Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Método semi-cego robusto de marca-d'água em imagens digitais baseado na Transformada Discreta de Fourier Quaternária

Lucas Jr. Ribas

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Bruno L. Macchiavello

Brasília
2025

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Método semi-cego robusto de marca-d'água em imagens digitais baseado na Transformada Discreta de Fourier Quaternária

Lucas Jr. Ribas

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

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

Prof. Dr. João J. C. Gondim
ENE/UnB

Prof.a Dr.a Edna Dias Canedo
CIC/UnB

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 3 de Julho de 2025

Dedicatória

Dedico este trabalho a toda minha família e à minha namorada, pois sem o apoio incondicional de todos essa conquista não seria possível.

Agradecimentos

Agradeço a Deus por me fortalecer ao longo desta jornada, agradeço aos meus professores pelos conhecimentos transmitidos, agradeço ao meu orientador Bruno Macchiavello, por me instruir ao longo deste trabalho e pela oportunidade de crescimento acadêmico, e por fim, agradeço por conseguir concluir mais esta etapa da minha vida.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Este trabalho apresenta um método semi-cego e robusto de marca-d'água para imagens digitais coloridas, que busca aprimorar a resistência contra distorções geométricas e aumentar a imperceptibilidade. O método proposto utiliza a Transformada Discreta de Fourier Quaternária - Quaternion Discrete Fourier Transform (QDFT) para incorporar as informações no domínio da frequência, onde antes da inserção, os bits da marca-d'água são embaralhados pela Transformada de Arnold por Bloco Iterativo - Iterative Block Arnold Transform (IBAT) com o intuito de aumentar a segurança da marca-d'água. Para garantir uma alta robustez contra ataques geométricos, o algoritmo de Transformada de Características Invariante à Escala - Scale Invariant Feature Transform (SIFT) é utilizado para extrair pontos de interesse que permitem a correção de distorções geométricas antes da extração da marca-d'água. Os resultados experimentais demonstram que o método alcança uma alta imperceptibilidade após o processo de incorporação, e uma alta robustez na extração após diversos tipos de ataques, como compressão JPEG, escalonamento, rotação escalonada e filtragem. Portanto, a combinação desses passos e técnicas resultam em um sistema de marca-d'água robusto e imperceptível, se tornando seguro para aplicações atuais de proteção de direitos autorais.

Palavras-chave: imagens digitais, imagens coloridas, marca-d'água, QDFT, quatérnios, transformada de arnold, pontos de interesse, SIFT, criptografia, chave mestra

Abstract

This study presents a semi-blind and robust watermarking method for color digital images, which seeks to improve resistance against geometric distortions and increase imperceptibility. The proposed method uses Quaternion Discrete Fourier Transform (QDFT) to embed information in the frequency domain, where before insertion, the watermark bits are scrambled by Iterative Block Arnold Transform (IBAT) in order to increase the security of the watermark. To ensure high robustness against geometric attacks, the Scale Invariant Feature Transform (SIFT) algorithm is used to extract keypoints that allow the correction of geometric distortions before watermark extraction. The experimental results demonstrate that the method achieves high imperceptibility after the embedding process and high robustness in extraction after various types of attacks, such as JPEG compression, scaling, scaled rotation, and filtering. Therefore, the combination of these steps and techniques results in a robust and imperceptible watermarking system, making it safe for current copyright protection applications.

Keywords: digital images, color images, watermark, QDFT, quaternions, arnold transform, arnold's cat map, keypoints, SIFT, encryption, master key

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AES Padrão de Criptografia Avançada - Advanced Encryption Standard.

AI Artificial Intelligence - Inteligência Artificial.

AT Transformada de Arnold - Arnold Transform.

BER Taxa de Erro de Bit - Bit Error Rate.

CDF Função de Distribuição Acumulada - Cumulative distribution function.

CSPRNG Gerador de Números Pseudoaleatórios Criptograficamente Seguros - Cryptographically Secure Pseudorandom Number Generator.

CTR Modo Contador - Counter Mode.

CTR-DRBG Gerador de Bits Aleatórios Determinísticos em Modo Contador - Counter Mode Deterministic Random Bit Generator.

dB Decibel.

DFT Transformada Discreta de Fourier - Discrete Fourier Transform.

DM Modulação por Dithering - Dither Modulation.

DoG Diferença de Gaussianas - Difference-of-Gaussian.

IBAT Transformada de Arnold por Bloco Iterativo - Iterative Block Arnold Transform.

IIBAT Transformada Inversa de Arnold por Bloco Iterativo - Inverse Iterative Block Arnold Transform.

Inf Valor representativo do infinito - Varia entre linguagens de programação.

IULPM Mapeamento Log-Polar Uniforme Aprimorado - Improved Uniform Log-polar Mapping.

IV Vetor de Inicialização - Initialization Vector.

JPEG Joint Photographic Experts Group.

KEK Chave de Criptografia de Chaves - Key Encryption Key.

LBM Modulação de bits *'baixos'* - Low-bit(s) Modulation.

LPM Mapeamento Log-Polar - Log-polar Mapping.

MSE Erro Quadrático Médio - Mean Square Error.

NC Correlação Normalizada - Normalized Correlation.

NIST Instituto Nacional de Padrões e Tecnologia - National Institute of Standards and Technology.

PRNG Gerador de Números Pseudoaleatórios - Pseudorandom Number Generator.

PSNR Relação Sinal-ruído de Pico - Peak Signal-to-Noise Ratio.

QDFT Transformada Discreta de Fourier Quaternária - Quaternion Discrete Fourier Transform.

QIM Modulação por Índice de Quantização - Quantization Index Modulation.

RANSAC Consenso por Amostragem Aleatória - Random Sample Consensus.

RGB Vermelho, Verde, e Azul - Red, Green, Blue.

RNG Gerador de Números Aleatórios - Random Number Generator.

SHA-256 Algoritmo de Hash Seguro 256 bits - Secure Hash Algorithm 256 bits.

SHA-512 Algoritmo de Hash Seguro 512 bits - Secure Hash Algorithm 512 bits.

SIFT Transformada de Características Invariante à Escala - Scale Invariant Feature Transform.

SSIM Índice de Similaridade Estrutural - Structural Similarity Index Measure.

ULPM Mapeamento Log-Polar Uniforme - Uniform Log-Polar Mapping.

Lista de Símbolos

α **Nível de significância** Probabilidade máxima de rejeitar a hipótese nula H_0 quando ela é verdadeira (falso positivo).

$\boldsymbol{\Delta}$ **Delta** Passo de quantização. Desempenha a mesma função tanto no Dither Modulation (DM) quanto no Quantization Index Modulation (QIM).

θ Angulo geral teta.

Nb Variável definida para representar o tamanho de um bloco ao utilizar a IBAT.

N Variável definida para representar os lados de uma imagem quadrada ao utilizar a AT.

h Altura de uma imagem, ou número de linhas da matriz de pixels..

w Largura de uma imagem, ou número de colunas da matriz de pixels..

x Coordenada ao longo do eixo x (linhas).

y Coordenada ao longo do eixo y (colunas).

imgs1 Conjunto 1 de imagens de testes (32 imagens 512 $\times$ 512).

imgs2 Conjunto 2 de imagens de testes (1000 imagens 384 $\times$ 256 e 256 $\times$ 384).

imgs3 Conjunto 3 de imagens de testes (256 imagens 512 $\times$ 512).

imgs4 Conjunto 4 de imagens de testes (256 imagens de alta resolução).

total_imgs Conjunto total de imagens de testes. Engloba todos os conjuntos de teste, imgs1, imgs2, imgs3 e imgs4.

V1 Marca-d'água de testes - Versão 1, com resolução base 612 $\times$ 612 (Figura 5.2, a).

V2 Marca-d'água de testes - Versão 2, com resolução base 250 $\times$ 250 (Figura 5.2, b).

Lista de Algoritmos

Capítulo 1

Introdução

Com o desenvolvimento da tecnologia, as questões de segurança da informação tornaram-se cada vez mais proeminentes e a conscientização das pessoas sobre a proteção da privacidade aumentou gradualmente [5]. O crescente uso de plataformas virtuais (como as redes sociais), aumenta a disseminação de imagens digitais, e com isso, é preciso atentar-se para novos riscos em relação a seus direitos autorais, pois pessoas com intenções maliciosas e ilegais podem falsificar, adulterar ou piratear tais imagens para seu próprio benefício e lucro [6]. Nesse cenário, há um destaque para o uso da Artificial Intelligence - Inteligência Artificial (AI), que apresenta vantagens inegáveis, porém, também traz consigo uma série de riscos e desafios que devem ser considerados e controlados [7]. As plataformas digitais se tornaram um cenário oportuno para a disseminação de fotografias geradas por AI, sendo que estas, em diversos casos, podem ser maliciosas e promover fraudes, desinformação e até mesmo causar a manipulação da reputação de um indivíduo [8]. Devido ao fato da geração dessas imagens se apresentar cada vez mais condizente com a realidade, torna-se urgente o investimento em soluções capazes de provar que tais imagens são falsas. Pensando nisso, dentre os métodos que podem auxiliar nessa questão, destacam-se os algoritmos para inserção de marcas-d'água.

Uma marca-d'água pode incorporar informações de direitos autorais em dados multimídia, por meio de certos algoritmos, sendo que tais informações incorporadas geralmente não são visíveis ou perceptíveis. A tecnologia de marca-d'água digital tem diversas aplicações: proteção, certificação, distribuição, anti-falsificação da mídia digital e etiqueta das informações do usuário [9]. O ideal é que uma marca-d'água sobreviva a possíveis ataques no arquivo marcado.

Visando a resolução das deficiências da tecnologia tradicional de segurança da informação, cada vez mais, os pesquisadores têm começado a estudar os procedimentos relacionados às marcas-d'água, especialmente as digitais. Assim sendo, essa temática tornou-se uma área de estudo muito importante [9]. Em seu artigo, Zhang et al [6] fazem

uma detalhada análise sobre os principais atributos, classificações e modelos de sistemas de marca-d'água em imagem digitais, destacando seus principais problemas. Por outro lado, em uma vertente mais prática, o trabalho de Bhatti et al [10] se propõe a implementar um sistema de marca-d'água em imagens coloridas, visando melhorar a robustez contra vários ataques. Já Qingtang et al [11], tem como objetivo implementar um método robusto adaptável de marca-d'água cega, para proteger os direitos autorais de imagens digitais coloridas e resistir a ataques geométricos. Além disso, Ouyang et al [3] também propõem um método de marca-d'água cega e robusta para imagens coloridas.

Neste trabalho, pensando em melhorar um algoritmo de marca-d'água já existente, foi proposto um método que envolve a utilização da QDFT, do SIFT e da Transformada de Arnold - Arnold Transform (AT). Inicialmente, no processo de incorporação, a marca-d'água passa por um processo de embaralhamento utilizando a IBAT (uma melhoria da AT), retornando seus bits embaralhados. Após isso, a imagem hospedeira é transformada para o domínio de frequência utilizando a QDFT, onde seus coeficientes são modificados para representar os bits da marca-d'água. Em seguida, através do SIFT, os pontos de interesse são extraídos e armazenados junto aos parâmetros, para que posteriormente possam ser utilizados para alinhar geometricamente a imagem para a extração. Além disso, todos os parâmetros do processo de incorporação são criptografados pelo Padrão de Criptografia Avançada - Advanced Encryption Standard (AES) utilizando uma chave mestra, onde esta é protegida por outra chave derivada de uma senha fornecida pelo usuário ao Argon2. Como resultado deste método proposto, e após diversos testes realizados, foi possível observar uma alta imperceptibilidade e uma alta robustez do sistema de marca-d'água frente a diversos tipos de ataque.

1.1 Apresentação do Problema

O método de marca-d'água proposto por Ouyang et al [3], adotado como referência neste trabalho, se destaca por utilizar de forma eficiente a QDFT ao processar a imagem colorida de forma holística, processando os três canais de cor de maneira correlacionada e simultânea. Essa abordagem resulta em uma menor distorção na imagem marcada após o processo de incorporação dos bits de informação. No entanto, o método apresenta uma perda de robustez frente a determinados tipos de transformações geométricas comuns no ciclo de vida de uma imagem digital, como translações ou rotações escalonadas, o que pode comprometer de forma significativa a extração da marca-d'água. Além disso, embora o método obtenha uma menor distorção ao utilizar a QDFT de forma holística, a estratégia adotada de incorporação dos bits compromete a imperceptibilidade da marca-d'água, resultando em artefatos visuais perceptíveis na imagem marcada. Dessa forma, o

método proposto neste trabalho visa resolver essas limitações, mantendo as vantagens do método de referência ao tratar a imagem de forma holística.

1.2 Objetivo

O objetivo principal deste trabalho é implementar melhorias de imperceptibilidade e robustez a ataques geométricos em um algoritmo de marca-d'água digital, que é apresentado no artigo denominado *"A Blind Robust Color Image Watermarking Method Using Quaternion Fourier Transform"* [3].

1.3 Objetivos secundários

- Analisar e implementar o método de referência [3] para estabelecer uma base de comparação.
- Incorporar métodos criptográficos ao sistema.
- Utilizar redundância.
- Manter a estratégia holística ao aplicar a QDFT.
- Desenvolver uma nova estratégia de incorporação dos bits.
- Acrescentar o método de Modulação por Dithering - Dither Modulation (DM).
- Acrescentar o método SIFT.
- Aprimorar a AT.

1.4 Organização da Monografia

Este trabalho é organizado da seguinte forma: no Capítulo 2, será abordada a fundamentação teórica, que apresenta os conceitos e definições de termos essenciais para o pleno entendimento deste trabalho; em seguida, o Capítulo 3 apresenta uma revisão bibliográfica, onde são mostrados artigos que embasaram o método proposto neste trabalho; o Capítulo 4, traz a metodologia utilizada; já o Capítulo 5, expõe os resultados encontrados; e por fim, o Capítulo 6 apresenta a conclusão, com as considerações finais e proposta de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

2.1 Segurança da Informação

2.1.1 Conceitos Básicos da Criptografia

A criptografia é uma ferramenta fundamental da segurança da informação baseada em algoritmos e chaves, que é aplicada em redes de comunicação através de protocolos de segurança para proteger os dados em trânsito. Existem muitos métodos de criptografia de informações, dentre eles, a criptografia de chave simétrica e a criptografia de chave assimétrica [12]. Os algoritmos simétricos utilizam uma única chave compartilhada entre duas partes. Já os algoritmos de chave pública, também conhecidos como algoritmos assimétricos, utilizam duas chaves: uma chave privada, conhecida apenas por uma das partes, e uma chave pública, disponível para as demais partes [13].

Os princípios da criptografia são: confidencialidade, integridade, autenticidade, e não repúdio. A confidencialidade garante que as informações sejam mantidas em segredo de todos, exceto das partes autorizadas; já a integridade, permite que as mensagens não sejam modificadas em trânsito; a autenticidade, atesta que o remetente de uma mensagem é autêntico; por fim, o não-repúdio assegura que o remetente de uma mensagem não consiga negar a criação da mensagem [14].

2.1.2 Geradores de Números Aleatórios Criptograficamente Seguros

Os Geradores de Números Aleatórios - Random Number Generators (RNGs) são classificados em três categorias. Os verdadeiros RNGs que produzem dados aleatórios com base em fontes físicas imprevisíveis, como o ruído de resistores elétricos. Os Geradores de Números Pseudoaleatórios - Pseudorandom Number Generators (PRNGs) que utilizam

um método determinístico para gerar números aleatórios, utilizando uma semente (*'seed'*) verdadeiramente aleatória para inicializar o algoritmo. E os PRNGs Criptograficamente Seguros (CSPRNGs), que são PRNGs especificamente projetados para serem resistentes a determinados ataques criptográficos, onde, mesmo que todo ou parte do estado interno do CSPRNG seja revelado, não deve ser possível deduzir os números gerados anteriormente [15].

2.1.3 AES - Advanced Encryption Standard

O AES é um algoritmo de criptografia simétrico que cifra os dados em blocos. No caso específico do AES, os dados são processados em blocos de tamanho fixo de 128 bits, tanto durante o processo de encriptação quanto na decriptação. O AES suporta chaves de 128, 192 e 256 bits, cada uma definindo o comportamento do algoritmo que é executado em rodadas. Para a chave de 128 bits, a quantidade de rodadas executadas é 10, para a chave de 192 bits são 12 rodadas, e para a chave de 256 bits o total de rodadas é 14. A divisão dos dados não varia de acordo com a chave, ou seja, os dados serão sempre divididos em blocos de 128 bits, mesmo se a chave fornecida for de 128, 192 ou 256 bits. Cada bloco de 128 bits é organizado em uma matriz de estado, com dimensões [4x4], onde cada elemento corresponde a 1 byte, ou 8 bits. A chave secreta é estruturada da mesma forma em uma matriz semelhante, porém as dimensões dessa matriz variam conforme o tamanho da chave passada para o AES [16].

Na primeira etapa, a matriz de estado que contém o bloco de dados a ser encriptado, é combinada com a chave principal por meio da $AddRoundKey$, que utiliza uma operação XOR para introduzir a primeira camada de confusão. Em cada rodada, são aplicadas quatro transformações: A $SubBytes$ que substitui cada byte da matriz de estado usando uma tabela não linear(S-Box), quebrando padrões estatísticos do texto original. A $ShiftRows$ que desloca as linhas da matriz de estado de forma cíclica, promovendo difusão. A $MixColumns$ que mistura as colunas da matriz de estado por meio de multiplicação matricial em um corpo finito, dispersando ainda mais os bits. E a $AddRoundKey$, que neste caso, utiliza uma sub-chave derivada da chave principal para aplicar a operação XOR, sendo que em cada rodada a sub-chave é única. Na última rodada, $MixColumns$ não é executado para simplificar a decriptação. A decriptação é o processo inverso de cada transformação utilizada na encriptação [16].

CTR - Counter Mode

O Modo Contador - Counter Mode (CTR), é um dos modos de operação do AES, funcionando como um sistema de criptografia de fluxo, ou $Stream\ Cipher$. Ao invés de operar

diretamente sobre os blocos dos dados de entrada, o modo CTR faz com que o AES cifre um contador, composto por um valor único, e um valor incremental. Esse contador é cifrado pelo AES, gerando um fluxo de chave, ou *keystream*, do mesmo tamanho do bloco, que é combinado com os dados de entrada por meio de uma operação XOR, resultando nos dados criptografados. O contador deve ser único para cada bloco e geralmente inclui um Vetor de Inicialização - Initialization Vector (IV) fixo, enquanto o restante é um valor incremental. Mesmo para mensagens diferentes cifradas com a mesma chave, o contador não deve se repetir. Isso permite criptografar mensagens de qualquer tamanho de forma eficiente e segura [17].

2.1.4 CTR-DRBG - Counter Mode Deterministic Random Bit Generator

O Gerador de Bits Aleatórios Determinísticos em Modo Contador - Counter Mode Deterministic Random Bit Generator (CTR-DRBG), como o próprio nome já diz, é um gerador determinístico de bits pseudoaleatórios baseado no CTR, sendo padronizado pelo Instituto Nacional de Padrões e Tecnologia - National Institute of Standards and Technology (NIST) detalhado no documento [18]. Ele utiliza, geralmente, algoritmos de cifra de bloco, como o AES no modo CTR para criptografar um contador incremental, que gera uma saída pseudoaleatória utilizada como fonte de números aleatórios [19]. O CTR-DRBG é um exemplo particular de um CSPRNG, atendendo aos requisitos de segurança criptográfica. Os dados externos, utilizados no AES em modo CTR, são uma forma de renovar a entropia do processo, sendo aplicados como re-sementeamento, que consiste em atualizar o estado interno do gerador com nova entropia, ou seja, dados aleatórios frescos são inseridos de tempos em tempos para garantir que a saída continue imprevisível, mesmo que o estado atual seja parcial ou totalmente comprometido.

2.1.5 SHA - Secure Hash Algorithm

Um conceito fundamental na criptografia, denominado *Hashing*, desempenha um papel essencial na garantia de integridade, autenticidade e segurança das informações. A aplicação de uma função hash criptográfica envolve transformar os dados de entrada de tamanho arbitrário em dados de tamanho fixo, chamados de *hash*. Essas funções *hash* são projetadas para produzir um valor único e consistente para qualquer dado de entrada, garantindo que até mesmo uma pequena alteração na entrada resulte em um valor *hash* significativamente diferente [20]. É importante enfatizar que as funções *hash* criptográficas são de via única, ou seja, os dados que passam pelo processo de *hashing* não podem

ser revertidos ao conteúdo original, servindo apenas para criar uma espécie de impressão digital dos dados de entrada.

SHA-2

Publicada pelo NIST em 2002 [21], a família do SHA-2 (Secure Hash Algorithm 2) representa um avanço significativo no *hashing* criptográfico em relação aos seus antecessores, MD5 (Message-Digest Algorithm 5) e SHA-1 (Secure Hash Algorithm 1). O SHA-2 oferece um conjunto de funções hash com diferentes tamanhos de saída, atendendo a diversas exigências de segurança [20].

Baseado na construção Merkle-Damgård, uma abordagem comum para construir funções hash a partir de funções de compressão unidirecionais, o SHA-2 é composto pelos seguintes componentes: $HashFunction$, que é a função central que processa uma mensagem e gera um valor hash de tamanho fixo. Ela opera em blocos de dados de tamanho fixo e utiliza uma série de funções de compressão para atualizar iterativamente o estado. $MessageSchedulev$, que prepara a mensagem para a função de compressão, manipulando as palavras da mensagem de uma maneira específica. Essa etapa intensifica o efeito avalanche, garantindo que uma pequena alteração na mensagem de entrada resulte em uma mudança significativa no valor *hash*. $CompressionFunction$, que processa um bloco de dados de tamanho fixo e o valor atual do estado para produzir um novo valor de estado. Essa operação é repetida para todos os blocos da mensagem, resultando no valor *hash* final. $PaddingScheme$, onde a mensagem é preenchida até atingir um comprimento específico divisível pelo tamanho do bloco. Isso garante um processamento consistente e facilita a adição de informações de comprimento para verificações de integridade da mensagem [20].

A principal diferença entre as variantes do SHA-2 está no número de palavras processadas no agendamento de mensagem e no número de rodadas da função de compressão. O Algoritmo de Hash Seguro 256 bits - Secure Hash Algorithm 256 bits (SHA-256) e o Algoritmo de Hash Seguro 512 bits - Secure Hash Algorithm 512 bits (SHA-512) diferenciam-se pelos seguintes aspectos: O SHA-256 processa blocos de mensagem de 512 bits utilizando palavras de 32 bits no agendamento de mensagem e emprega 64 rodadas na função de compressão, produzindo um valor *hash* de 256 bits. O SHA-512 processa blocos de mensagem de 1024 bits utilizando palavras de 64 bits no agendamento de mensagem e utiliza 80 rodadas na função de compressão, gerando um valor *hash* de 512 bits [20].

2.1.6 Argon2

O Argon2 tem como uma de suas aplicações principais o *hashing* de senhas, sendo projetado para ser *memory-hard*, o que exige uma grande quantidade de memória para obter o *hash* de forma eficiente. O objetivo principal é dificultar ataques de força bruta, fazendo com que o processo de quebra de senhas se torne extremamente lento e caro para atacantes que utilizam hardwares especializados. Além de manter um bom desempenho quando executado em CPUs modernas [22].

2.2 Imagens Digitais

Uma imagem pode ser definida como uma função bidimensional, $\boldsymbol{f(x, y)}$, onde $\boldsymbol{x}$ e $\boldsymbol{y}$ são coordenadas espaciais (de um plano), e a amplitude de $\boldsymbol{f}$ em qualquer par de coordenadas $\boldsymbol{(x, y)}$ é chamada de intensidade ou nível de cinza da imagem naquele ponto. Quando $\boldsymbol{x}$, $\boldsymbol{y}$ e os valores de amplitude de $\boldsymbol{f}$ são todos quantidades finitas e discretas, chamamos a imagem de imagem digital [23].

Note que uma imagem digital é composta por um número finito de elementos, cada um com uma localização e um valor específicos. Esses elementos são chamados de elementos de imagem, elementos de quadro, ou mais comummente usado, pixels.

O pixel representa o menor elemento em uma imagem digital, e o seu valor, como já mencionado, é representado pela amplitude de uma coordenada no plano, definindo a intensidade ou nível de cinza da imagem. O termo nível de cinza, refere-se a uma medida escalar de intensidade que varia do preto, passando pelos tons de cinza, até o branco [23].

Supondo que uma imagem digital tenha $\boldsymbol{h}$ linhas e $\boldsymbol{w}$ colunas, os valores das coordenadas na origem são $\boldsymbol{(x, y) = (0, 0)}$. Os valores seguintes ao longo da primeira linha da imagem são representados como $\boldsymbol{(x, y) = (0, 1)}$. E para os valores seguintes ao longo da primeira coluna temos $\boldsymbol{(x, y) = (1, 0)}$. Os valores das coordenadas $\boldsymbol{x}$ e $\boldsymbol{y}$ são sempre inteiros positivos.

De maneira simples, ao observar a Figura 2.1, é percebido que ao mover as coordenadas ao longo do eixo $\boldsymbol{x}$, a coluna é percorrida pulando de linha em linha, e quando as coordenadas são movidas ao longo do eixo $\boldsymbol{y}$, a linha é percorrida pulando de coluna em coluna.

Uma imagem digital em níveis de cinza tem pixels com valores inteiros igualmente espaçados no intervalo $\boldsymbol{[0, L-1]}$, onde $\boldsymbol{L = 2^k}$ é o número total de níveis de cinza. Uma imagem com $\boldsymbol{2^k}$ níveis é chamada de imagem de k-bits, por exemplo, uma imagem de 8 bits tem 256 níveis de cinza [23]. A extensão do intervalo total de valores, conhecido como faixa dinâmica, determina o quanto um pixel difere dos demais em relação à intensidade de cinza. Imagens que utilizam uma parte significativa dessa faixa apresentam alto contraste,

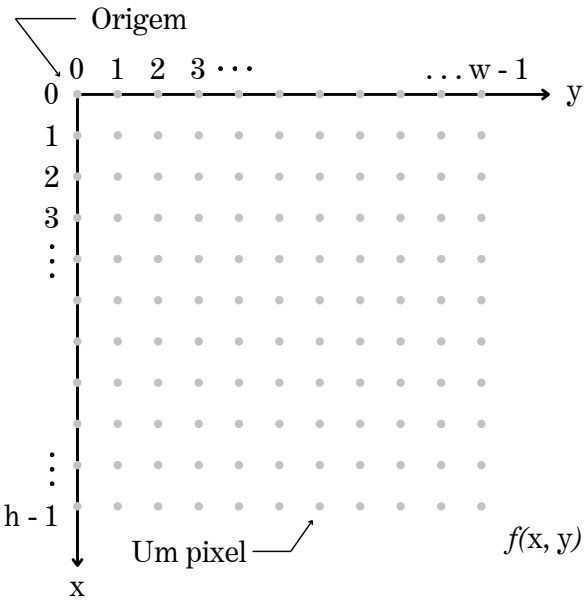


Figura 2.1: Convenção adotada para representação das coordenadas de uma imagem digital. Figura Modificada. (Fonte: [23]).

enquanto aquelas com uma faixa dinâmica reduzida tendem a parecer desbotadas. A quantidade de bits exigidos para armazenar uma imagem em nível de cinza é definida como $\boldsymbol{b = h \times w \times k}$, e no caso de imagens quadradas onde $\boldsymbol{h = w}$, a definição é simplificada para $\boldsymbol{b = N^2 k}$.

2.2.1 Modelo de Cor RGB

Os componentes primários no modelo de cores RGB são vermelho (R), verde (G) e azul (B). Esse modelo é bem estruturado, organizado em um sistema de coordenadas cartesianas, com as cores representadas como pontos em um cubo tridimensional completo [23]. Os vértices do cubo correspondem a diferentes cores: preto na origem $\boldsymbol{(R = G = B = 0)}$, branco no vértice mais distante $\boldsymbol{(R = G = B = 1)}$, as cores primárias (R, G, B) em três vértices e as cores secundárias (ciano, magenta e amarelo) nos outros três. A linha entre preto e branco, onde $\boldsymbol{(R = G = B)}$, não indica ausência de cor, mas sim a escala de cinza. Os pontos nessa linha têm valores no intervalo $\boldsymbol{(0 \leq R, G, B \leq 1)}$. Para simplificar, assume-se que os valores de cor estão normalizados, tornando o cubo na Figura Figura 2.2 um cubo unitário, porém, na prática os valores utilizados estão no intervalo de $\boldsymbol{[0, 2^{k-bits} - 1]}$.

Uma imagem RGB é composta por três planos de imagem (ou camadas), cada um representando uma cor primária. A combinação desses elementos produz a imagem resultante, que é exibida em um monitor RGB. O número de cores possíveis depende da

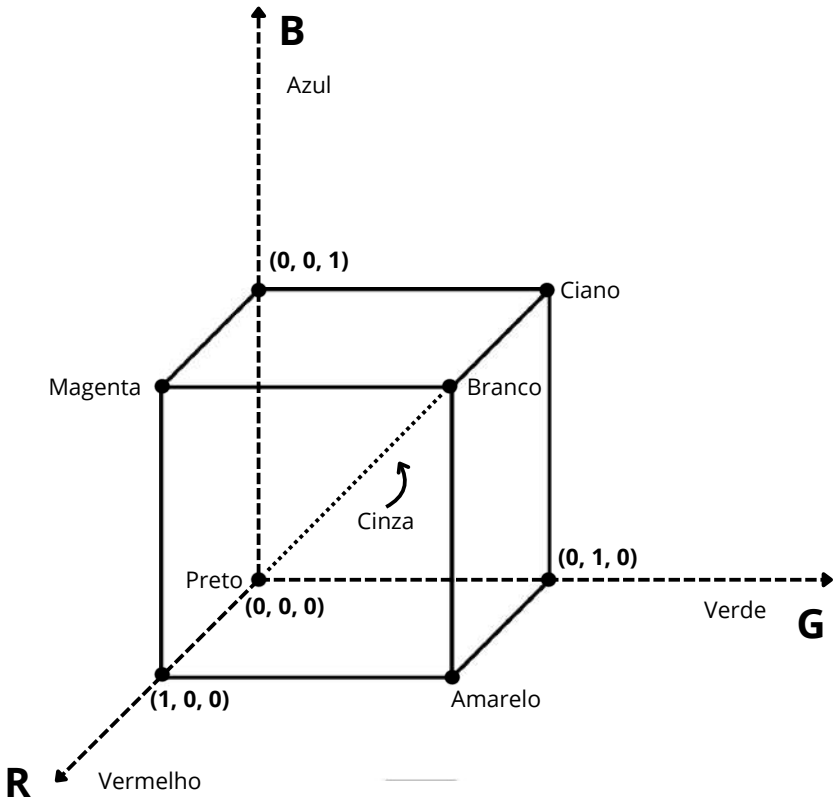


Figura 2.2: Esquema do cubo de cores RGB. Figura Modificada. (Fonte: [23]).

quantidade de bits usados para representar cada pixel. Por exemplo, em uma imagem RGB de 24 bits (8 bits por componente), cada pixel é definido por um triplo (R, G, B), o que significa que é possível representar $\mathbf{(2^8)^3 = 16.777.216}$ cores. Essas imagens são geralmente chamadas de imagens de cores reais ou imagem de cores completas [23].

2.3 Transformada de Arnold

As imagens possuem características como 'alta correlação entre pixels, grande capacidade e alta redundância'. Essas características dificultam a aplicação de métodos criptográficos tradicionais em imagens e tornam o processo mais lento. Atender aos requisitos de segurança de aplicações que envolvem imagens apresenta uma grande dificuldade na prática. As técnicas de criptografia de imagens, popularmente conhecidas como métodos de embaralhamento de imagens, são aplicadas para codificar a imagem em uma forma distorcida. Essas técnicas transformam a imagem em uma forma ininteligível que não pode ser percebida diretamente. Os métodos de embaralhamento, como a Transformada de Arnold, oferecem uma alternativa aos sistemas de segurança baseados em senhas [24].

Apesar de não ser uma criptografia teoricamente segura, a Transformada de Arnold, também conhecida como 'Arnold's cat mapping', oferece aplicações impressionantes em watermarking devido às suas propriedades únicas ao lidar com imagens. Ela embaralha a marca-d'água enquanto preserva seu padrão espacial. Isso pode ser facilmente integrado à imagem hospedeira sem comprometer o domínio visual. A Transformada de Arnold é prática e computacionalmente eficiente, especialmente no contexto de imagens, devido à sua simplicidade. Ela dificulta a identificação visual da marca-d'água, sendo inquestionável frente a ataques triviais.

Isso a torna adequada para situações em que o objetivo é dificultar a remoção ou detecção da marca-d'água, mas não para garantir confidencialidade absoluta.

A Transformada de Arnold foi proposta pelo matemático russo V. I. Arnold, sendo definida como um processo de reposicionamento que realinha os pixels de uma imagem digital. [25] conforme mencionado por Wu [26].

A Transformada de Arnold é definida como:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad mod \quad N, \tag{2.1}$$

onde $\boldsymbol{x}$ e $\boldsymbol{y}$ correspondem às coordenadas originais de um pixel, enquanto as coordenadas $\boldsymbol{x'}$ e $\boldsymbol{y'}$ representam a nova posição dos pixels na imagem criptografada, já a letra $\boldsymbol{N}$ equivale ao tamanho da imagem quadrada ($h = w = N$) [24]. Essa técnica muda a posição de dois pixels de maneira iterativa e, assim, uma imagem desordenada pode ser gerada [26].

Para recuperar a imagem original, a Transformada de Arnold Inversa é executada com o mesmo número de iterações que foi usado para gerar a imagem criptografada [24].

A Transformada de Arnold Inversa é definida como [26]:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad mod \quad N, \tag{2.2}$$

onde as coordenadas $\boldsymbol{x'}$ e $\boldsymbol{y'}$ da imagem criptografada são multiplicadas pela matriz de transformação inversa à matriz da equação Equação 2.1, retornando o pixel às suas coordenadas originais $\boldsymbol{x}$ e $\boldsymbol{y}$.

Uma definição alternativa da Transformada de Arnold Inversa é dada como [27]:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \left(\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} N \\ N \end{bmatrix} \right) \quad mod \quad N. \tag{2.3}$$

Embora o custo computacional aumente, a adição do deslocamento $[N;\ N]$ ajuda a prevenir problemas com valores negativos que ocorrem quando a transformação depende de intervalos válidos. Quando as variáveis são positivas, a operação de módulo N garante um valor dentro do intervalo válido [0, N-1], porém, quando elas são negativas o resultado irá depender da implementação. Em algumas linguagens de programação, como Python ou MATLAB, a operação $(\pm A \ mod\ N)$ é sempre positivo quando N é positivo, enquanto em outras, como C ou Java, o operação $(\pm A \ mod\ N)$ pode retornar valores negativos quando A for negativo.

Esse deslocamento $[N;\ N]$ resolve essa diferença, e de maneira geral, simplifica e torna a implementação mais robusta.

2.4 QDFT - Quaternion Discrete Fourier Transform

Em processamento de imagens, operações comuns como filtragem, compressão, ou análise de texturas, são mais eficientes e precisas no domínio da frequência, onde é analisado como as variações de intensidade dos pixels se distribuem em diferentes padrões espaciais. A Transformada Discreta de Fourier - Discrete Fourier Transform (DFT) converte uma função de coordenadas espaciais em uma função de frequências, o que revela componentes de baixa frequência (variações lentas de intensidade) e componentes de alta frequência (bordas, detalhes finos). Dessa forma, a DFT e sua inversa são definidas como [23]:

$$\begin{aligned} F(u,v) &= \sum_{x=0}^{h-1}\sum_{y=0}^{w-1} e^{-j2\pi\left(\frac{ux}{h}+\frac{vy}{w}\right)} f(x,y) \\ f(x,y) &= \tfrac{1}{\sqrt{h\times w}} \sum_{u=0}^{h-1}\sum_{v=0}^{w-1} e^{j2\pi\left(\frac{ux}{h}+\frac{vy}{w}\right)} F(u,v) \end{aligned} \tag{2.4}$$

onde $F(u,v)$ é a matriz complexa contendo os coeficientes de frequência, e j é a unidade imaginária $j = \sqrt{-1}$.

Na matriz complexa, o coeficiente $F(0,0)$ fica no canto superior esquerdo do espectro, de modo que as baixas frequências ficam concentradas nas bordas da matriz. Para facilitar a análise do sinal, é feito um $shift$ que centraliza as baixas frequências e afasta as altas frequências para a borda.

Para introduzir a QDFT, é necessário apresentar os números quaternários, que são uma extensão não comutativa dos números complexos para o espaço quaternário [28]. Dessa forma, um quatérnio é definido como:

$$q = a + bi + cj + dk \tag{2.5}$$

onde a, b, c e d são números reais, e i, j, e k são os operadores imaginários, que seguem a propriedade:

$$i^2 = j^2 = k^2 = ijk = -1. \tag{2.6}$$

Os quatérnios permitem estender a DFT para o domínio dos quatérnios, possibilitando tratar sinais multidimensionais. Dessa forma, a DFT pode ser generalizada para quatérnios puros, logo a QDFT e sua inversa são definidas como [3]:

$$\begin{aligned} F(u,v) &= \tfrac{1}{\sqrt{h\times w}} \sum_{x=0}^{h-1} \sum_{y=0}^{w-1} e^{-2\mu\pi\left(\frac{ux}{h}+\frac{vx}{w}\right)} f(x,y) \\ f(x,y) &= \tfrac{1}{\sqrt{h\times w}} \sum_{u=0}^{h-1} \sum_{v=0}^{w-1} e^{2\mu\pi\left(\frac{ux}{h}+\frac{vx}{w}\right)} F(u,v) \end{aligned} \tag{2.7}$$

onde $\mu^2 = -1$ é um quatérnio puro unitário. Assim, cada pixel (RGB) é tratado como um quatérnio puro permitindo o processamento holístico de imagens coloridas. Logo

$$f(x,y) = 0 + R(x,y)i + G(x,y)j + B(x,y)k. \tag{2.8}$$

2.5 Domínio Log-Polar

O Mapeamento Log-Polar - Log-polar Mapping (LPM) possui várias propriedades importantes que o tornam útil como uma estrutura de amostragem. O mapeamento de dois padrões regulares, como mostrado na Figura 2.3, resulta em padrões igualmente regulares no outro domínio [29].

A partir da Figura 2.3, os círculos concêntricos no plano da imagem tornam-se linhas verticais no plano cortical. Um único círculo mapeia para uma única linha vertical, pois o raio constante **r** em todos os ângulos θ do círculo fornece uma coordenada constante de $\boldsymbol{\rho}$ para todas as coordenadas $\boldsymbol{\theta_{cortical}}$, sendo $\boldsymbol{\rho}$ a valor da distância radial. Da mesma forma, uma imagem de linhas radiais que possuem ângulo constante, mas raio variável, resulta em um mapeamento de linhas horizontais.

Essas características de mapeamento são fundamentais para algumas propriedades, como a invariância à rotação e ao escalonamento. A rotação e o escalonamento resultam em deslocamentos ao longo dos eixos $\boldsymbol{\theta_{cortical}}$ e $\boldsymbol{\rho}$, respectivamente.

Para a invariância à rotação, observe que todas as orientações angulares possíveis de um ponto em um dado raio irão se mapear para a mesma linha vertical. Assim, se um

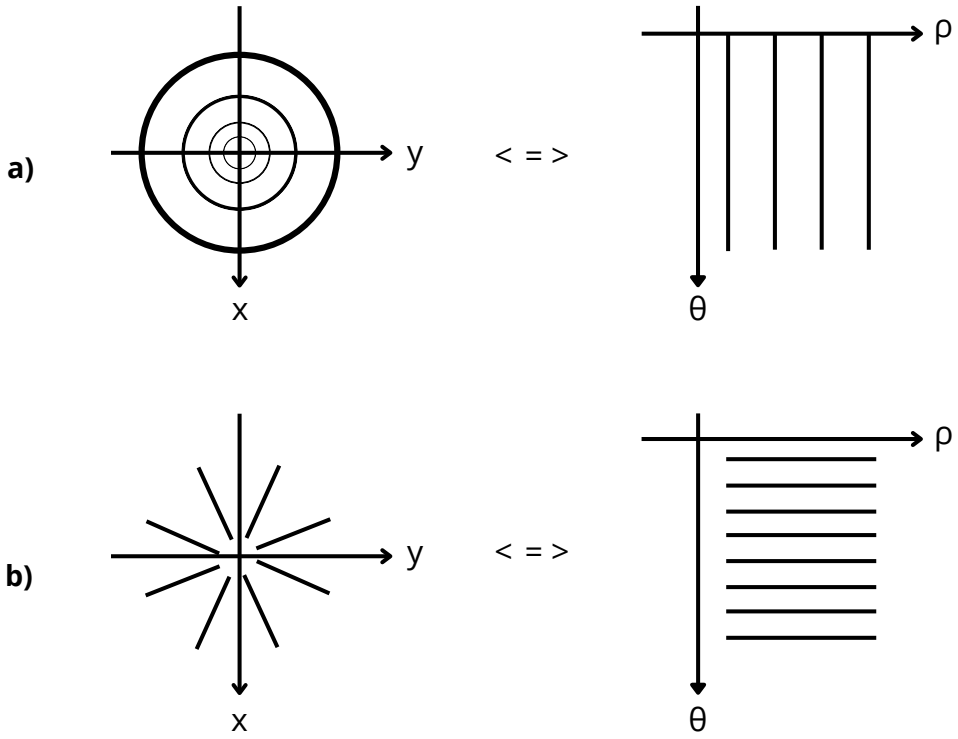


Figura 2.3: O LPM aplicado a padrões regulares. (a) Quando aplicado a círculos concêntricos no plano da imagem, são mapeados em linhas verticais no plano cortical. (b) Quando aplicado a linhas radiais no plano da imagem, são mapeados em linhas horizontais no plano cortical. Figura Modificada. (Fonte: [29]).

objeto for rotacionado ao redor da origem, isso resultará apenas em um deslocamento vertical da imagem mapeada. Esse mesmo resultado é válido para as linhas radiais. À medida que uma linha radial gira em torno da origem, seu mapeamento horizontal inteiro se move apenas verticalmente.

A invariância ao escalonamento é outra característica desse mapeamento log-polar. A partir da Figura 2.3, vemos que à medida que um ponto se afasta da origem ao longo de uma linha radial, seu mapeamento permanece na mesma linha horizontal, movendo-se da esquerda para a direita. Os mapeamentos dos círculos concêntricos permanecem linhas verticais e se movem apenas horizontalmente à medida que os círculos mudam de tamanho [29].

Propriedades

Um ponto $\boldsymbol{z(x,\ y)}$ nas coordenadas cartesianas $\boldsymbol{x}$ e $\boldsymbol{y}$ pode ser expresso como um número complexo:

$$z_{complexo} = x + jy \tag{2.9}$$

onde $\boldsymbol{x}$ representa a coordenada ao longo do eixo real, $\boldsymbol{j}$ é a unidade imaginária sendo igual à $j^2 = -1$, ou, $j = \sqrt{-1}$. Por fim, $\boldsymbol{y}$ indica a coordenada ao longo do eixo imaginário.

Ao utilizar $\boldsymbol{z}$ em sua forma cartesiana $\boldsymbol{(x,\ y)}$, ele representa um vetor que se estende da origem do plano cartesiano até o ponto $\boldsymbol{z(x,\ y)}$. O módulo de $\boldsymbol{|z|}$ corresponde à distância euclidiana do ponto $\boldsymbol{z(x,\ y)}$ até a origem. Essa distância radial $\boldsymbol{\rho}$ é calculada da seguinte forma:

$$\rho = |z| = \sqrt{x^2 + y^2} \tag{2.10}$$

Para expressar $\boldsymbol{z_{complexo}}$ na forma polar, além de $\boldsymbol{\rho}$ (distância radial), é necessário determinar o ângulo polar θ, que indica a direção do vetor em relação ao eixo real positivo. O cálculo do ângulo polar deve ser feito utilizando $\boldsymbol{atan2()}$ em vez de $\boldsymbol{arctg()}$, uma vez que $\boldsymbol{atan2()}$ considera o quadrante onde o ponto $\boldsymbol{z(x,\ y)}$ está localizado, evitando erros quando $\boldsymbol{x}$ é negativo ou nulo. Dessa forma, tem-se:

$$\theta = atan2(y, x). \tag{2.11}$$

Com a distância radial $\boldsymbol{\rho}$ e o ângulo polar θ, é possível representar $\boldsymbol{z_{complexo}}$ em sua forma polar:

$$z_{polar} = \rho * e^{j\theta} \tag{2.12}$$

onde: $e^{j\theta} = \cos\theta + j\sin\theta$.

O termo $\boldsymbol{\rho}$ representa a distância radial, ou seja, a distância do ponto $\boldsymbol{z(x,\ y)}$ até a origem. O fator $\boldsymbol{e^{j\theta}}$ fornece a direção do vetor no espaço, baseada no ângulo polar θ.

Com esses elementos, obtém-se a forma polar de $\boldsymbol{z_{complexo}}$. Para prosseguir com o mapeamento rumo às coordenadas Log-polares, deve-se aplicar a função logarítmica a $\boldsymbol{z_{polar}}$.

A LPM é definido por:

$$z_{logPolar} = \ln(z_{polar}) \tag{2.13}$$

A aplicação do logaritmo a um número complexo segue a definição:

$$z_{logPolar} = \ln(\rho * e^{j\theta}), \tag{2.14}$$

que é igual a:

$$z_{logPolar} = \ln(\rho) + \ln(e^{j\theta}), \tag{2.15}$$

Separando as partes real e imaginária, tem-se:

$$\begin{aligned} Real(z_{logPolar}) &= \ln(\rho) \\ Imag(z_{logPolar}) &= \ln(e^{j\theta}). \end{aligned} \tag{2.16}$$

O logaritmo da parte exponencial $\boldsymbol{e^{j\theta}}$ resulta no próprio $\boldsymbol{j\theta}$, de acordo com a propriedade do logaritmo, que afirma que $\boldsymbol{\ln(e^x) = x}$ para qualquer número real ou complexo. Assim, obtém-se:

$$\begin{aligned} Real(z_{logPolar}) &= \ln(\rho) \\ Imag(z_{logPolar}) &= j\theta. \end{aligned} \tag{2.17}$$

Resultando na forma:

$$z_{logPolar} = \ln(\rho) + j\theta, \tag{2.18}$$

A parte real $\boldsymbol{\ln(\rho)}$ representa a distância radial do ponto original ao centro, agora expressa em uma escala logarítmica. E a parte imaginária θ preserva a informação angular, indicando o ângulo do ponto em relação ao eixo real.

Para garantir que o mapeamento de $\boldsymbol{z_{polar}}$ para $\boldsymbol{z_{logPolar}}$ seja único, é necessário restringir o intervalo de θ para $\boldsymbol{[\theta, 2\pi)}$, evitando ambiguidade devido à periodicidade de $\boldsymbol{e^{j\theta}}$. Isso ocorre porque adicionar múltiplos inteiros de $\boldsymbol{2\pi}$ ao ângulo não altera a posição do ponto no plano complexo.

2.5.1 ULPM - Uniform Log-Polar Mapping

Como descrito, o LPM possui propriedades úteis que permitem o mapeamento eficiente de padrões regulares, os quais são invariantes a rotações e escalonamentos. Para uso prático, no entanto, é necessário discretizar o contínuo em segmentos radiais e angulares.

