento do algoritmo de remoção de marcas adicionando considerações prévias de possíveis variações na marca, não só por meio das técnicas descritas e aplicadas neste trabalho, mas também de outras possíveis transformações geométricas na marca, a exemplo das marcas aleatoriamente rotacionadas, as quais o algoritmo de remoção não está preparado para lidar, e portanto é ineficaz em sua remoção. Apesar desta possibilidade, uma marca d'água de maior robustez representará uma maior barreira de esforço necessário para a sua remoção, e assim diminui

o risco da aplicação mal-intencionada de algoritmos que busquem remover marcas de forma sistemática e generalizada, uma vez que estes algoritmos, para apresentarem resultados satisfatórios em marcas robustas, necessariamente serão mais complexos e lentos, tornando-os menos viáveis para aplicação em massa.

Verificou-se então que ainda há muitas possibilidades para a melhora de marcas d'água atualmente utilizadas, que são vulneráveis a ataques de remoção que podem ser inteiramente automatizados, buscando de maneira autônoma imagens marcadas na internet e removendo suas marcas. Entre estas possibilidades de melhora, a simples distorção aleatória ou a rotação da marca antes de ser aplicada já representa um grande passo no aumento da robustez destas marcas. Além das marcas melhoradas analisadas neste trabalho, também podem ser estudadas em trabalhos futuros marcas d'água digitais não visíveis que funcionem como indicadores ocultos dos direitos autorais da imagem, bem como marcas d'água aplicadas nos diferentes domínios de uma imagem, como o da frequência.

REFERÊNCIAS

- Biermann, C. J. (1996). 7 - paper and its properties. In Biermann, C. J., editor, *Handbook of Pulping and Papermaking (Second Edition)*, pages 158 – 189. Academic Press, San Diego, second edition edition.
- Burrus, C. S. (2012). Iterative reweighted least squares. *OpenStax CNX*. <http://cnx.org/contents/92b90377-2b34-49e4-b26f-7fe572db78a1@12>.
- Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8:679 – 698.
- Courant, R. and Hilbert, D. (1989). *Methods of Mathematical Physics, Vol. 1*. Wiley-VCH.
- Dekel, T., Rubinstein, M., Liu, C., and Freeman, W. T. (2017). On the effectiveness of visible watermarks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6864–6872.
- Fred Mintzer, Gordon W. Braudaway, M. M. Y. (1997). Effective and ineffective digital watermarks. In *Proceedings of International Conference on Image Processing*. IEEE.
- Frosh, P. (2001). Inside the image factory: stock photography and cultural production. *Media, Culture & Society*, 23(5):625–646.
- Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Kankanhalli, M. S., Rajmohan, and Ramakrishnan, K. R. (1999). Adaptive visible watermarking of images. In *Proceedings IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 568–573 vol.1.
- Lagae, A., Lefebvre, S., Cook, R., DeRose, T., Drettakis, G., Ebert, D., Lewis, J., Perlin, K., and Zwicker, M. (2010). A survey of procedural noise functions. *Computer Graphics Forum*, 29(8):2579–2600.
- Levin, A., Lischinski, D., and Weiss, Y. (2008). A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242.
- Minerva M. Yeung, Frederick C. Mintzer, G. W. B. A. R. (1997). Digital watermarking for high-quality imaging. In *Proceedings of First Signal Processing Society Workshop on Multimedia Signal Processing*. IEEE.

- Mohanty, S. P., Ramakrishnan, K. R., and Kankanhalli, M. S. (2000). A dct domain visible watermarking technique for images. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, volume 2, pages 1029–1032 vol.2.
- Pérez, P., Gangnet, M., and Blake, A. (2003). Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318.
- Perlin, K. (1985). An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296.
- Perlin, K. (2002). Improving noise. *ACM Trans. Graph.*, 21(3):681–682.
- Potdar, V. M., Han, S., and Chang, E. (2005). A survey of digital image watermarking techniques. In *INDIN '05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, pages 709–716.
- Sen, O., Chemudugunta, C., and Gopi, M. (2003). Silhouette-opaque transparency rendering. In *Computer Graphics and Imaging*.
- Shen, J., Jin, X., and Zhou, C. (2005). Gradient based image completion by solving poisson equation. In Ho, Y.-S. and Kim, H. J., editors, *Advances in Multimedia Information Processing - PCM 2005*, pages 257–268, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Sindhu, V. (2013). A review on image halftone processing.
- Veryovka, O. and Buchanan, J. W. (1999). Halftoning with image-based dither screens. In *Graphics Interface*.
- Zhou Wang, Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Apêndice:
Exemplos adicionais do desempenho do algoritmo de
remoção de marcas d'água