No LPM, no plano cartesiano, a distância $\mathbf{r}$ pode ter qualquer valor definido no intervalo. Para discretizar, é selecionado um intervalo de interesse $[\boldsymbol{r_{min}}, \boldsymbol{r_{max}}]$, que indica a distância mínima $\boldsymbol{r_{min}}$ da origem e a distância máxima $\boldsymbol{r_{max}}$. O objetivo é dividir esse intervalo em $\boldsymbol{T}$ partes discretas ao longo da distância radial $\boldsymbol{\rho}$, resultando em $\boldsymbol{T-1}$ pontos radiais que mantêm a relação logarítmica do LPM. No entanto, no LPM, os incrementos entre os pontos radiais crescem exponencialmente à medida que se afastam da origem, o que leva a uma distribuição desigual dos pontos. Na discretização angular isso não ocorre, pois o ângulo θ é tratado de forma linear. O intervalo completo de ângulos $\boldsymbol{[\theta, 2\pi)}$ é di-

vidido uniformemente em $\boldsymbol{A}$ partes iguais, resultando em uma separação constante entre cada ponto angular.

Portanto, em casos onde uma representação mais uniforme e uma distribuição mais equilibrada dos pontos ao longo do raio são necessárias, o Mapeamento Log-Polar Uniforme - Uniform Log-Polar Mapping (ULPM) fornece um método para tornar os pontos espaçados de maneira quase uniforme. Isso é feito ajustando a base logarítmica utilizada para a discretização radial.

Intervalos em escala log-polar

Temos os pontos dados em escala log-polar como [30]: $\boldsymbol{a^{i-1}}$, $\boldsymbol{a^{i}}$, $\boldsymbol{a^{i+1}}$ para $i = 0, 1, 2, 3... T-1$ intervalos. A diferença (intervalo) entre dois pontos vizinhos na escala log-polar é dada por:

$$a^i - a^{i-1} \tag{2.19}$$

Onde a razão entre dois intervalos adjacentes é:

$$\frac{a^{i+1}-a^i}{a^i-a^{i-1}} = a \tag{2.20}$$

,

mostrando que há uma proporção fixa entre intervalos consecutivos ao avançar de i para $i+1$.

Portanto, se $\boldsymbol{a}$ for um valor próximo de 1, $\boldsymbol{a^{i}}$, $\boldsymbol{a^{i-1}}$ estarão muito próximos. Assim, haverá mais pontos entre $\boldsymbol{r_{min}}$ e $\boldsymbol{r_{max}}$. Na prática, isso significa que, ao escolher a próximo de 1, você obtém uma distribuição "quase uniforme" dos pontos em uma escala logarítmica.

De acordo com [30], é definido $\boldsymbol{a = b^{1/T}}$, onde $\boldsymbol{b = \frac{r_{min}}{r_{max}} > 1}$ e $\boldsymbol{T}$ é o número de intervalos log-polares entre $\boldsymbol{r_{min}}$ e $\boldsymbol{r_{max}}$. Dado $\boldsymbol{b = 2}$ temos:

$$a = 2^{1/T} \tag{2.21}$$

.

Ao definir $\boldsymbol{a = 2^{1/T}}$, teremos $\boldsymbol{T}$ subdivisões (ou T intervalos) entre $\boldsymbol{r_{min}}$ e $\boldsymbol{r_{max}}$, o que aumentará o número de posições possíveis no domínio log-polar.

Por fim o mapeamento uniforme log-polar definido como:

$$\begin{aligned} L1 &= floor(log_a(\tfrac{\rho_i}{R})) + \tfrac{T}{2} \\ L2 &= floor(\tfrac{A \times \theta_i}{\pi}) \end{aligned} \tag{2.22}$$

onde $\boldsymbol{L1}$ é a coordenada log-polar radial uniforme, e $\boldsymbol{L2}$ a coordenada log-polar angular. $\boldsymbol{\rho_i}$ é a i-ésima distância radial, $\boldsymbol{\theta_i}$ é i-ésimo ângulo polar resultante da Equação 2.11, $\boldsymbol{R}$ é um valor empírico que define a faixa cartesiana mapeada para o domínio log-polar, $\boldsymbol{floor(.)}$ é uma função de arredondamento de piso, ou seja, arredondar para o menor inteiro próximo. $\boldsymbol{T}$ é o número de intervalos radiais, e $\boldsymbol{A}$ o número de intervalos angulares.

2.6 Correlação de Fase e o Padrão de Rastreamento Bipolar

Uma maneira eficiente de ressincronizar um padrão embutido em uma imagem, evitando falsos alarmes causados pela grande magnitude dos coeficientes de baixa frequência dessa imagem, é adaptar o método de correlação de fase mantendo a magnitude do padrão de rastreamento. Nesse sentido, [30] introduz a correlação de fase customizada, que se difere da correlação cruzada normal e da correlação de fase convencional, porque aproveita as propriedades do padrão de rastreamento. A característica do padrão de rastreamento $\boldsymbol{g(x, y)}$ é que seus elementos podem assumir apenas valores discretos variando entre $\boldsymbol{-1}$, $\boldsymbol{0}$ e $\boldsymbol{1}$, o que é essencial para reduzir o alcance dinâmico do sinal, o que, por sua vez, ajuda a minimizar o chamado ruído de correspondência gerado pela imagem na qual o padrão de rastreamento foi inserido. Esse tipo específico de padrão permite que a informação de magnitude do padrão de rastreamento seja mantida durante o cálculo da correlação, preservando assim a distinção entre os valores bipolares (-1, 0 e 1).

De acordo com [23], a correlação cruzada é uma operação semelhante à convolução, sem a inversão do kernel, que é comumente usada em aplicações de processamento de sinais e imagens para medir a semelhança entre dois sinais em diferentes posições relativas. A correlação cruzada entre duas funções $\boldsymbol{f(x, y)}$ e $\boldsymbol{h(x, y)}$ é definida como:

$$f(x,y) \circ h(x,y) = \frac{1}{h \times w} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} f^*(i,\ j)\ h(x+i, y+j) \tag{2.23}$$

.

Nessa Equação 2.23, $\boldsymbol{h}$e $\boldsymbol{w}$representam as dimensões da função $\boldsymbol{f}$, $\boldsymbol{f^*(i,\ j)}$ indica o complexo conjugado de $\boldsymbol{f}$ em cada ponto $\boldsymbol{(i,\ j)}$, e a operação $\circ$ denota a correlação cruzada. O fator $\frac{\boldsymbol{1}}{\boldsymbol{h \times w}}$ normaliza o resultado, garantindo que a correlação seja independente da escala das funções.

O conjugado de um número complexo é definido como o número complexo que possui a mesma parte real e uma parte imaginária com sinal oposto. Mais formalmente, se você tiver um número complexo $\boldsymbol{c = c_{real} + c_{imag} i}$, (onde $\boldsymbol{c_{real}}$ é a parte real e $\boldsymbol{c_{imag}}$ é a parte

imaginária), então o conjugado complexo de $\boldsymbol{c}$, denotado por $\bar{\boldsymbol{c}}$, é $\bar{\boldsymbol{c}} = \boldsymbol{c_{real}} - \boldsymbol{c_{imag}}\boldsymbol{i}$. Ou seja, basicamente, você mantém a parte real e inverte o sinal da parte imaginária.

No domínio espacial, a correlação cruzada entre uma função bidimensional $\boldsymbol{f(x,y)}$ e um padrão de rastreamento $\boldsymbol{g(x,y)}$, consiste em deslocar o padrão sobre a função e calcular a soma ponderada dos produtos em cada deslocamento. Tal operação pode ser realizada de forma eficiente no domínio da frequência utilizando a Transformada de Fourier e o teorema da correlação, que afirma que a correlação cruzada no domínio espacial é igual à transformada inversa do produto espectral das transformadas de Fourier das funções envolvidas, sendo uma delas conjugada complexa, conforme descrito a seguir na equação:

$$r(k1, k2) = IDFT[F(u,v)G^*(u,v)] \tag{2.24}$$

.

onde $\boldsymbol{F(u,v) = DFT(f(x,y))}$ e $\boldsymbol{G(u,v) = DFT(g(x,y))}$, com a $\boldsymbol{IDFT}$ representando a transformada discreta inversa de Fourier. O símbolo $*$ indica o conjugado complexo de $\boldsymbol{G(u,v)}$, e $\boldsymbol{k1}$ e $\boldsymbol{k2}$ representam as coordenadas do ponto de máxima correlação. Essa abordagem clássica permite localizar a melhor correspondência entre um padrão de rastreamento e uma função, buscando a posição de deslocamento que maximize a correlação.

Entretanto, abordagens tradicionais de correlação cruzada podem ser suscetíveis a falsos positivos devido à magnitude extremamente variável dos coeficientes de baixa frequência [30]. Para resolver esse problema, foram propostas técnicas baseadas em correlação de fase, como em [31], que sugere o uso da correlação de fase apenas, ou seja, quando se considera apenas a fase das transformadas de Fourier dos sinais ou imagens a serem correlacionados, descartando a magnitude. O motivo por trás disso é que a correspondência se torna mais fácil de ser realizada, pois ao trabalhar apenas com a fase, reduz-se a interferência da grande variação da magnitude, que frequentemente pode gerar ruídos e falsos positivos durante o processo de correspondência. A informação da fase se refere à estrutura (principalmente espacial) do sinal, enquanto a magnitude está relacionada apenas à intensidade do sinal. Como resultado, a correlação de fase é útil para encontrar o padrão, mesmo quando a magnitude varia significativamente.

Diante disso, [30] propôs manter a magnitude do padrão de rastreamento $\boldsymbol{g(x,y)}$, ao invés de usar apenas a fase de $\boldsymbol{F(u,v)}$, descartando a magnitude de ambos os sinais (como ocorre quando se utiliza apenas correlação de fase). A razão disso é devido ao fato dos elementos de $\boldsymbol{g(x,y)}$ estarem em um intervalo bipolar limitado de $\boldsymbol{-1}$, $\boldsymbol{0}$, $\boldsymbol{1}$, fazendo com que esse intervalo minimize os efeitos do ruído proveniente da imagem na qual foi inserido o padrão.

Os valores bipolares $\boldsymbol{-1}$ e $\boldsymbol{1}$ têm a propriedade de serem simétricos em torno de zero,

o que faz com que haja uma maior probabilidade de cancelamento (valores positivos e negativos se anulam mutuamente) quando multiplicados pelos coeficientes da imagem no cálculo da correlação. Além disso, os **zeros** no padrão de rastreamento faz com que certas regiões da imagem não contribuam para a correlação, ajudando a diminuir o impacto das regiões de alta magnitude da imagem no resultado.

Se apenas a fase de $\boldsymbol{g(x, y)}$ fosse usada, essa vantagem seria perdida, pois a fase sozinha não contém informações sobre a natureza bipolar do sinal. Manter a magnitude de $\boldsymbol{g(x, y)}$ permite preservar essa característica, que, como já mencionado anteriormente, ajuda a suprimir o ruído de correspondência atenuando as interferências causadas pela alta intensidade da imagem que poderiam levar a falsos positivos.

A correlação de fase customizada é, portanto, um ajuste que mantém a informação de fase da transformada de Fourier $\boldsymbol{F(u, v)}$, mas preserva o magnitude do padrão de rastreamento $\boldsymbol{g(x, y)}$ para melhorar a robustez contra o ruído e reforçar a correspondência correta [30]. A definição para a correlação de fase customizada é dada como:

$$\begin{aligned} r''(k1, k2) &= IDFT[F'_\phi(u, v)G^*(u, v)] \\ F'_\phi(u, v) &= e^{jF_\phi(u,v)} \end{aligned} \tag{2.25}$$

onde $\boldsymbol{F'_\phi(u, v)}$ é a exponencial complexa da fase $\boldsymbol{F_\phi(u, v)}$ da $\boldsymbol{F(u, v)}$, sendo $\boldsymbol{F(u, v)}$ a trasnformada discreta Fourier de $\boldsymbol{f(x, y)}$. Além disso, $\boldsymbol{G^*(u, v)}$ é o conjugado complexo de $\boldsymbol{G(u, v) = DFT(g(x, y))}$, e $\boldsymbol{IDFT}$ representa a transformada discreta inversa de Fourier.

2.7 Marcas-d'água em Imagens Digitais

O processo de inserção de marcas-d'água em imagens digitais é uma técnica para incorporar informações em uma determinada imagem com o propósito de determinar a origem da imagem [32]. Por meio do algoritmo de marca-d'água digital, padrões de bits, chamados de assinatura, são inseridos na imagem [33]. A assinatura pode ser usada para detectar as informações de direitos autorais, sendo armazenada dentro do próprio arquivo de imagem, dificultando a detecção e a remoção da marca-d'água [33].

O procedimento de marca-d'água geralmente inclui dois processos: incorporação e extração de marca-d'água. No primeiro caso, o algoritmo recebe a imagem original e a marca-d'água, que pode ser qualquer tipo de dado, como uma imagem secreta, e depois realiza a incorporação, produzindo uma imagem com marca-d'água, sendo esta transmitida em domínio público. O último processo, ou seja, a extração, é aplicado à imagem com marca-d'água com o objetivo de extrair a imagem secreta [24].

A marca d'água de uma imagem deve conter algumas propriedades como por exemplo: imperceptibilidade, sendo invisível ao Sistema Visual Humano (HVS) e não causando muita distorção na imagem original; carga útil, onde fornece capacidade máxima de incorporação para garantir a recuperação completa da marca-d'água durante o processo de extração; robustez, que é a capacidade de extrair a marca-d'água, mesmo quando a imagem digital é exposta a vários ataques geométricos e de processamento de sinal; segurança, que é a propriedade de suportar ataques hostis e permitir que somente a entidade autenticada consiga modificar a marca-d'água; complexidade computacional, que lida com o quantum de tempo levado pelos algoritmos de marca-d'água para codificar e decodificar; e eficácia da incorporação, onde existe probabilidade de detectar a marca d'água imediatamente após a incorporação [24].

Com base em seus algoritmos de extração, os métodos de marca-d'água de imagem são divididos em três abordagens: cego, não cego e semi-cego. Um algoritmo é cego se a marca-d'água puder ser extraída apenas utilizando a imagem marcada. Já o modelo não cego precisa da imagem original e da própria marca-d'água para realizar a extração. A abordagem de marca-d'água semi-cega não necessita da imagem original, porém utiliza a marca-d'água original para extrair a marca-d'água da imagem marcada [34]. Além disso, os sistemas de marca-d'água podem ser classificados como frágil ou robusto. A marca-d'água frágil é projetada para ser sensível a qualquer alteração, de modo que o detentor dos direitos possa garantir a integridade e autenticidade da mídia marcada, funcionando como uma assinatura digital. Por outro lado, a marca-d'água robusta é feita para ser resistente a interferências e processamentos na mídia, sendo utilizada geralmente para proteção de direitos autorais [6].

2.8 Métodos de Ocultação de Dados

2.8.1 LBM - Low-bit(s) Modulation

A Modulação de bits *'baixos'* - Low-bit(s) Modulation (LBM) é uma técnica de incorporação de informação, onde os bits menos significativos de um sinal quantizado são substituídos pelos bits da mensagem. Essa estratégia de inserção pode ser generalizada em uma classe de sistemas de quantização e substituição, denominados LBM generalizados [35], logo

$$s = q(g) + d(m) \tag{2.26}$$

onde $\boldsymbol{g}$ é o sinal original e $\boldsymbol{q(.)}$ um quantizador grosseiro que representa os bits mais significativos, que é somado a um deslocamento $\boldsymbol{d(m)}$ de acordo com a mensagem $\boldsymbol{m}$.

A fórmula a seguir, é a estratégia utilizada pelo método de referência [3] para incorporar os bits da marca-d'água, onde essa estratégia se encaixa dentro da classe generalizada dos LBM. Dessa forma:

$$s = \begin{cases} \underbrace{2\Delta \times round(g \ / \ 2\Delta)}_{q(g)} + \underbrace{\Delta/2, \quad if \ (m == 1)}_{d(m)} \\ \underbrace{2\Delta \times round(g \ / \ 2\Delta)}_{q(g)} - \underbrace{\Delta/2, \quad if \ (m == 0)}_{d(m)} \end{cases} \tag{2.27}$$

onde $\boldsymbol{g}$ representa os coeficientes da parte real da QDFT, $\boldsymbol{\Delta}$ é o passo de quantização e $\boldsymbol{m}$ representa o bit da marca-d'água.

2.8.2 DM - Dither Modulation

O DM é a implementação prática da Modulação por Índice de Quantização - Quantization Index Modulation (QIM), em que a base teórica central do QIM é utilizar a mensagem $\boldsymbol{m}$ para determinar um quantizador específico dentre vários disponíveis, de modo que esse quantizador selecionado é utilizado para quantizar o sinal hospedeiro. O DM implementa isso de um modo eficiente, onde em vez de projetar vários quantizadores complexos, ele utiliza um único quantizador de base $\boldsymbol{q(.)}$, e desloca esse quantizador utilizando um vetor de Dither $\boldsymbol{d(m)}$ que é determinado pela mensagem $\boldsymbol{m}$. Logo,

$$s(g;\ m) = q(g + d(m)) - d(m) \tag{2.28}$$

onde $\boldsymbol{g}$ é o sinal original, $\boldsymbol{m}$ é a mensagem a ser inserida, $\boldsymbol{d(m)}$ é o vetor de Dither correspondente a mensagem, $\boldsymbol{q(.)}$ é o quantizador base e $\boldsymbol{s}$ é o sinal com a mensagem incorporada.

Na prática, para cada mensagem $\boldsymbol{m}$, o sinal hospedeiro $\boldsymbol{g}$ é quantizado por uma grade de quantização diferente, sendo esta a grade do quantizador base deslocada por valores diferentes em $\boldsymbol{d(m)}$. Dessa forma, o DM implementa de forma prática e eficiente a ideia de múltiplos quantizadores do QIM.

2.9 SIFT - Scale Invariant Feature Transform

O algoritmo SIFT, proposto inicialmente por Lowe [36], é amplamente utilizado na área de correspondência de características em imagens. As características extraídas pelo SIFT são projetadas para serem invariantes a escala, rotação e translação. O processo de extração envolve quatro etapas para extração das características, sendo estas a geração do espaço de

escala, a detecção dos extremos no espaço de escala, a localização dos pontos de interesse e a geração do descritor SIFT [37].

O objetivo da primeira etapa é criar representações da imagem em múltiplas escalas. Para isso, é feita a convolução da imagem $I(x, y)$ com filtros Gaussianos $G(x, y, \sigma)$ de diferentes escalas s_K. Logo,

$$\begin{aligned} G(x, y, \sigma) &= \tfrac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \\ L(x, y, (s_K)\sigma) &* I(x, y), \end{aligned} \tag{2.29}$$

onde $L(x, y, (s_K)\sigma)$ é a chamada pirâmide Gaussiana.

A segunda etapa é detectar os extremos no espaço de escala, onde são identificados os potenciais pontos de interesse que são invariantes à escala. De maneira eficiente, isso é feito utilizando a função de Diferença de Gaussianas - Difference-of-Gaussian (DoG), onde a imagem DoG é calculada pela diferença entre duas escalas próximas na pirâmide Gaussiana, logo

$$D(x, y, \sigma) = L(x, y, (s_K)\sigma) - L(x, y, \sigma). \tag{2.30}$$

A Figura 2.4 demonstra o processo, onde para cada oitava do espaço de escala, a imagem inicial é convoluída repetidamente com funções Gaussianas para gerar o conjunto de imagens do espaço de escala mostrado à esquerda. Imagens Gaussianas adjacentes são subtraídas entre si para produzir as imagens de diferença de Gauss (DoG), demonstradas à direita. Após cada oitava, a imagem DoG é reamostrada com um fator de redução de 2, e o processo é repetido.

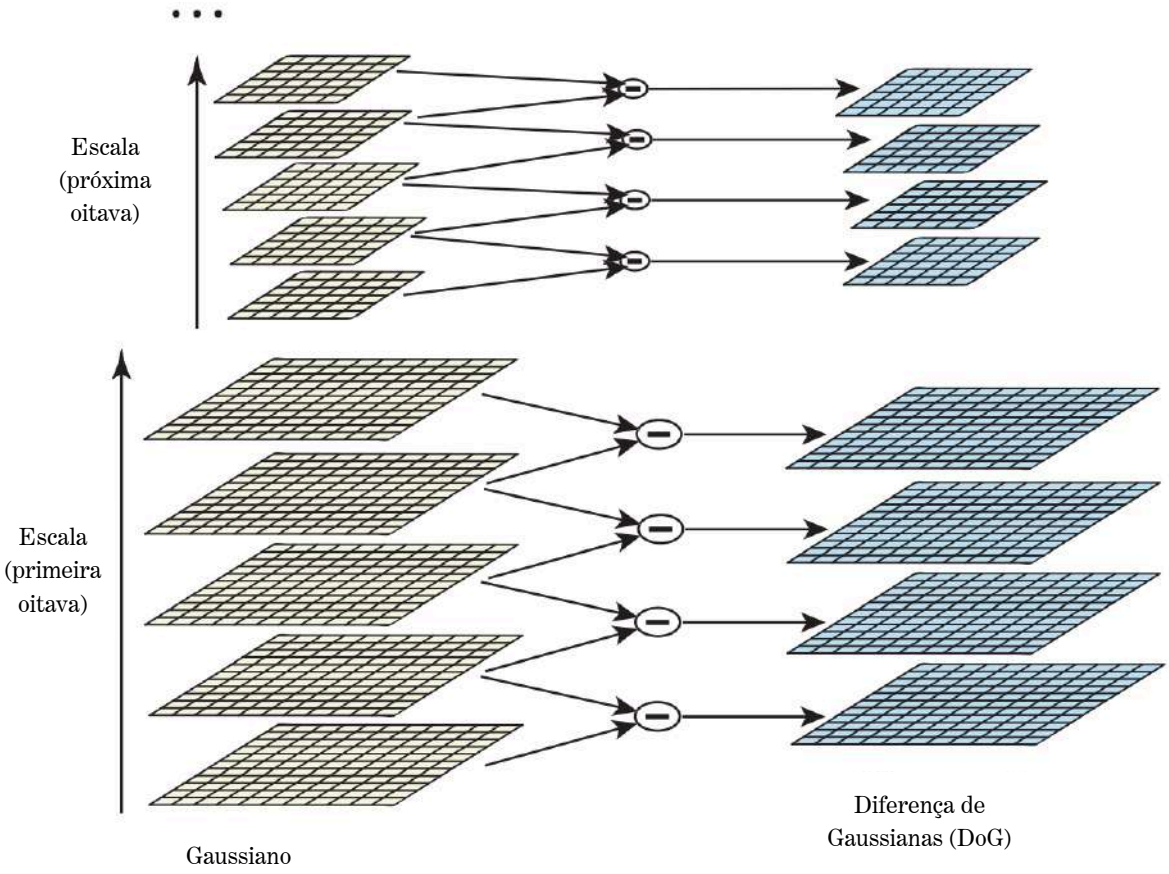


Figura 2.4: Espaço de escala. (Fonte: [36]).

Para determinar os extremos locais (máximos e mínimos), cada pixel na imagem DoG é comparado com seus 26 vizinhos, onde 8 são na mesma imagem, 9 na imagem da escala acima e 9 da escala abaixo, conforme demonstrado na 2.5.

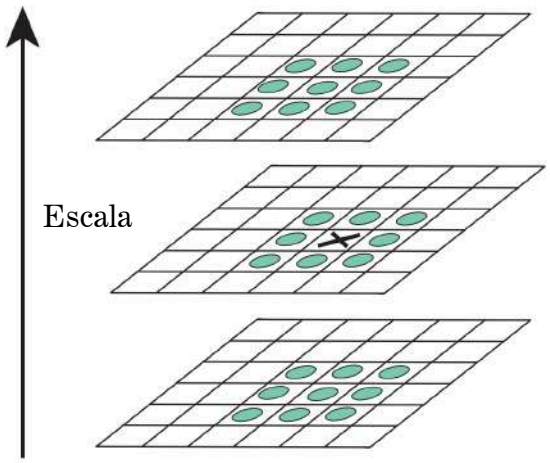


Figura 2.5: Máximos e mínimos DoG (Fonte: [36]).

Na etapa de otimização da localização do ponto de interesse, o ponto candidato encontrado tem sua localização e escala refinados, além de eliminar pontos instáveis. O ajuste preciso é feito utilizando uma expansão de Taylor da função de espaço de escala $D(x, y, \sigma)$ para calcular a localização interpolada do extremo. Os pontos com baixo contraste são sensíveis a ruído e, portanto, são instáveis, levando à sua rejeição se o valor da função no extremo $|D(X_M)|$ for menor que um limiar(0.03). Pontos mal localizados ao longo de uma aresta também são eliminados calculando as curvaturas principais por meio da matriz Hessiana, se a razão entre essas curvaturas for maior que um limiar(≥ 10), o ponto é descartado.

Após os pontos de interesse estáveis terem sido identificados, um descritor é criado para cada um deles. Primeiro é feita a atribuição de orientação de modo a alcançar a invariância à rotação. Cada ponto de interesse tem uma orientação consistente atribuída com base nas propriedades locais da imagem. Dessa forma, um histograma de orientações de gradiente é formado a partir de uma região ao redor do ponto de interesse, onde o pico mais alto do histograma define a orientação do ponto. Outros picos próximos de 80% do pico mais alto podem ser usados para criar pontos de interesse adicionais na mesma localização, porém com novas orientações. Depois da definição das orientações dos pontos de interesse, um vetor de características é criado, onde um bloco de 16 x 16 pixels ao redor do ponto de interesse é selecionado e dividido em 16 sub-regiões de 4 x 4 pixels. Cada sub-região tem um histograma de 8 direções de gradiente calculado, de modo que a concatenação desses 16 histogramas resulta em um vetor de características de 128 dimensões (16 sub-regiões x 8 direções). Por fim, para reduzir os efeitos das mudanças de iluminação, o vetor de características é normalizado (normalização L2) para ter um comprimento unitário.

A Figura 2.6 apresenta uma versão simplificada do bloco amostrado ao redor do ponto de interesse, onde em vez de um bloco de 16 $\times$ 16 pixels com sub-regiões de 4 $\times$ 4 pixels, é ilustrado um bloco de 8 $\times$ 8 pixels dividido em sub-regiões de 2 $\times$ 2 pixels.

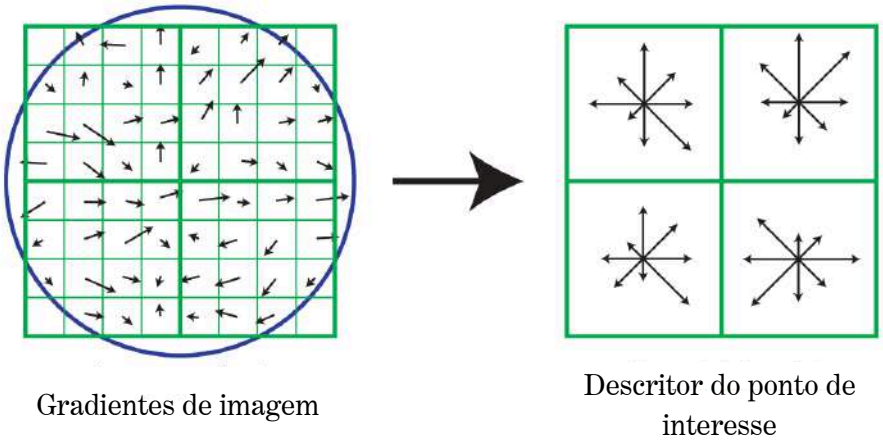


Figura 2.6: Pontos de interesse (Fonte: [36]).

2.9.1 RootSIFT

Para que a correspondência entre pontos de interesse funcione de maneira consistente, o vetor de características passa por uma normalização L2, ajustando o vetor para que sua norma Euclidiana seja igual 1. No entanto, para a comparação de histogramas, a norma L2 possui um desempenho inferior a outras técnicas, como a distância de Hellinger.

O RootSIFT é uma atualização para a comparação de descritores SIFT que aumenta de forma significativa o desempenho e a precisão. A melhoria é feita ao realizar um pré-processamento do vetor de características (descritor) SIFT, já normalizados em L2 por padrão, e realizar a normalização L1 seguida pela raiz quadrada de cada elemento. Esse simples ajuste faz com que a comparação dos novos descritores RootSIFT, usando a distância Euclidiana padrão, seja matematicamente equivalente a comparar os descritores SIFT originais com o kernel de Hellinger, que é mais eficaz para histogramas [38].

2.10 Ataques em Sistemas de Marca-d'água

Na maioria das aplicações de marca-d'água, a chance dos dados protegidos serem processados de algum jeito é muito alta. Esse processamento pode incluir aprimoramento de sinal, compressão com perdas ou filtragens. Analisando nesse aspecto, a marca-d'água inserida pode ser prejudicada tanto intencionalmente quanto de forma acidental, assim, processamentos podem ser aplicados com o objetivo claro de remover a marca-d'água. Portanto, no contexto de marcas-d'água, considera-se por ataque, qualquer processamento que possa comprometer a detecção da marca-d'água ou a comunicação das informações transmitidas por ela [39].

2.11 Métricas

Para medir a eficácia em algoritmos de watermarking, os métodos comumente usados são a Relação Sinal-ruído de Pico - Peak Signal-to-Noise Ratio (PSNR), o Índice de Similaridade Estrutural - Structural Similarity Index Measure (SSIM), a Correlação Normalizada - Normalized Correlation (NC) e a Taxa de Erro de Bit - Bit Error Rate (BER). Sendo o PSNR e o SSIM métricas para avaliar o impacto de mudanças perceptuais entre a imagem hospedeira e a imagem com marca-d'água, e o BER e NC para avaliar a robustez da marca-d'água extraída frente a ataques comuns em processamento de imagens [40].

2.11.1 PSNR - Peak Signal-to-Noise Ratio

O PSNR mede a relação do pico máximo de um sinal(no caso especifico, imagens) e o ruído presente, ou seja, ele mede o quão forte é o sinal em relação a um ruido inserido. O valor do PSNR é expresso em Decibel (dB).

Quanto maior o valor do PSNR, menor é o ruído introduzido e mais fiel é imagem modificada em relação a imagem original. Um valor representado como infinito(Inf) do PSNR indica que as imagens são 100% idênticas.

Definição [4]:

$$PSNR(H_1, H_2) = 10 \log_{10} \left(3 \times h \times w \ \frac{MAX_H^2}{MSE} \right) \tag{2.31}$$

onde $\boldsymbol{h}$ e $\boldsymbol{w}$ são as dimensões das imagens processadas. Note que ambas as dimensões devem ser iguais, da imagem original e da imagem comparada. $\boldsymbol{MAX_H}$ representa o valor máximo possível que um pixel pode assumir, no caso de imagens com 8 bits de profundidade por canal, $\boldsymbol{MAX_H = 255}$.

O Erro Quadrático Médio - Mean Square Error (MSE) é dado como:

$$MSE = \sum_{k=1}^{3} \sum_{x=1}^{h} \sum_{y=1}^{w} [H_1^k(x, y) - H_2^k(x, y)]^2 \tag{2.32}$$

onde $\boldsymbol{H_1}$ e $\boldsymbol{H_2}$ são a imagem original e a imagem comparada. $\boldsymbol{H_1^k(x, y)}$ e $\boldsymbol{H_2^k(x, y)}$ representam os valores dos pixels no canal $\boldsymbol{k}$ das imagens.

2.11.2 SSIM - Structural Similarity Index Measure

O SSIM é uma métrica que leva em conta características perceptuais, considerando a luminância, contraste e estrutura espacial da imagem. O valor do SSIM esta no intervalo [0, 1], também podendo ser exibido em porcentagem de similaridade. Quanto maior o valor do SSIM, maior a fidelidade entre imagens comparadas.

Definição [40]:

$$SSIM(H_1, H_2) = \frac{(2\mu H_1 \mu H_2 + C1)(2\sigma H_1 H_2 + C2)}{(\mu_{H_1}^2 + \mu_{H_2}^2 + C1)(\sigma_{H_1}^2 + \sigma_{H_2}^2 + C2)} \tag{2.33}$$

onde $\boldsymbol{\mu H_1}$ e onde $\boldsymbol{\mu H_2}$ são a média local da imagem original e da imagem comparada, respectivamente. Além disso, $\boldsymbol{\sigma^2_{H_1}}$ e $\boldsymbol{\sigma^2_{H_2}}$ são as disparidades entre $\boldsymbol{H_1}$ e $\boldsymbol{H_2}$. $\boldsymbol{C1}$ e $\boldsymbol{C2}$ são as duas variáveis para estabilizar a divisão, evitando divisões por zero.

2.11.3 NC - Normalized Correlation

O NC mede a correlação normalizada entre dois sinais, como imagens, sequências binárias ou matrizes de dados, avaliando o grau de similaridade entre seus valores numéricos. O NC resultante varia no intervalo entre 0 e 1 ([0, 1]), onde 1 indica que os sinais são totalmente idênticos, e 0 indica nenhuma correlação.

Definição [4]:

$$NC = \frac{\sum_{i=1}^{p}\sum_{j=1}^{q}[mark(i,j) \times mark'(i,j)]^2}{\left(\sqrt{\sum_{i=1}^{k_1}\sum_{j=1}^{k_2}[mark(i,j)]^2}\sqrt{\sum_{i=1}^{k_1}\sum_{j=1}^{k_2}[mark'(i,j)]^2}\right)} \tag{2.34}$$

2.11.4 BER - Bit Error Rate

Para comparar a marca-d'água extraída com a marca-d'água original, é calculado a taxa de erro de bits entre marcas utilizando o BER. Quanto menor o BER, maior é fidelidade da marca extraída. A faixa de valores para o BER esta no intervalo [0, 1], e se ele for zero, a marca-d'água foi recuperada 100%.

Definição [40]:

$$BER = \frac{1}{PQ}\left(\sum_{i=1}^{P}\sum_{j=1}^{Q}[W(i,j) - W^{©}(i,j)]^2\right) \tag{2.35}$$

onde $\boldsymbol{W}$ e $\boldsymbol{W^{©}}$ são a marca-d'água original e a marca-d'água extraída, respectivamente.

2.11.5 Intervalos

Valores altos de PSNR e de SSIM indicam que a distorção introduzida é baixa, o que implica em uma boa qualidade visual. Porém, é importante destacar que nem sempre valores altos irão garantir a imperceptibilidade da marca-d'água, sendo necessário uma avaliação conjunta do aspecto visual e dos níveis de PSNR e SSIM. As Tabela 2.1 e

Tabela 2.2 definem intervalos de valores que classificam e estimam o impacto da distorção introduzida [2].

Tabela 2.1: Intervalos PSNR. (Fonte: [2]).

PSNR(dB)	**Qualidade**
Abaixo de 15 dB ($\boldsymbol{dB <= 15}$)	Inaceitável
Entre 15 e 25 dB ($\boldsymbol{15 < dB <= 25}$)	Qualidade ruim. Distorções ou artefatos perceptíveis.
Entre 25 e 30 dB ($\boldsymbol{25 < dB <= 30}$)	Qualidade média. Aceitável para algumas aplicações, mas pode não ser suficiente para necessidades de alta qualidade.
Entre 30 e 35 dB ($\boldsymbol{30 < dB <= 35}$)	Boa qualidade. Aceitável para a maioria das aplicações.
Entre 35 e 40 dB ($\boldsymbol{35 < dB <= 40}$)	Qualidade muito boa.
Acima de 40 dB ($\boldsymbol{dB > 40}$)	Excelente qualidade. Diferenças quase imperceptíveis em relação à imagem original.

Tabela 2.2: Intervalos SSIM. (Fonte: [2]).

SSIM	**Qualidade**
Igual à 0 ($\boldsymbol{ssim = 0}$)	Nenhuma informação estrutural compartilhada entre as duas imagens.
Entre 0 e 0.5 ($\boldsymbol{0 < ssim <= 0.5}$)	Baixa similaridade. Há diferenças estruturais ou distorções significativas.
Entre 0.5 e 0.8 ($\boldsymbol{0.5 < ssim <= 0.8}$)	Similaridade moderada. Podem haver distorções perceptíveis, mas a estrutura geral permanece consistente com a imagem de referência.
Entre 0.8 e 1 ($\boldsymbol{0.8 < ssim < 1}$)	Alta similaridade entre as duas imagens.
Igual à 1 ($\boldsymbol{ssim = 1}$)	A imagem de teste é idêntica à referência. Similaridade estrutural perfeita.

2.12 Distribuição Binomial

A distribuição binomial calcula a probabilidade de ocorrer exatamente $\boldsymbol{k}$ sucessos em $\boldsymbol{n}$ tentativas. Cada tentativa é independente e possui duas possibilidades, sucesso ou fracasso, sendo a probabilidade de sucesso $\boldsymbol{p}$ constante em todas as tentativas. A Função de Distribuição Acumulada - Cumulative distribution function (CDF) binomial percorre

os possíveis valores de $\boldsymbol{k}$ e acumula as probabilidades até encontrar o menor $\boldsymbol{k}$ cuja soma atenda ao nível de confiança exigido.

A probabilidade de obter exatamente $\boldsymbol{k}$ sucessos em $\boldsymbol{n}$ tentativas é dado por:

$$P(K = k) = \binom{n}{k} \times p^k \times (1-p)^{n-k}, \tag{2.36}$$

onde

$$\binom{n}{k} = \frac{n!}{k! \times (n-k)!}. \tag{2.37}$$

A CDF binomial é a soma das probabilidades de 0 até $\boldsymbol{k}$:

$$F(k) = P(K \leq k) = \sum_{i=0}^{k} \binom{n}{k} \times p^i \times (1-p)^{n-i}. \tag{2.38}$$

Na inversa da CDF binomial, em vez de receber um número fixo de sucessos $\boldsymbol{k}$, a função recebe uma probabilidade acumulada desejada e retorna o menor valor de $\boldsymbol{k}$ que atinge ou excede essa probabilidade. Dessa forma, a probabilidade acumulada, definida pelo nível de confiança $(\boldsymbol{1 - \alpha})$ é passada, retornando um menor inteiro $\boldsymbol{k}$ tal que:

$$P(K \leq k) \geq (1 - \alpha), \tag{2.39}$$

onde $\boldsymbol{\alpha}$ representa o nível de significância.

2.13 RANSAC - Random Sample Consensus

O algoritmo Consenso por Amostragem Aleatória - Random Sample Consensus (RANSAC) pode ser definido como uma técnica de estimativa robusta que é amplamente utilizada em uma variedade de problemas de visão computacional, onde é frequentemente aplicado para correspondência de características, registro de imagens e cálculo de matrizes fundamentais, essenciais para recuperação de postura de câmera, estrutura a partir de movimento e localização baseada em características [41, 42]. Esse algoritmo atua em uma estrutura de hipótese e verificação, trabalhando com dados que contêm uma grande proporção de outliers. Primeiro, o RANSAC seleciona aleatoriamente um subconjunto mínimo de dados correspondente a uma hipótese e estima o modelo usando esse subconjunto mínimo de dados. O modelo resultante é então avaliado em todo o conjunto de dados, onde o número de pontos de dados consistentes com o modelo é usado como

suporte para o próprio modelo [42]. Com isso, o ciclo de hipótese e verificação é repetido até que o número máximo de iterações seja atingido, sendo encerrado apenas quando a probabilidade de encontrar um modelo com suporte maior do que o melhor modelo atual, ficar abaixo de um limite especificado.

O próximo capítulo (Capítulo 3) apresentará a revisão bibliográfica, com o resumo dos métodos de referência utilizados como base para o método proposto neste trabalho.

Capítulo 3

Revisão Bibliográfica

Este capítulo resume os artigos que embasam o método proposto neste trabalho, destacando os aspectos que serão mantidos e as melhorias planejadas para implementação.

3.1 A Blind Robust Color Image Watermarking Method Using Quaternion Fourier Transform

O artigo *'A Blind Robust Color Image Watermarking Method Using Quaternion Fourier Transform'* [3] descreve o método que foi a principal fonte de motivação para o método proposto. Em suas pesquisas, acerca de trabalhos anteriores, os autores encontraram majoritariamente artigos sobre o desenvolvimento de tecnologias de marca-d'água voltadas para as imagens digitais em tons de cinza. Além disso, dentre os artigos encontrados, os que exploravam marcas-d'água em imagens coloridas apresentavam algumas limitações nos métodos usados. Até o momento de publicação do artigo referido [3], a maioria dos esquemas de marca-d'água baseados na QDFT realizavam a divisão da imagem em blocos de tamanho 8×8 pixels, o que levava esses esquemas a precisarem de mais tempo de execução e a não serem robustos o suficiente para resistir aos cortes e ataques comuns. Foi constatado por Ouyang, et al [3] que poucos esquemas de marca-d'água baseados em QDFT consideraram outras abordagens diferentes de dividir a imagem em blocos, e por isso, eles exploram esse problema. Diante disso, Ouyang, et al [3] propôs um esquema de marca-d'água cega para imagens coloridas, baseado na QDFT e no ULPM, onde uma marca-d'água binária é embaralhada pela AT e incorporada nas frequências médias da parte real dos coeficientes QDFT. Depois, visando à robustez a ataques geométricos, um padrão de rastreamento bipolar é inserido na magnitude da QDFT utilizando o domínio log-polar do ULPM. Esse padrão de rastreamento é regenerado no momento da extração, e é correlacionado utilizando a correlação de fase customizada (Equação 2.25) para estimar

e corrigir uma transformação geométrica. Esse método se mostra bastante eficaz para transformações geométricas mais simples, como rotações e escalonamentos pequenos. No entanto, em casos de translação, o padrão de rastreamento correlacionado no domínio do ULPM não consegue detectar deslocamentos nos eixos angular e radial, o que leva à não reversão da distorção geométrica e consequentemente a falha da extração. Além disso, embora afirmem utilizar QIM, os autores recorrem a uma implementação do LBM generalizado (subseção 2.8.1) para modular os coeficientes, introduzindo artefatos perceptíveis na imagem marcada, o que é adicionalmente agravado pela incorporação agrupada nas frequências médias. Diante disso, é proposto neste trabalho substituir o LBM pelo DM, alterar a modificação localizada incorporando cópias de redundância da marca-d'água, e substituir o ULPM pelo SIFT a fim de reverter as transformações geométricas com mais precisão. Por fim, a estratégia de utilização da AT é aprimorada com o objetivo de elevar o nível de segurança do embaralhamento da marca-d'água antes da incorporação.

3.2 Color Image Watermarking Based on Quaternion Fourier Transform and Improved Uniform Log-polar Mapping

O artigo *'Color Image Watermarking Based on Quaternion Fourier Transform and Improved Uniform Log-polar Mapping'* [4] é uma atualização do artigo referido anteriormente em 3.1, na qual os mesmos autores mantiveram todo o processo inalterado exceto pelo aprimoramento do método ULPM, desenvolvendo o IULPM. A precisão angular do ULPM era de 0.5°, ou seja, qualquer ataque de rotação na imagem marcada que caísse entre esses passos era automaticamente arredondada para o ponto mais próximo. Por exemplo, em um ataque de 30.2°, o ULPM iria detectar 30.0° ou 30.5°, introduzindo um erro de 0.3 que comprometia a recuperação correta da marca-d'água. Na versão aprimorada, Ouyang, et al [4], aumentou a precisão para 0.1° ao ampliar o número de intervalos angulares no domínio log-polar de 360 para 1800. Dessa forma, rotações de 30.2° passaram a ser detectadas e corrigidas com precisão. Outro ponto relevante é a introdução de testes com vários conjuntos de imagens, demonstrando resultados consistentes, algo ausente na versão anterior que estringiu seus testes a apenas duas imagens. No entanto, essa versão atualizada e aprimorada ainda mantém a mesma limitação anterior, ou seja, translações ainda não são detectadas pelo ULPM/IULPM, portanto, as mesmas propostas de melhorias apresentadas em 3.1 estão mantidas.

O próximo capítulo (Capítulo 4) apresentará a metodologia e os passos para implementação do método proposto.

Capítulo 4

Metodologia

Este capítulo descreve o método proposto, abrangendo integralmente o fluxo de incorporação e extração da marca-d'água. Inicialmente, são abordados os mecanismos de segurança que envolvem a geração de chaves e IV's para o CSPRNG, responsável por fornecer a aleatoriedade necessária durante todo o processo. Em seguida, é feito o cálculo da capacidade de bits de informação que a imagem hospedeira pode carregar, que orienta o pré-processamento da marca-d'água, englobando ajustes dimensionais e o embaralhamento utilizando uma variação aprimorada da AT, denominada IBAT. Na etapa de incorporação, a imagem é convertida ao domínio da frequência pela QDFT, e três cópias redundantes dos bits da marca-d'água são inseridas na parte real dos coeficientes quaternários, estágio esse que inclui também a definição de coordenadas e a detecção de pontos de interesse via SIFT, que são armazenados juntamente com os demais parâmetros e com a marca-d'água embaralhada, servindo como referência para validação durante a extração. Para a extração, os pontos de interesse detectados pelo SIFT contidos nos parâmetros armazenados são utilizados para reverter distorções geométricas antes de submeter a imagem novamente à QDFT e recuperar os bits incorporados na parte real. Após isso, o pós-processamento seleciona a cópia da marca-d'água extraída que apresenta a maior similaridade com a marca-d'água de referência, utilizando a taxa de acerto para aplicar a inversa da CDF binomial e validar estatisticamente a presença da marca-d'água. Por fim, as cópias redundantes restantes são então combinadas para corrigir erros e aprimorar a qualidade final da marca-d'água recuperada.

4.1 Segurança do Sistema

Como o método proposto trata-se de um sistema semi-cego de marca-d'água, abordado na Seção 2.7, os parâmetros que foram utilizados durante a incorporação dos bits devem ser armazenados para tornar possível a extração. Apesar dos parâmetros não conterem

nenhuma localização explícita de onde foi inserida a marca-d'água, eles podem facilitar uma possível tentativa de remoção mais especializada, utilizando os parâmetros como vantagem. Diante disso, todos os parâmetros utilizados na incorporação são criptografados pelo AES.

No contexto de um sistema robusto de marca-d'água, abordado na Seção 2.7, é possível classificar os ataques em dois escopos principais, onde o primeiro é relacionado à robustez, que avalia a capacidade da marca-d'água de sobreviver a processamentos de sinal, sendo intencionais ou não, e o segundo aborda a segurança criptográfica, que engloba o processo de geração e proteção dos parâmetros de incorporação dos bits da marca-d'água.

Embora o método proposto utilize métodos criptográficos para a proteção dos parâmetros de incorporação e para geração de números pseudoaleatórios, a análise de resistência a ataques direcionados a essa segurança não será o foco. Esta decisão delimita o escopo para validar o requisito fundamental de um sistema de marca-d'água robusto, que é a sua capacidade de sobreviver a ataques de processamento de sinal aplicados diretamente à imagem marcada. Consequentemente, não serão demonstrados testes contra ataques especializados que exploram vulnerabilidades do algoritmo ou seus parâmetros para remover a marca-d'água de forma estratégica.

Portanto, o foco será nos ataques de processamento de sinal mantendo o escopo demonstrado na literatura acadêmica sobre marcas-d'água.

4.1.1 Ataques

Para validar a robustez do sistema de marca-d'água, diversos ataques de processamento de sinal serão aplicados à imagem marcada, onde esses ataques foram selecionados para simular cenários realistas, englobando tanto manipulações não intencionais que são comuns no compartilhamento de imagens digitais, quanto manipulações intencionais que tem como objetivo a remoção da marca-d'água. Dessa forma, a robustez do sistema será quantificada na seção de Resultados pela capacidade de detecção da marca-d'água após cada um desses testes. Os ataques que serão demonstrados nos resultados consistem em variações de processamentos de sinal comuns no contexto de imagens digitais, assim, a Tabela 4.1 detalha e categoriza cada conjunto desses ataques, em que o método proposto foi projetado para resistir.

Tabela 4.1: Descrição dos Ataques

Manipulação Aplicada	Categoria	Descrição
Compressão JPEG	Ataque de Compressão	Reduz o tamanho do arquivo da imagem descartando informações, o que pode degradar a marca-d'água.
Adição de Ruído (Gaussiano, Salt & Pepper)	Ataque de Ruído	Insere perturbações estáticas na imagem, o que pode corromper o sinal da marca-d'água.
Filtragem Espacial (Média, Mediana, Gaussiano)	Ataque de Filtragem	Aplica operações de suavização ou desfoque que atenuam as altas frequências, corrompendo a marca-d'água nessa faixa.
Ajuste de Brilho	Ataque de Intensidade	Altera uniformemente a luminância geral da imagem, deixando a imagem mais clara ou mais escura.
Ajuste de Contraste	Ataque de Intensidade	Aumenta ou diminui a diferença entre as áreas claras e escuras da imagem.
Rotação	Ataque Geométrico	Gira a imagem em torno do seu centro por um determinado ângulo.
Escala	Ataque Geométrico	Redimensiona a imagem, aumentando ou diminuindo seu tamanho total.
Corte (Cropping)	Ataque Geométrico	Remove partes das da imagem, descartando uma parte de sua área.
Translação	Ataque Geométrico	Desloca a imagem inteira em uma direção, sendo horizontal e/ou vertical.
Reflexão (Flip)	Ataque Geométrico	Inverte a imagem ao longo do seu eixo, sendo horizontal ou vertical.
Manipulações conjuntas	Ataques Combinados	Aplicação sequencial de dois ou mais ataques distintos para simular uma degradação mais severa.

4.1.2 Números Pseudoaleatórios

O processo de inserção e extração da marca-d'água envolve várias etapas que dependem de números pseudoaleatórios, como o embaralhamento dos bits, a definição de suas posições, a quantização dos coeficientes de frequência da imagem hospedeira, entre outras etapas. Diante disso, para garantir a aleatoriedade de forma segura, um CSPRNG baseado no CTR-DRBG é utilizado como fonte de números pseudoaleatórios, que além de serem utilizados nos procedimentos de inserção e extração, são utilizados para gerar as chaves e IV's necessários durante o processo. O padrão para esse tipo de gerador (CSPRNG) são chaves de 128, 192, ou 256 bits e IV's de 128 bits, onde esse mesmo formato é fornecido ao AES para criptografar todos os parâmetros utilizados no sistema.

A adoção de um CSPRNG baseado no CTR-DRBG na maioria das etapas do processo se deve à necessidade de reproduzir exatamente a mesma sequência de números aleatórios

durante a extração da marca-d'água. Porém, para casos em que a reprodução não é necessária, como no caso da geração de um salt ou de um IV específico, é utilizado um CSPRNG nativo do sistema operacional, obtendo o máximo de entropia a partir do pool de ruído do kernel.

4.1.3 Gerador de Chaves e IV's

Para gerar novas chaves ou IV's, uma chave e um IV são usados como *seed* no CSPRNG determinístico que produz um vetor de números pseudoaleatórios. Logo, o

$$vetor_{rand} = VetorRandCSPRNG(0, 255,\ (n_{chaves} \times n_{bytes}),\ chave,\ iv), \tag{4.1}$$

,

onde $\boldsymbol{n_{bytes}}$ define o tamanho da chave, e o intervalo [0 255] garante um inteiro de 8 bits sem sinal. Dessa forma, para obter as chaves, o $\boldsymbol{vetor_{rand}}$ é reorganizado em uma matriz em que cada linha representa uma chave individual, conforme ilustrado na Figura 4.1.

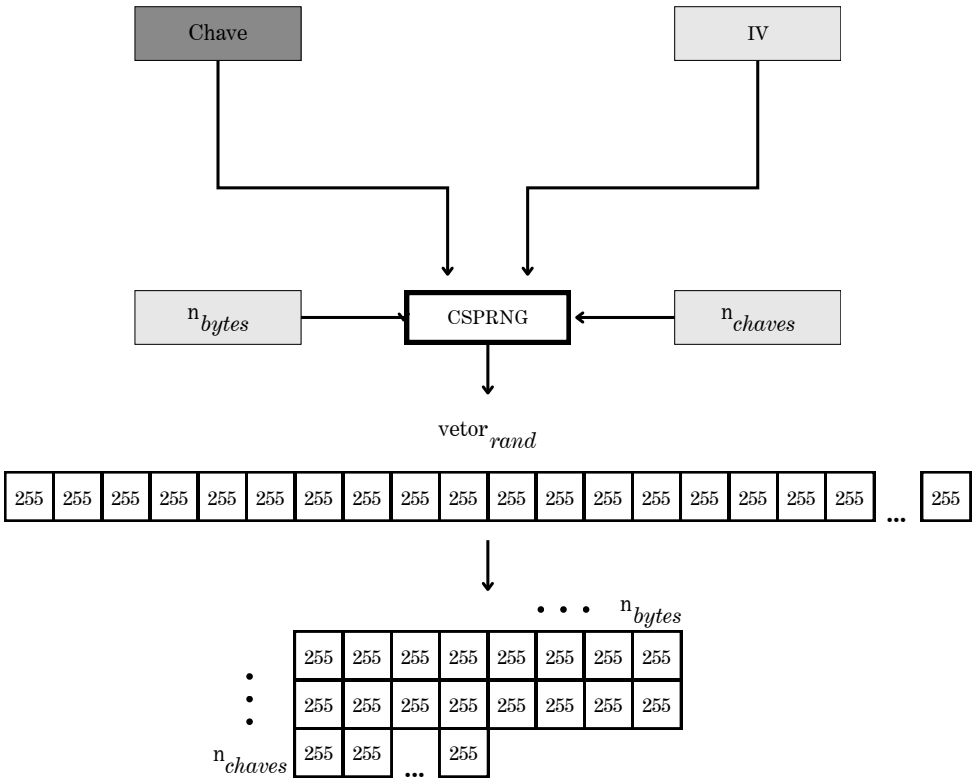


Figura 4.1: Gerador de chaves e IV's. (Fonte: Autoral).

4.1.4 Chaves do Sistema

Como o sistema depende da geração de números pseudoaleatórios, cada etapa deve usar uma chave e um IV exclusivos, de modo a garantir uma alta entropia dos números gerados. A chave mestra (256 bits) é obtida diretamente do gerador criptográfico nativo do sistema operacional, e é então utilizada para gerar as chaves do sistema. No total, são geradas 7 chaves de 32 bytes (256 bits) e 7 IV's de 16 bytes (128 bits), conforme a Figura 4.2.

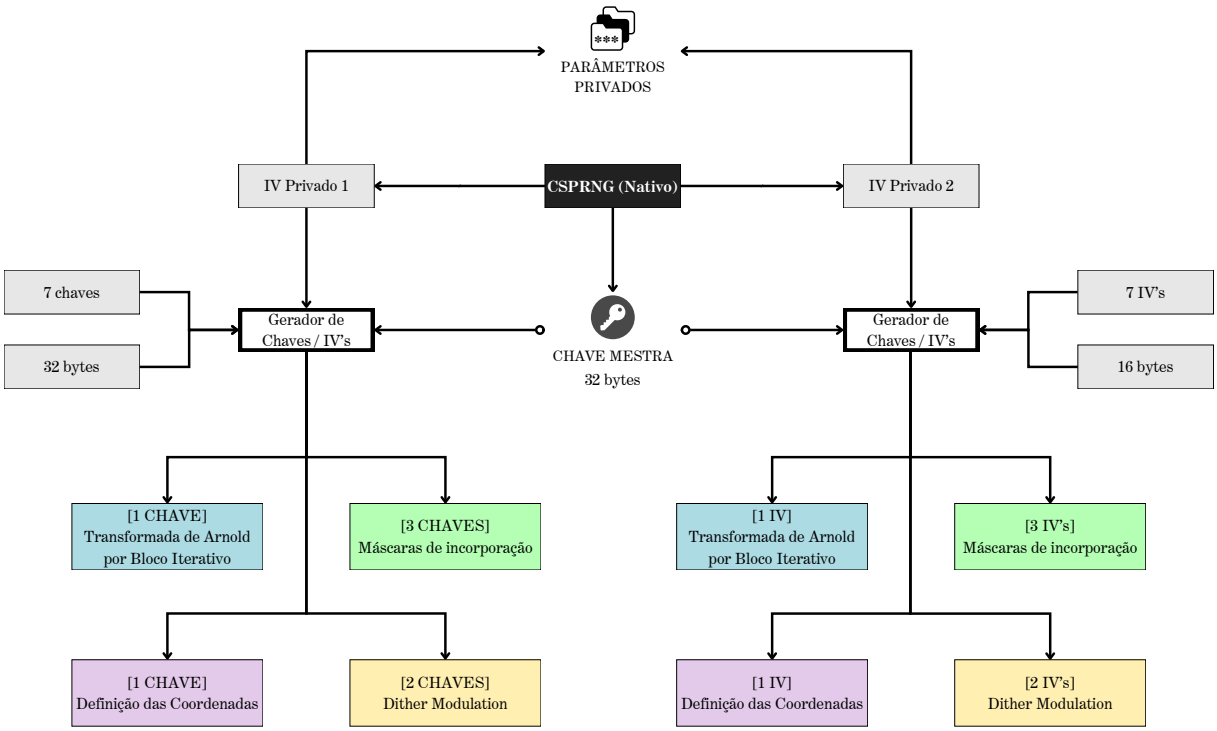


Figura 4.2: Chaves do sistema. (Fonte: Autoral).

4.2 Capacidade da Imagem Hospedeira

O cálculo da capacidade de inserção de bits na imagem hospedeira foi baseado na ideia contida no método de referência [3], onde o número de bits que podem ser inseridos é calculado de acordo com as dimensões da imagem. Para estimar a capacidade da imagem hospedeira, os autores se basearam na quantidade de bits inseridos por bloco, onde se o bloco possui a dimensão (8 x 8) pixels, ele tem a capacidade de inserção de 1 bit de informação, se a dimensão é (16 x 16) pixels a capacidade é de 4 bits de dados, e assim por diante. Com base nessa ideia, foi possível deduzir uma fórmula que representasse essa relação, que é definida como:

$$capacidade_{base} = round(2^{(2 \times (log_2(min(h,\ w)) - 3))}) \tag{4.2}$$

,

onde $\boldsymbol{h}$ e $\boldsymbol{w}$ são as dimensões de altura e largura da imagem, respectivamente. A operação $\boldsymbol{min(.)}$ retorna o menor dos valores, e $\boldsymbol{round(.)}$ é uma função de arredondamento para o valor mais próximo.

4.2.1 Capacidade por Faixa de Frequência

A faixa de baixa frequência do espectro QDFT concentra os coeficientes responsáveis por capturar mudanças muito suaves na intensidade dos pixels, por exemplo, regiões uniformes, gradientes de iluminação e transições lentas de cinza. Esses coeficientes de baixa frequência correspondem à maior parte da energia da imagem e, qualquer modificação pode causar distorções, como variações de brilho ou manchas perceptíveis. Diante disso, essa faixa é utilizada como referência para estimar o número de bits que podem ser inseridos na imagem hospedeira sem ultrapassar o limiar de imperceptibilidade. Assim, basear a capacidade de inserção nessa banda mais sensível garante que artefatos visíveis sejam minimizados ou evitados.

O valor da $\boldsymbol{capacidade_{base}}$ é definido utilizando como base apenas as dimensões de altura e largura, calculando um número fixo de bits independente do conteúdo da imagem. Diante disso, é necessário adaptar esse valor de acordo com a quantidade de baixas frequências presentes. Assim, é preciso quantificar a energia contida nas baixas frequências e utilizar esse limiar para adaptar o valor de $\boldsymbol{capacidade_{base}}$.

A quantificação da energia associada às baixas frequências em uma imagem, diretamente no domínio da frequência (QDFT), é um grande desafio, uma vez que não existe um critério universalmente estabelecido para definir os limiares que separam baixas e altas frequências no espectro. Nesse contexto, o uso do operador de Sobel, que funciona como um detector de gradientes locais, mostrou ser uma abordagem mais robusta e eficiente para estimar, de forma relativa, o conteúdo de baixas frequências entre diferentes imagens. Dessa forma, imagens com grandes regiões homogêneas e mudanças suaves na iluminação tendem a ter mais energia concentrada nas baixas frequências, enquanto imagens com muitas texturas, detalhes finos e bordas abruptas costumam apresentar uma maior presença de componentes de altas frequências.

Com base nessas informações, o operador Sobel foi utilizado como ferramenta para estimar a proporção de áreas uniformes e gradientes suaves presentes em uma imagem. Assim, ao utilizar esse valor como um parâmetro, foi possível obter uma aproximação do nível de modificação que pode ser aplicado na faixa de baixa frequência do espectro QDFT. Ou seja, se a imagem hospedeira possui uma alta proporção de áreas suaves, o valor da $\boldsymbol{capacidade_{base}}$ é reduzido para evitar distorções na imagem hospedeira.

Imagem original (nível de cinza) | Gradiente na direção vertical (Gx) | Gradiente na direção horizontal (Gy)

Figura 4.3: Gradientes calculados com o operador Sobel : Imagem (512x512). (Fonte: Autoral).

Os gradientes $\boldsymbol{Gx}$ e $\boldsymbol{Gy}$, que representam as mudanças de intensidade nas direções vertical e horizontal, respectivamente, são calculados utilizando o operador Sobel, conforme a Figura 4.3.

Com os valores $\boldsymbol{Gx}$ e $\boldsymbol{Gy}$ podemos obter a magnitude do gradiente, definida como:

$$MAG = \sqrt{G_x^2 + G_y^2} \tag{4.3}$$

A magnitude do gradiente representa a intensidade das variações locais na imagem, sendo tradicionalmente utilizada para a detecção de bordas e transições abruptas de intensidade. No entanto, no caso da análise de baixas frequências, os menores valores da magnitude do gradiente podem ser utilizados para identificar regiões homogêneas ou com variações suaves, que correspondem a áreas capturadas por componentes de baixa frequência no domínio da frequência. Com base em testes empíricos, foi adotado um limiar fixo de **10** para a magnitude do gradiente, que se mostrou adequado para quantificar essas regiões suaves. Assim, o valor da suavidade é definido como:

$$sv = \sum_{x=1}^{h} \sum_{y=1}^{w} (MAG(x, y) < 10) \tag{4.4}$$

onde $\boldsymbol{sv}$ é a contabilização de todos os pixels com uma variação de intensidade baixa.

Agora, com o valor da $\boldsymbol{sv}$, é possível calcular a proporção de pixels com variações lentas em relação ao total de pixels da imagem, e assim estimar o grau de homogeneidade da imagem. O $\boldsymbol{grau_{hg}}$ é definido por:

$$grau_{hg} = \frac{sv}{h \times w} \tag{4.5}$$

Onde **h** e **w** são as dimensões de altura e largura, respectivamente. O valor de $\boldsymbol{grau_{hg}}$ é utilizado para ajustar dinamicamente a capacidade base de inserção, $\boldsymbol{capacidade_{base}}$, definida na Equação 4.2. Portanto, a capacidade adaptada é definida como:

$$capacidade_{adaptada} = round\left(capacidade_{base} \times (1 - grau_{hg})\right) \tag{4.6}$$

onde o complemento de $\boldsymbol{grau_{hg}}$ é utilizado para definir a nova capacidade.

Para evitar uma redução drástica na capacidade de inserção, foi definido um valor mínimo como 50% da $\boldsymbol{capacidade_{base}}$:

$$capacidade_{min} = round(capacidade_{base} \times 0.5). \tag{4.7}$$

Assim, caso uma imagem possua níveis muito altos de suavidade, implicando em um valor pequeno para $\boldsymbol{capacidade_{adaptada}}$, essa assumirá o valor mínimo.

4.2.2 Redundância

Ao utilizar a faixa de baixa frequência como referência, definindo a $\boldsymbol{capacidade_{adaptada}}$, as outras faixas, média frequência e alta frequência, podem ser modificadas com a mesma quantidade de bits sem causar grandes distorções. Com isso, é possível triplicar a capacidade de inserção da imagem hospedeira mantendo a mesma qualidade visual. No entanto, ao invés de utilizar esse espaço extra para inserir uma marca-d'água com mais bits de informação, optou-se por manter a capacidade de inserção ($\boldsymbol{capacidade_{adaptada}}$), e inserir a mesma marca-d'água três vezes como redundância, uma cópia da marca-d'água para cada faixa de frequência, intensificando a robustez do sistema. Assim, a capacidade total é definida como:

$$capacidade_{total} = 3 \times capacidade_{adaptada} \tag{4.8}$$

onde cada faixa recebe a mesma quantidade de bits, sendo uma cópia dos bits da marca-d'água para cada faixa, conforme a Figura 4.4.

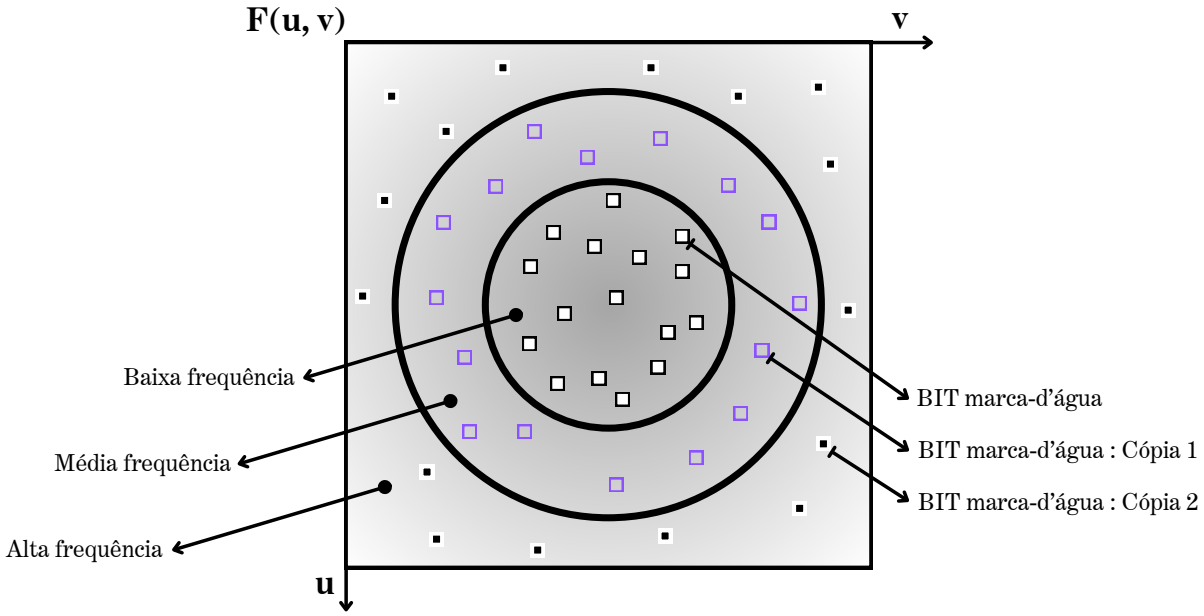


Figura 4.4: Inserção dos bits da marca-d'água nas três bandas de frequência. (Fonte: Autoral).

4.3 Pré-Processamento da Marca-d'água

Para atender os requisitos do sistema, a marca-d'água deve ser uma imagem binária, com dimensões verticais e horizontais iguais $(\boldsymbol{h_m = w_m})$, onde $\boldsymbol{h_m}$ e $\boldsymbol{w_m}$ representam as dimensões de altura e largura da marca-d'água, respectivamente. Para sua inserção na imagem hospedeira, a marca-d'água é ajustada proporcionalmente de acordo com a estimativa da capacidade de bits. Além disso, a Transformada de Arnold, abordada em [XX], é aplicada para embaralhar os bits, adicionando uma camada de segurança ao sistema e garantindo uma distribuição equilibrada dos bits, reduzindo o impacto visual.

4.3.1 Ajuste da Marca-d'água

Ao importar a marca-d'água, o ajuste de suas dimensões é feito com base na quantidade de bits que podem ser inseridos na imagem, definidos pela $\boldsymbol{capacidade_{adaptada}}$. Caso a marca-d'água esteja em um formato RGB ou em tons de cinza, ela é convertida para uma imagem binária. O Algoritmo **1** demonstra esse processo.

Algoritmo 1 Importar marca-d'água

if $h_m = w_m$ **then**

  if $marcaDagua \neq Binaria$ **then**

    $marcaDagua \leftarrow ImagemBinaria(marcaDagua)$

  end if

  $resolucao_{marca} \leftarrow floor(\sqrt{capacidade_{adaptada}})$

  $marcaDagua \leftarrow Redimensionar(marcaDagua,\ resolucao_{marca})$

end if

4.3.2 Embaralhamento

A correta aplicação da Transformada de Arnold - Arnold Transform (AT) requer que a imagem tenha dimensões iguais, ou seja, uma imagem quadrada. Para facilitar o entendimento, as dimensões da imagem de marca d'água serão tratadas como $\boldsymbol{N}$.

O comportamento cíclico da AT, define que ao aplicar a transformação um número repetido de vezes, a imagem "embaralhada" eventualmente retornará à sua forma original. Isso gera uma grande vulnerabilidade, pois se o atacante conseguir recuperar a marca-d'água embaralhada, conhecer $\boldsymbol{N}$ e a matriz de transformação, a marca poderá ser facilmente decodificada por força bruta se $\boldsymbol{N}$ for pequeno. Portanto, depender do número de iterações e utilizar uma matriz de transformação fixa não é uma boa estratégia, principalmente em imagens de marcas-d'água que geralmente possuem dimensões reduzidas.

Para elevar a segurança da AT, a matriz $\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$, comumente utilizada, é substituída por uma matriz semelhante que atenda aos requisitos de unimodularidade e hiperbolicidade. Essas características garantem a dinâmica caótica típica da AT e definem a base para a geração de uma nova matriz. Dessa forma, define-se a base para a unimodularidade e hiperbolicidade como

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad Det(A) = 1, \quad |a + d| > 2, \tag{4.9}$$

onde $\boldsymbol{Det(.)}$ calcula o determinante, $\boldsymbol{Det(A) = 1}$ caracteriza a matriz como unimodular, e o traço da matriz $\boldsymbol{|a + d| > 2}$, define a matriz como hiperbólica.

A AT realiza operações modulares que mantém o intervalo das coordenadas dentro das dimensões da imagem, assim cada operação da AT é realizada de acordo com o

módulo $\boldsymbol{N}$. Isso faz com que não importe o valor absoluto dos componentes da matriz de transformação, ela sempre será reduzida a $\boldsymbol{N}$. Sendo assim, ao gerar as matrizes de forma aleatória o espaço de possibilidades é restrito a $\boldsymbol{N}$, que define o nível de segurança da AT. De forma clara, quanto menor é o valor de $\boldsymbol{N}$, menor é a quantidade de matrizes possíveis que respeitam as condições da Equação 4.9.

A estratégia principal é gerar matrizes de forma aleatória que atendam aos requisitos da Equação 4.9, no entanto, as imagens que representam marcas-d'água geralmente possuem dimensões $\boldsymbol{N}$ reduzidas, que impactam diretamente no nível de segurança da AT. Devido a esse fato, mesmo que uma matriz seja gerada de forma aleatória de acordo com uma chave, um atacante, de posse do valor de $\boldsymbol{N}$, pode testar todas as matrizes possíveis dentro do espaço $\boldsymbol{N}$. Para contornar esse problema, uma novo método para melhorar a segurança da AT foi desenvolvido, a IBAT.

IBAT - Iterative Block Arnold Transform

Esse método consiste em aplicar a AT à um bloco de tamanho aleatório $\boldsymbol{Nb}$ utilizando uma matriz de transformação unimodular e hiperbólica gerada de forma aleatória, com um número aleatório de iterações. Isso é feito para $\boldsymbol{n_{blocos}}$ distribuídos aleatoriamente por toda a área da imagem, onde cada bloco possui uma matriz de transformação única, e tamanho $\boldsymbol{Nb}$ variado. O número de iterações da AT aplicada a um bloco, varia de bloco para bloco, e cada bloco é sobreposto a outro bloco, fazendo com que a AT aplicada a um bloco utilize uma área que já foi transformada por outro bloco, ou seja, um pixel em uma coordenada é dependente de inúmeras iterações de transformadas de blocos anteriores. A Figura 4.5 demonstra visualmente a distribuição dos blocos ao aplicar a IBAT.

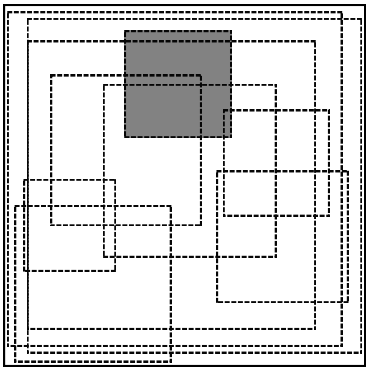


Figura 4.5: Distribuição aleatória de blocos ao longo da área da imagem, onde cada bloco é trasnformado pela AT (Fonte: Autoral).

A escolha do número de blocos $(\boldsymbol{n_{blocos}})$ utilizados durante a IBAT é crucial, pois quanto maior $\boldsymbol{n_{blocos}}$, melhor é a distribuição dos blocos, e maior é a área transformada.

Além disso, como cada bloco é dependente do outro, a ordem de transformação dos blocos importa, ou seja, quando maior $\boldsymbol{n_{blocos}}$ maior é o número de combinações possíveis dos blocos. Portanto, a aplicação da IBAT depende de vários parâmetros que elevam a segurança do processo de embaralhamento da marca-d'água, sendo estes, a matriz de transformação de cada bloco, a dimensão de cada bloco, o número de iterações aplicado a cada bloco, as coordenadas de cada bloco, e a ordem de transformação de todos os blocos.

Antes da transformação dos blocos, a AT é aplicada na imagem completa no início e no fim do processo, afim de embaralhar todos os blocos em uma única transformação. Ambas as transformações, inicial e final, utilizam matrizes únicas e iterações variadas. A Figura 4.6 e a Figura 4.7 oferecem uma visão detalhada de todo o processo de aplicação da IBAT.

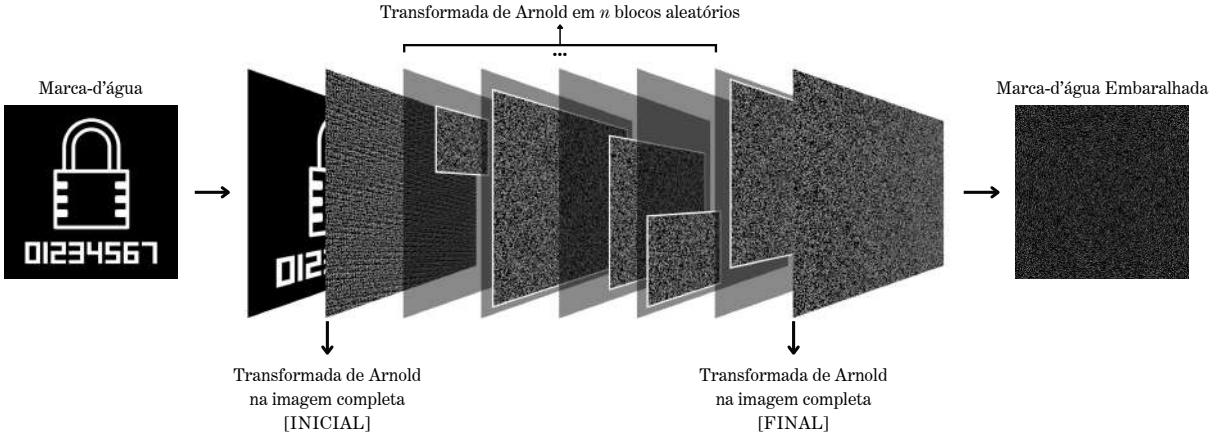


Figura 4.6: Transformada de Arnold por Bloco Iterativo - Iterative Block Arnold Transform (IBAT). (Fonte: Autoral).

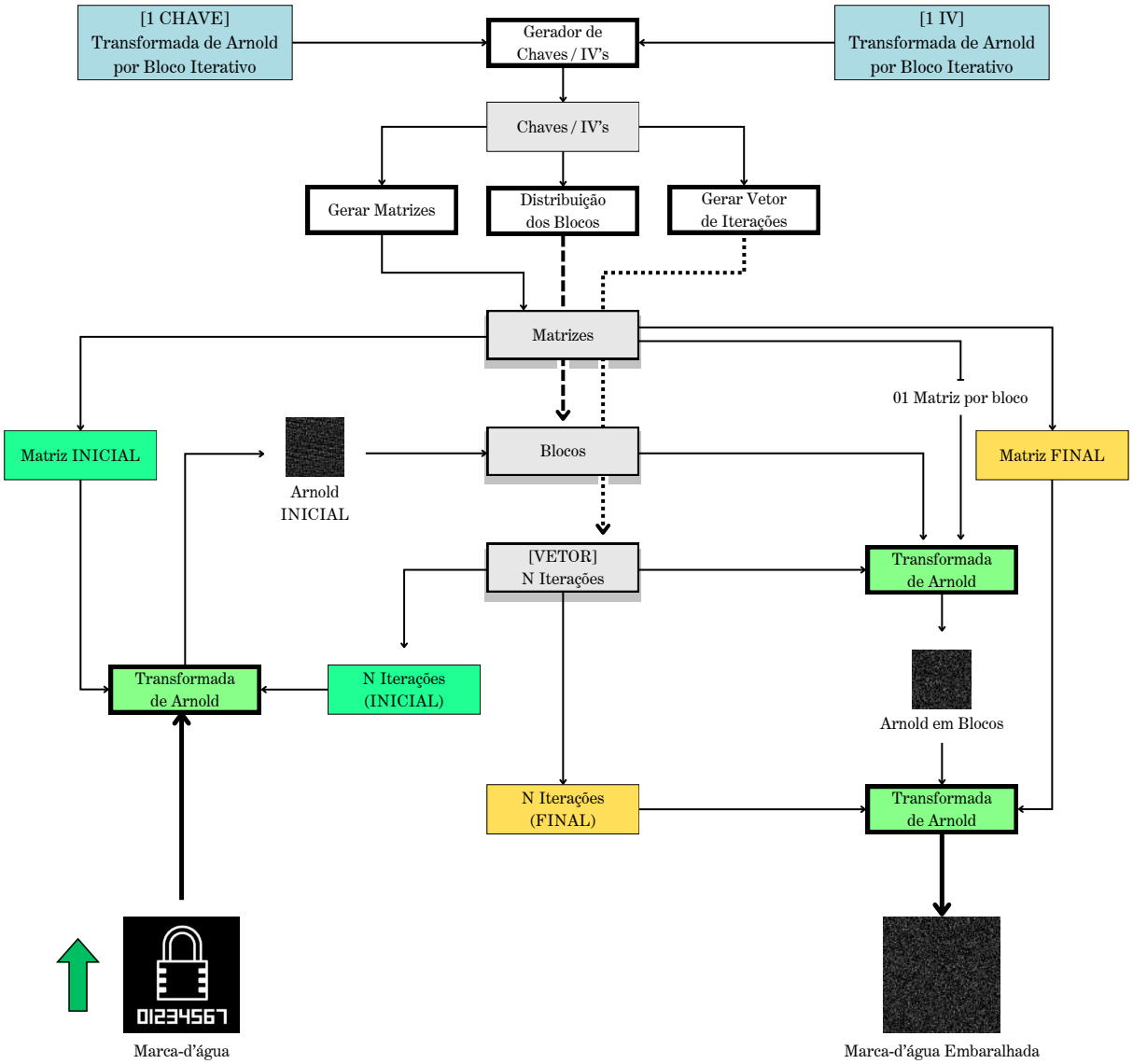


Figura 4.7: Esquema da IBAT. (Fonte: Autoral).

4.3.3 Sequência de Bits

Após a marca-d'água ter suas dimensões ajustadas de acordo com a imagem hospedeira e ser embaralhada pela IBAT, ela é transformada em uma sequência de bits. Em seguida, essa sequência de bits é concatenada com duas cópias redundantes, gerando uma sequência tripla ($\boldsymbol{sequência}_{\boldsymbol{tripla}}$) pronta para ser embutida na imagem, conforme ilustrado na Figura 4.8.

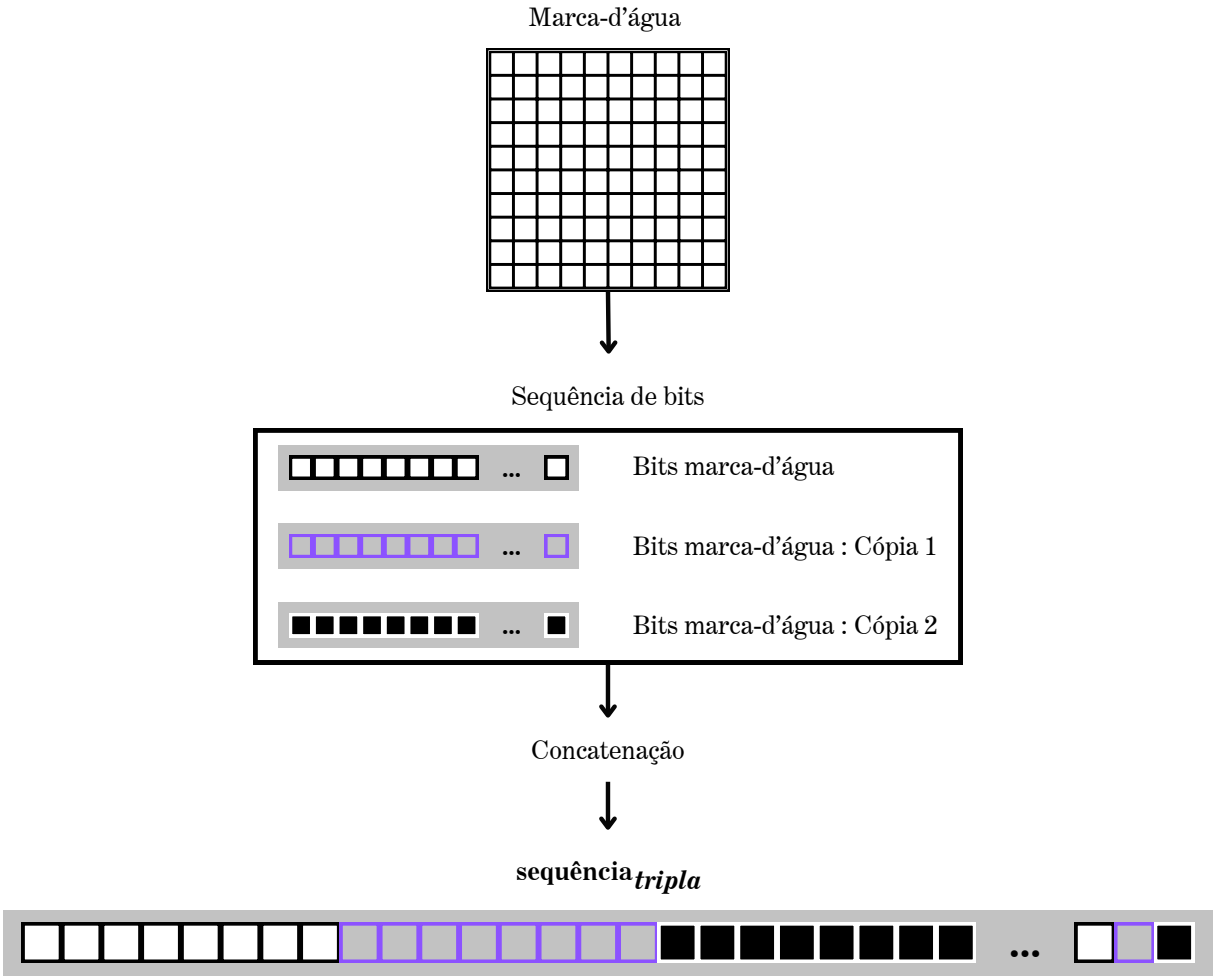


Figura 4.8: Concatenação dos dos bits da marca-d'água com suas cópias de redundância. (Fonte: Autoral).

4.4 Incorporação da Marca-d'água

Para inserir os bits da marca-d'água, a imagem hospedeira é transformada para o domínio da frequência por meio da QDFT ao longo do eixo quaterniônico $\mu = (i + j + k)/\sqrt{3}$, destacando-se a parte real para a incorporação dos bits. A parte real é selecionada por representar, nesse eixo, a intensidade luminosa total, que é uma estimativa robusta da luminância perceptiva que resiste melhor a operações de compressão e filtragem, enquanto as componentes imaginárias codificam crominância e fase, que são menos resistentes a distorções. A Figura 4.9 resume o esquema de incorporação da marca-d'água.

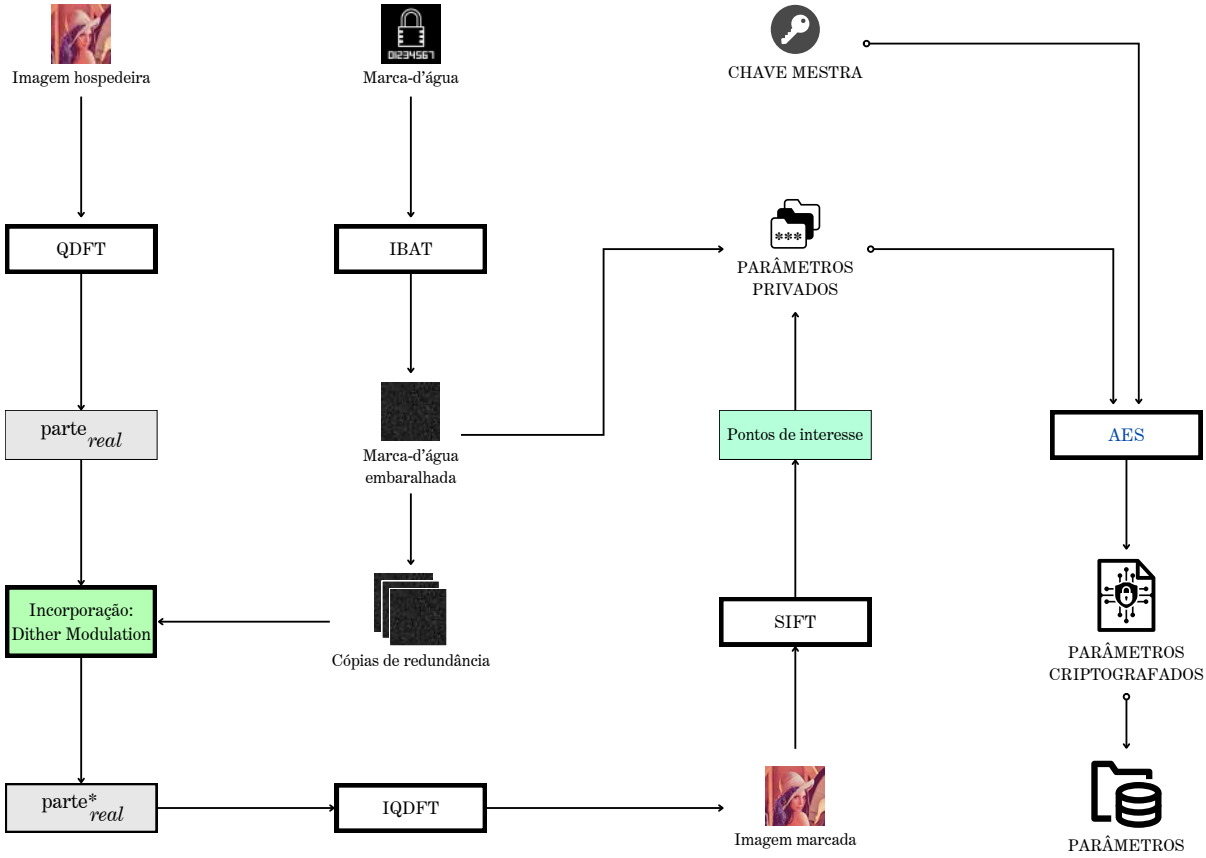


Figura 4.9: Esquema resumido de incorporação da marca-d'água. (Fonte: Autoral).

4.4.1 Máscaras de Incorporação

Uma máscara é definida por uma matriz lógica ou binária, onde cada elemento determina se uma operação deve ser aplicada ou não na coordenada correspondente. Dessa forma, quando o valor da máscara é verdadeiro (ou 1), a operação é realizada naquele ponto, e quando é falso (ou 0), nenhuma operação é realizada. Na prática, uma máscara funciona como uma imagem binária, composta por zeros e uns, que seleciona regiões específicas para processamento.

No contexto da incorporação dos bits da marca-d'água, se o valor da máscara for verdadeiro (ou 1), o bit será embutido, caso contrário, não haverá modificação.

Estrutura da Máscara

A estrutura da máscara de incorporação baseia-se em formas anelares com bordas definidas por ruído pseudoaleatório. A Figura 4.10 ilustra um exemplo de máscara anelar com essa característica.

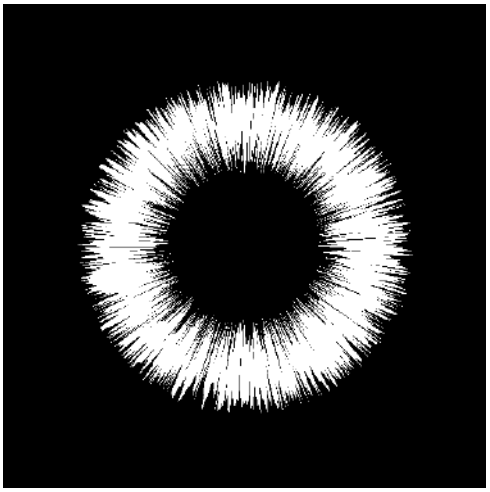


Figura 4.10: Máscara anelar com ruído pseudoaleatório. (Fonte: Autoral).

A espessura do anel é definida pelo raio da circunferência (ou elipse) formada, sendo um raio interno e outro externo, conforme a Figura 4.11.

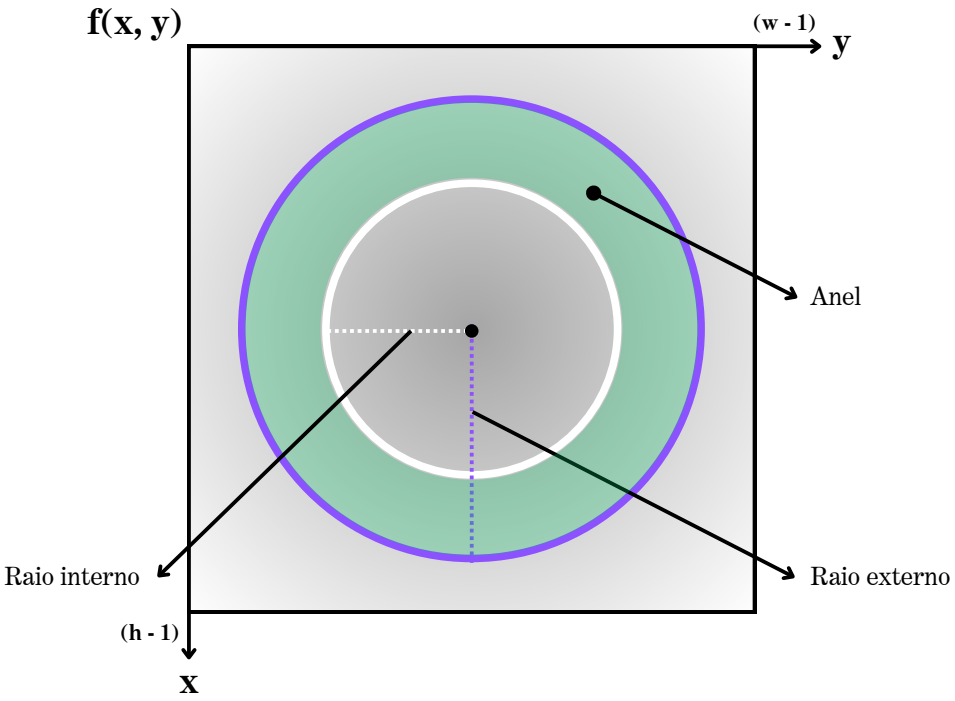


Figura 4.11: Estrutura base do anel. (Fonte: Autoral).

O raio é calculado de forma adaptativa com base em um valor percentual definido como parâmetro. Esse valor permite que o raio seja ajustado proporcionalmente a qualquer dimensão. Tanto o raio interno quanto o externo são calculados da mesma forma, sendo

que o raio interno deve, obrigatoriamente, ser menor que o raio externo. A Equação 4.10 apresenta a forma geral para converter o valor percentual no raio correspondente em pixels. Assim, o raio interno e o raio externo são definidos como $\boldsymbol{r_{min}}$ e $\boldsymbol{r_{max}}$, respectivamente. Diante disso,

$$\begin{aligned} r_{min} &= round\left(\left(\frac{min(h,w)\times \boldsymbol{porcentagem\_raio_{interno}}}{100}\right) \div 2\right), \\ r_{max} &= round\left(\left(\frac{min(h,w)\times \boldsymbol{porcentagem\_raio_{externo}}}{100}\right) \div 2\right), \end{aligned} \tag{4.10}$$

onde $\boldsymbol{h}$ e $\boldsymbol{w}$ são as dimensões de altura e largura da imagem hospedeira, respectivamente. A operação $\boldsymbol{min(.)}$ retorna o menor dentre os valores, e $\boldsymbol{round(.)}$ é uma função de arredondamento para o valor mais próximo.

Ao definir a espessura do anel, um ruído é adicionado às suas bordas internas e externas. O nível de ruído adicionado é definido de forma similar ao raio, através de um valor percentual. Além disso, o ruído de cada borda é gerado de forma pseudoaleatória utilizando um CSPRNG, onde cada borda, interna e externa, utiliza chaves e vetores de inicialização diferentes. Dessa maneira, o ruído é definido como:

$$\begin{aligned} ruido_{interno} &= \left|round\left(\frac{min(h,w)\times \boldsymbol{porcentagem\_ruido_{interno}}}{100}\right) \times RandnCSPRNG(key_1, iv_1)\right| \\ ruido_{externo} &= -\left|round\left(\frac{min(h,w)\times \boldsymbol{porcentagem\_ruido_{externo}}}{100}\right) \times RandnCSPRNG(key_2, iv_2)\right| \end{aligned} \tag{4.11}$$

onde $\boldsymbol{RandnCSPRNG(.)}$ é um CSPRNG com distribuição normal padrão.