(a) Imagem Marcada Original



(b) Imagem Processada



(c) Imagem Marcada Original



(d) Imagem Processada



(e) Imagem Marcada Original



(f) Imagem Processada



(a) Imagem Marcada Original



(b) Imagem Processada



(c) Imagem Marcada Original



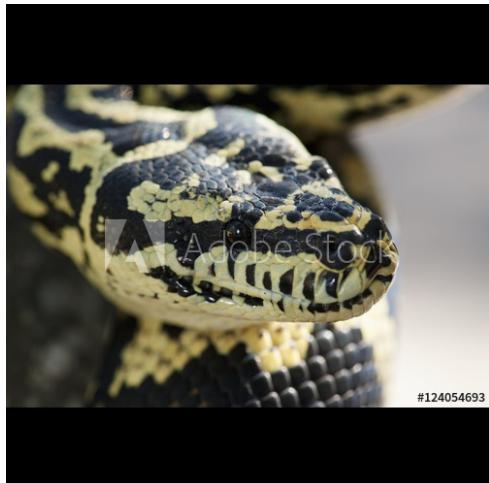
(d) Imagem Processada



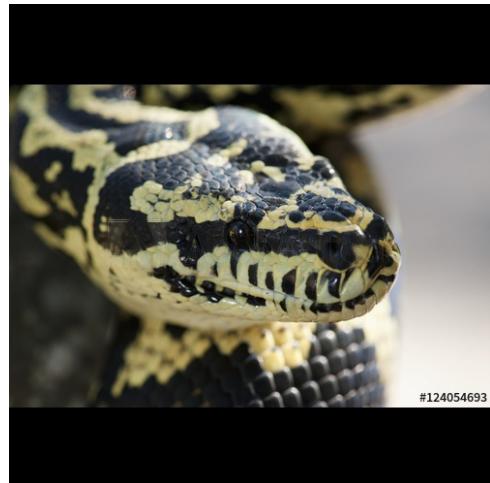
(e) Imagem Marcada Original



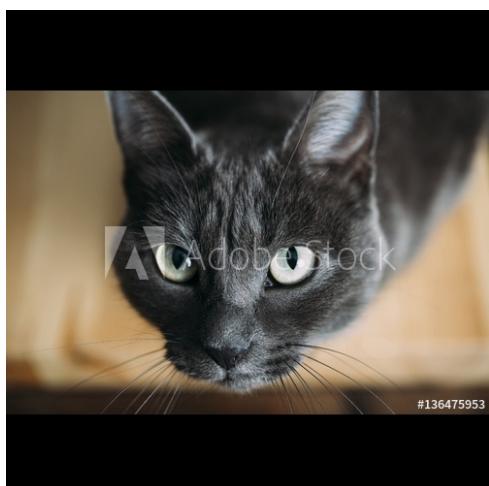
(f) Imagem Processada



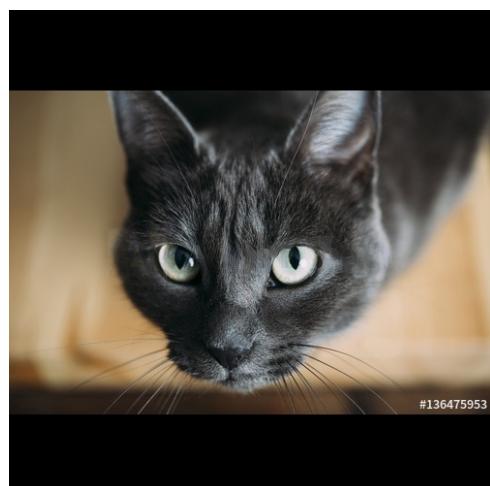
(a) Imagem Marcada Original



(b) Imagem Processada



(c) Imagem Marcada Original



(d) Imagem Processada



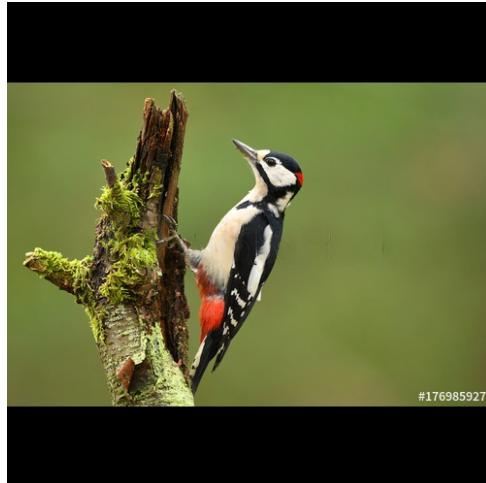
(e) Imagem Marcada Original



(f) Imagem Processada



(a) Imagem Marcada Original



(b) Imagem Processada



(c) Imagem Marcada Original



(d) Imagem Processada



(e) Imagem Marcada Original



(f) Imagem Processada



(a) Imagem Marcada Original (Ampliação)



(b) Imagem Processada (Ampliação)