Caso o nível de ruído não seja definido (ou seja, $porcentagem\_ruido_{interno/externo} = 0$), os de valores $\boldsymbol{r_{min}}$ e $\boldsymbol{r_{max}}$ serão constantes em todo ângulo θ no intervalo de $[0, 2\pi)$, o que implica que as bordas do anel não apresentarão nenhuma perturbação, conforme a Figura 4.12. Assim,

$$\begin{aligned} r(\theta)_{min} &= r_{min} + 0, \forall\ \theta \in [0, 2\pi), \\ r(\theta)_{max} &= r_{max} + 0, \forall\ \theta \in [0, 2\pi). \end{aligned} \tag{4.12}$$

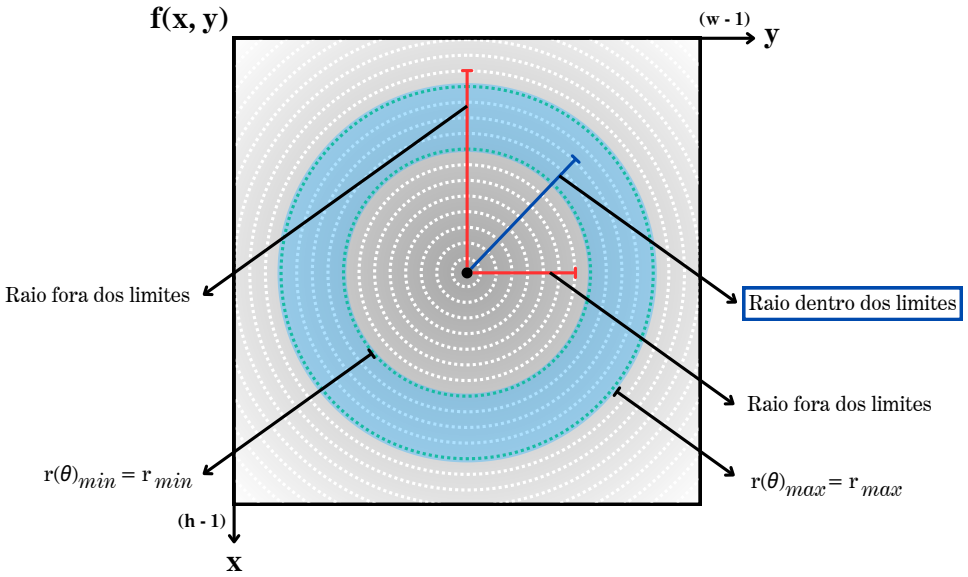


Figura 4.12: Estrutura anelar com $\boldsymbol{r_{min}}$ e $\boldsymbol{r_{max}}$ constantes. (Fonte: Autoral).

Quando o ruído for definido (ou seja, $porcentagem\_ruido_{interno/externo} \neq 0$), os valores de $\boldsymbol{r_{min}}$ e $\boldsymbol{r_{max}}$ em cada ângulo θ dentro do intervalo de $[0, 2\pi)$, terão um valor pseudoaleatório diferente. Logo,

$$\begin{aligned} r(\theta)_{min} &= r_{min} + ruido(\theta)_{interno}, \forall\ \theta \in [0, 2\pi), \\ r(\theta)_{max} &= r_{max} + ruido(\theta)_{externo}, \forall\ \theta \in [0, 2\pi). \end{aligned} \tag{4.13}$$

O efeito ruidoso originado por essas variações é demonstrado na Figura 4.10 e na Figura 4.13.

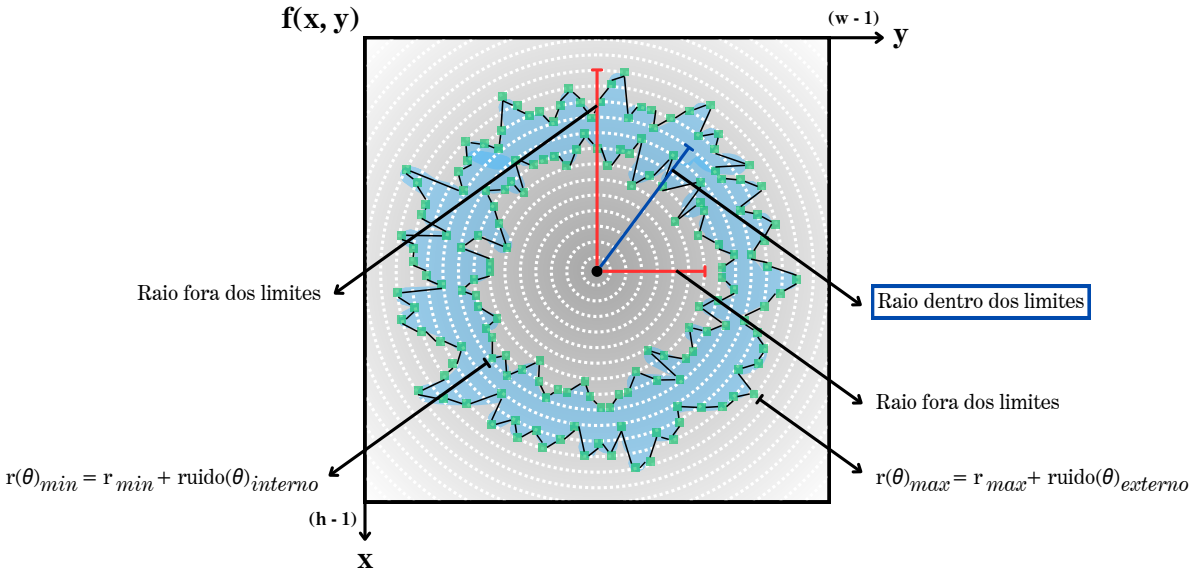


Figura 4.13: Estrutura anelar ruidosa. (Fonte: Autoral).

Máscara Principal

A partir da estrutura apresentada para a geração das máscaras anelares, é possível construir a máscara principal que servirá de base para a inserção dos bits da marca-d'água. Essa máscara é formada pela combinação de duas máscaras anelares, a máscara (a) e a máscara (b). A máscara (a) possui valores percentuais fixos para os raios interno e externo ($\boldsymbol{porcentagem\_raio_{interno}}$ e $\boldsymbol{porcentagem\_raio_{externo}}$, referidos na Equação 4.10), sendo 0% para o raio interno e 142% para o externo, conforme demonstrado pela Figura 4.14.

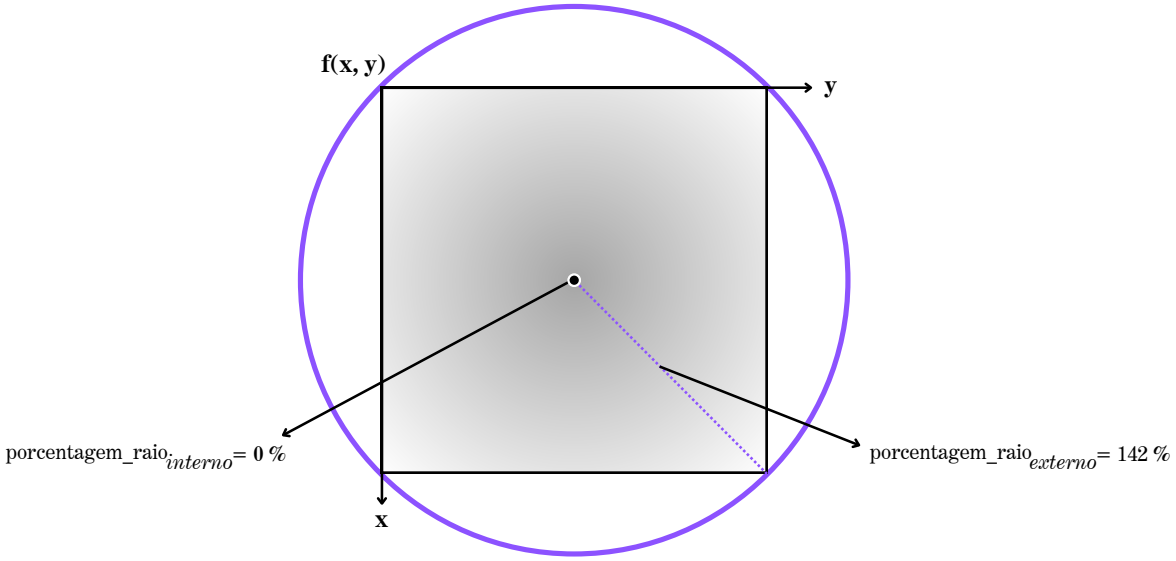


Figura 4.14: Máscara (a). (Fonte: Autoral).

O valor 142% é o percentual necessário para que o raio atinja o valor da diagonal, cobrindo toda a área disponível. A derivação desse valor parte do princípio do cálculo da diagonal de um quadrado, dado como

$$diagonal = l \times \sqrt{2}, \tag{4.14}$$

onde $\boldsymbol{l}$ é o lado do quadrado. Porém, para manter a proporção em qualquer dimensão, os valores são definidos em porcentagem, logo

$$100 \times \sqrt{2} \approx 142\%. \tag{4.15}$$

A máscara (b) é gerada com os valores percentuais do raio fixos em 50% e 100%, de modo que o anel possa tangenciar as bordas da máscara, conforme ilustrado na Figura 4.15.

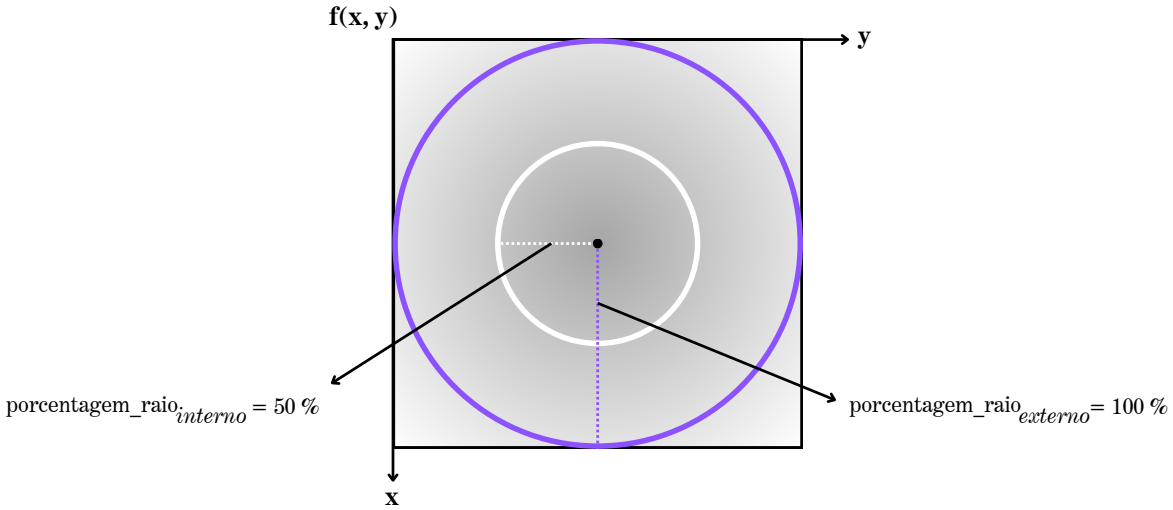


Figura 4.15: Máscara (b). (Fonte: Autoral).

Os níveis de ruído (***porcentagem_ruido***$_{interno}$ e ***porcentagem_ruido***$_{externo}$, referidos na Equação 4.11) são definidos da mesma maneira que o raio, utilizando valores percentuais. A máscara (a) tem seu ruído definido apenas para borda interna, logo o ruído interno é fixo em 10% e o ruído externo em 0%. Já na máscara (b), ambas as bordas possuem ruído adicionado, fixos em 10% para a borda interna e 12% para a borda externa. Além disso, as chaves requeridas pela Equação 4.11, e derivadas na subseção 4.1.4, são utilizadas para modular a aleatoriedade do ruído. A Figura 4.16 demonstra o esquema para gerar o ruído.

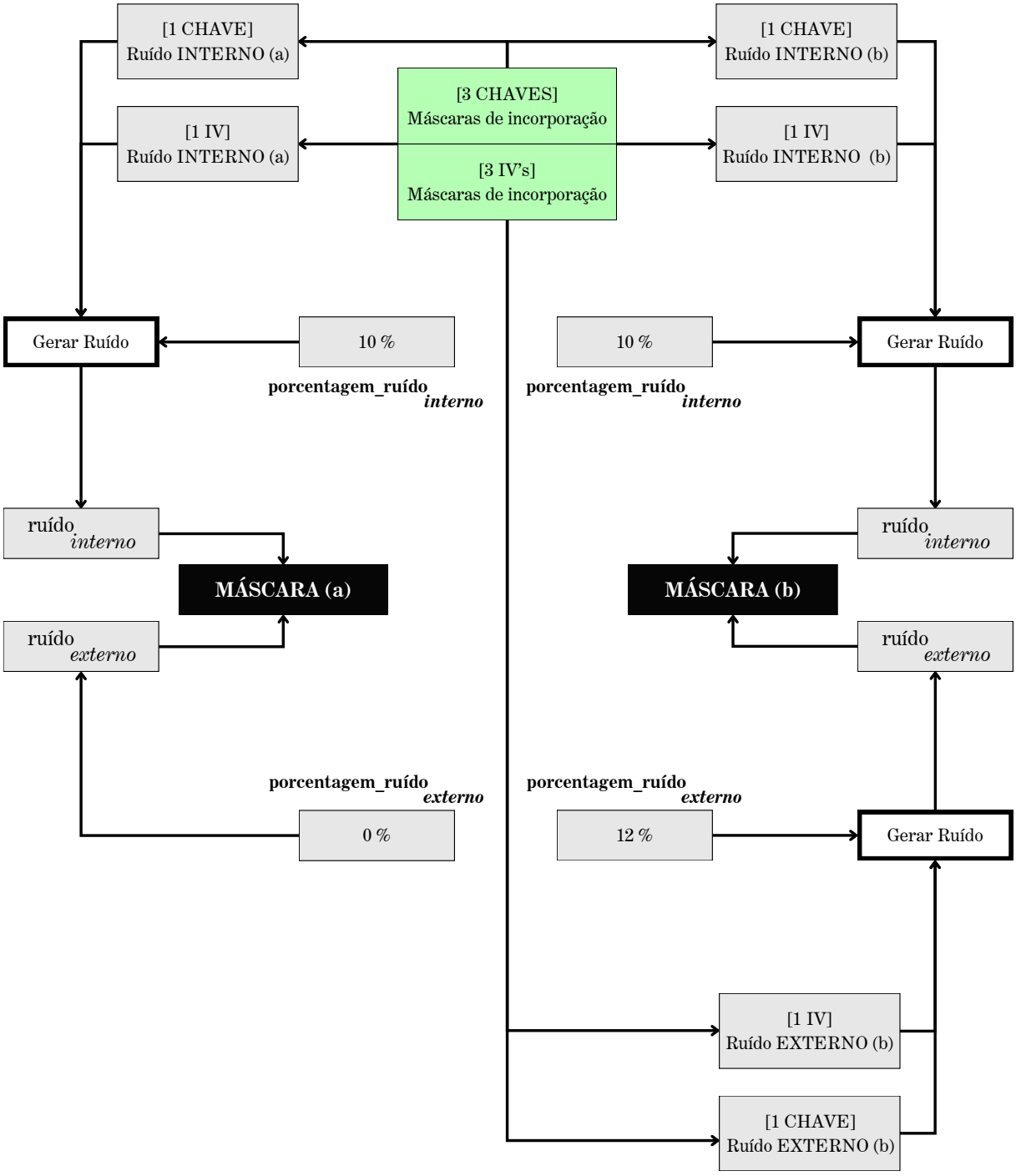
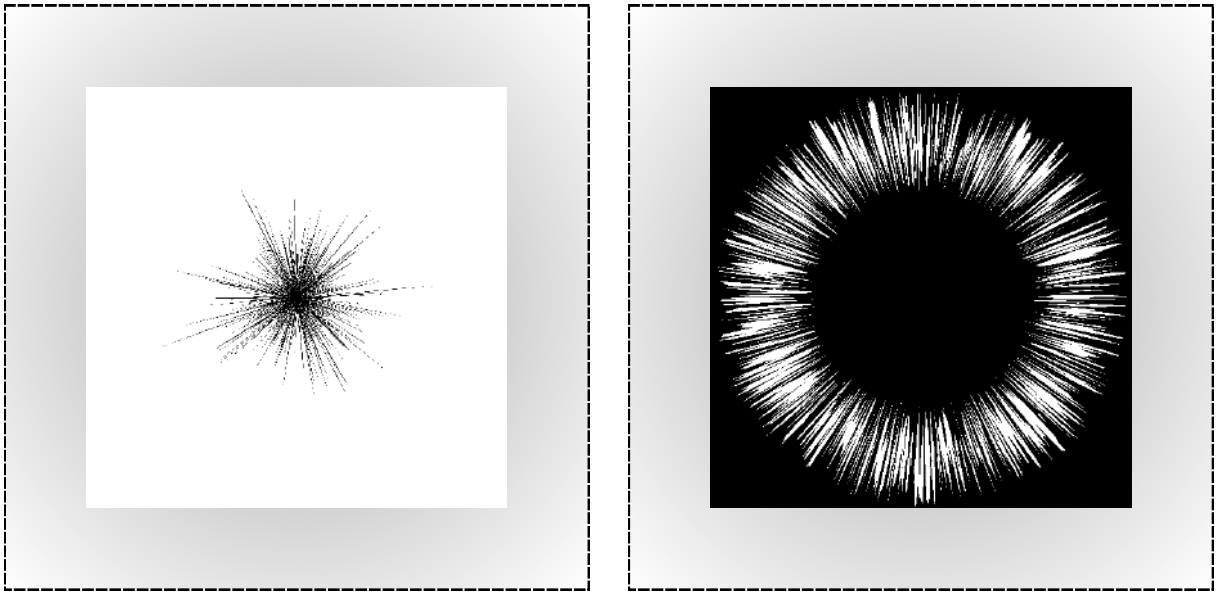


Figura 4.16: Esquema para gerar o ruído. (Fonte: Autoral).

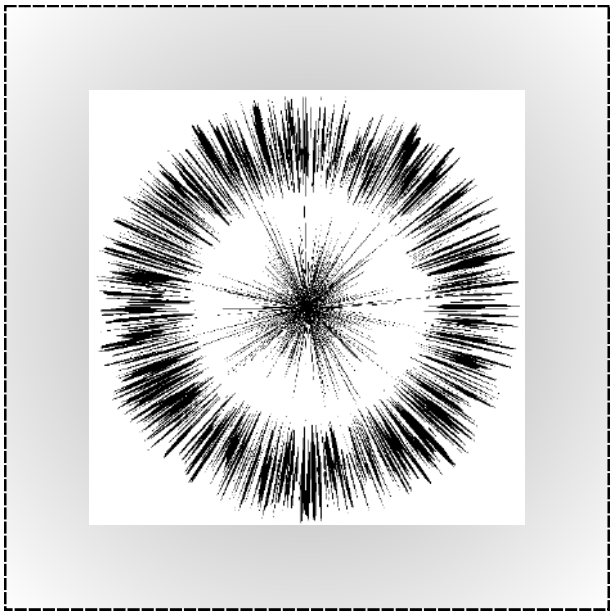
A Figura 4.17 exibe as máscaras (a) e (b), obtidas com base nos parâmetros especificados até este ponto do processo.



Máscara (a) Máscara (b)

Figura 4.17: Máscaras binárias (a) e (b). (Fonte: Autoral).

A máscara principal é obtida simplesmente subtraindo a máscara (b) da máscara (a). A Figura 4.18 ilustra a máscara principal resultante.



Máscara principal

Figura 4.18: Máscara binária principal. (Fonte: Autoral).

Conforme discutido anteriormente, as baixas frequências são as mais sensíveis a modificações, e além do centro do espectro, as faixas centrais concentram grande parte desses componentes de baixa frequência. Por esse motivo, a máscara principal é ajustada subtraindo as regiões correspondentes às faixas centrais da máscara, resultando em quatro quadrantes distintos. Dessa forma, ao sobrepor uma cruz binária que remove essas faixas centrais, as baixas frequências presentes nessas áreas não são alteradas, preservando a integridade das regiões de maior sensibilidade no domínio de frequência. A Figura 4.19 demonstra a máscara principal após o ajuste.

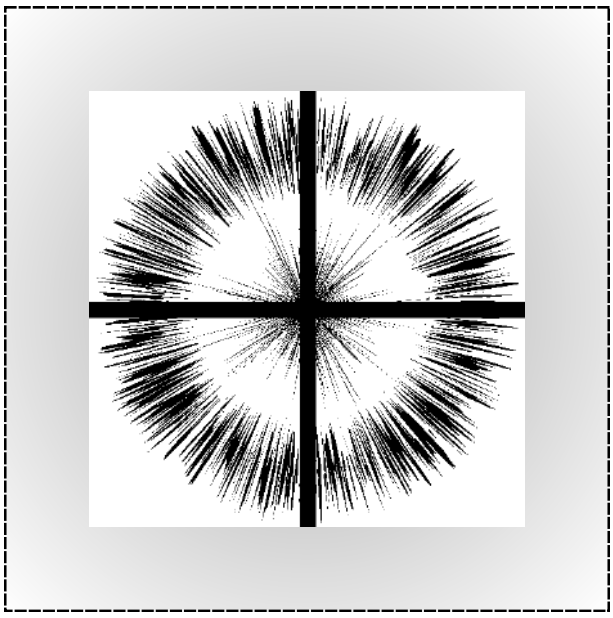


Figura 4.19: Máscara binária principal após a remoção das faixas centrais. (Fonte: Autoral).

A distribuição das coordenadas (tópico que será detalhado na próxima subseção (4.4.2)) é demonstrada pela Figura 4.20 que ilustra essa distribuição na região definida pela máscara principal e destaca os pontos a serem modificados no domínio de frequência.

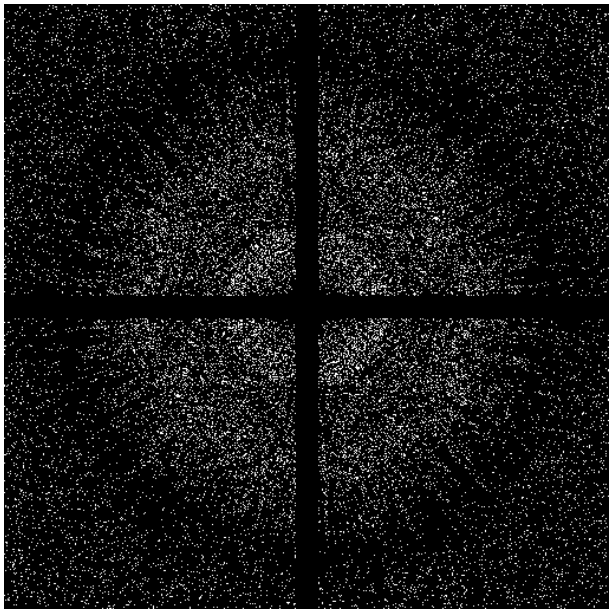
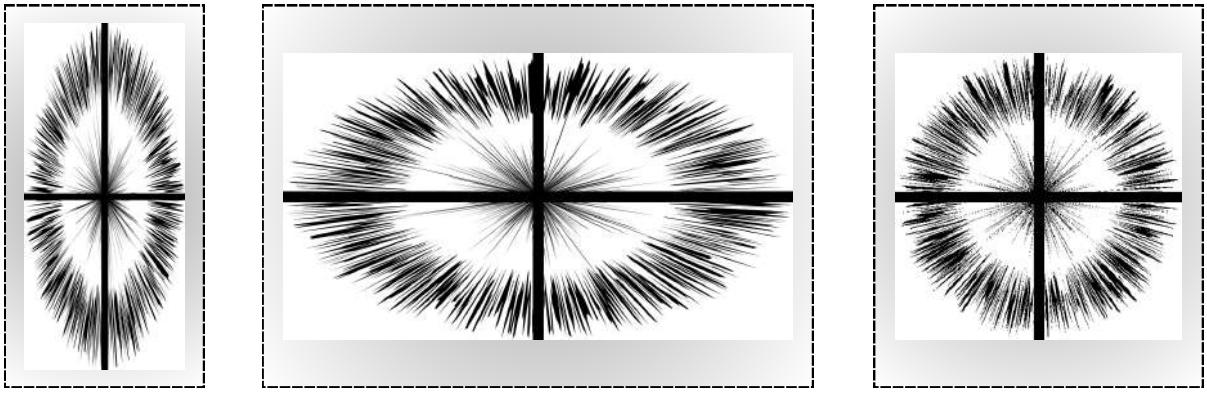


Figura 4.20: Distribuição das coordenadas de incorporação dentro da área delimitada pela máscara principal. (Fonte: Autoral).

Tabela 4.2: Resumo dos parâmetros. (Fonte: Autoral).

Parâmetro	**Máscara (a)**	**Máscara (b)**
$\boldsymbol{porcentagem\_raio_{interno}}$	0 %	50 %
$\boldsymbol{porcentagem\_raio_{externo}}$	142 %	100 %
$\boldsymbol{porcentagem\_ruido_{interno}}$	10 %	10 %
$\boldsymbol{porcentagem\_ruido_{externo}}$	0 %	12 %

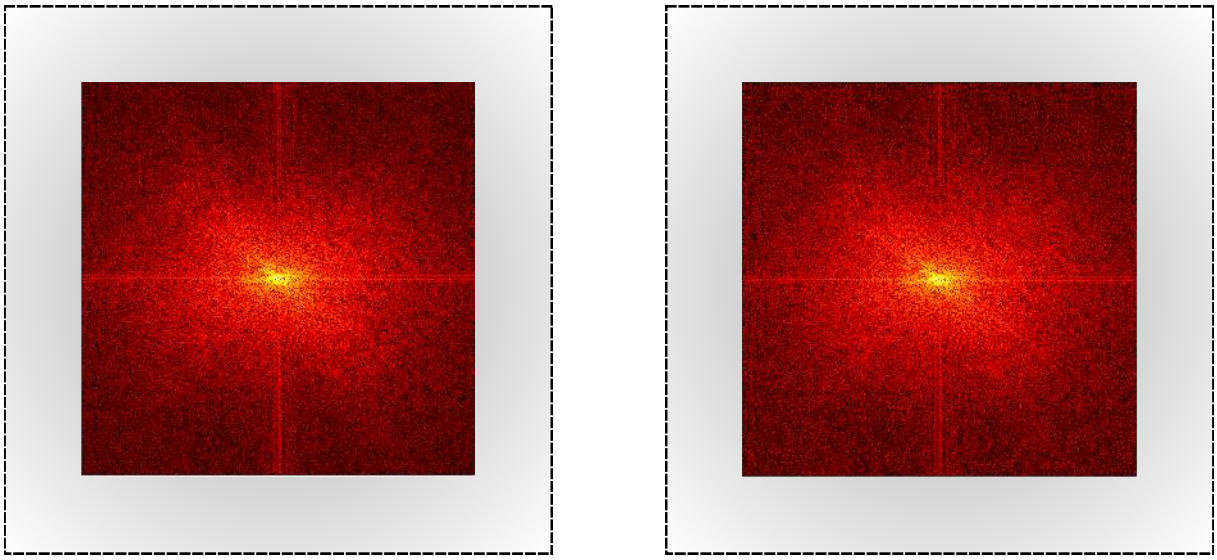
A Tabela 4.2 resume os parâmetros utilizados na geração das máscaras (a) e (b) que compõem a máscara principal. Esses parâmetros são fixos em todos os casos e definem a região de incorporação. Entretanto, o ruído é gerado de forma pseudoaleatória a partir das chaves fornecidas ao CSPRNG. Dessa forma, embora a estrutura básica da máscara permaneça a mesma, cada instância será única quando chaves diferentes forem utilizadas. A Figura 4.21 ilustra alguns exemplos de máscaras em diferentes dimensões.



(a) (b) (c)

Figura 4.21: **(a)** 4000 × 1844, **(b)** 2160 × 3840, **(c)** 512 × 512. (Fonte: Autoral).

A escolha desses parâmetros foi feita de modo a integrar de forma natural a região de inserção ao espectro de frequência, simulando a distribuição dos coeficientes de frequência e minimizando evidências de modificações, conforme demonstrado pela Figura 4.22. Essa figura ilustra o espectro modificado a partir das coordenadas distribuídas na região definida pela máscara principal.



Espectro original Espectro modificado

Figura 4.22: Exemplos de espectros de frequência da parte real da QDFT. (Fonte: Autoral).

Segmentação da Máscara Principal

Conforme introduzido em 4.2, duas cópias adicionais dos bits da marca-d'água são inseridas como redundância, totalizando uma inserção tripla. Dessa forma, cada faixa de frequência recebe uma cópia da marca-d'água, assim, a máscara principal é dividida em três faixas correspondentes às frequências baixa, média e alta, cada uma destinada à inserção de uma cópia dos bits. Além disso, a faixa de baixa frequência é subdividida em duas regiões, permitindo ajustar a intensidade das modificações nessa banda mais sensível.

Para efetuar a divisão, o mesmo método de geração das máscaras é utilizado, porém sem a inserção de ruído (ou seja, $porcentagem\_ruido_{interno/externo} = 0$). Dessa forma, cada faixa pode ser delimitada com precisão em sua respectiva área. A Tabela 4.3 exemplifica os parâmetros utilizados para gerar as faixas.

Tabela 4.3: Parâmetros utilizados na segmentação da máscara principal em faixas de frequência. (Fonte: Autoral).

Parâmetros	$\boldsymbol{Subfaixa_1}$ **Baixa**	$\boldsymbol{Subfaixa_2}$ **Baixa**	**Faixa Média**	**Faixa Alta**
$\boldsymbol{porcentagem\_raio_{interno}}$	0 %	27 %	50 %	76 %
$\boldsymbol{porcentagem\_raio_{externo}}$	27 %	50 %	76 %	142 %

Embora os parâmetros que definem as fronteiras entre as faixas sejam idênticos, não ocorre sobreposição em razão do arredondamento aplicado durante a conversão para pixels. A Figura 4.23 ilustra as faixas segmentadas a partir da máscara principal.

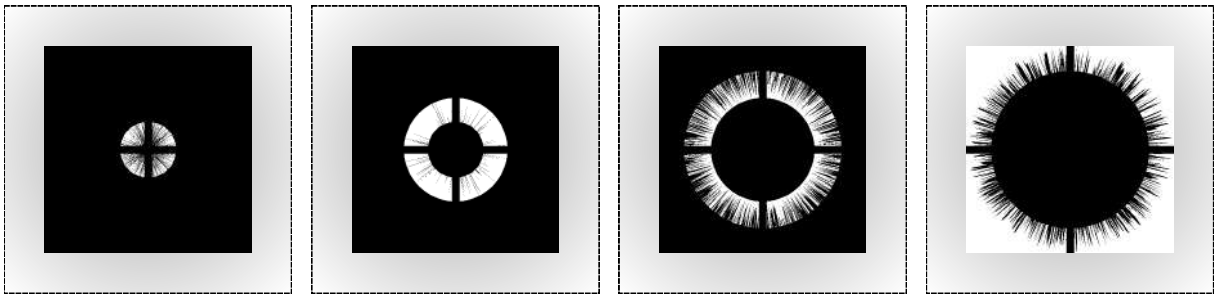


Figura 4.23: Máscaras que delimitam a área de inserção de cada faixa de frequência. (Fonte: Autoral).

4.4.2 Definição das Coordenadas

A máscara principal e suas subdivisões constituem a base para determinar as coordenadas de incorporação, auxiliando na definição das coordenadas e suas correspondentes simétricas. Além disso, cada faixa é associada a um fator de ponderação ($\boldsymbol{f_{\Delta}}$) que ajusta o valor de $\boldsymbol{\Delta}$ (passo de quantização do DM), de acordo com sua faixa de frequência.

Coordenadas Simétricas

Antes de gerar as coordenadas de incorporação, é fundamental compreender que as modificações nos coeficientes devem respeitar a lógica de simetria da parte real da QDFT, onde para cada coeficiente em uma coordenada, existe um coeficiente correspondente em uma coordenada simétrica que deve ser igualmente modificado, conforme ilustrado na Figura 4.24.

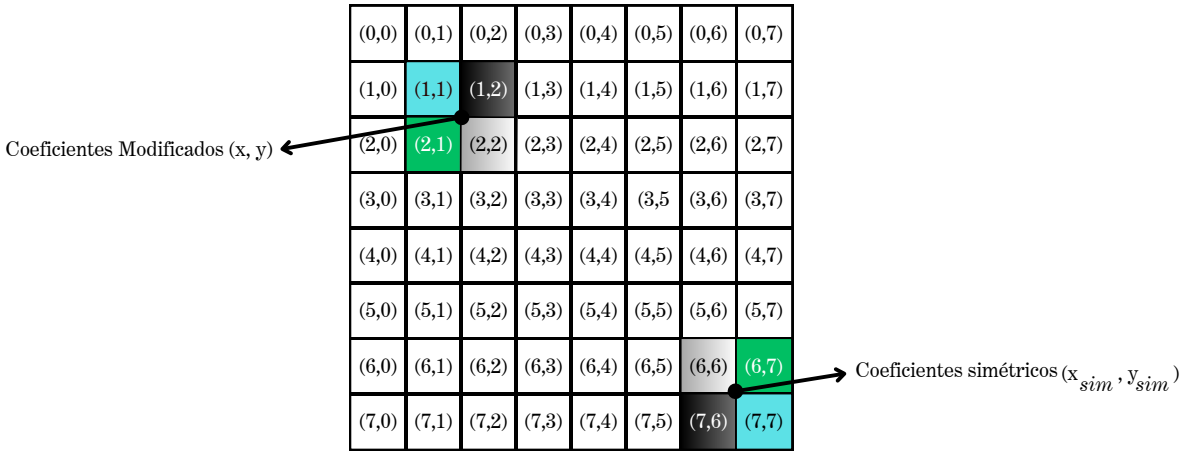


Figura 4.24: Exemplo de modificação simétrica. Figura Modificada (Fonte: [28]).

Para calcular as coordenadas simétricas, é necessário verificar separadamente se as dimensões da imagem (altura e largura) são pares ou ímpares. Se for par, a coordenada é subtraída de sua respectiva dimensão, e a operação ***mod*** é aplicada para controlar o intervalo. Caso seja ímpar, além de subtrair a coordenada, é preciso subtrair o valor 1 para ajustar o centro simétrico corretamente. Esse processo é detalhado pelo Algoritmo 2, onde $\boldsymbol{h}$ e $\boldsymbol{w}$ são as dimensões de altura e largura, respectivamente, e $\boldsymbol{x}$ e $\boldsymbol{y}$ são as coordenadas de inserção. As coordenadas simétricas são definidas como $\boldsymbol{x_{sim}}$ e $\boldsymbol{y_{sim}}$.

Algoritmo 2 Coordenadas simétricas

if $h = par$ **then**
  $x_{sim} \leftarrow (h - x) \mod h$
else
  $x_{sim} \leftarrow ((h - x) - 1) \mod h$
end if

if $w = par$ **then**
  $y_{sim} \leftarrow (w - y) \mod w$
else
  $y_{sim} \leftarrow ((w - y) - 1) \mod w$
end if

Coordenadas de Incorporação

Para todas as coordenadas ($\boldsymbol{x}$, $\boldsymbol{y}$) na máscara binária principal cujo valor seja 1, calcula-se a posição simétrica conforme o Algoritmo 2. Em seguida, cada par ($\boldsymbol{x}$, $\boldsymbol{y}$) é armazenado junto com sua correspondente simétrica ($\boldsymbol{x_{sim}}$, $\boldsymbol{y_{sim}}$) em uma matriz de coordenadas ($\boldsymbol{matriz_{coordenadas}}$ [$\boldsymbol{n} \times \mathbf{4}$]). A Figura 4.25 ilustra o processo.

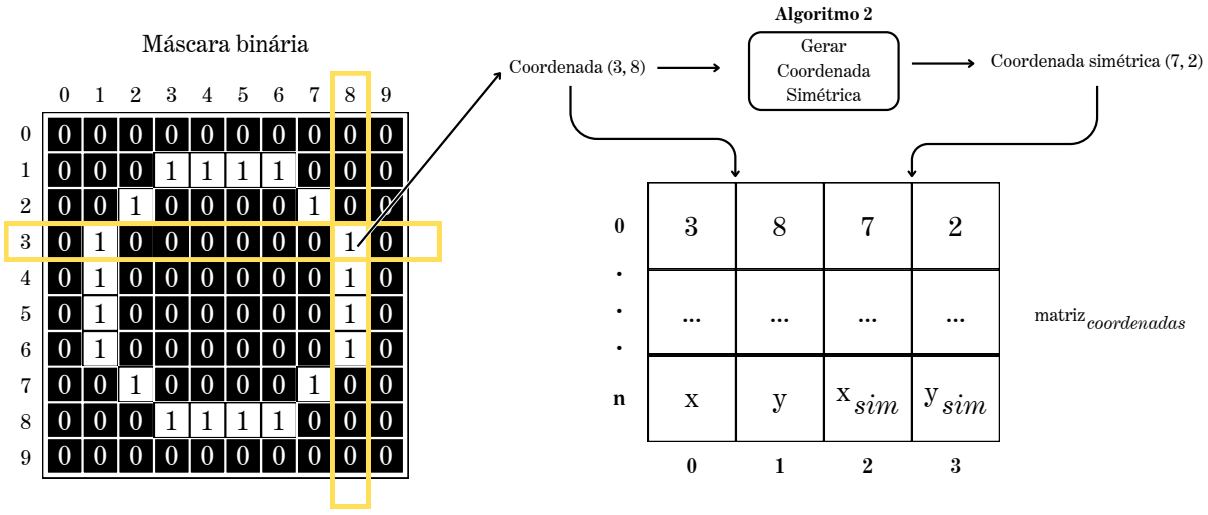


Figura 4.25: Processo de geração das coordenadas a partir da máscara binária. (Fonte: Autoral).

Como as coordenadas são geradas sequencialmente a partir da máscara principal, é necessário permutá-las para assegurar a dispersão máxima dos bits da marca-d'água. Dessa forma, a função $\boldsymbol{RandPermCSPRNG(.)}$ recebe como entrada a chave e o IV

gerados na subseção 4.1.4 para definição das coordenadas, e produz um vetor de índices pseudoaleatórios que são utilizados na permutação das linhas da matriz de coordenadas. Assim,

$$indices_{rand} = RandPermCSPRNG(n_{linhas},\ key_{coords},\ iv_{coords}), \tag{4.16}$$

onde $\boldsymbol{n_{linhas}}$ é o número de linhas da matriz de coordenadas ($\boldsymbol{matriz_{coordenadas}}$).

Após o embaralhamento, as coordenadas presentes em $\boldsymbol{matriz_{coordenadas}}$ são distribuídas entre as diferentes faixas de frequência com base no tamanho da sequência de bits da marca-d'água.

A sequência de bits ($\boldsymbol{sequência_{tripla}}$), abordada em 4.3.3, é composta por três subsequências correspondentes aos bits da marca-d'água, das quais duas são cópias de redundância. Cada subsequência de $\boldsymbol{sequência_{tripla}}$ possui $\boldsymbol{n_{bits}}$, valor que determina a quantidade de coordenadas atribuídas a cada faixa de frequência. Na faixa de baixa frequência, uma das subsequências é particionada em três partes, sendo uma delas alocada à $subfaixa_1$, enquanto as duas restantes são destinadas à $subfaixa_2$. Logo, a quantidade de coordenadas alocadas em cada faixa é definida como

$$\begin{aligned} nbits_{fxbaixa} &= \begin{cases} subfaixa_1 = round\left(\frac{n_{bits}}{3}\right) \\ subfaixa_2 = n_{bits} - subfaixa_1 \end{cases}, \\ nbits_{fxmedia} &= n_{bits}, \\ nbits_{fxalta} &= n_{bits}. \end{aligned} \tag{4.17}$$

Para alocar as coordenadas nas faixas, a matriz de coordenadas ($\boldsymbol{matriz_{coordenadas}}$) é percorrida iterativamente linha por linha, analisando cada coordenada ($\boldsymbol{x}$, $\boldsymbol{y}$) e sua correspondente simétrica ($\boldsymbol{x_{sim}}$, $\boldsymbol{y_{sim}}$). Inicialmente, verifica-se se a coordenada atual ($\boldsymbol{x}$, $\boldsymbol{y}$) não é uma coordenada simétrica de si mesma ou de outra coordenada já processada anteriormente. Após essa verificação, determina-se a qual faixa a coordenada pertence, com base nas regiões segmentadas (Figura 4.23) da máscara principal. Simultaneamente, a matriz de coordenadas ($\boldsymbol{matriz_{coordenadas}}$) é reestruturada com a adição de duas colunas, uma para o fator de ponderação ($\boldsymbol{f_{\Delta}}$) e outra para o identificador da faixa de frequência.

O fator de ponderação ($\boldsymbol{f_{\Delta}}$) ajusta o valor de $\boldsymbol{\Delta}$ com o objetivo de atenuar as modificações dos coeficientes de frequência em regiões mais sensíveis. Conforme a Tabela 4.4, cada faixa possui um $\boldsymbol{f_{\Delta}}$ empiricamente ajustado para equilibrar robustez e imperceptibilidade.

Tabela 4.4: Valores para o fator de ponderação (f_{Δ}) conforme as faixas de frequência. (Fonte: Autoral).

Id	Faixas	f_{Δ}
0	Baixa frequência : $subfaixa_1$	0.68
1	Baixa frequência : $subfaixa_2$	0.75
2	Média frequência	1.00
3	Alta frequência	1.00

O procedimento é repetido até que cada faixa seja preenchida com o número de coordenadas correspondente à quantidade de bits da marca-d'água, conforme especificado na Equação 4.17. Concluída a distribuição, a matriz de coordenadas ($\boldsymbol{matriz_{coordenadas}}$) é então ordenada de acordo com o identificador de faixa. A Figura 4.26 ilustra o processo.

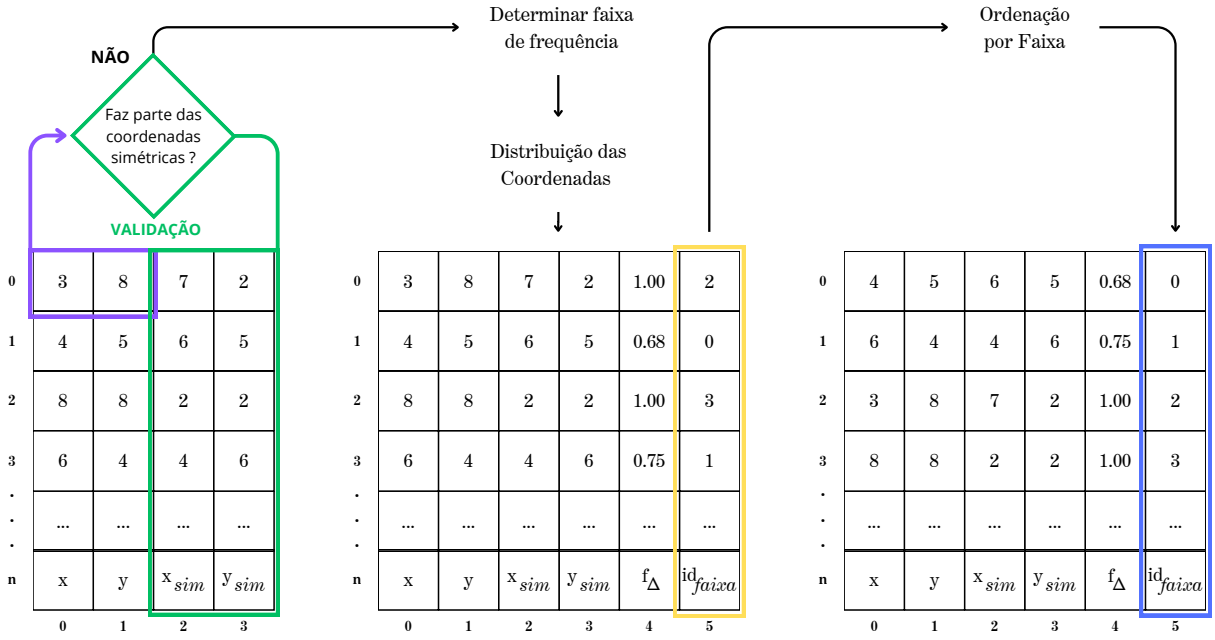


Figura 4.26: Distribuição das coordenadas. (Fonte: Autoral).

A validação apresentada na Figura 4.26 garante que cada coordenada ($\boldsymbol{x}$, $\boldsymbol{y}$) não se sobreponha a nenhuma de suas simétricas ($\boldsymbol{x_{sim}}$, $\boldsymbol{y_{sim}}$), evitando conflitos na aplicação das modificações.

Para efeitos de ilustração, a Figura 4.27 exibe as coordenadas resultantes após o processo de distribuição. Os pontos amarelos indicam as posições ($\boldsymbol{x}$, $\boldsymbol{y}$) enquanto os pontos azuis correspondem às suas coordenadas simétricas ($\boldsymbol{x_{sim}}$, $\boldsymbol{y_{sim}}$).

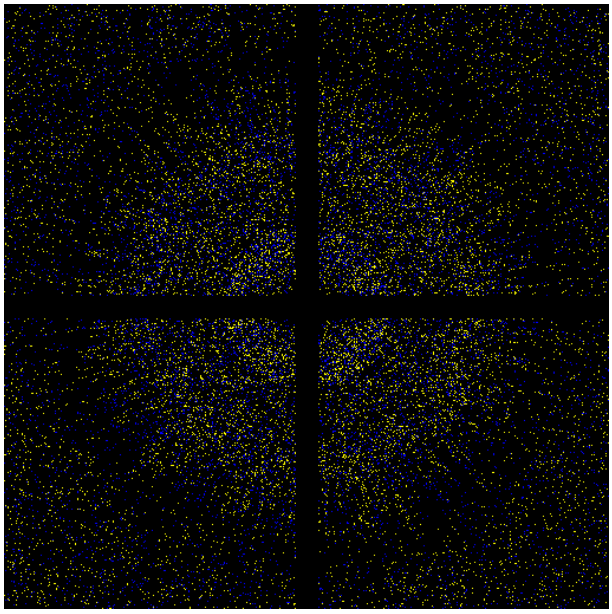


Figura 4.27: Representação visual da distribuição das coordenadas. (Fonte: Autoral).

4.4.3 Incorporação

A matriz de coordenadas ($\boldsymbol{matriz_{coordenadas}}$) é utilizada para gerar um vetor coluna contendo os valores de Δ ponderados de acordo com a faixa correspondente. Para isso, a coluna da matriz que contém os pesos f_Δ é multiplicada pelo valor de Δ, resultando no vetor coluna $\boldsymbol{\Delta_{efetivo}}$, conforme ilustrado na Figura 4.28.

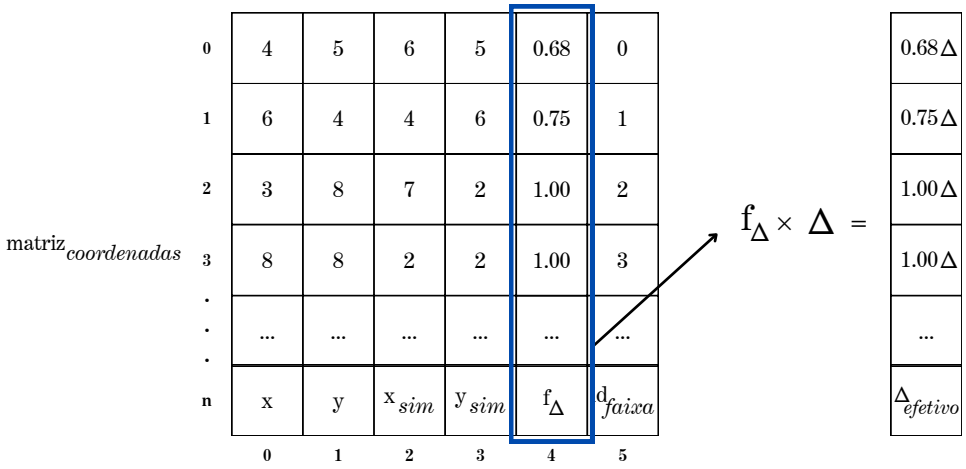


Figura 4.28: Vetor coluna $\Delta_{efetivo}$ com os valores de Δ ponderados. (Fonte: Autoral).

Conforme discutido na subseção 2.8.2, o método de quantização DM modula os coeficientes por meio da adição de um ruído artificial denominado *Dither*. Para aumentar

a robustez e minimizar o impacto perceptual da quantização, cada bit da marca é associado a um valor pseudoaleatório distinto de *Dither*, gerado dentro de um intervalo proporcional ao $\Delta_{efetivo}$. Assim,

$$\begin{aligned} d_0 &= randvCSPRNG(\ \tfrac{\Delta_{efetivo}}{5},\quad \tfrac{\Delta_{efetivo}}{4},\ key_{1_{dither}},\ iv_{1_{dither}}), \\ d_1 &= randvCSPRNG(-\tfrac{\Delta_{efetivo}}{4},\ -\tfrac{\Delta_{efetivo}}{5},\ key_{2_{dither}},\ iv_{2_{dither}}), \end{aligned} \tag{4.18}$$

onde $\boldsymbol{d_0}$ e $\boldsymbol{d_1}$ são vetores que contêm os valores de *Dither* associados aos bits 0 e 1, respectivamente. A função $randvCSPRNG(.)$ gera vetores de valores fracionários pseudoaleatórios dentro do intervalo definido e com base nas chaves geradas na subseção 4.1.4 para o DM. Na prática, o DM é aplicado da seguinte forma:

$$coef^* = \begin{cases} round((coef + d0)/\Delta_{efetivo}) \times \Delta_{efetivo} - d0, & if\ (bit == 0) \\ round((coef + d1)/\Delta_{efetivo}) \times \Delta_{efetivo} - d1, & if\ (bit == 1) \end{cases} \tag{4.19}$$

onde $\boldsymbol{coef}$ é o coeficiente a ser modificado pelo bit da marca-d'água.

O processo de incorporação consiste em alterar os coeficientes de frequência da parte real da QDFT conforme os bits da marca-d'água, por meio do método de quantização DM. A parte real é obtida ao transformar a imagem hospedeira para o domínio da frequência por meio da QDFT, logo:

$$F_{QDFT} = QDFT(imagem_{hospedeira},\ \mu), \tag{4.20}$$

onde $\boldsymbol{F_{QDFT}}$ denota a matriz quaterniónica de coeficientes resultante da transformação, e $\boldsymbol{\mu}$ é o quatérnio puro, definido como:

$$\mu = \tfrac{(i+j+k)}{\sqrt{3}}. \tag{4.21}$$

Após a QDFT, realiza-se um deslocamento de quadrantes na matriz F_{QDFT} para centralizar o coeficiente de frequência zero (DC) e as componentes de baixa frequência, ao mesmo tempo em que as altas frequências são redistribuídas para as bordas da matriz. Dessa forma,

$$F_{QDFT_{shift}}(u, v) = F_{QDFT}((u + \tfrac{h}{2})\ mod\ h,\quad (v + \tfrac{w}{2})\ mod\ w), \tag{4.22}$$

onde $\boldsymbol{u}$ varia de $\mathbf{0}$ a $\boldsymbol{h - 1}$ e $\boldsymbol{v}$ de $\mathbf{0}$ a $\boldsymbol{w - 1}$. Para obter a parte real, decompõe-se a matriz quaterninônica em suas quatro componentes, logo:

$$(parte_{real},\ parte_i,\ parte_j,\ parte_k) = parts(F_{QDFT_{shift}}). \tag{4.23}$$

Considerando o número de coordenadas definidas em $\boldsymbol{matriz_{coordenadas}}$, que corresponde exatamente à quantidade de bits na sequência ($\boldsymbol{sequência_{tripla}}$), o processo de inserção se resume em percorrer a matriz de coordenadas linha por linha, modificando diretamente cada coeficiente e seu correspondente simétrico, conforme o bit associado. A fim de preservar a coerência espectral na parte real, cada coeficiente alterado deve ter seu correspondente simétrico igualmente modificado, porém com sinal invertido. O Algoritmo 3 detalha o processo, onde $ditherModulation(.)$ é a aplicação da Equação 4.19.

Para manter a uniformidade e obedecer à notação padrão no domínio da frequência, adota-se o par de índices (u, v) para designar as coordenadas (x, y) em $matriz_{coordenadas}$.

Algoritmo 3 Incorporação

for $i = 1$ to n_{linhas} **do**

$u,\ v,\ u_{sim},\ v_{sim} \leftarrow matriz_{coordenadas}(i)$

$bit \leftarrow sequência_{tripla}(i)$

$coef \leftarrow parte_{real}(u, v)$

$coef_{modificado} \leftarrow ditherModulation(coef,\ bit,\ \Delta_{efetivo}(i),\ d_0(i),\ d_1(i))$

$parte^*_{real}(u,\ v) \leftarrow coef_{modificado}$

$parte^*_{real}(u_{sim},\ v_{sim}) \leftarrow -coef_{modificado}$

end for

Finalizada a inserção de todos os bits, a parte real da QDFT é atualizada, sendo substituída por sua versão modificada ($\boldsymbol{parte^*_{real}}$), logo,

$$F^*_{QDFT_{shift}} = quaternion(parte^*_{real},\ parte_i,\ parte_j,\ parte_k). \tag{4.24}$$

Concluída a atualização da parte real e sua reinserção na matriz quaterniônica, as frequências são reorganizadas para sua disposição original por meio da inversa da função $fftshift$. Em seguida, aplica-se a transformada inversa da QDFT para a reconstrução da imagem hospedeira. Assim, obtém-se:

$$imagem_{marcada} = IQDFT(shift(F^*_{QDFT_{shift}}),\ \mu), \tag{4.25}$$

onde $\boldsymbol{shift}$ refere-se a aplicação da Equação 4.22. A Figura 4.29 demonstra a imagem marcada resultante ao fim do processo.

Imagem Original Imagem Marcada

Figura 4.29: Imagem marcada (512 x 512). (Fonte: Autoral).

4.4.4 Detecção de Características

A precisão na extração da marca-d'água depende diretamente da disposição espacial da imagem marcada, logo, para que o processo de extração ocorra corretamente a imagem deve manter a mesma configuração espacial existente no momento da inserção da marca. Logo, qualquer tipo de processamento que mude essa disposição (como rotações, redimensionamentos ou outras transformações geométricas) deve ser revertido, de modo a garantir a correta recuperação dos bits da marca-d'água.

Diante dessa dependência, pontos de interesse são extraídos da imagem marcada após o fim do processo de incorporação. Esses pontos servem como referência para que a imagem possa ser realinhada à sua disposição espacial original. A extração desses pontos é feita pelo algoritmo SIFT, que identifica automaticamente locais de elevada informação na imagem, reconhecendo pontos onde ocorrem variações acentuadas de intensidade em múltiplas direções (como cantos e interseções de bordas), resultando em pontos de referência precisos para o processo de alinhamento geométrico.

Pre-processamento

Para aumentar a precisão na detecção dos pontos de interesse durante a aplicação do SIFT, a imagem marcada passa por um pré-processamento com o objetivo de reforçar as variações de intensidade e atenuar componentes indesejados, como ruídos ou variações artificiais de iluminação, resultando na detecção de pontos mais robustos e estáveis. Como o SIFT opera exclusivamente em imagens em nível de cinza, é necessário converter a

imagem marcada de RGB para escala de cinza, garantindo que o processo de detecção ocorra de forma adequada. A Figura 4.30 resume as etapas do processo.

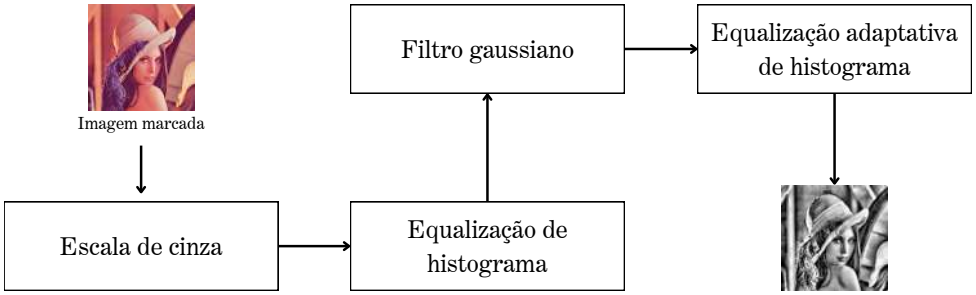


Figura 4.30: Etapas do pré-processamento. (Fonte: Autoral).

Regiões muito escuras ou muito claras podem dificultar a detecção de variações locais pelo SIFT, assim, a imagem marcada passa por uma equalização de histograma, que redistribui os níveis de intensidade ao longo de toda a faixa dinâmica disponível, aumentando o contraste e evidenciando detalhes em áreas que anteriormente eram pouco perceptíveis.

Ao realçar globalmente o contraste com a equalização de histograma, ruídos de alta frequência podem surgir, resultando em pontos de interesse instáveis e potencialmente prejudiciais ao desempenho do detector. Dessa forma, aplica-se um filtro gaussiano para atenuar essas flutuações, preservando os contornos mais relevantes e tornando a detecção do SIFT mais precisa e direcionada a características consistentes. O filtro Gaussiano é aplicado com desvio padrão $\sigma = 2$ e tamanho de kernel ajustado automaticamente.

Uma equalização local é realizada por meio da equalização adaptativa de histograma, a fim de melhorar o contraste em regiões que, apesar da equalização global, permaneceram com baixo contraste local. A equalização adaptativa divide a imagem em pequenos blocos e ajusta o contraste de cada região individualmente, evitando a amplificação excessiva de ruídos. Esse realce local destaca pequenas variações de intensidade mesmo em áreas aparentemente homogêneas, favorecendo a detecção de pontos de interesse pelo SIFT de forma mais equilibrada e distribuída ao longo de toda a imagem. O tamanho dos blocos utilizados na equalização adaptativa é de (8×8) pixels, com um limite máximo de realce de contraste fixado em 0.02. A Figura 4.31 demonstra os efeitos de cada etapa do pré-processamento.

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

Figura 4.31: **(a)** Escala de cinza, **(b)** Equalização de histograma, **(c)** Filtro gaussiano, **(d)** Equalização adaptativa de histograma. (Fonte: Autoral).

Detecção

Concluído o pré-processamento, a imagem marcada é submetida ao processo de detecção de pontos de interesse, dos quais são selecionados os $\boldsymbol{n_{pontos\,fortes}}$ mais relevantes. A Figura 4.32 ilustra o processo.

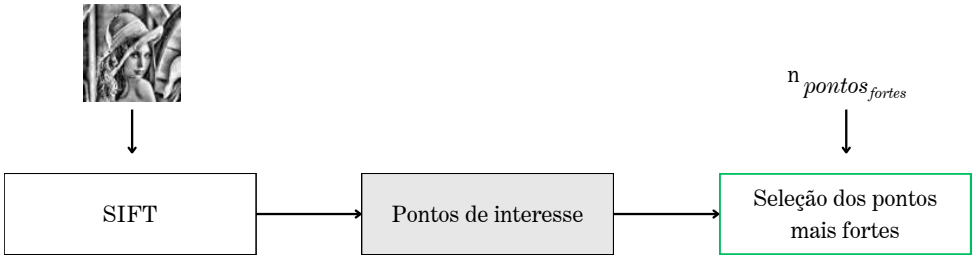
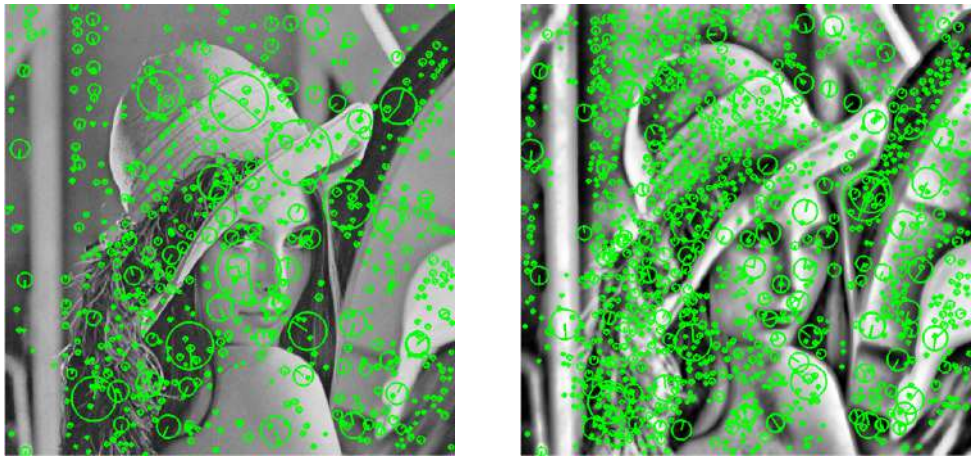


Figura 4.32: Esquema de detecção. (Fonte: Autoral).

Testes empíricos indicaram que armazenar, em média, 1000 pontos de interesse é suficiente para garantir uma elevada taxa de sucesso no alinhamento geométrico em diferentes dimensões de imagem. Esse valor atua como um limite superior, ou seja, se a detecção identificar mais de 1000 pontos, somente os 1000 mais significativos são armazenados. Logo,

$$n_{pontos\,fortes} \leq 1000. \tag{4.26}$$

A Figura 4.33 ilustra a detecção de pontos de interesse em diferentes abordagens, sem pré-processamento e com um pré-processamento. Sem a aplicação do pré-processamento (exceto pela conversão obrigatória para nível de cinza) o SIFT detecta um número menor de pontos de interesse em comparação ao cenário com pré-processamento completo, demonstrando sua necessidade.



(a) (b)

Figura 4.33: Pontos de interesse. **(a)** Sem pré-processamento. **(b)** Com pré-processamento. (Fonte: Autoral).

4.4.5 Processamento dos Parâmetros

Todos os parâmetros utilizados durante o processo de inserção, e classificados como privados são criptografados pelo AES com a chave mestra. Além disso, para proteger a chave mestra, uma Chave de Criptografia de Chaves - Key Encryption Key (KEK) é derivada de uma senha (ou chave) fornecida pelo usuário utilizando o Argon2, onde a KEK é então utilizada para criptografar a chave mestra. Após isso, a KEK é passada ao SHA-256 para criar um hash, necessário para validação posterior da senha (ou chave) do usuário. A Figura 4.34 demonstra os parâmetros privados, enquanto que a Figura 4.35 ilustra todo o processo criptografia e armazenamento dos parâmetros.

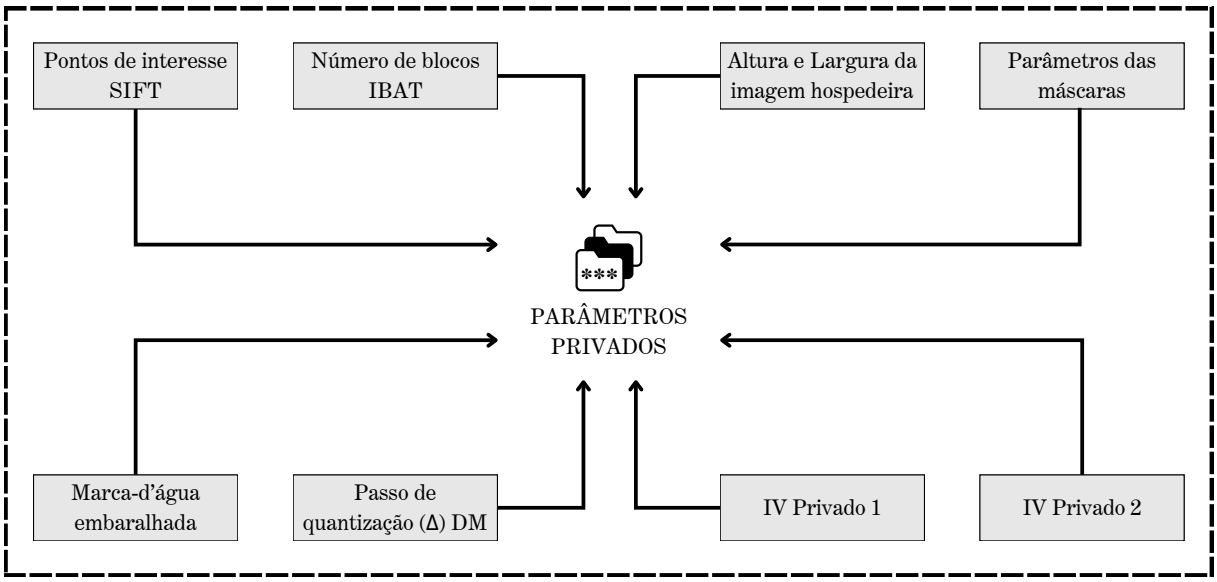


Figura 4.34: Parâmetros privados. (Fonte: Autoral).

Parâmetros Privados

- Pontos de interesse extraídos pelo algoritmo SIFT.
- Número de blocos da IBAT.
- Altura e largura da imagem hospedeira.
- Parâmetros responsáveis pela formação e segmentação das máscaras de incorporação (Tabela 4.2 e Tabela 4.3).
- A marca-d'água embaralhada, necessária para autenticação após o processo de extração.
- $\boldsymbol{\Delta}$ (passo de quantização do DM).
- IV's usados na geração das chaves do sistema (subseção 4.1.4).

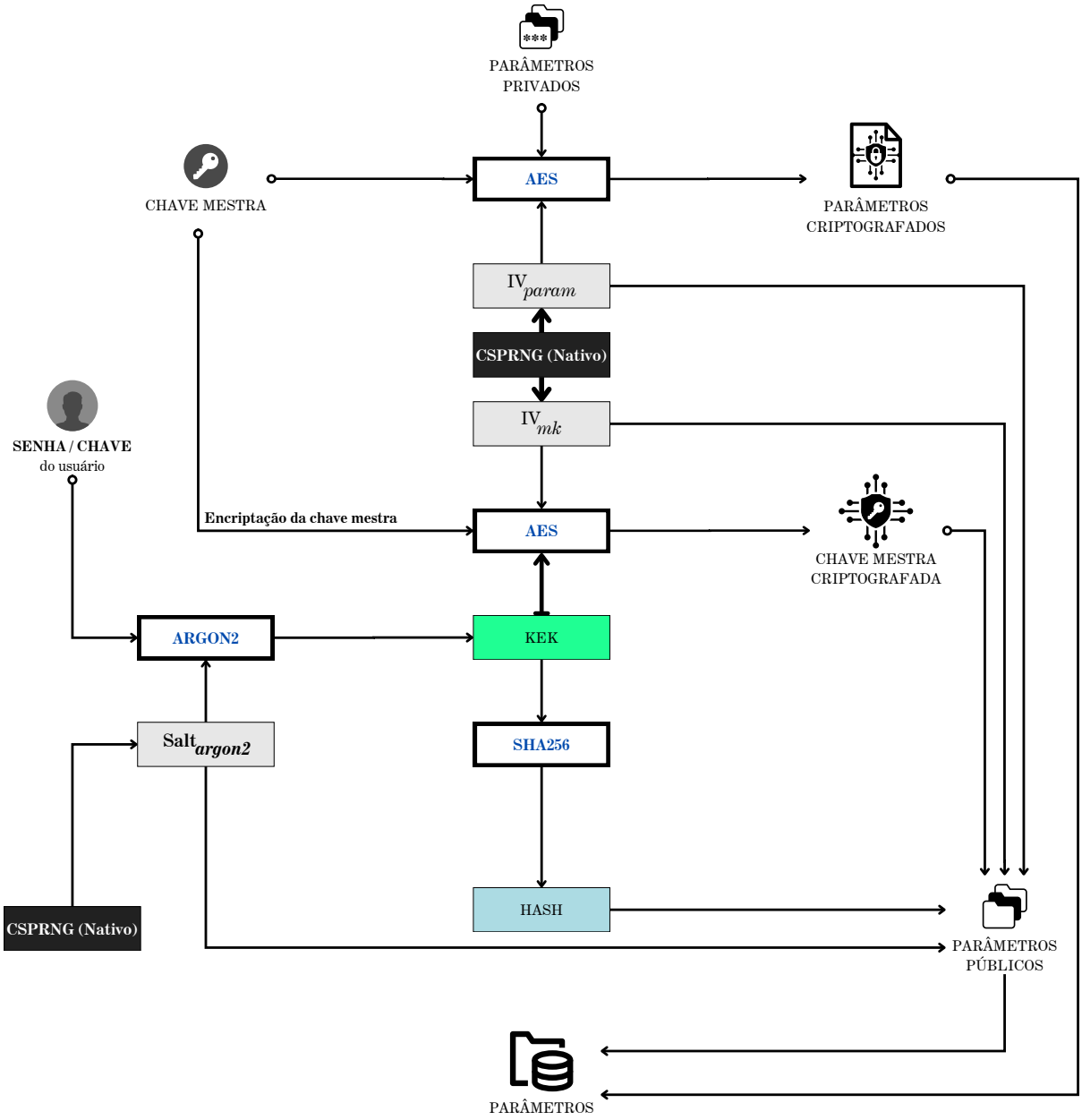


Figura 4.35: Armazenamento dos parâmetros. (Fonte: Autoral).

Parâmetros Públicos

- Hash da KEK, utilizado para verificação da senha (ou chave) do usuário.
- $Salt_{argon2}$, utilizado na derivação da KEK.
- O IV_{mk} utilizado na criptografia da chave mestra.
- O IV_{param} utilizado na criptografia dos parâmetros privados.
- Chave mestra criptografada pela KEK.

4.5 Extração da Marca-d'água

Como o método proposto adota um esquema semi-cego de marca-d'água, são aproveitadas algumas informações do processo de incorporação, em especial os pontos de interesse que descrevem características da imagem hospedeira, e também a marca-d'água original para garantir maior precisão na extração. Dessa forma, os pontos de interesse armazenados nos parâmetros são utilizados para reverter possíveis distorções geométricas na imagem marcada candidata (imagem potencialmente portadora da marca-d'água). Em seguida, a imagem corrigida é passada para o domínio da frequência aplicando a QDFT, assim obtendo a parte real na qual os bits da marca-d'água são extraídos. A Figura 4.36 apresenta um resumo do esquema de extração.

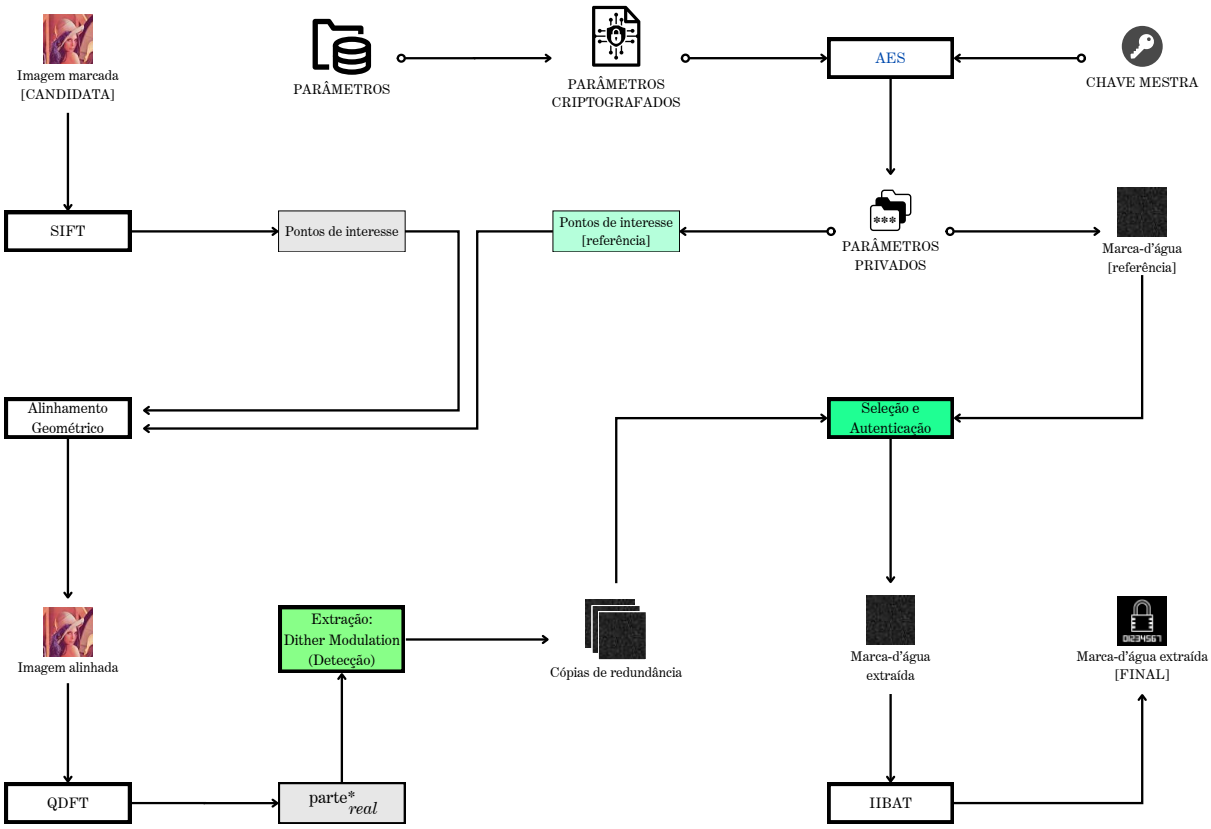


Figura 4.36: Resumo do esquema de extração da marca-d'água. (Fonte: Autoral).

4.5.1 Decodificação dos Parâmetros

Para efetuar a extração, é preciso regenerar a KEK, que é utilizada para descriptografar a chave mestra contida nos parâmetros. A KEK é obtida através do Argon2, onde a senha (ou chave) do usuário é passada junto ao IV (IV_{argon2}) utilizado anteriormente no processo de incorporação. Dessa forma, a KEK regenerada passa pelo SHA-256 e o hash obtido é

comparado ao hash armazenado nos parâmetros. Se os hashs forem iguais, então a KEK regenerada é considerada válida, e é então utilizada para descriptografar a chave mestra. O processo é ilustrado pela Figura 4.37.

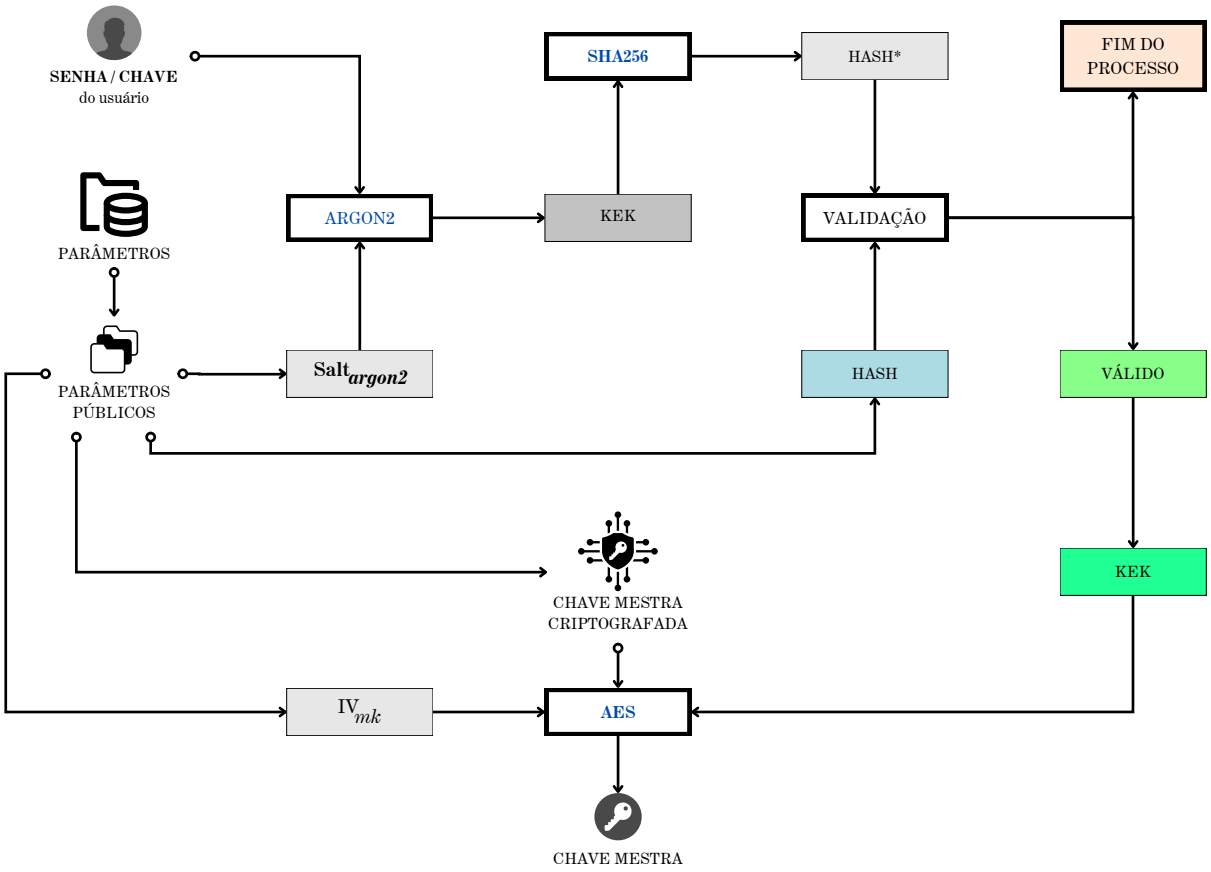


Figura 4.37: Recuperação da chave mestra. (Fonte: Autoral).

A chave mestra recuperada é usada para descriptografar os parâmetros privados e prosseguir com o processo de extração da marca-d'água, conforme demonstrado na Figura 4.38.

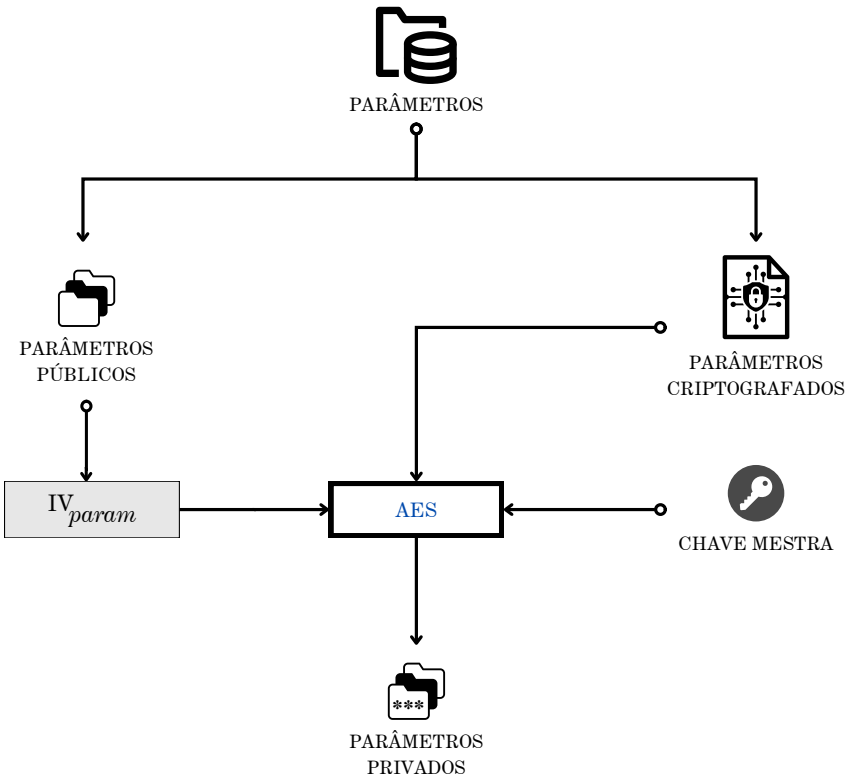


Figura 4.38: Recuperação dos parâmetros privados. (Fonte: Autoral).

4.5.2 Correspondência de Características

Para realizar a correspondência de características, é necessário primeiro extrair os pontos de interesse da imagem marcada candidata seguindo o mesmo processo descrito na subseção 4.4.4. Os vetores de características desses pontos, denominados descritores, são comparados aos descritores dos pontos de referência previamente armazenados, permitindo estimar uma matriz de transformação capaz de reverter eventuais alterações geométricas.

Os descritores extraídos e os descritores de referência são comparados por meio da distância euclidiana, validando as correspondências de acordo com um limiar de distância. Antes desse pareamento, cada vetor de características é submetido ao processo RootSIFT que reduz a influência de picos acentuados e valoriza variações médias e baixas no histograma de gradientes, melhorando a robustez frente a variações de iluminação e contraste.

No momento do pareamento, o *Ratio Test de Lowe* calcula a menor (d_1) e a segunda menor (d_2) distância entre os descritores, considerando a correspondência válida apenas se a razão entre elas for inferior a um limiar, fixo em 0.7. Logo, o ponto de interesse é considerado válido se

$$\frac{d_1}{d_2} < 0.7. \tag{4.27}$$

Ao final do processo de pareamento, obtém-se um conjunto de correspondências entre os pontos de interesse. No entanto, esse conjunto pode conter um número significativo de outliers, uma vez que o pareamento entre descritores é realizado de forma local, ponto a ponto, sem considerar a consistência geométrica global das correspondências. O algoritmo RANSAC é utilizado para identificar esses outliers avaliando cada correspondência em relação ao tipo de modelo geométrico (homografia, afim ou similaridade), definido previamente e fornecido ao RANSAC, que passa a iterar sobre subconjuntos mínimos de correspondências selecionados aleatoriamente para estimar seus parâmetros e avaliar o grau de consenso. Aqueles pares que não se ajustam ao modelo estimado são marcados como outliers e descartados, restando finalmente o conjunto de inliers mais consistente e os parâmetros da transformação que definem a matriz.

Como cada tipo de modelo geométrico abrange diferentes tipos de transformação, o RANSAC é aplicado separadamente utilizando os três modelos mais comuns, homografia, afim e similaridade, com o erro de projeção máximo fixo em 1.5 pixels. Dessa forma, a matriz de transformação correspondente ao modelo que gerar o maior número de inliers é selecionada para realinhar a imagem marcada candidata. A Figura 4.39 demonstra o fluxo completo.

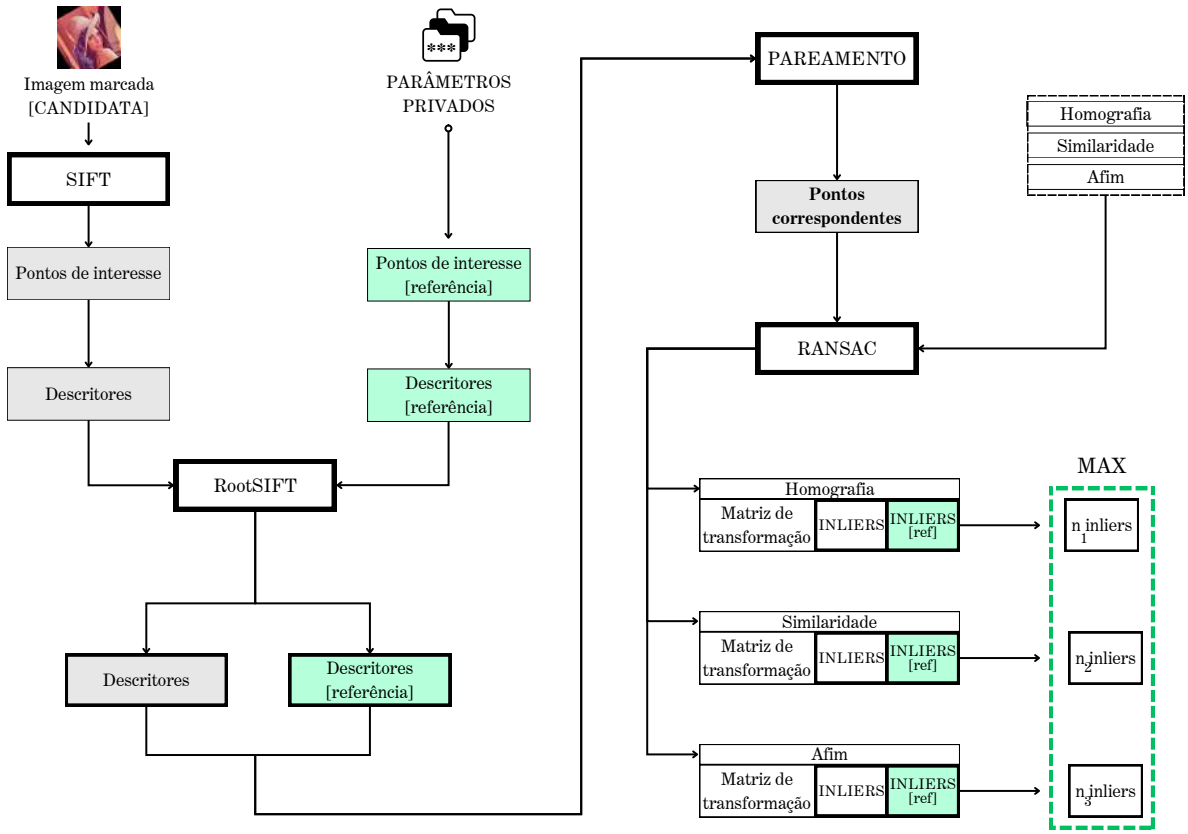


Figura 4.39: Correspondência de Características. (Fonte: Autoral).

Além da matriz de transformação ($\boldsymbol{matriz_{tform}}$), o modelo geométrico selecionado inclui dois conjuntos de inliers com a mesma quantidade de pontos, o conjunto de pontos válidos de referência ($\boldsymbol{pontos_{válidosREF}}$) e o conjunto de pontos correspondentes válidos ($\boldsymbol{pontos_{válidos}}$) na imagem marcada candidata, conforme demonstrado na Figura 4.40.

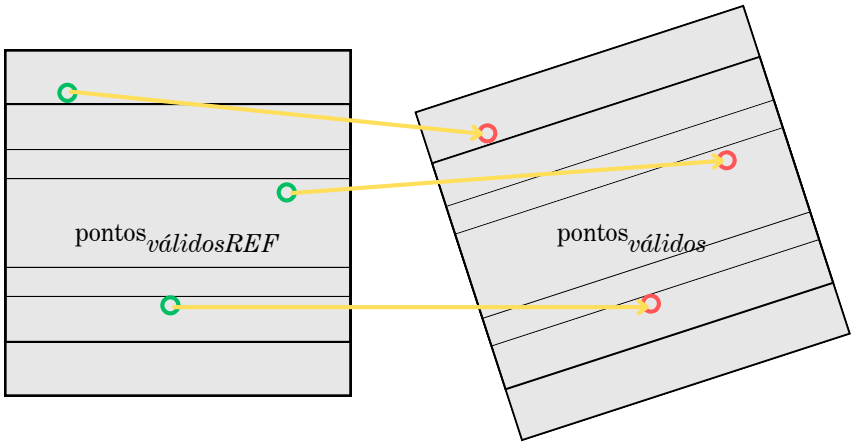


Figura 4.40: **(a)** Pontos válidos de referência. **(b)** Pontos válidos da imagem marcada candidata. (Fonte: Autoral).

A confiabilidade dos pontos correspondidos na imagem marcada candidata é verificada pela sua quantidade em relação ao número total de pontos de referência armazenados. Considerando que foram armazenados $\boldsymbol{n_{pontos_{REF}}}$, o cenário ideal seria que o maior número possível desses pontos seja pareado na imagem marcada candidata durante o processo de correspondência de características. Diante disso, um número mínimo de pontos é necessário para que as correspondências sejam consideradas confiáveis. Essa verificação é realizada por meio da razão entre a quantidade de pontos pareados ($\boldsymbol{n_{inliers}}$) e o total de pontos de referência previamente armazenados ($\boldsymbol{n_{pontos\,REF}}$). Logo,

$$\frac{n_{inliers}}{n_{pontos_{REF}}} \geq 0.1 \ , \tag{4.28}$$

onde o valor fixo em **0.1** define a razão mínima de pontos necessária para assegurar que a matriz de transformação represente uma correspondência geométrica válida e precisa.

4.5.3 Alinhamento Geométrico

Para confirmar que a imagem marcada candidata foi efetivamente transformada e evitar alinhamentos geométricos desnecessários, é feita uma projeção da transformação nos pontos de referência utilizando a matriz de transformação estimada. Assim, cada ponto (x', y') projetado pela matriz de transformação é definido como:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = matriz_{tform} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{4.29}$$

logo, os pontos de referência projetados são dados como:

$$pontos^{*}_{válidosREF} = matriz_{tform} \times pontos_{válidosREF}. \tag{4.30}$$

Ao aplicar a transformação geométrica estimada diretamente nos pontos de referência, é analisado o deslocamento imposto pela transformação. Se houver movimentação significativa dos pontos de referência, a imagem marcada candidata sofreu uma transformação e exige realinhamento. Caso a transformação seja próxima da identidade e os pontos permanecerem próximos ou no mesmo lugar, o alinhamento geométrico não é necessário. Dessa forma, o deslocamento é definido pela distância euclidiana entre pontos de referência e os pontos de referência projetados pela matriz, gerando o vetor de distâncias ($\boldsymbol{d}$). Logo,

$$d_i = \sqrt{(x'_i - x_i)^2 + (y'_i - y_i)^2} \quad \Longrightarrow$$

$$\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n_{inliers}} \end{pmatrix} = \begin{pmatrix} \|ponto^{*}_{válidoREF_1} - ponto_{válidoREF_1}\|_2 \\ \|ponto^{*}_{válidoREF_2} - ponto_{válidoREF_2}\|_2 \\ \vdots \\ \|ponto^{*}_{válidoREF_{n_{inliers}}} - ponto_{válidoREF_{n_{inliers}}}\|_2 \end{pmatrix} \tag{4.31}$$

onde $||.||_2$ representa a norma-2 (distância euclidiana) entre os pontos.

O status da transformação é baseado no desvio padrão do vetor de distâncias ($\boldsymbol{d}$) e na maior distância dentro do vetor. Logo,

$$\begin{aligned} distancia_{dp} &= std(d), \\ distancia_{max} &= max(d). \end{aligned} \tag{4.32}$$

onde $\boldsymbol{std(.)}$ calcula o desvio padrão, e $\boldsymbol{max(.)}$ é a função que retorna o maior valor dentro do vetor.

Se a matriz de transformação causar um deslocamento nos pontos de referência, o desvio padrão das distâncias será alto devido as diferentes magnitudes de movimento entre os pontos. Porém, se não houver deslocamento significativo os valores de distância contidos em ($\boldsymbol{d}$) permanecerão próximos de zero, resultando em um desvio padrão baixo. No caso de os pontos se moverem de forma relativamente uniforme, gerando um desvio

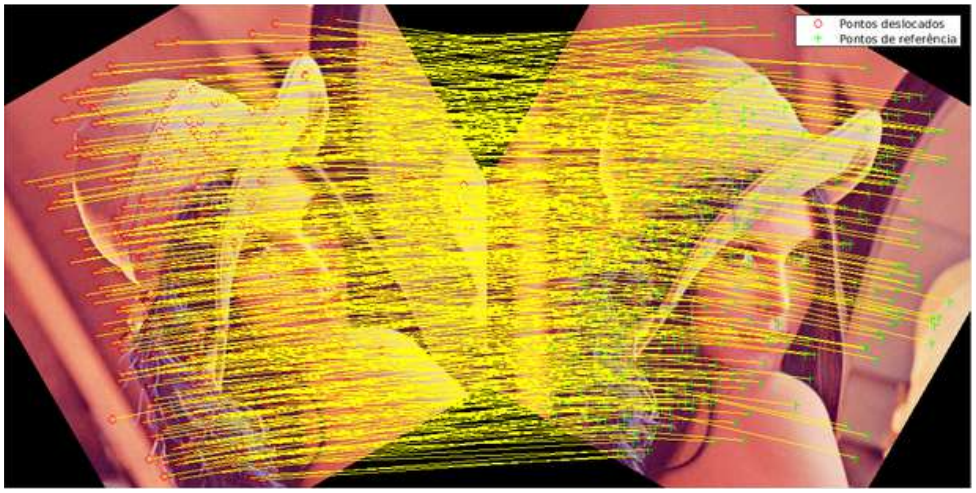
padrão baixo mesmo na presença de uma transformação, a distância máxima é utilizada como critério adicional para verificar a ocorrência de uma transformação geométrica.

Os limiares, determinados com base em testes empíricos, são fixos em **0.55** para desvio padrão, e **3.5** para a distância máxima. O Algoritmo 4 descreve o processo de decisão, onde se o status resultante for verdadeiro, conclui-se que a imagem marcada candidata sofreu uma transformação geométrica e precisa ser realinhada.

Algoritmo 4 Status da Transformação

$status \leftarrow falso$
if $distancia_{dp} > 0.55$ **then**
    $status \leftarrow verdadeiro$
end if
if $distancia_{max} > 3.5$ **then**
    $status \leftarrow verdadeiro$
end if

Caso seja confirmado que a imagem marcada candidata passou por uma transformação geométrica, ela é realinhada utilizando a matriz de transformação ($\boldsymbol{matriz_{tform}}$). A Figura 4.41 ilustra um exemplo de alinhamento geométrico, com correção rotacional de $-30.0°$. Os pontos em verde são os pontos de referência que foram pareados com os pontos considerados válidos na imagem marcada candidata, ilustrados em vermelho.



(a) (b)

Figura 4.41: **(a)** Imagem marcada candidata rotacionada em 30.0°. **(b)** Imagem após o alinhamento geométrico. (Fonte: Autoral).

4.5.4 Extração

Após o alinhamento da imagem marcada candidata, é possível dar início ao processo de extração que segue um fluxo semelhante ao de incorporação da marca-d'água. As máscaras de incorporação (agora utilizadas para localizar os bits), a definição das coordenadas, o ajuste da matriz de coordenadas com base no fator de ponderação ($\boldsymbol{f_\Delta}$), a geração dos vetores de Dither ($\boldsymbol{d_0}$ e $\boldsymbol{d_1}$), e a obtenção da parte real da QDFT seguem exatamente os mesmos procedimentos descritos nas Seções 4.4.1, 4.4.2 e 4.4.3, onde todos os parâmetros e chaves utilizados devem ser rigorosamente os mesmos.

Os vetores de Dither ($\boldsymbol{d_0}$ e $\boldsymbol{d_1}$) regenerados são utilizados para a decodificação dos bits da marca-d'água, logo:

$$\begin{aligned} dist_0 &= min((coef^* + d0)\ mod\ \Delta_{efetivo},\ \ \Delta_{efetivo} - (coef^* + d0)\ mod\ \Delta_{efetivo}) \\ dist_1 &= min((coef^* + d1)\ mod\ \Delta_{efetivo},\ \ \Delta_{efetivo} - (coef^* + d1)\ mod\ \Delta_{efetivo}) \end{aligned} \tag{4.33}$$

onde $\boldsymbol{dist_0}$ e $\boldsymbol{dist_1}$ mede as distâncias da qual o coeficiente está dos seus possíveis pontos de reconstrução, onde a menor distância indica o bit correto. Dessa forma:

$$bit_{extraido} = \begin{cases} 0, & if\ (dist_0 \leq dist_1) \\ 1, & if\ (dist_0 > dist_1). \end{cases} \tag{4.34}$$

Embora a geração das coordenadas simétricas seja indispensável para gerar a matriz de coordenadas, durante a extração dos bits da marca-d'água apenas as coordenadas principais são de fato utilizadas. Assim, a matriz de coordenadas é percorrida linha a linha e a sequência tripla é recuperada utilizando modo de detecção do DM. O Algoritmo 5 apresenta os detalhes do procedimento de extração, onde $ditherModulationDetect(.)$ aplica a Equação 4.33 e a Equação 4.34.