(c) Imagem Marcada Original (Ampliação)



(d) Imagem Processada (Ampliação)



(e) Imagem Marcada Original (Ampliação)



(f) Imagem Processada (Ampliação)



(a) Imagem Marcada Original (Ampliação)



(b) Imagem Processada (Ampliação)



(c) Imagem Marcada Original (Ampliação)



(d) Imagem Processada (Ampliação)



(e) Imagem Marcada Original (Ampliação)



(f) Imagem Processada (Ampliação)



(a) Imagem Marcada Original (Ampliação)



(b) Imagem Processada (Ampliação)



(c) Imagem Marcada Original (Ampliação)



(d) Imagem Processada (Ampliação)



(e) Imagem Marcada Original (Ampliação)



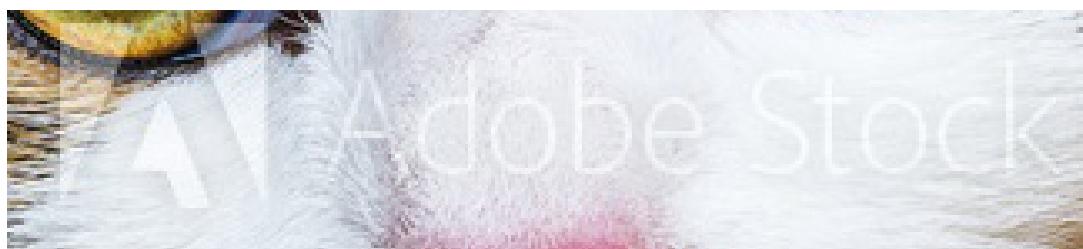
(f) Imagem Processada (Ampliação)



(a) Imagem Marcada Original (Ampliação)



(b) Imagem Processada (Ampliação)



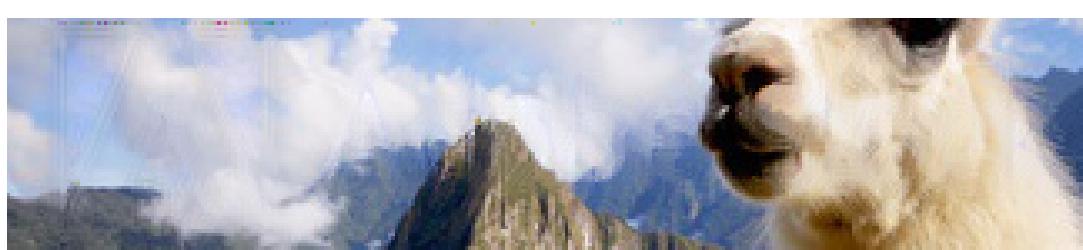
(c) Imagem Marcada Original (Ampliação)



(d) Imagem Processada (Ampliação)



(e) Imagem Marcada Original (Ampliação)



(f) Imagem Processada (Ampliação)