Algoritmo 5 Extração

for $i = 1$ to n_{linhas} **do**

$u,\ v\ \leftarrow\ matriz_{coordenadas}(i)$

$coef_{modificado} \leftarrow parte^*_{real}(u,\ v)$

$bit_{extraido} \leftarrow ditherModulationDetect(coef_{modificado},\ \Delta_{efetivo}(i),\ d_0(i),\ d_1(i))$

$sequência_{tripla_{EX}}(i) \leftarrow bit_{extraido}$

end for

4.6 Pós-Processamento da Marca-d'água

Uma vez extraída a sequência tripla ($\boldsymbol{sequência}_{\boldsymbol{tripla}_{\boldsymbol{EX}}}$), é feito um pós-processamento para verificar se os bits recuperados correspondem efetivamente à marca-d'água inserida ou decorrem apenas de ruído aleatório.

4.6.1 Seleção

Para selecionar a melhor cópia redundante da marca-d'água dentro da sequência tripla, são utilizados os bits da marca-d'água original embaralhados, que foram previamente armazenados nos parâmetros na fase de incorporação. A cópia que exibir a menor Taxa de Erro de Bit - Bit Error Rate (BER) é então escolhida como a provável marca-d'água, conforme demonstrado na Figura 4.42.

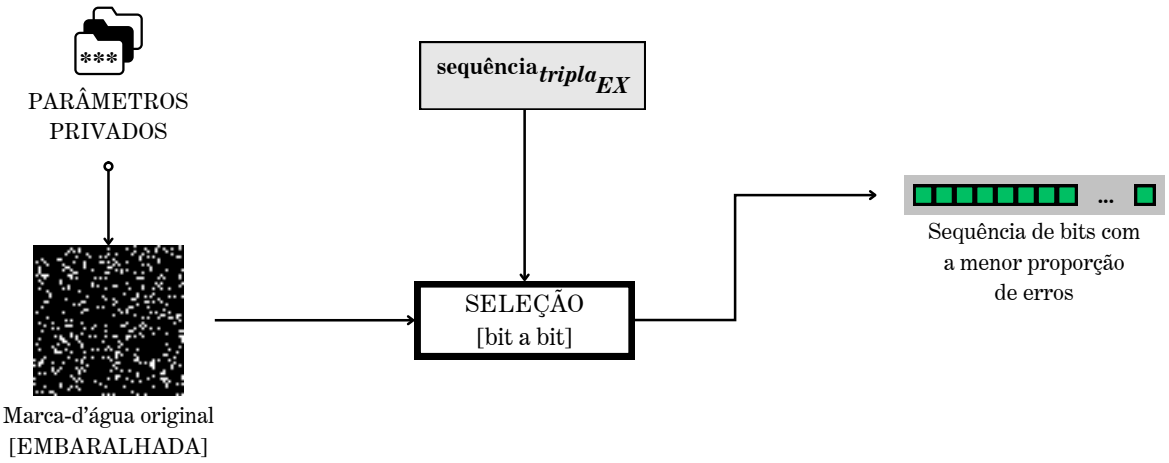
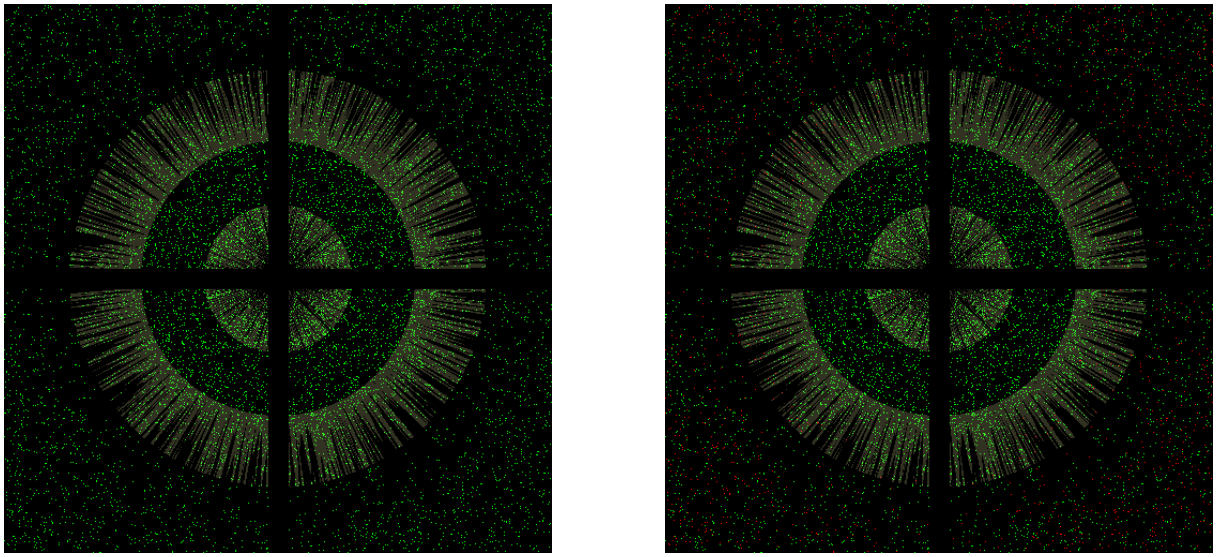


Figura 4.42: Esquema de seleção dos bits com menor BER. (Fonte: Autoral).

A Figura 4.43 ilustra as coordenadas e faixas de onde os bits foram extraídos da imagem marcada, tanto em seu estado intacto quanto após um ataque de compressão JPEG-80. Os pontos verdes indicam extrações bem-sucedidas, enquanto os vermelhos sinalizam falhas na recuperação dos bits.



(a) (b)

Figura 4.43: Coordenadas de extração. **(a)** Imagem marcada intacta. **(b)** Imagem após compressão JPEG-80. (Fonte: Autoral).

4.6.2 Autenticidade

O número de bits corretos ($\boldsymbol{nbits_{corretos}}$) na subsequência de menor BER é utilizado para validar a autenticidade dos bits extraídos e evitar falsos positivos. Para isso, é exigido um número mínimo de bits que foram corretamente extraídos, garantindo que os acertos observados não ocorreram por mera coincidência. Esse limiar mínimo é calculado utilizando a inversa da CDF binomial, que determina a quantidade mínima ($\boldsymbol{k}$) de acertos necessária para um determinado nível de confiança ($\mathbf{1-\alpha}$), garantindo que os acertos não são aleatórios. Dessa forma, o número mínimo ($\boldsymbol{k_{min}}$) de bits corretos que comprovam a existência da marca-d'água é dado como:

$$k_{min} = inversaBinomial(1-\alpha, \quad n_{bits}, \quad p_0) + 1 \tag{4.35}$$

onde $\boldsymbol{\alpha}$ representa o nível de significância, que é fixo **0.00003**. O número de bits da marca-d'água é definido por $\boldsymbol{n_{bits}}$, e $\boldsymbol{p_0}$, fixo em **0.5**, representa a probabilidade de acerto sob hipótese nula (aleatoriedade). Dessa forma, se

$$nbits_{corretos} \geq k_{min}, \tag{4.36}$$

a presença da marca-d'água é confirmada.

4.6.3 Refino da Extração e Desembaralhamento

Confirmada a presença da marca-d'água, a subsequência com menor BER é refinada com base nas demais subsequências descartadas por apresentarem taxas de erro mais altas. Apesar de não terem sido selecionadas, essas cópias ainda contêm bits que foram corretamente recuperados, os quais são utilizados para corrigir os possíveis bits incorretos na subsequência principal (com menor BER), aprimorando a qualidade final da extração.

Concluído o refinamento, a marca-d'água é desembaralhada por meio da transformação inversa da IBAT, utilizando a chave e o IV previamente armazenados nos parâmetros. Para isso, as operações descritas na subseção 4.3.2 são executadas em ordem reversa, restaurando a marca-d'água. A Figura 4.44 demonstra o processo.

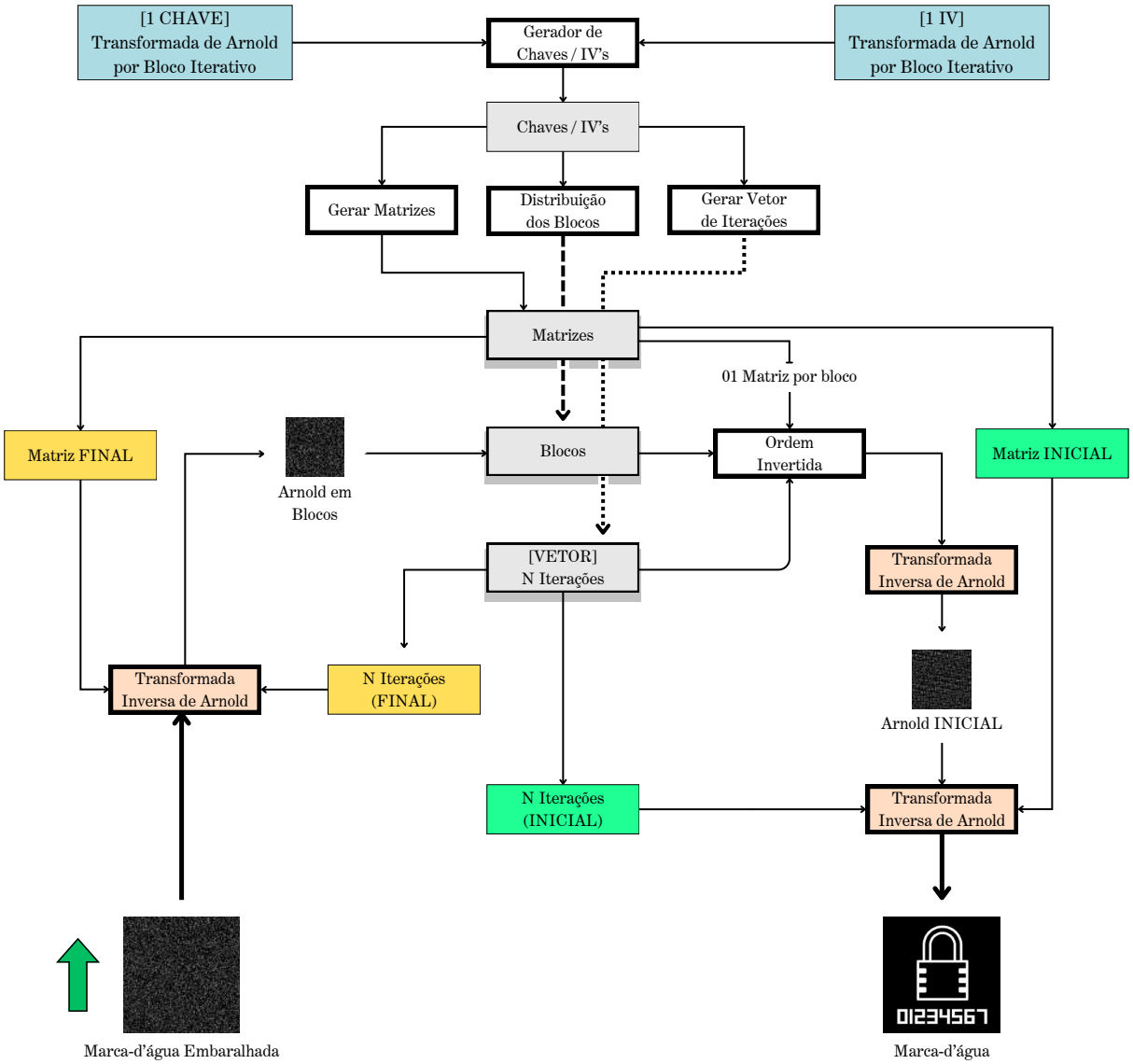


Figura 4.44: Esquema da Transformada Inversa de Arnold por Bloco Iterativo - Inverse Iterative Block Arnold Transform (IIBAT). (Fonte: Autoral).

A Figura 4.45 apresenta a marca-d'água após o refinamento, extraída da imagem marcada submetida à compressão com perdas (JPEG-50).

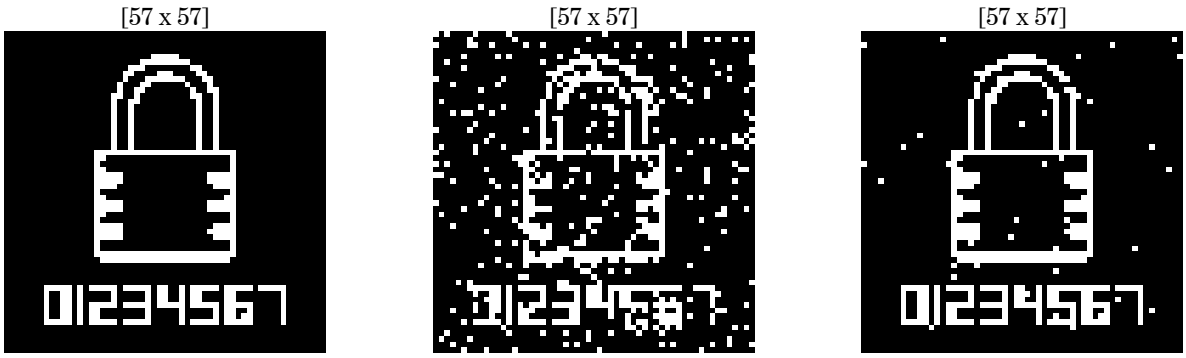


Figura 4.45: Marca-d'água extraída. (Fonte: Autoral).

O próximo capítulo (Capítulo 5) apresentará os resultados obtidos em termos de robustez e imperceptibilidade do método proposto.

Capítulo 5

Resultados

Este capítulo apresenta os resultados referentes aos testes realizados, avaliando o desempenho individual e comparando os métodos de referência [3, 4] com método proposto neste trabalho.

5.1 Ambiente de Testes

As operações de processamento de imagens foram implementadas em MATLAB, enquanto que as funções de criptografia, como AES, o CSPRNG (CTR-DRBG) e o Argon2, foram implementados em linguagem C para aproveitar a aceleração por hardware. Essas rotinas foram integradas ao MATLAB por meio do MEX (MATLAB Executable), que permite a chamada direta de funções externas escritas em $C/C++$ como se fossem funções nativas. O desenvolvimento e execução dos testes foram realizados utilizando o sistema operacional Ubuntu 22.04.5 LTS (64 bits).

5.1.1 Parâmetros Ajustáveis

Os parâmetros exibidos na Tabela 5.1 foram ajustados de forma a equilibrar desempenho, robustez e imperceptibilidade do sistema de marca-d'água.

Tabela 5.1: Parâmetros

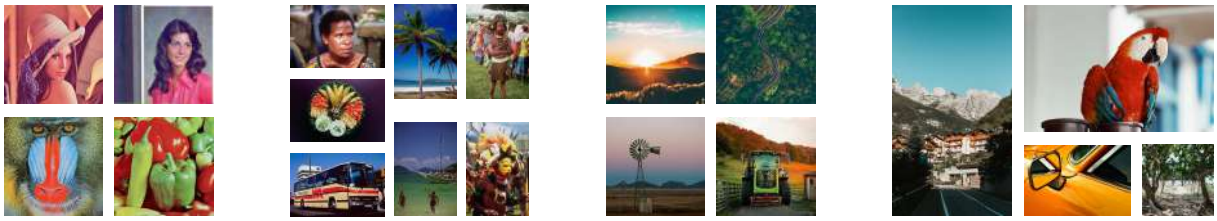
Parâmetros	Valor	Descrição
Senha (ou chave) do usuário	**'senha-facil-123'**	String de qualquer tamanho, podendo ser outra chave criptográfica.
Quantidade blocos IBAT	**80**	Número de blocos transformados internamente pela AT. Quanto mais blocos, mais seguro, porém com um tempo maior de execução.
Passo de quantização ($\boldsymbol{\Delta}$)	**68**	Controla o nível de alterações. Um $\boldsymbol{\Delta}$ maior é mais robusto, porém causa maior distorção na imagem.
$n_{pontos\,fortes}$ SIFT	**1000**	Quantidade máxima de pontos de interesse armazenados. Mais pontos garante um alinhamento geométrico mais preciso, porem aumenta o tamanho (bytes) dos parâmetros armazenados.

5.1.2 Imagens Hospedeiras

Como o método de referência [3], restringiu seus testes a apenas duas imagens, parte dos conjuntos de imagens de teste de sua versão aprimorada [4] serão utilizados.

A versão aprimorada utilizou dois conjuntos de testes, $'Data1'$ com 64 imagens (512×512) comumente usadas em pesquisas de processamento de imagens, e $'Data2'$ com 1000 imagens de resolução 384×256 ou 256×384 pixels. No entanto, devido a referência ao conjunto $'Data1'$ estar offline, foi preciso buscar as imagens separadamente, das quais 32 das 64 imagens de $'Data1'$ foram recuperadas e agrupadas no conjunto de imagens de teste ***imgs1***. Dessa forma, para manter a variedade e atingir um número considerável de imagens, o conjunto ***imgs1*** foi complementado com imagens adicionais de 512×512 pixels, formando o ***imgs3***. Por outro lado, a referência ao conjunto $'Data2'$ com as 1000 imagens foi encontrado online, o que permitiu que todas as imagens fossem reunidas no conjunto de imagens de teste ***imgs2***. Portanto, os conjuntos ***imgs1***, ***imgs2*** e ***imgs3*** formam a base de imagens de teste para realizar a comparação entre os métodos de referência e o método proposto neste trabalho.

Além dos testes de comparação, são realizados testes individuais para melhor embasar a robustez e imperceptibilidade do método proposto. Diante disso, um conjunto extra de imagens de teste foi formado, o ***imgs4***, composto por inúmeras imagens de alta resolução, ampliando significativamente a variedade da base de testes. A Figura 5.1 demonstra uma prévia desses grupos, e a Tabela 5.2 exibe detalhes adicionais acerca dos conjuntos de imagens de teste.



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

Figura 5.1: Prévia dos conjuntos de imagens de teste. **(a)** *imgs1*. **(b)** *imgs2*. **(c)** *imgs3*. **(d)** *imgs4*.

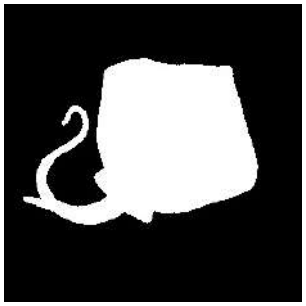
Tabela 5.2: Detalhes dos conjuntos de teste.

Conjunto de imagens de teste	**Resolução**	**Quantidade**	**Fonte**
imgs1	**Fixa**: (512 x 512)	32	STI - Standard Test Images [1]
imgs2	**Fixa**: (384 x 256) e (256 x 384)	1000	Corel 1000 Dataset [43, 1]
imgs3	**Fixa**: (512 x 512)	256	Image Pack [1]
imgs4	**Variada**: (1664 x 2496, 1362 x 2043, 1919 x 1280, ..., **h** x **w**)	256	Image Pack [1]

5.1.3 Imagens de Marca-d'água

Foram realizados diversos testes com diferentes imagens binárias para a representação da marca-d'água, porém, para a apresentação dos resultados, duas imagens foram selecionadas. A primeira imagem binária (Marca-d'água de testes - Versão 1 (V1)) é utilizada nos testes individuais do método proposto. A segunda imagem binária (Marca-d'água de testes - Versão 2 (V2)) é a mesma marca-d'água utilizada pelos autores do método de referência [3] e sua versão aprimorada [4], sendo esta utilizada como forma justa na comparação dos resultados entre os métodos. Mais detalhes são demonstrados na Figura 5.2.

(a)



(b)

Figura 5.2: **(a)** Marca-d'água de testes - **Versão 1 (V1)** (612 x 612). (Fonte: Autoral). **(b)** Marca-d'água de testes - **Versão 2 (V2)** (250 x 250). (Fonte: [44, 3, 4]).

Como especificado na Figura 5.2, as imagens de marca-d'água V1 e V2 possuem resoluções base (612 x 612) e (250 x 250) pixels, respectivamente. Diante disso, no caso do método proposto, essas proporções são adaptadas automaticamente de acordo com as características e resolução da imagem hospedeira, conforme abordado em 4.2.

5.2 Resultados e Testes Individuais

5.2.1 Imperceptibilidade

Para avaliar o impacto visual causado pela incorporação da marca-d'água, são utilizadas as métricas PSNR e SSIM, descritas na seção 2.11, as quais indicam o nível de distorção introduzido. A interpretação desses valores é feita com base em limiares definidos na subseção 2.11.5.

Ao comparar visualmente a imagem original com a imagem marcada, conforme a Figura 5.3, é possível concluir que a incorporação da marca-d'água V1 não introduziu nenhum artefato ou distorção evidente na imagem hospedeira.

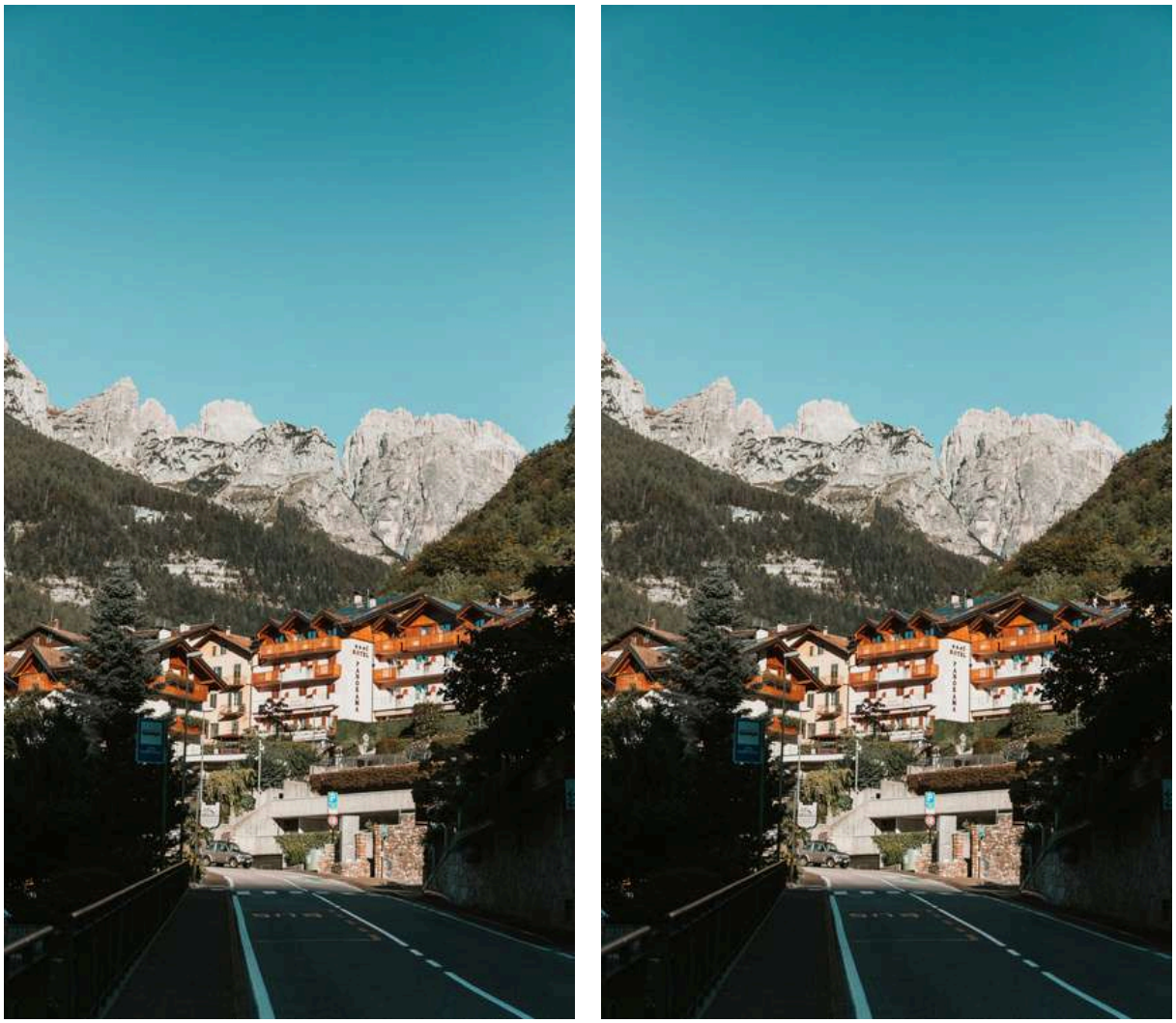
(a) Imagem original

(b) Imagem marcada: **Método proposto**

Figura 5.3: Demonstração da imagem hospedeira original, e sua versão após a incorporação da marca-d'água V1 (**45 x 45**). **(a)** Imagem original (512 x 960). **(b)** Imagem marcada (PSNR: 44,8785 dB, SSIM: 0,9758). (Imagem utilizada no processo: UnB, Autoral).

Além disso, os altos valores de PSNR(44,8785 dB) e SSIM(0.9758) confirmam que o ruído introduzido não foi capaz de causar distorções perceptíveis, mantendo a imagem

marcada em alta qualidade em relação à imagem original. A Figura 5.4 demonstra outro exemplo de alta imperceptibilidade após a incorporação da marca-d'água.



(a) Imagem original

(b) Imagem marcada: **Método proposto**

Figura 5.4: Demonstração da imagem hospedeira original, e sua versão após a incorporação da marca-d'água V1 (**98 x 98**). **(a)** Imagem original (1881 x 1058). **(b)** Imagem marcada (PSNR: 43,6425 dB, SSIM: 0,9738). (Imagem utilizada no processo: Molveno, Georgia de Lotz [1].

Tabela 5.3: Valores médios de PSNR e SSIM.

Conjunto de imagens de teste	$PSNR_{médio}$	$SSIM_{médio}$
imgs1	40.7150 dB	0.9595
imgs2	41.7830 dB	0.9752
imgs3	40.6380 dB	0.9587
imgs4	42.6052 dB	0.9693

A Tabela 5.3 demonstra que após a incorporação da marca-d'água V1 em todas as imagens de teste, a média dos valores de PSNR permanece acima de 40 dB, bem como a média dos valores do SSIM se aproxima de 1 em todos os conjuntos de teste, comprovando que o método se mantém consistente em diferentes classes e tipos de imagens, mantendo a imperceptibilidade da marca-d'água.

Para uma análise mais detalhada do impacto perceptual, outros exemplos de imagens marcadas estão disponíveis no apêndice, em A.1.

5.2.2 Autenticação

A utilização da marca-d'água original como referência permite através da comparação com as cópias redundantes determinar o número de bits corretos. Nesse contexto, para evitar falsos positivos, uma porcentagem mínima de acertos é necessária para que a sequência da marca-d'água extraída não seja considerada ruído aleatório.

Em um cenário completamente aleatório, cada bit tem 50% de chance de coincidir com o original, assim como em um lançamento de uma moeda. Porém, no caso de uma série de eventos (10 lançamentos de moeda ou 10 bits comparados) a taxa de acertos pode variar de forma considerável em torno dos 50%, ou seja, quanto menor o tamanho da amostra, maior é essa oscilação em torno do valor central, o que faz com que obter 7 caras em 10 lançamentos seja perfeitamente aceitável como resultado aleatório.

O ponto central é que considerar uma taxa de acerto de 50% como limiar de aleatoriedade depende diretamente do tamanho da amostra. Por exemplo, uma taxa de acerto de 55% em relação à marca-d'água original poderia ser interpretada como válida por ultrapassar 50%, sugerindo que a sequencia extraída não é aleatória. No entanto, essa conclusão esta incorreta, pois 55% de acertos em uma sequência de 10 bits está dentro da variabilidade esperada da aleatoriedade, enquanto que os mesmos 55% em uma sequência de 4096 bits já não seriam considerados aleatórios devido a variância relativa ser pequena. Diante disso, utilizar a inversa da CDF binomial é a melhor estratégia, pois o limiar percentual mínimo de acertos é adaptado dinamicamente de acordo com o tamanho da sequencia de

bits da marca-d'água, evitando que bits aleatórios sejam confundidos com a marca-d'água original.

Para embasar essa teoria, todos os conjuntos de imagens de teste (***imgs1***, ***imgs2***, ***imgs3***, ***imgs4***) foram submetidos diretamente ao processo de extração sem nenhuma marca-d'água incorporada. Assim, foi possível verificar se a marca-d'água seria detectada mesmo sem nenhuma marca-d'água inserida. Os gráficos a seguir demonstram a taxa de acerto de todas as imagens de teste em relação ao seu limiar mínimo.

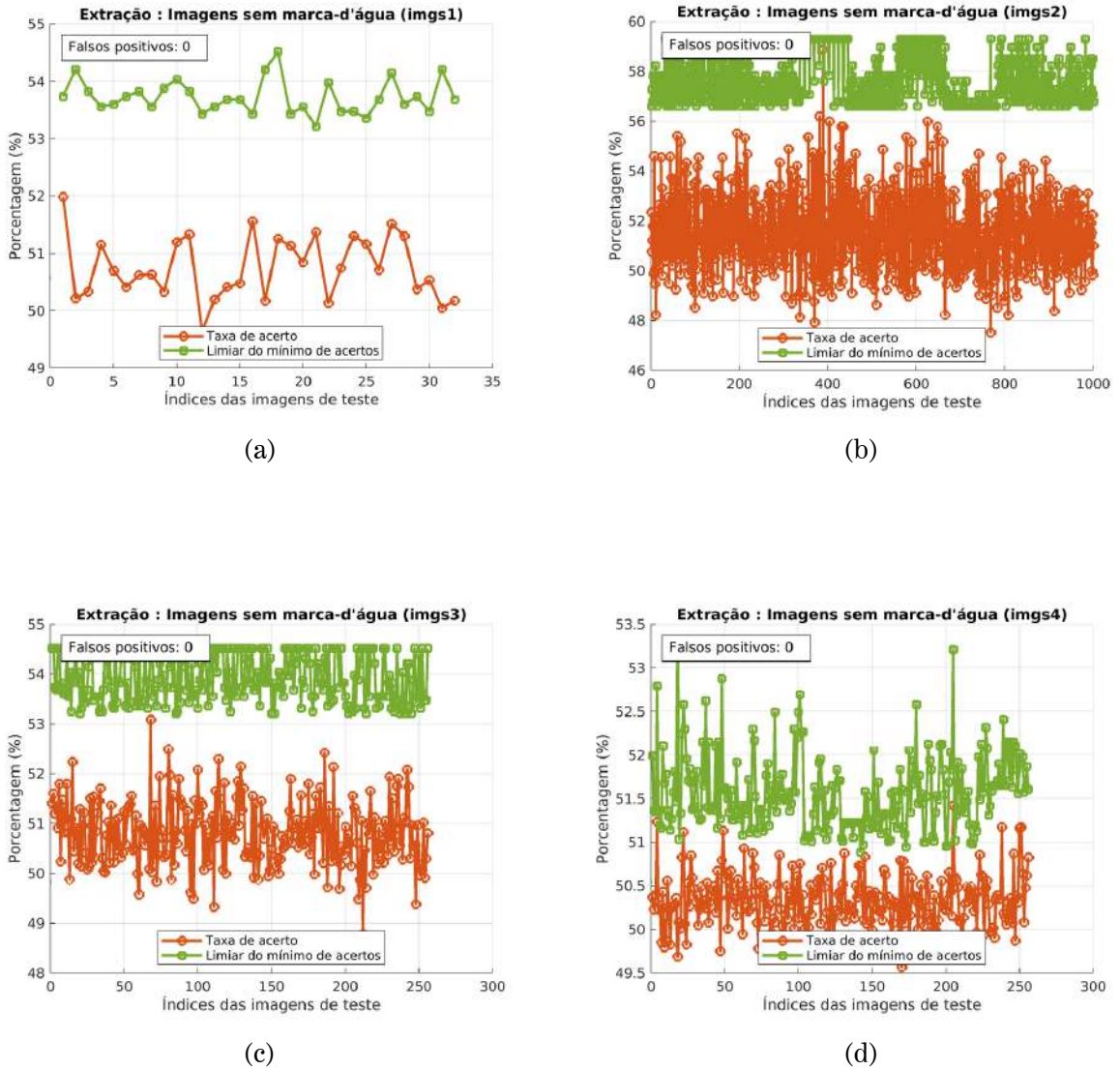


Figura 5.5: Extração da marca-d'água nos conjuntos de imagens originais (sem marca-d'água). **(a)** *imgs1*. **(b)** *imgs2*. **(c)** *imgs3*. **(d)** *imgs4*.

Conforme ilustrado na Figura 5.5, a taxa de acerto obtida ao tentar extrair a marca-d'água de imagens não marcadas permanece consistentemente abaixo do limiar mínimo exigido para que uma sequência seja considerada válida. Esse resultado comprova que

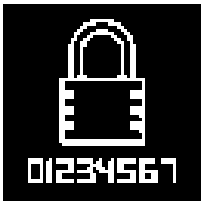
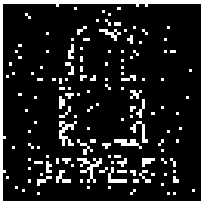
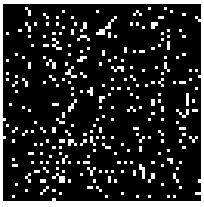
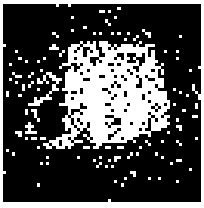
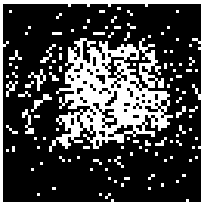
o nível de confiança (1 - α) de $99,997\%$ garante um limiar seguro de acertos, capaz de evitar falsos positivos.

5.2.3 Robustez

Para avaliar a robustez do método proposto, todos os conjuntos de imagens de teste foram submetidos a diferentes tipos de ataques. Nesse contexto, a métrica mais adequada para medir a robustez é a NC, que é capaz de avaliar o grau de similaridade dos dados binários da marca-d'água extraída de forma direta. Como já abordado em 2.11.3, o NC varia no intervalo [0, 1], onde o valor 1 indica a máxima correlação entre a marca-d'água original e a marca-d'água extraída, demonstrando a total recuperação dos bits da marca-d'água. Além disso, o BER, apresentado em 2.11.4, é utilizado para medir a taxa de erro dos bits extraídos.

A Tabela 5.4 demonstra a relação entre o valor de NC e a qualidade perceptível da marca-d'água extraída. Os valores de NC das marcas-d'água extraídas evidenciam uma perda na qualidade visual conforme o NC diminui. Essa análise demonstra que, em média, para valores de NC inferiores a 0.5, a marca-d'água torna-se progressivamente irreconhecível.

Tabela 5.4: Exemplos de valores NC.

			
NC: 1.0000	NC: 0.8627	NC: 0.5646	NC: 0.2445
			
NC: 1.0000	NC: 0.8119	NC: 0.6887	NC: 0.3765

O conjunto de imagens ***total_imgs***, formado pelos grupos de imagens de teste ***imgs1***, ***imgs2***, ***imgs3*** e ***imgs4***, foi submetido a uma série de ataques com o objetivo de estimar o impacto dessas alterações sobre um número significativo de imagens, avaliando o quanto comprometem a integridade da marca-d'água. A Figura 5.6 demonstra alguns exemplos de ataques utilizados.

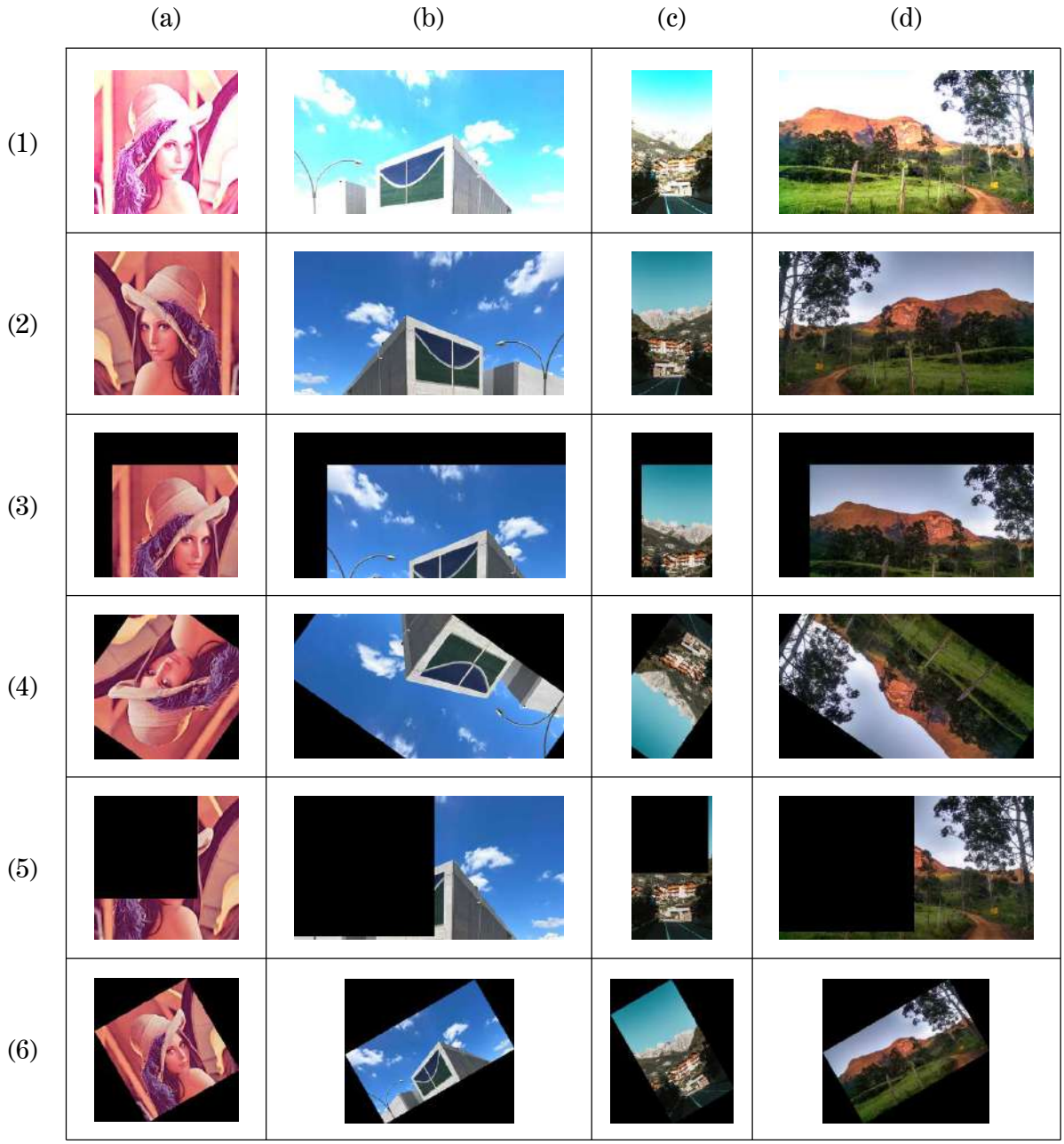


Figura 5.6: Alguns exemplos de ataque às imagens de teste. **(a)** Lenna. **(b)** UnB. **(c)** Molveno. **(d)** Pedra do Pato. **(1)** Aumento da iluminação (2.0 x). **(2)** Reflexão horizontal. **(3)** Translação x(22 %) y(12 %). **(4)** Rotação ($-215.0°$) com recorte. **(5)** Corte (1/2). **(6)** Rotação ($30.0°$) com expansão.

Todos os resultados referentes ao ataques são especificados nas tabelas de ataques. Essas tabelas demonstram as marcas-d'água extraídas das imagens de exemplo (Lenna, UnB, Molveno, Pedra do Pato) e uma média dos valores de NC, calculados entre a marca-d'água extraída e a marca-d'água original após cada ataque. Cada marca-d'água (V1)

exemplificada nas colunas Lenna, UnB, Molveno, e Pedra do Pato possui uma resolução única que foi adaptada de acordo com a imagem.

A Tabela 5.5 demonstra o impacto na qualidade da marca-'água após ataques de compressão com perdas à imagem marcada, considerando diferentes níveis de compressão JPEG e seus respectivos fatores de compressão. É notado que, mesmo com um fator de compressão agressivo (como a redução para 20% do tamanho original), a marca-d'água é recuperada com um alto valor médio de NC (0,7353) em todos os conjuntos de imagens de teste, se mantendo visualmente reconhecível.

Tabela 5.5: Ataques de compressão. ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques de compressão	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Compressão JPEG (80)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 0.9994	M_{nc}: 0.9989 DP: 0.0074
Compressão JPEG (50)	[57 x 57]  NC: 0.9613	[45 x 45]  NC: 0.8857	[98 x 98]  NC: 0.9125	[265 x 265]  NC: 0.9872	M_{nc}: 0.9302 DP: 0.0798
Compressão JPEG (40)	[57 x 57]  NC: 0.9080	[45 x 45]  NC: 0.8345	[98 x 98]  NC: 0.8960	[265 x 265]  NC: 0.9704	M_{nc}: 0.9103 DP: 0.0872
Compressão JPEG (20)	[57 x 57]  NC: 0.7764	[45 x 45]  NC: 0.7641	[98 x 98]  NC: 0.7563	[265 x 265]  NC: 0.8061	M_{nc}: 0.7353 DP: 0.1027

Já nas Tabelas 5.6 a 5.10, são apresentados os valores de NC após os ataques geométricos, que modificam a disposição espacial dos pixels, alterando a estrutura global da imagem marcada. Os testes demonstram que na maioria dos ataques geométricos o SIFT é capaz de realinhar a imagem marcada de modo que a extração ocorra sem mai-

ores problemas, demonstrando uma alta robustez do método proposto frente a ataques geométricos. As tabelas a seguir dividem os ataques geométricos em 5 partes.

Tabela 5.6: Ataques geométricos (**Parte 1**). ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques geométricos	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Reflexão vertical	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 0.8767 DP: 0.2796
Reflexão horizontal	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 0.8782 DP: 0.2788
Translação x(0.4%) y(0.4%)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9930	[265 x 265]  NC: 1.0000	M_{nc}: 0.9948 DP: 0.0123
Translação x(5%) y(7%)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 0.9590	M_{nc}: 0.9860 DP: 0.0273
Translação x(22%) y(12%)	[57 x 57]  NC: 0.9843	[45 x 45]  NC: 0.9962	[98 x 98]  NC: 0.9956	[265 x 265]  NC: 0.9949	M_{nc}: 0.9363 DP: 0.0674
Translação x(2%) y(0.19%)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9952	[265 x 265]  NC: 1.0000	M_{nc}: 0.9919 DP: 0.0206

Tabela 5.7: Ataques geométricos (**Parte 2**). ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques geométricos	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Rotação (30.0°) com expansão	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9960	[265 x 265]  NC: 0.9991	M_{nc}: 0.9995 DP: 0.0042
Rotação (90.5°) com expansão	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 0.9999 DP: 0.0006
Rotação (58.0°) com recorte	[57 x 57]  NC: 0.9976	[45 x 45]  NC: 0.9962	[98 x 98]  NC: 0.9758	[265 x 265]  NC: 0.9969	M_{nc}: 0.9394 DP: 0.0617
Rotação (−215.0°) com recorte	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 0.9981	[98 x 98]  NC: 0.9908	[265 x 265]  NC: 0.9919	M_{nc}: 0.9529 DP: 0.0533
Rotação (334.0°) com recorte	[57 x 57]  NC: 0.9988	[45 x 45]  NC: 0.9962	[98 x 98]  NC: 0.9965	[265 x 265]  NC: 1.0000	M_{nc}: 0.9635 DP: 0.0462
Rotação (30.0°) com recorte + Escala (0.8)	[57 x 57]  NC: 0.9819	[45 x 45]  NC: 0.9777	[98 x 98]  NC: 0.9288	[265 x 265]  NC: 0.9924	M_{nc}: 0.8724 DP: 0.1095
Escala (0.8)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9987	[265 x 265]  NC: 1.0000	M_{nc}: 0.9923 DP: 0.0153

Tabela 5.8: Ataques geométricos (**Parte 3**). ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques geométricos	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Escala (0.9)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 0.9985 DP: 0.0046
Escala (1.2)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 1.0000 DP: 0.0003
Escala (1.5)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 1.0000 DP: 0.0005
Escala (2.0)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 1.0000 DP: 0.0008
Escala (2.5)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 0.9999 DP: 0.0017
Rotação ($-50.5°$) com recorte	[57 x 57]  NC: 0.9976	[45 x 45]  NC: 0.9981	[98 x 98]  NC: 0.9758	[265 x 265]  NC: 0.9996	M_{nc}: 0.9443 DP: 0.0588
Rotação ($-90.0°$) com recorte	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9969	[265 x 265]  NC: 1.0000	M_{nc}: 0.9837 DP: 0.0294

Tabela 5.9: Ataques geométricos (**Parte 4**). ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques geométricos	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Rotação ($-120.1°$) com recorte	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 0.9925	[98 x 98]  NC: 0.9813	[265 x 265]  NC: 0.9993	M_{nc}: 0.9380 DP: 0.0631
Rotação (30.0°) com recorte	[57 x 57]  NC: 0.9988	[45 x 45]  NC: 0.9981	[98 x 98]  NC: 0.9987	[265 x 265]  NC: 0.9952	M_{nc}: 0.9569 DP: 0.0499
Rotação (90.5°) com recorte	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 0.9981	[98 x 98]  NC: 0.9829	[265 x 265]  NC: 0.9994	M_{nc}: 0.9564 DP: 0.0565
Rotação (30.0°) com recorte + Ajuste de iluminação ([0.2, 0.8] -> [0,1])	[57 x 57]  NC: 0.9964	[45 x 45]  NC: 0.9869	[98 x 98]  NC: 0.8665	[265 x 265]  NC: 0.9767	M_{nc}: 0.8605 DP: 0.1649
Escala (0.8) + Filtro Gaussiano (3 x 3)	[57 x 57]  NC: 0.9976	[45 x 45]  NC: 0.9943	[98 x 98]  NC: 0.9860	[265 x 265]  NC: 0.9998	M_{nc}: 0.9478 DP: 0.0546
Rotação (6.4°) com recorte + Aumento da iluminação (2.0 x)	[57 x 57]  NC: 0.9951	[45 x 45]  NC: 0.9833	[98 x 98]  NC: 0.9617	[265 x 265]  NC: 0.9999	M_{nc}: 0.8669 DP: 0.1924
Corte (1/2)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 0.9943	[98 x 98]  NC: 0.9878	[265 x 265]  NC: 1.0000	M_{nc}: 0.9484 DP: 0.0585

Tabela 5.10: Ataques geométricos (**Parte 5, final**). ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques geométricos	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Corte (1/3)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 0.9822 DP: 0.0321
Corte (1/4)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 0.9921 DP: 0.0199

Diversos ataques, demonstrados nas Tabelas 5.11 a 5.13, testam a robustez do método frente à incorporação de ruído aleatório, mudanças de contraste, mudanças de iluminação, suavização e cortes abruptos. Nesse contexto, o método proposto é capaz de recuperar a marca-d'água com sucesso na maioria dos ataques, evidenciando sua robustez e consistência mesmo diante de diferentes tipos de distorções. As tabelas a seguir dividem a demonstração de diversos ataques em 3 partes.

Tabela 5.11: Ataques diversos (**Parte 1**). ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques diversos	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Ruído 'Salt & Pepper' (0.01)	[57 x 57]  NC: 0.9902	[45 x 45]  NC: 0.9887	[98 x 98]  NC: 0.9921	[265 x 265]  NC: 0.9884	M_{nc}: 0.9852 DP: 0.0115
Ruído Gaussiano (0.005)	[57 x 57]  NC: 0.9648	[45 x 45]  NC: 0.9482	[98 x 98]  NC: 0.9546	[265 x 265]  NC: 0.9630	M_{nc}: 0.9465 DP: 0.0725

Tabela 5.12: Ataques diversos (**Parte 2**). ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques diversos	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Filtro Mediano (3 x 3)	[57 x 57]  NC: 0.9610	[45 x 45]  NC: 0.9868	[98 x 98]  NC: 0.8262	[265 x 265]  NC: 0.9780	M_{nc}: 0.7980 DP: 0.1867
Ajuste de iluminação ([0.2, 0.8] -> [0,1])	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9996	[265 x 265]  NC: 1.0000	M_{nc}: 0.9642 DP: 0.0764
Ajuste de iluminação ([0,1] -> [0.2, 0.8])	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9974	[265 x 265]  NC: 1.0000	M_{nc}: 0.9823 DP: 0.0292
Aumento do contraste (2.0 x)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9873	[265 x 265]  NC: 1.0000	M_{nc}: 0.9537 DP: 0.0583
Redução do contraste (0.5 x)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9974	[265 x 265]  NC: 1.0000	M_{nc}: 0.9845 DP: 0.0248
Filtro de Média (3 x 3)	[57 x 57]  NC: 0.9404	[45 x 45]  NC: 0.9753	[98 x 98]  NC: 0.8649	[265 x 265]  NC: 0.9438	M_{nc}: 0.8377 DP: 0.0920
Filtro Gaussiano (3 x 3)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 1.0000	[265 x 265]  NC: 1.0000	M_{nc}: 0.9990 DP: 0.0042

Tabela 5.13: Ataques diversos (**Parte 3, final**). ($\boldsymbol{M_{nc}}$: média NC, $\boldsymbol{DP}$: desvio padrão)

Ataques diversos	**Lenna**	**UnB**	**Molveno**	**Pedra do Pato**	***total_imgs***
Compressão JPEG (80) + Ruído 'Salt & Pepper' (0.01)	[57 x 57]  NC: 0.9319	[45 x 45]  NC: 0.9465	[98 x 98]  NC: 0.9177	[265 x 265]  NC: 0.9184	M_{nc}: 0.9096 DP: 0.0456
Filtro Gaussiano (3 x 3) + Redução do contraste (0.5 x)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9974	[265 x 265]  NC: 1.0000	M_{nc}: 0.9874 DP: 0.0190
Filtro de Média (3 x 3) + Redução do contraste (0.5 x)	[57 x 57]  NC: 0.8917	[45 x 45]  NC: 0.9080	[98 x 98]  NC: 0.7990	[265 x 265]  NC: 0.8913	M_{nc}: 0.7944 DP: 0.0798
Filtro gaussiano (3 x 3) + Ruído Salt & Pepper 0.01 + Contraste reduzido (0.5 x)	[57 x 57]  NC: 0.9517	[45 x 45]  NC: 0.9529	[98 x 98]  NC: 0.8799	[265 x 265]  NC: 0.9349	M_{nc}: 0.8680 DP: 0.0577
Corte (1/4) + Redução da iluminação (0.5 x)	[57 x 57]  NC: 1.0000	[45 x 45]  NC: 1.0000	[98 x 98]  NC: 0.9974	[265 x 265]  NC: 1.0000	M_{nc}: 0.9744 DP: 0.0390

O desempenho individual de cada conjunto de imagens de teste (***imgs1***, ***imgs2***, ***imgs3***, ***imgs4***), pode ser analisado separadamente no apêndice, em A.2.

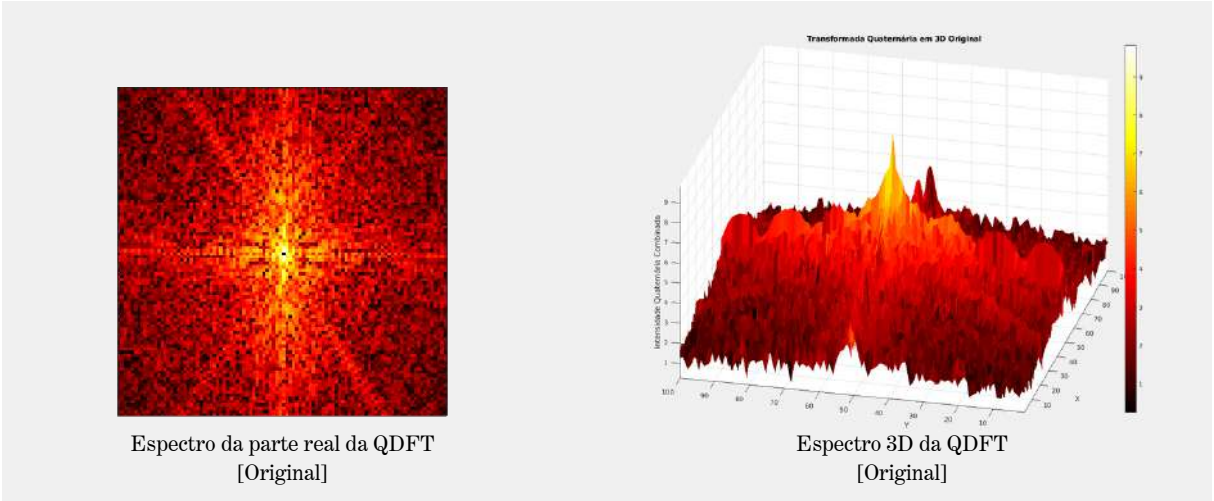
5.3 Comparação entre os Métodos

Neste tópico, os resultados são apresentados de forma a comparar a robustez e a imperceptibilidade do método de referência [3, 4] com as do método proposto neste trabalho.

5.3.1 Imperceptibilidade (Análise Comparativa)

A imperceptibilidade não tem relação direta com o tipo de abordagem (cego ou semi-cego), pois ela é determinada pela forma como as informações são inseridas e pelo impacto visual resultante na qualidade da imagem. O método proposto utiliza a QDFT da mesma forma que o método de referência [3, 4], quantizando os coeficientes de frequência da parte real dos quatérnios para embutir os bits. No entanto, em vez de utilizar o LBM, é adotado o DM, que provoca uma menor distorção na imagem ao quantizar esses coeficientes.

Ao utilizar o LBM, o coeficiente é quantizado diretamente para um ponto da grade e depois é deslocado pela adição de um deslocamento. Isso faz com que a energia fique permanentemente no sinal, o que contribui para distorção. Além disso, ao quantizar os coeficientes diretamente, muitos irão cair no mesmo ponto da grade, refletindo uma alteração regular quando analisados em conjunto. No método de referência, esse efeito é agravado pela incorporação agrupada dos bits no domínio da frequência. Como resultado, além de destacar as modificações nos coeficientes, esse procedimento gera artefatos perceptíveis na imagem ao retornar para o domínio espacial. A Figura 5.7 demonstra o espectro de frequência sem modificações, enquanto a Figura 5.8 ilustra esse mesmo espectro após ser quantizado pelo LBM de acordo com o método de referência.



Espectro da parte real da QDFT [Original]

Espectro 3D da QDFT [Original]

Figura 5.7: Espectro da parte real da QDFT e o espectro 3D da QDFT completa (100×100). (Fonte: Autoral).

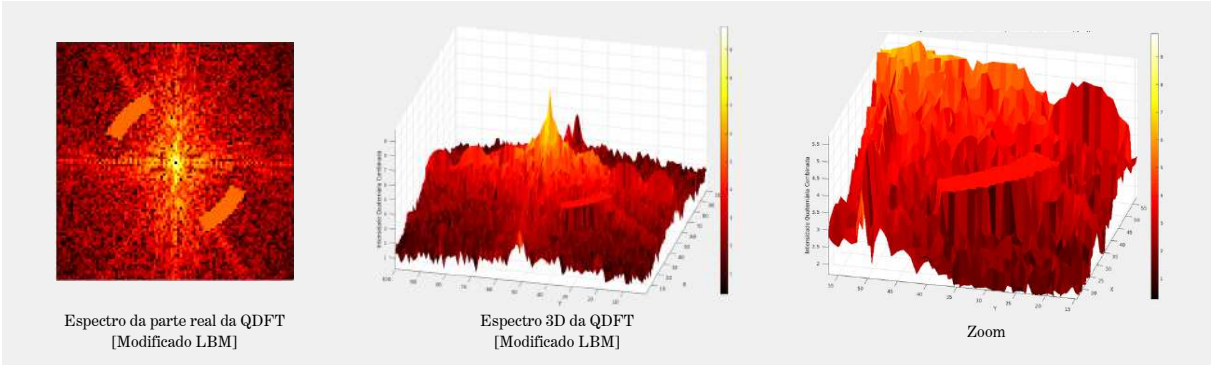


Figura 5.8: Espectro modificado pelo LBM (100×100). A área mais destacada é o local quantizado para representar os bits da marca-d'água. (Fonte: Autoral).

No artigo que descreve o método de referência [3], a imagem marcada (Lenna) exibia artefatos ondulatórios que não haviam sido explicados. Após inúmeros testes, variando padrões e locais de incorporação, ficou claro que a incorporação dos bits de forma agrupada, na área indicada pela Figura 5.8, aliada ao uso do LBM, era o principal causador dos artefatos.

A Figura 5.9 apresenta a imagem marcada extraída do artigo de referência e a imagem marcada obtida ao inserir os bits de forma agrupada na área indicada pela Figura 5.8.

(a) Imagem do artigo de referência

(b) Imagem marcada seguindo a estratégia agrupada dos bits

Figura 5.9: **(a)** 40 dB (512×512) **(b)** 40.0717 dB (512×512). (Fonte: Autoral).

Diante dos artefatos visuais perceptíveis gerados pelo método de referência, foi preciso buscar uma alternativa. Primeiramente, o LBM foi substituído pelo DM, com o objetivo de reduzir os padrões regulares introduzidos pela quantização direta. Ao contrário do LBM, o DM adiciona um deslocamento (ou Dither) aleatório ao coeficiente antes da quantização, que depois é quantizado para o ponto mais próximo da grade de quantização base. Após isso, o mesmo deslocamento é subtraído do coeficiente quantizado, fazendo com que a energia do deslocamento não contribua diretamente para a distorção final, o que suaviza as modificações e evita a formação de padrões. Além disso, a adição do deslocamento aleatório contribui para o aumento da robustez ao introduzir variações naturais entre os coeficientes. A Figura 5.10 demonstra o efeito da substituição direta do LBM pelo DM, onde é possível observar que, apesar da modificação ainda ser evidente, a superfície resultante apresenta uma distribuição mais irregular, reduzindo os padrões uniformes característicos do LBM.

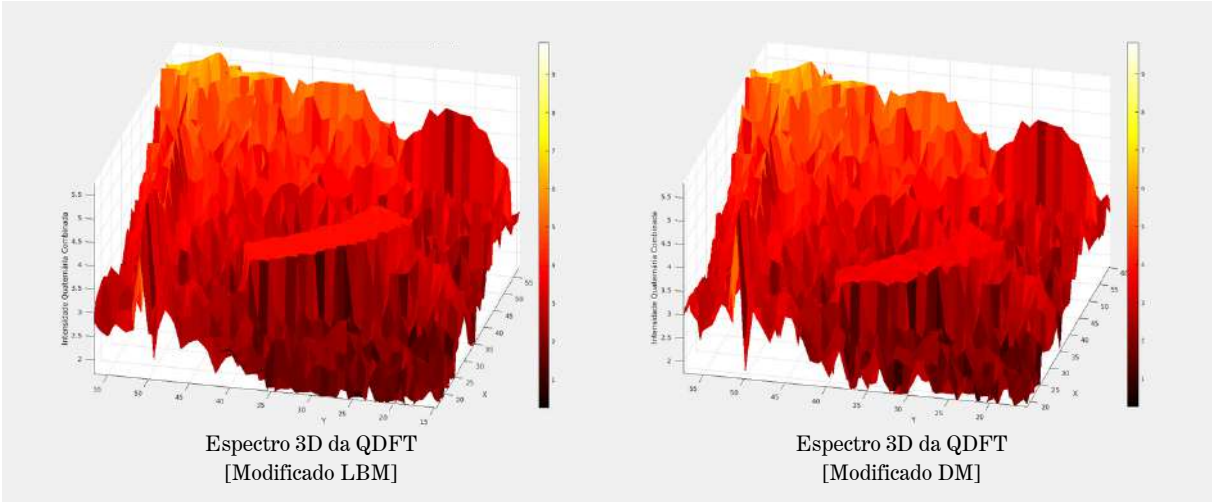


Figura 5.10: LBM versus DM. (Fonte: Autoral).

Portanto, após a substituição do LBM pelo DM, a estratégia final foi dispersar os bits por todo o espectro, conforme descrito na metodologia. A Figura 5.11 ilustra o espectro original sem modificações comparado ao espectro modificado utilizando o método proposto, demonstrando uma incorporação mais natural ao espectro.

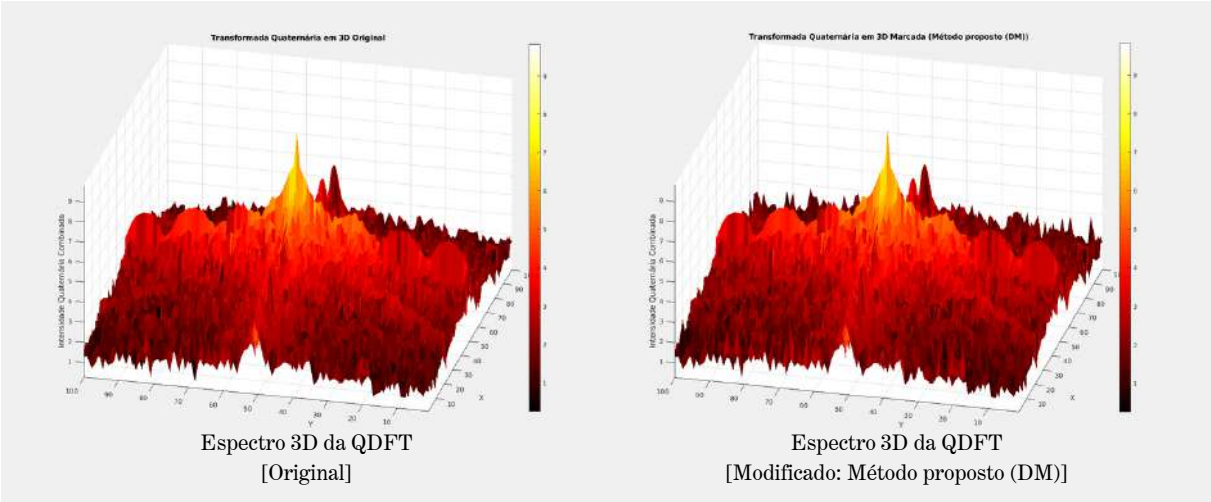
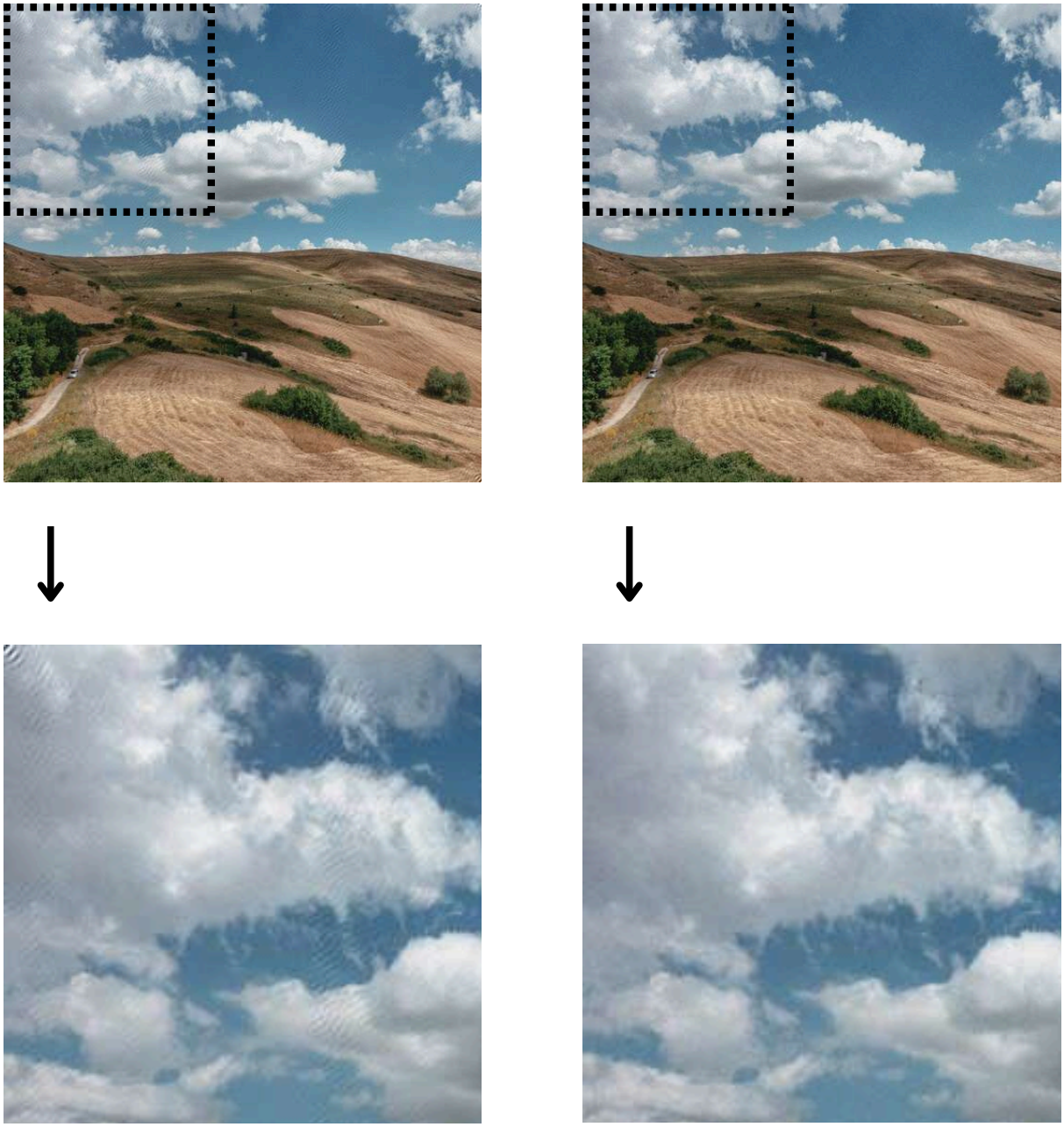


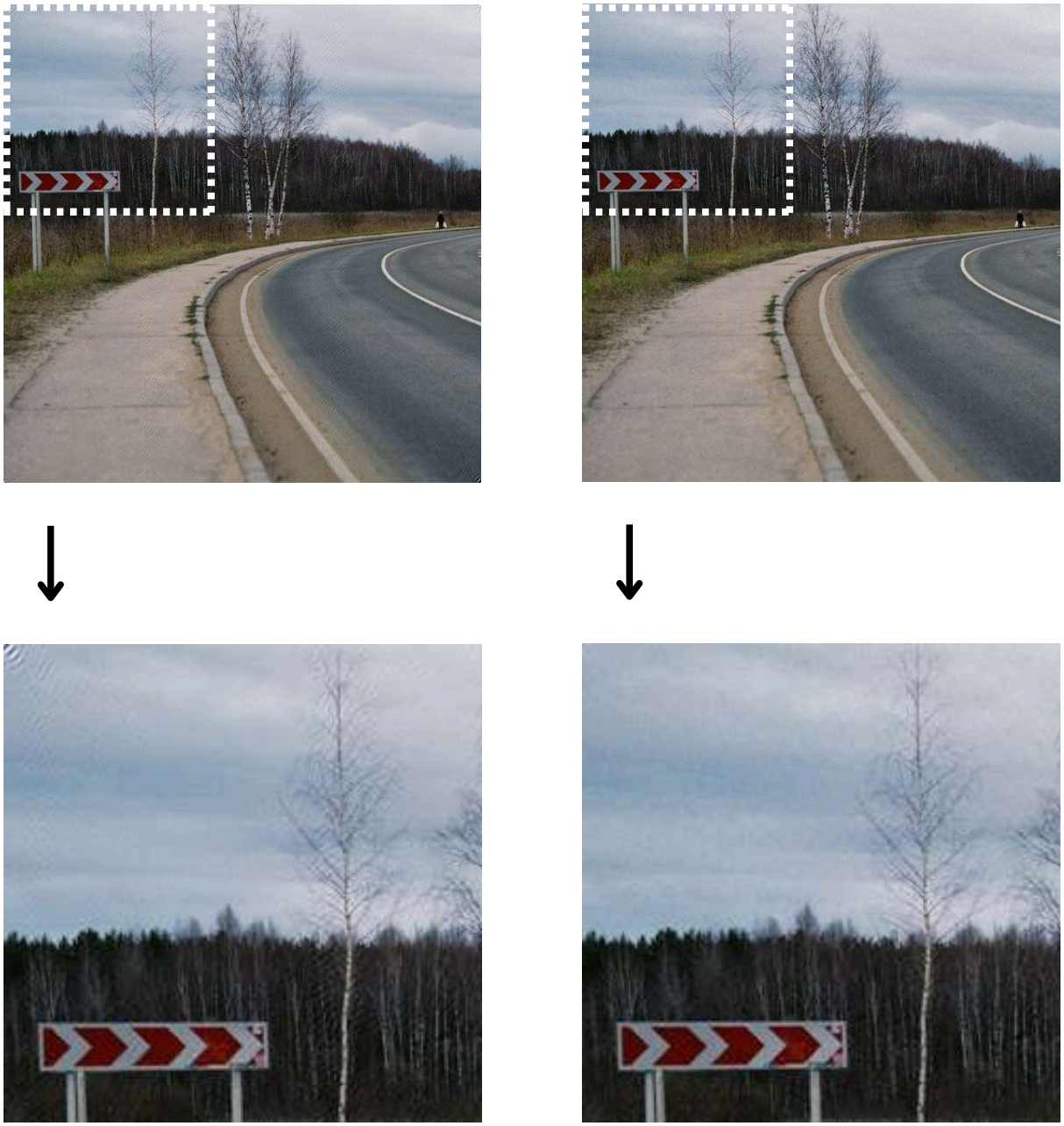
Figura 5.11: Espectro orignal versus espectro modificado pelo método proposto. (Fonte: Autoral).

As imagens a seguir ilustram o impacto visual da melhoria perceptual introduzida pelo método proposto. Apesar dos valores de PSNR serem próximos, o nível de distorção perceptível introduzida pelo método de referência é evidentemente maior.



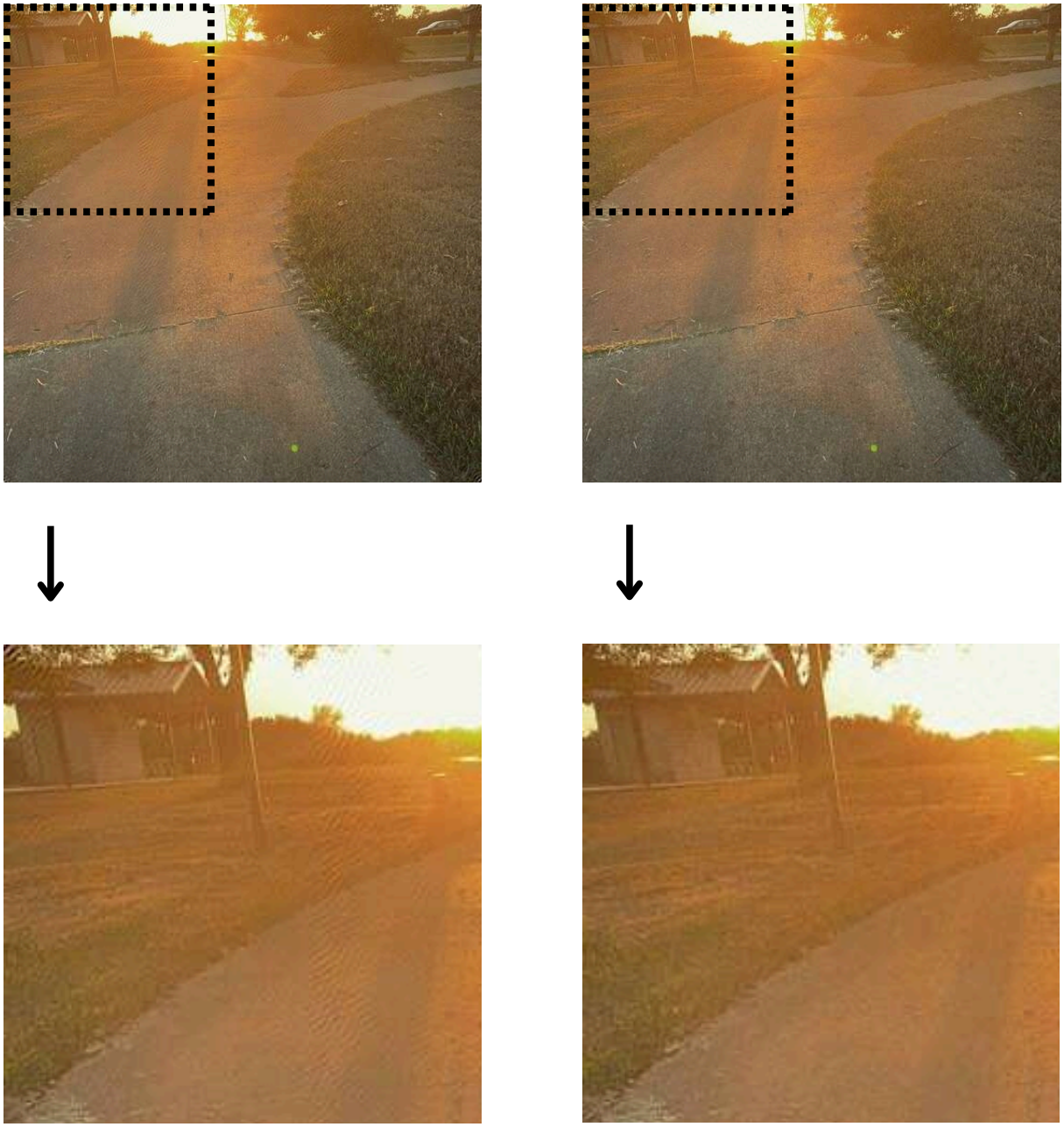
(a) Imagem marcada: QDFT-ULPM **(b)** Imagem marcada: **Método proposto**

Figura 5.12: **(a)** PSNR 39.3070 dB **(b)** PSNR 40.3260 dB. (Imagem utilizada no processo: Accadia, Josh Chiodo [1]).



(a) Imagem marcada: QDFT-ULPM **(b)** Imagem marcada: **Método proposto**

Figura 5.13: **(a)** PSNR 39.0832 dB **(b)** PSNR 40.2489 dB. (Imagem utilizada no processo: Dubna, Ivan Stepanov [1]).



(a) Imagem marcada: QDFT-ULPM **(b)** Imagem marcada: **Método proposto**

Figura 5.14: **(a)** PSNR 39.2085 dB **(b)** PSNR 39.9391 dB. (Imagem utilizada no processo: Street, Isabella Cassady [1]).

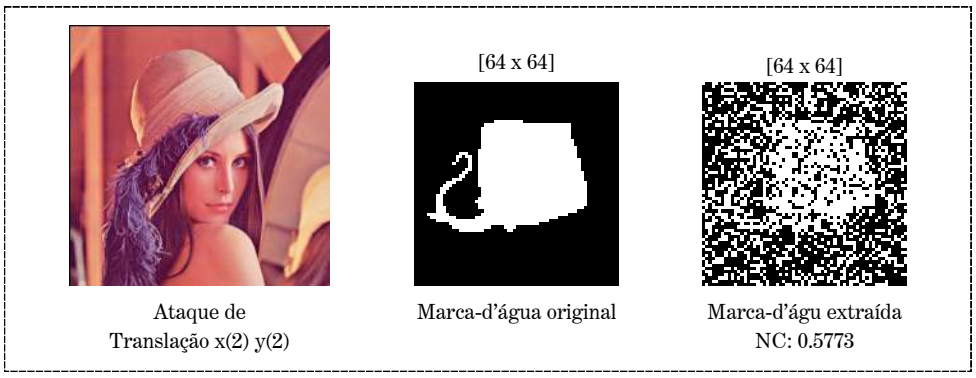
5.3.2 Robustez (Análise Comparativa)

A abordagem cega do método de referência [3, 4] se mostra mais vulnerável às transformações em cenários práticos de compartilhamento de imagens (compressão JPEG, redimensionamentos, translações, entre outros), o que pode comprometer a detecção da

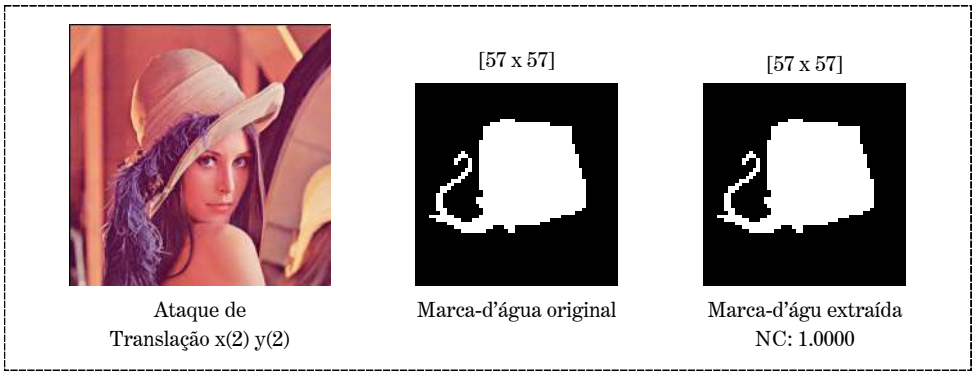
marca-d'água. Por outro lado, a abordagem semi-cega do método proposto mantém um desempenho superior mesmo nesses cenários.

O método de referência utiliza o ULPM para reverter transformações geométricas, onde um padrão de rastreamento é embutido na magnitude da QDFT e correlacionado posteriormente para estimar transformações geométricas. Como já abordado, o ULPM converte rotações para deslocamentos ao longo do eixo angular e escalonamentos em deslocamentos no eixo radial. No entanto, em casos de translação o ULPM não é capaz de detectar ou compensar esse tipo de transformação, pois a translação não é interpretada como um deslocamento no domínio log-polar, levando à falha total na extração da marca-d'água.

No caso de translações muito pequenas, como deslocamentos de 1 ou 2 pixels, a distribuição de frequências na QDFT não se altera de forma significativa, o que permite a recuperação parcial dos bits pelo método de referência, conforme demonstrado na Figura 5.15.



(a) Método de referência

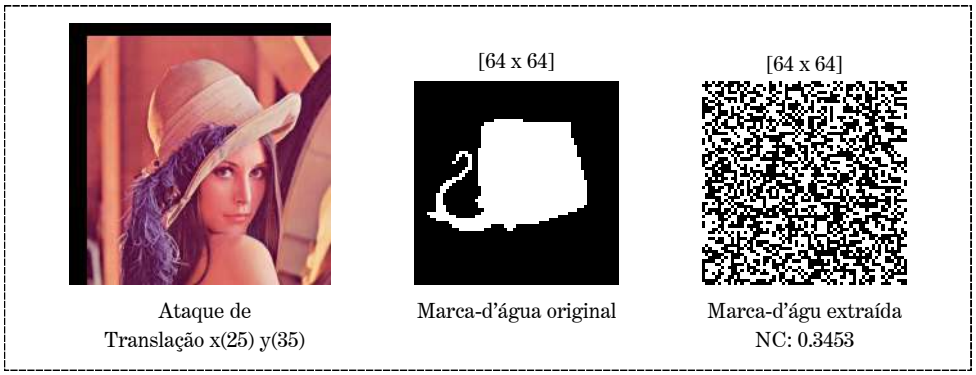


(b) Método proposto

Figura 5.15: **(a)** Extração da marca-d'água pelo método de referência, com 68.5059% da sequencia de bits recuperada. **(b)** Extração pelo método proposto. (Fonte: Autoral).

Porém, em deslocamentos maiores a estrutura espectral da QDFT é alterada, fazendo

com que a extração da marca-d'água falhe totalmente. A Figura 5.16 apresenta o resultado da extração da marca-d'água em ambos os métodos após uma translação de 25 pixels no eixo x e 35 pixels no eixo y.



(a) Método de referência

(b) Método proposto

Figura 5.16: **(a)** Extração da marca-d'água pelo método de referência, com 48.6084% da sequencia de bits recuperada. **(b)** Extração pelo método proposto. (Fonte: Autoral).

No caso de uma rotação com expansão, o ULPM consegue detectar e corrigir a rotação, porém o movimento do eixo radial causado pela escala não é detectado em decorrência do movimento de translação, resultando na imagem corrigida angularmente mas com a resolução incorreta. A Figura 5.17 demonstra tentativa de correção pelo ULPM.

Figura 5.17: Imagem marcada rotacionada e corrigida pelo ULPM (Fonte: Autoral).

A Figura 5.18 apresenta a marca-d'água extraída após a correção pelo ULPM, demonstrando o impacto de um ataque de rotação com expansão, onde a imagem é escalada e transladada para o centro, e depois é rotacionada.

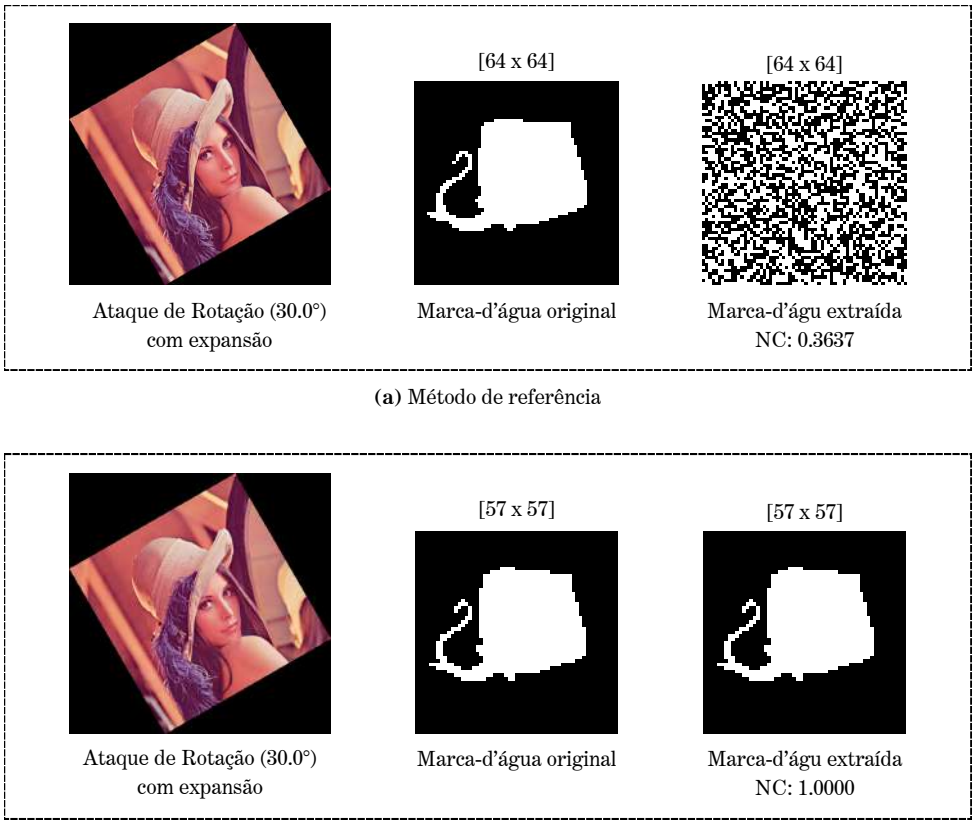


Figura 5.18: **(a)** Extração da marca-d'água pelo método de referência, com 50% da sequencia de bits recuperada. **(b)** Extração pelo método proposto. (Fonte: Autoral).

A combinação do ULPM com a QDFT transforma o método de referência em uma abordagem totalmente cega de marca-d'água, o que representa uma grande vantagem por dispensar o armazenamento de informações da imagem original. No entanto, o ULPM apresenta uma grande limitação, a incapacidade de detectar translações, onde translações mínimas já comprometem a extração da marca-d'água, enquanto que translações mais significativas podem corrompê-la totalmente. A versão aprimorada do método de referência propõe melhorias ao ULPM, dando origem ao IULPM, porém, a melhoria é apenas na precisão de detecção dos ângulos de rotação, passando de 0,5 graus no ULPM para 0,1 grau no IULPM. Portanto, tanto o método de referência [3] quanto sua versão aprimorada [4] mantêm a mesma limitação, que os torna vulneráveis a transformações geométricas comuns em cenários práticos de compartilhamento de imagens, prejudicando a confiabilidade da detecção e extração da marca-d'água.

Diante desse cenário, a utilização do SIFT, para extrair as características da imagem marcada e armazená-las junto aos demais parâmetros mostrou ser uma estratégia robusta para lidar com transformações geométricas. No entanto, isso faz com que a combinação do SIFT com a QDFT transforme o método proposto em uma abordagem semi-cega de marca-d'água, pois informações referentes à imagem original são utilizadas para ajustar a imagem para o processo de extração. A Tabela 5.14 apresenta a comparação entre o método de referência e o método proposto frente a ataques geométricos, listando a média dos valores de NC da marca-d'água extraída em cada método.

Tabela 5.14: NC médio; Ataques de geométricos (Método de referência [3] vs Método proposto). Conjunto de testes: ***imgs1***, ***imgs2*** e ***imgs3***. Marca-d'água inserida: **V2**

Ataques Geométricos	**QDFT-ULPM [3]**	**Método proposto**
Reflexão vertical	0.3672	**0.8933**
Reflexão horizontal	0.3672	**0.8979**
Translação x(0.4%) y(0.4%)	0.1371	**0.9981**
Translação x(5%) y(7%)	0.3554	**0.9939**
Translação x(22%) y(12%)	0.3835	**0.9651**
Translação x(2%) y(0.19%)	0.2830	**0.9962**
Rotação (30.0°) com expansão	0.3666	**0.9999**
Rotação (900.5°) com expansão	0.3592	**1.0000**
Rotação (58.0°) com recorte	0.8479	**0.9689**
Rotação (−215.0°) com recorte	0.8328	**0.9757**
Rotação (334.0°) com recorte	0.9029	**0.9813**
Escala (0.8)	0.9888	**0.9961**
Escala (0.9)	0.9937	**0.9993**
Escala (1.2)	0.9995	**1.0000**
Escala (1.5)	0.9998	**1.0000**
Escala (2.0)	0.3774	**1.0000**
Escala (2.5)	0.3603	**1.0000**
Rotação (30.0°) com recorte + Escala (0.8)	0.6385	**0.9272**

A utilização da redundância foi um ponto crucial, pois incorporar cópias da marca-d'água em todas as faixas de frequência deixou o método robusto a diferentes tipos de ataques. Por exemplo, se a imagem marcada passar por uma compressão JPEG, as altas frequências serão afetadas diretamente, porém as baixas frequências ficarão intactas, ou seja, apenas uma cópia foi corrompida deixando as outras duas intactas. Agora no caso de um filtro passa-altas, as baixas frequências seriam afetadas e as altas frequências ficariam intactas. Quando há uma combinação de ambos os ataques, as altas e baixas frequências são afetadas ao mesmo tempo, fazendo com que a faixa de média frequência seja a cópia sobrevivente. A determinação da cópia sobrevivente poderia ser feita com um método de votação majoritária entre os bits das cópias de redundância, porém, considerando que as

características da imagem já estavam sendo exploradas por meio do SIFT, já classificando o método como semi-cego, optou-se por utilizar a marca-d'água original como referência para selecionar a cópia com menor taxa de erro, aumentando a precisão da detecção. Mesmo com essa adição, o sistema permanece classificado como semi-cego, pois apenas informações parciais da imagem original são utilizadas. A Tabela 5.15 demonstra que o método proposto é superior ao método de referência em casos onde a compressão é mais agressiva.

Tabela 5.15: NC médio; Ataques de compressão (Método de referência [3] vs Método proposto). Conjunto de testes: ***imgs1***, ***imgs2*** e ***imgs3***. Marca-d'água inserida: **V2**

Ataques de compressão	**QDFT-ULPM [3]**	**Método proposto**
Compressão JPEG (80)	0.9954	**0.9997**
Compressão JPEG (50)	0.8269	**0.9662**
Compressão JPEG (40)	0.7314	**0.9568**
Compressão JPEG (20)	0.4603	**0.8545**

As tabelas a seguir reúnem os resultados comparativos entre o método de referência, sua versão aprimorada e o método proposto.

Tabela 5.16: NC médio; Ataques diversos em imagens 512 × 512 (Método de referência [3] (e sua versão aprimorada [4]) vs Método proposto). Conjunto de testes: ***imgs1*** e ***imgs3***. Marca-d'água inserida: **V2**

Ataques diversos	**QDFT-ULPM [3]**	**QDFT-IULPM [4]**	**Método proposto**
Rotação (−50.5°) com recorte	0.8320	0.7585	**0.9857**
Rotação (−90.0°) com recorte	0.9930	0.9853	**1.0000**
Rotação (−120.1°) com recorte	0.7544	0.7536	**0.9873**
Rotação (6.4°) com recorte + Aumento da iluminação (2.0 x)	0.5628	0.7144	**0.9463**
Escala (0.8) + Filtro Gaussiano (3 x 3)	0.9548	0.7073	**0.9850**
Corte (1/3)	0.8882	0.9835	**0.9923**
Corte (1/2)	0.7689	0.9796	**0.9806**
Escala (1.2)	0.9984	0.9768	**1.0000**
Escala (0.9)	0.9831	**0.9999**	0.9997
Corte (1/4) + Redução da iluminação (0.5 x)	0.9124	**0.9921**	0.9892

Tabela 5.17: NC médio; Ataques diversos (Método de referência [3] (e sua versão aprimorada [4]) vs Método proposto). Conjunto de testes: ***imgs1***, ***imgs2*** e ***imgs3***. Marca-d'água inserida: **V2**

Ataques diversos	**QDFT-ULPM [3]**	**QDFT-IULPM [4]**	**Método proposto**
Ruído 'Salt & Pepper' (0.01)	0.9692	0.9388	**0.9939**
Ruído Gaussiano (0.005)	0.9437	0.8460	**0.9733**
Ajuste de iluminação ([0.2, 0.8] -> [0,1])	0.8798	0.9540	**0.9790**
Aumento do contraste (2.0 x)	0.8346	0.8365	**0.9765**
Redução do contraste (0.5 x)	0.9842	0.9597	**0.9925**
Filtro Gaussiano (3 x 3)	0.9902	0.9912	**0.9996**
Compressão JPEG (80)	0.9954	0.9559	**0.9997**
Compressão JPEG (50)	0.8269	0.8040	**0.9662**
Compressão JPEG (80) + Ruído 'Salt & Pepper' (0.01)	0.9209	0.7864	**0.9589**
Filtro Gaussiano (3 x 3) + Redução do contraste (0.5 x)	0.9916	0.9773	**0.9939**
Ajuste de iluminação ([0,1] -> [0.2, 0.8])	0.9842	**0.9997**	0.9912
Filtro Mediano (3 x 3)	0.8871	**0.8743**	0.8640
Filtro gaussiano (3 x 3) + Ruído Salt & Pepper 0.01 + Redução do contraste (0.5 x)	0.9002	**0.9537**	0.9342
Filtro de Média (3 x 3)	**0.9916**	0.9618	0.9121
Filtro de Média (3 x 3) + Redução do contraste (0.5 x)	**0.9916**	0.9389	0.8882

O próximo capítulo (Capítulo 6) apresentará a conclusão, formalizando os pontos mais importantes ao longo do projeto.

Capítulo 6

Conclusão

O objetivo principal do método proposto neste trabalho foi implementar melhorias ao método de referência [3, 4] em relação a imperceptibilidade e robustez a determinados tipos de transformações geométricas. O método de referência demonstrou problemas com artefatos perceptíveis na imagem resultante do processo de incorporação dos bits da marca-d'água. Além disso, a robustez contra transformações geométricas que envolvem translações se demonstrou insuficiente, implicando em falhas na extração da marca-d'água. Diante disso, a proposta de uma nova estratégia de inserção de bits apresentou um alto desempenho em termos de robustez e imperceptibilidade, onde a utilização de redundância foi um ponto crucial para resistir a ataques combinados. Adicionalmente, a utilização do SIFT se mostrou uma solução robusta e eficiente em lidar com transformações geométricas mais agressivas, obtendo em testes altas taxas de recuperação da marca-d'água. Portanto, a proposta deste trabalho em resolver as limitações do método de referência se mostrou promissora e gerou resultados satisfatórios, de forma que amplos testes em diferentes conjuntos de imagem evidenciaram a robustez e imperceptibilidade superior do método proposto. O objetivo em trabalhos futuros é estender e otimizar o método para o ambiente de produção.

Referências

[1] Ribas, Lucas: *Images for Research Purposes: Um Banco de Imagens para Estudos em Processamento de Imagens Digitais*. `https://github.com/ribas858/images-for-research-purposes`, 2025. xii, xiii, 90, 93, 110, 111, 112, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137

[2] Zhu, Yuehan, Tomohiro Fukuda e Nobuyoshi Yabuki: *A mixed reality design system for interior renovation: Inpainting with 360-degree live streaming and generative adversarial networks after removal*. Technologies, 12(1), 2024, ISSN 2227-7080. `https://www.mdpi.com/2227-7080/12/1/9`. xiv, 30

[3] Ouyang, Junlin, Huazhong Shu, Xingzi Wen, Jiasong Wu, Fan Liao e Gouenou Coatrieux: *A blind robust color image watermarking method using quaternion fourier transform*. Em *2013 6th International Congress on Image and Signal Processing (CISP)*, volume 01, páginas 485–489, 2013. xiv, xv, 3, 4, 14, 23, 33, 39, 88, 89, 90, 91, 105, 106, 107, 112, 116, 117, 118, 119, 120, 121

[4] Ouyang, Junlin, Gouenou Coatrieux, Beijing Chen e Huazhong Shu: *Color image watermarking based on quaternion fourier transform and improved uniform log-polar mapping*. Computers & Electrical Engineering, 46:419–432, 2015, ISSN 0045-7906. `https://www.sciencedirect.com/science/article/pii/S0045790615000725`. xv, 28, 29, 34, 88, 89, 90, 91, 105, 106, 112, 116, 119, 120, 121

[5] Zhu, Runze: *Information Security and Privacy Protection Based on Intelligent Encryption and Cryptography* . Em *2023 International Conference on Applied Physics and Computing (ICAPC)*, páginas 395–400, Los Alamitos, CA, USA, dezembro 2023. IEEE Computer Society. `https://doi.ieeecomputersociety.org/10.1109/ICAPC61546.2023.00080`. 2

[6] Zhang, Xiaomei, Hu Guan, Ying Huang e Shuwu Zhang: *Research on digital image watermarking technology*. Em *2020 International Conference on Culture-oriented Science & Technology (ICCST)*, páginas 330–335, 2020. 2, 22

[7] Vartumyan, A.A., G.G. Mikhailov, E.V. Galdin, T.N. Lavrova e V.N. Orobinskaya: *Risks associated with the use of artificial intelligence in various fields of science*. Em *2023 5th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, páginas 434–438, 2023. 2

[8] Kolomeets, Maxim, Han Wu, Lei Shi e Aad van Moorsel: *The face of deception: The impact of ai-generated photos on malicious social bots*. IEEE Transactions on Computational Social Systems, 12(3):1080–1091, 2025. 2

[9] Xuehua, Jiang: *Digital watermarking and its application in image copyright protection.* Em *2010 International Conference on Intelligent Computation Technology and Automation*, volume 2, páginas 114–117, 2010. 2

[10] Bhatti, Uzair Aslam, Zhaoyuan Yu, Jingbing Li, Saqib Ali Nawaz, Anum Mehmood, Kun Zhang e Linwang Yuan: *Hybrid watermarking algorithm using clifford algebra with arnold scrambling and chaotic encryption.* IEEE Access, 8:76386–76398, 2020. 3

[11] Su, Qingtang, Decheng Liu e Yehan Sun: *A robust adaptive blind color image watermarking for resisting geometric attacks.* Information Sciences, 606:194–212, 2022, ISSN 0020-0255. `https://www.sciencedirect.com/science/article/pii/S0020025522004789`. 3

[12] Dong, Ming: *The problems and countermeasures of computer network information security.* Em *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, páginas 1806–1809, 2020. 5

[13] Stallings, William: *Cryptography and Network Security Principles and Practices, Fourth Edition.* Prentice Hall, 4ª edição, November 2005, ISBN 0131873164, 9780131873162. 5

[14] Paar, Christof e Jan Pelzl: *Understanding Cryptography: A Textbook for Students and Practitioners.* Springer Publishing Company, Incorporated, 1st edição, 2009, ISBN 3642041000. 5

[15] McEvoy, Robert, James Curran, Paul Cotter e Colin Murphy: *Fortuna: Cryptographically secure pseudo-random number generation in software and hardware.* Em *2006 IET Irish Signals and Systems Conference*, páginas 457–462, 2006. 6

[16] Boussif, Mohamed: *On the security of advanced encryption standard (aes).* Em *2022 8th International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, páginas 83–88, 2022. 6

[17] Ketata, Rim, Lobna Kriaa, Leila Azzouz Saidane e Gérard Chalhoub: *Detailed analysis of the aes ctr mode parallel execution using openmp.* Em *2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*, páginas 1–9, 2016. 7

[18] Barker, Elaine e John Kelsey: *Recommendation for random number generation using deterministic random bit generators.* Relatório Técnico SP 800-90A Rev. 1, National Institute of Standards and Technology (NIST), June 2015. `https://doi.org/10.6028/NIST.SP.800-90Ar1`. 7

[19] Cohney, Shaanan, Andrew Kwong, Shahar Paz, Daniel Genkin, Nadia Heninger, Eyal Ronen e Yuval Yarom: *Pseudorandom black swans: Cache attacks on ctr_drbg.* Em *2020 IEEE Symposium on Security and Privacy (SP)*, páginas 1241–1258, 2020. 7

[20] Sharma, Shashwat: *A comprehensive study of cryptographic hash functions.* Em *Defence Materials and Stores Research and Development Establishment*, June 2024. 7, 8

[21] Standards, National Institute of e Technology (NIST): *Secure hash signature standard (shs) (fips pub 180-2).* Relatório Técnico FIPS PUB 180-2, National Institute of Standards and Technology, Gaithersburg, MD, August 2002. `https://csrc.nist.gov/files/pubs/fips/180-2/final/docs/fips180-2.pdf`, Federal Information Processing Standard Publication. 8

[22] Biryukov, Alex, Daniel Dinu, Dmitry Khovratovich e Simon Josefsson: *Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications.* RFC 9106, setembro 2021. `https://www.rfc-editor.org/info/rfc9106`. 9

[23] Gonzalez, Rafael C. e Richard E. Woods: *Digital image processing 2Nd Ed.* Prentice-Hall, 2. ed. edição, 2002. 9, 10, 11, 13, 19

[24] Kulkarni, Pranesh K e Girish Kulkarni: *A copyright protection scheme for grayscale images using wavelet transform and arnold transform.* Em *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*, páginas 1–6, 2020. 11, 12, 21, 22

[25] Wei, Ding: *Digital image transformation and information hiding and disguising technology.* Chinese Journal of Computers, 1998. `https://api.semanticscholar.org/CorpusID:64080530`. 12

[26] Wu, Lingling, Jianwei Zhang, Weitao Deng e Dongyan He: *Arnold transformation algorithm and anti-arnold transformation algorithm.* Em *2009 First International Conference on Information Science and Engineering*, páginas 1164–1167, 2009. 12

[27] Wang, Huanying, Zihan Yuan, Siyu Chen e Qingtang Su: *Embedding color watermark image to color host image based on 2d-dct.* Optik, 274:170585, 2023, ISSN 0030-4026. `https://www.sciencedirect.com/science/article/pii/S0030402623000815`. 12

[28] Wang, Xiang yang, Chun peng Wang, Hong ying Yang e Pan pan Niu: *A robust blind color image watermarking in quaternion fourier transform domain.* Journal of Systems and Software, 86(2):255–277, 2013, ISSN 0164-1212. `https://www.sciencedirect.com/science/article/pii/S0164121212002312`. 13, 62

[29] Araujo, H. e J.M. Dias: *An introduction to the log-polar mapping.* Em *Proceedings II Workshop on Cybernetic Vision*, páginas 139–144, 1996. 14, 15

[30] Kang, Xiangui, Jiwu Huang e Wenjun Zeng: *Efficient general print-scanning resilient data hiding based on uniform log-polar mapping.* IEEE Transactions on Information Forensics and Security, 5(1):1–12, 2010. 18, 19, 20, 21

[31] Zheng, D., J. Zhao e A. El Saddik: *Rst-invariant digital image watermarking based on log-polar mapping and phase correlation.* IEEE Transactions on Circuits and Systems for Video Technology, 13(8):753–765, 2003. 20

[32] Cayre, F., C. Fontaine e T. Furon: *Watermarking security: theory and practice.* IEEE Transactions on Signal Processing, 53(10):3976–3987, 2005. 21

[33] Panchal, Urvi H. e Rohit Srivastava: *A comprehensive survey on digital image watermarking techniques*. Em *2015 Fifth International Conference on Communication Systems and Network Technologies*, páginas 591–595, 2015. 21

[34] Bt, Azizah, Mazdak Zamani, Sagheb Kohpayeh e Tanya Koohpayeh: *Taxonomy and performance evaluation of feature based extraction techniques in digital image watermarking*. Em *Third International Conference on Advances In Computing, Control And Networking -ACCN 2015*, 2015. 22

[35] Chen, B. e G.W. Wornell: *Quantization index modulation: a class of provably good methods for digital watermarking and information embedding*. IEEE Transactions on Information Theory, 47(4):1423–1443, 2001. 22

[36] Lowe, David G.: *Distinctive image features from scale-invariant keypoints*. Int. J. Comput. Vision, 60(2):91–110, novembro 2004, ISSN 0920-5691. `https://doi.org/10.1023/B:VISI.0000029664.99615.94`. 23, 25, 27

[37] Daixian, Zhu: *Sift algorithm analysis and optimization*. Em *2010 International Conference on Image Analysis and Signal Processing*, páginas 415–419, 2010. 24

[38] Arandjelović, Relja e Andrew Zisserman: *Three things everyone should know to improve object retrieval*. Em *2012 IEEE Conference on Computer Vision and Pattern Recognition*, páginas 2911–2918, 2012. 27

[39] Varshney, Yukti: *Attacks on digital watermarks: Classification, implications, benchmarks*. International Journal on Emerging Technologies (Special Issue NCETST-2017), páginas 229–235, 2017. `https://www.researchtrend.net/ijet/pdf/58-F-731.pdf`. 27

[40] Meselhy Eltoukhy, Mohamed, Ayman E. Khedr, Mostafa M. Abdel-Aziz e Khalid M. Hosny: *Robust watermarking method for securing color medical images using slant-svd-qft transforms and otp encryption*. Alexandria Engineering Journal, 78:517–529, 2023, ISSN 1110-0168. `https://www.sciencedirect.com/science/article/pii/S1110016823006580`. 28, 29

[41] Raguram, Rahul, Jan Michael Frahm e Marc Pollefeys: *Exploiting uncertainty in random sample consensus*. Em *2009 IEEE 12th International Conference on Computer Vision*, páginas 2074–2081, 2009. 31

[42] Yu, Huili, Shalini Keshavamurthy, He Bai, Sameer Sheorey, Hieu Nguyen e Clark N. Taylor: *Uncertainty estimation for random sample consensus*. Em *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, páginas 395–400, 2014. 31, 32

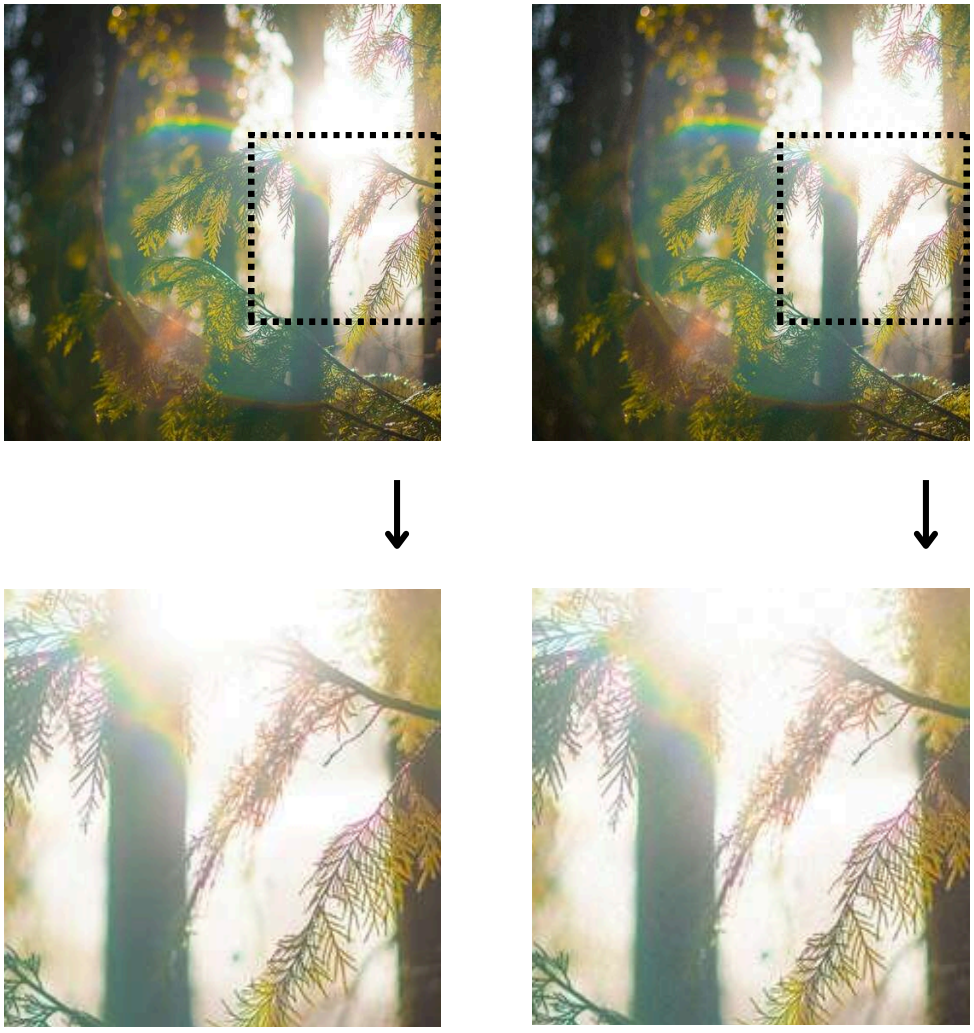
[43] Dataset, Corel 1000: *Corel 1000 dataset*. `https://wang.ist.psu.edu/docs/related/`, 2001. 1000 imagens de teste. 90

[44] Hills, J. e A. Bagnall: *MPEG-7 CE Shape-1 Part B*. `https://timeseriesclassification.com/description.php?Dataset=ShapesAll`. Dataset derivado do MPEG-7 CE Shape-1 Part B, parte do UCR/UEA Time Series Classification Archive. 91

Apêndice A

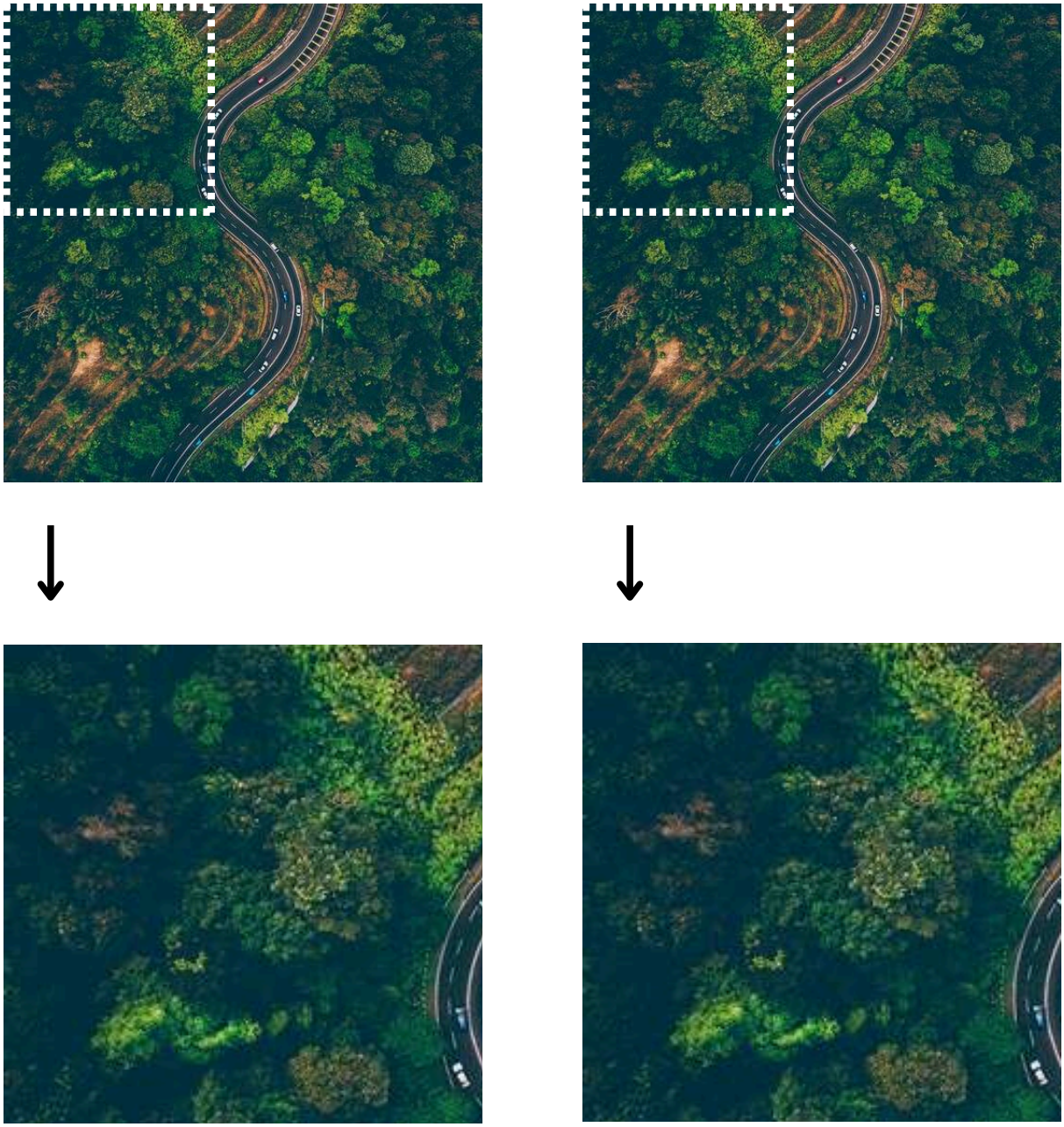
Complemento aos Resultados

A.1 Imagens Marcadas



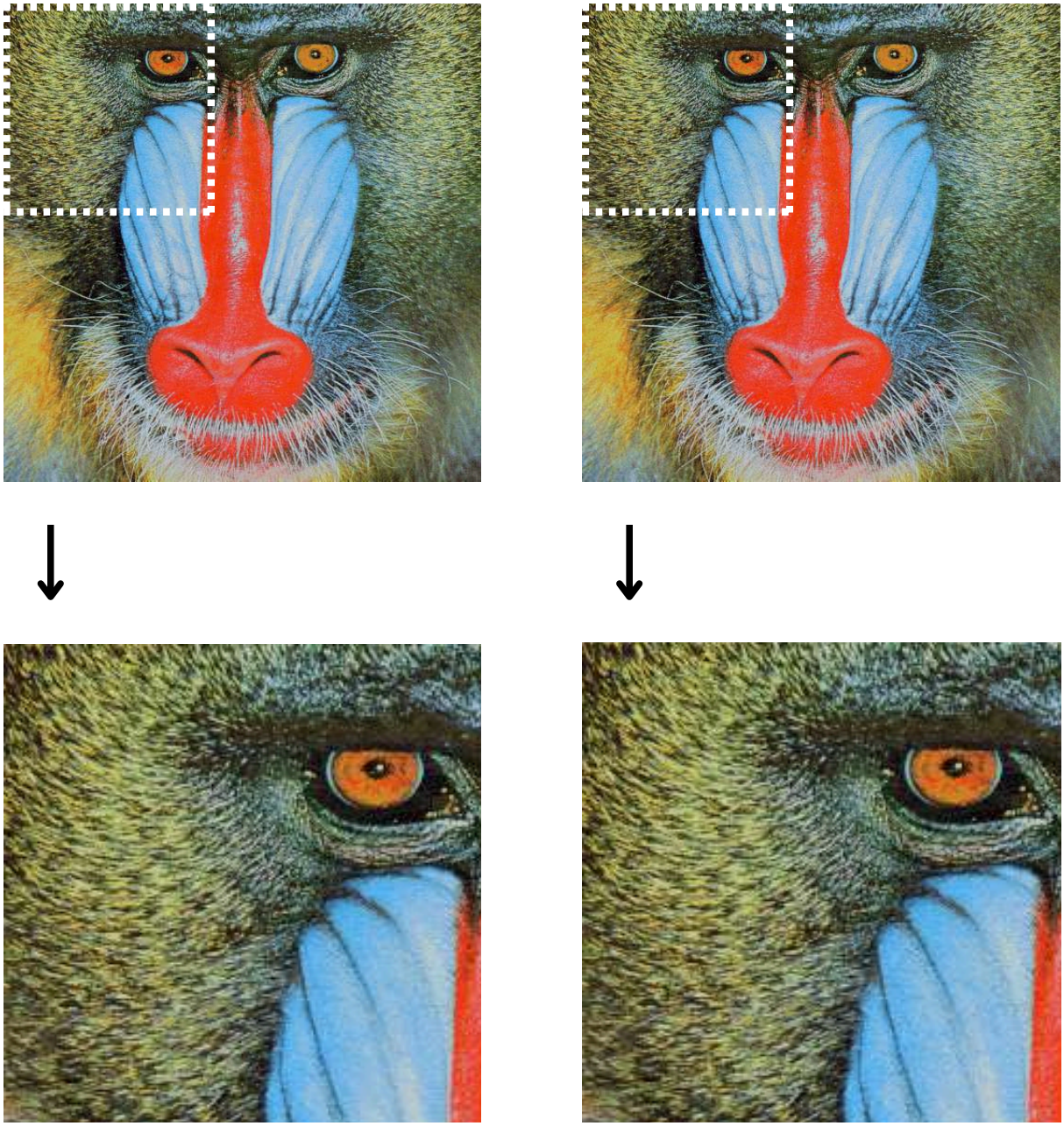
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.1: PSNR 40.5793 dB. (Imagem utilizada no processo: Light, Caspar Rae [1]).



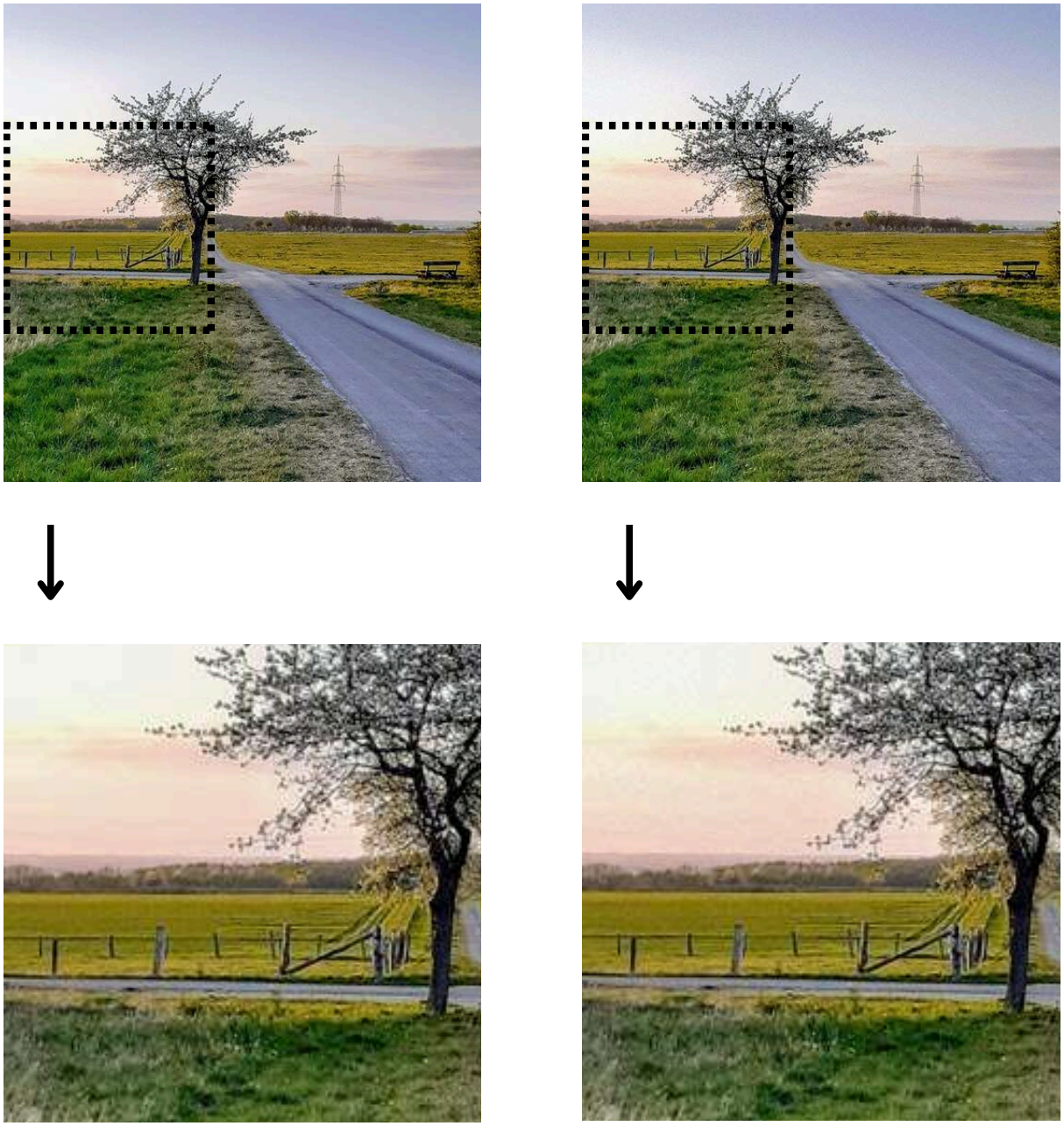
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.2: PSNR 39.0364 dB. (Imagem utilizada no processo: Malaysia, Deva Darshan [1]).



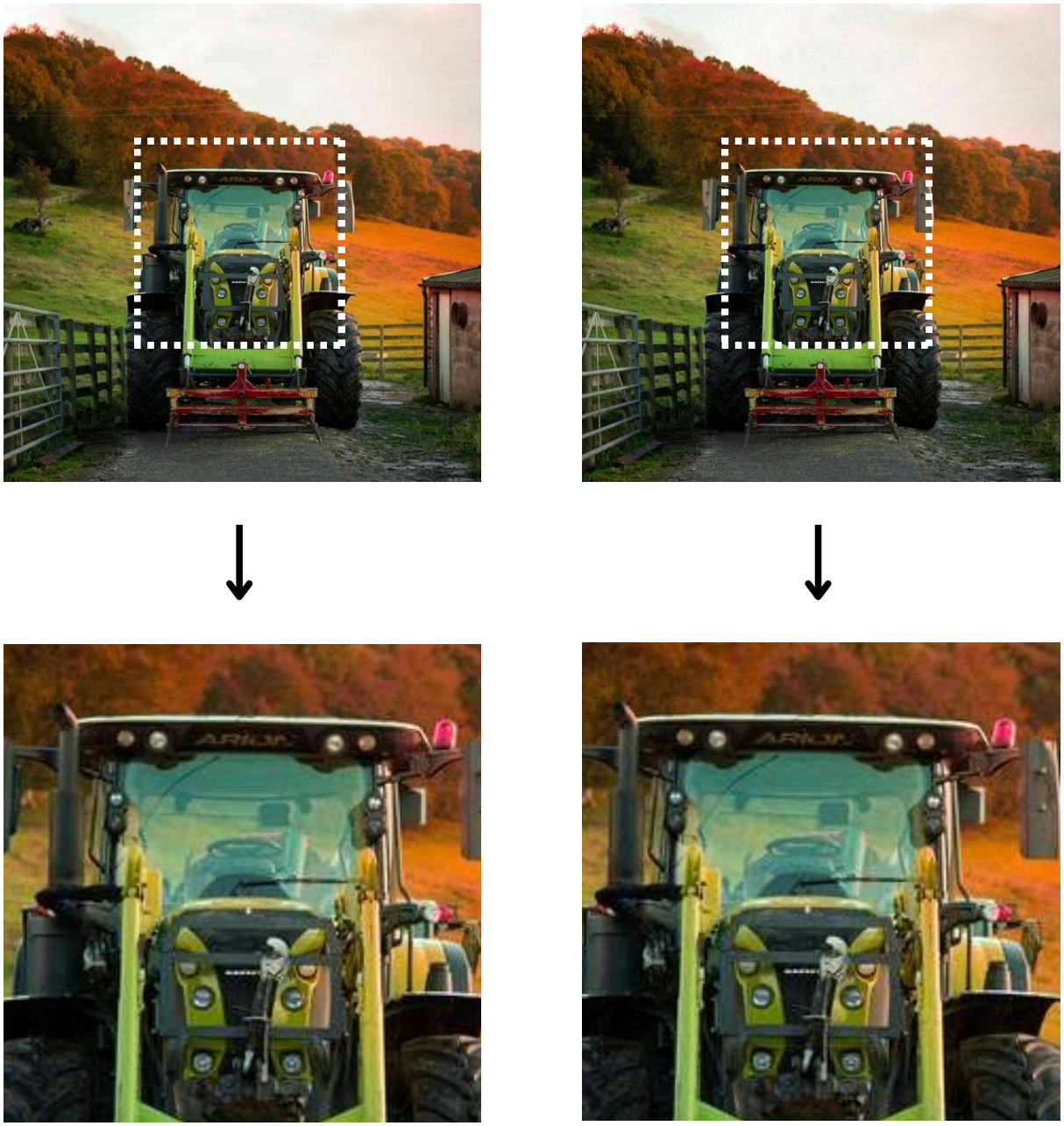
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.3: PSNR 39.3854 dB. (Imagem utilizada no processo: Mandril [1]).



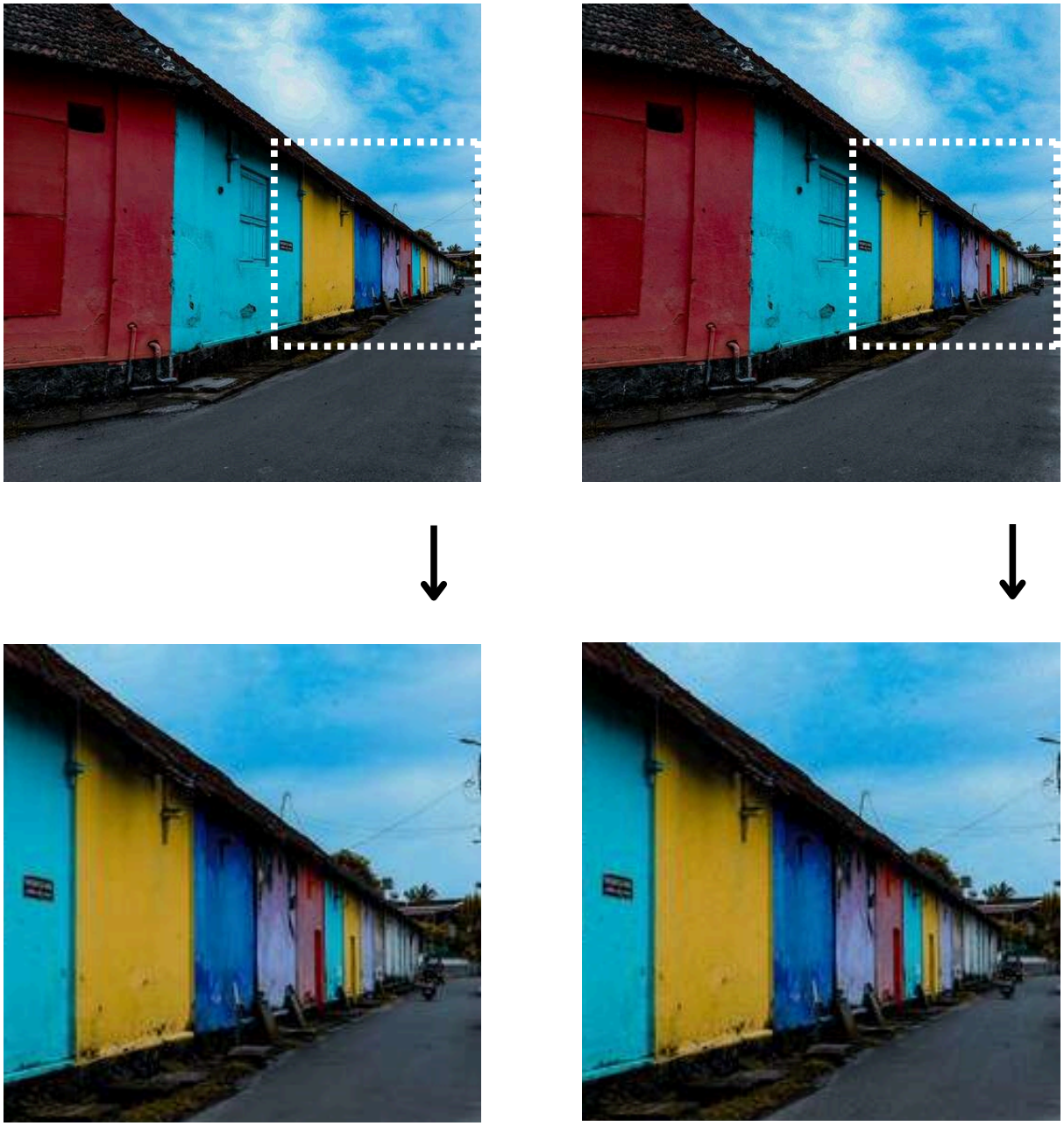
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.4: PSNR 40.2226 dB. (Imagem utilizada no processo: Rüthen, Andrea Hagenhoff [1]).



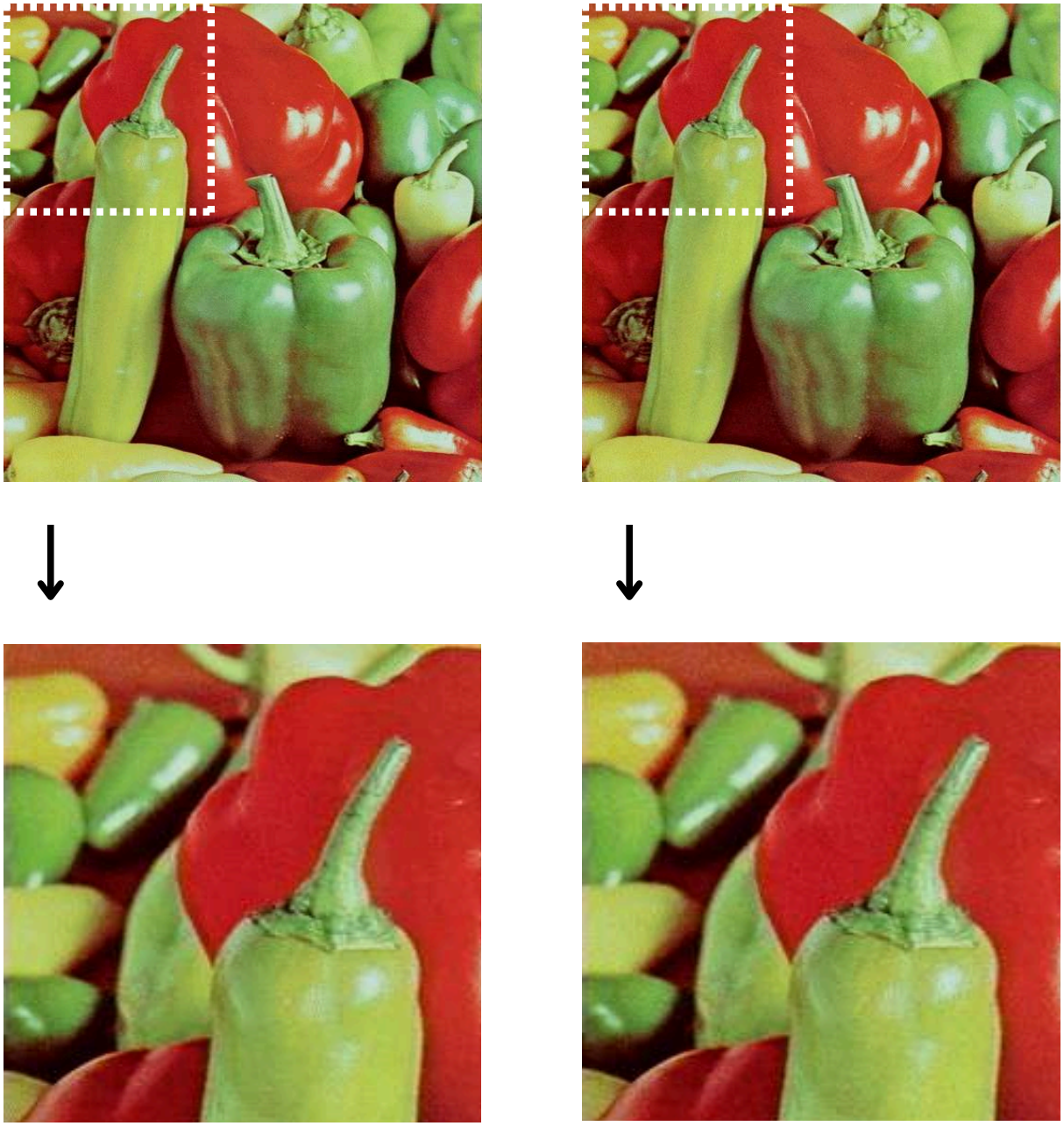
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.5: PSNR 40.0555 dB. (Imagem utilizada no processo: Tractor, Ajeet Panesar [1]).



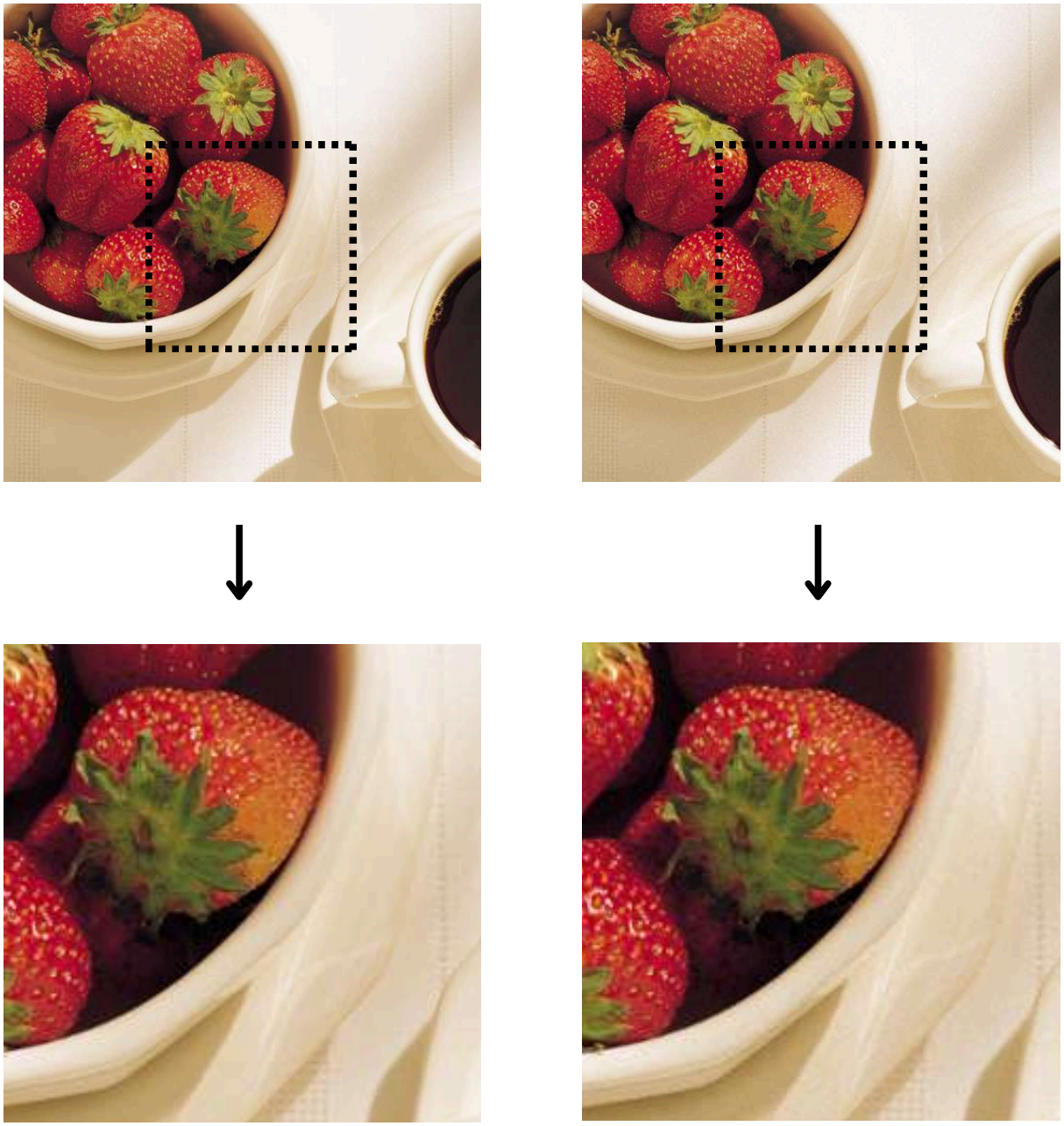
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.6: PSNR 40.8631 dB. (Imagem utilizada no processo: Mattancherry, Aby Zachariah [1]).



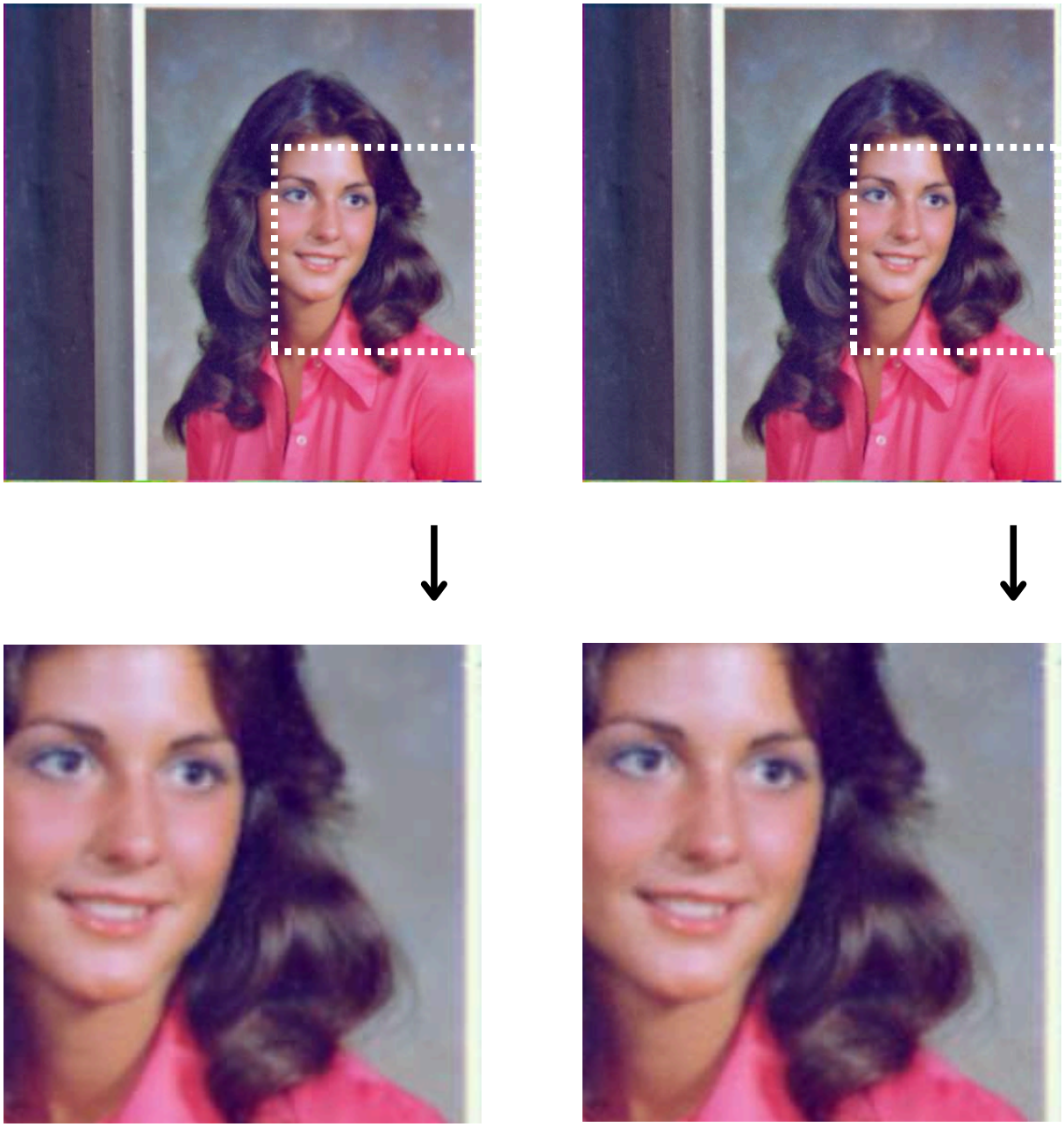
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.7: PSNR 40.0113 dB. (Imagem utilizada no processo: Peppers [1]).



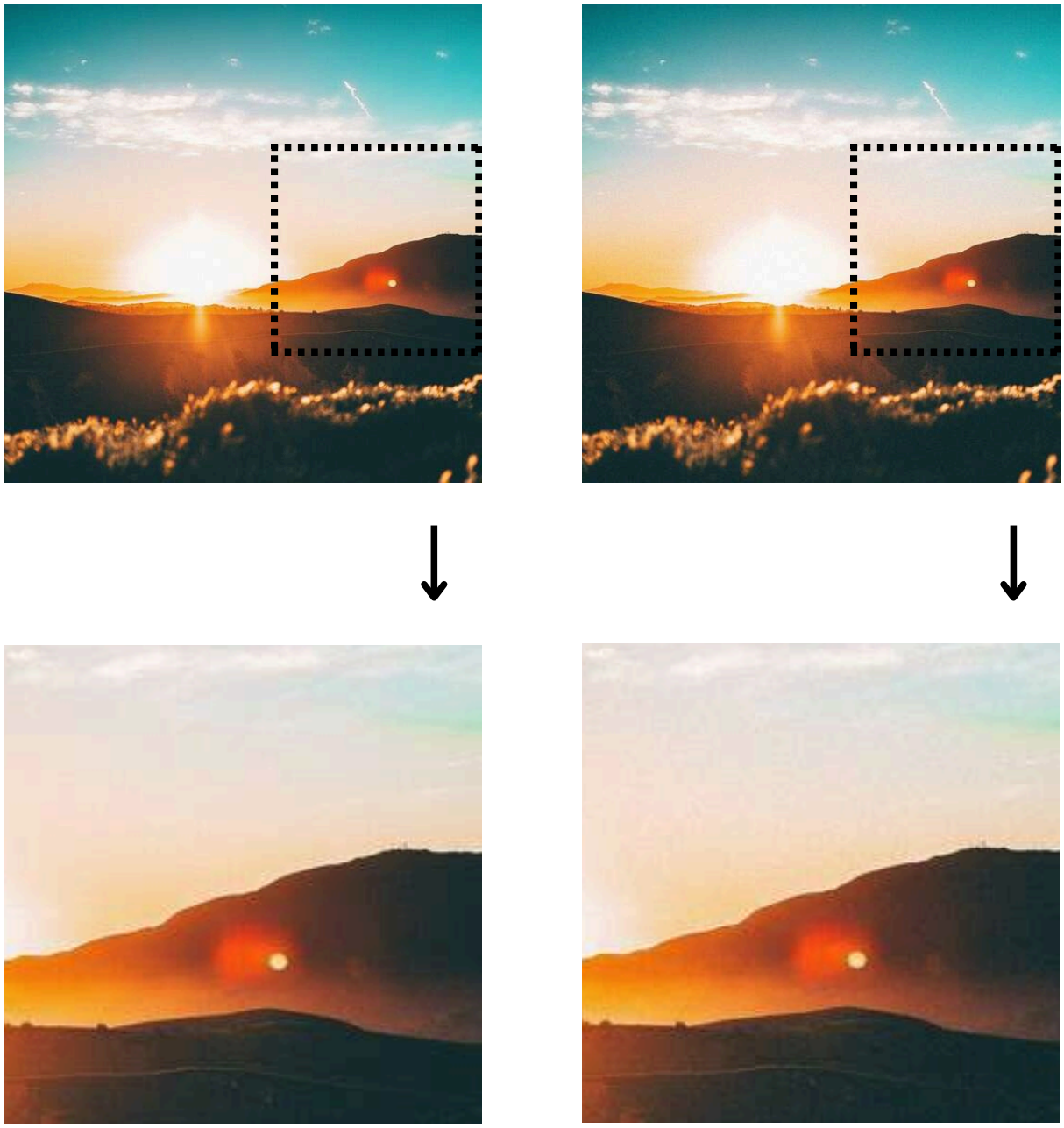
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.8: PSNR 41.3577 dB. (Imagem utilizada no processo: Strawberries Coffee [1]).



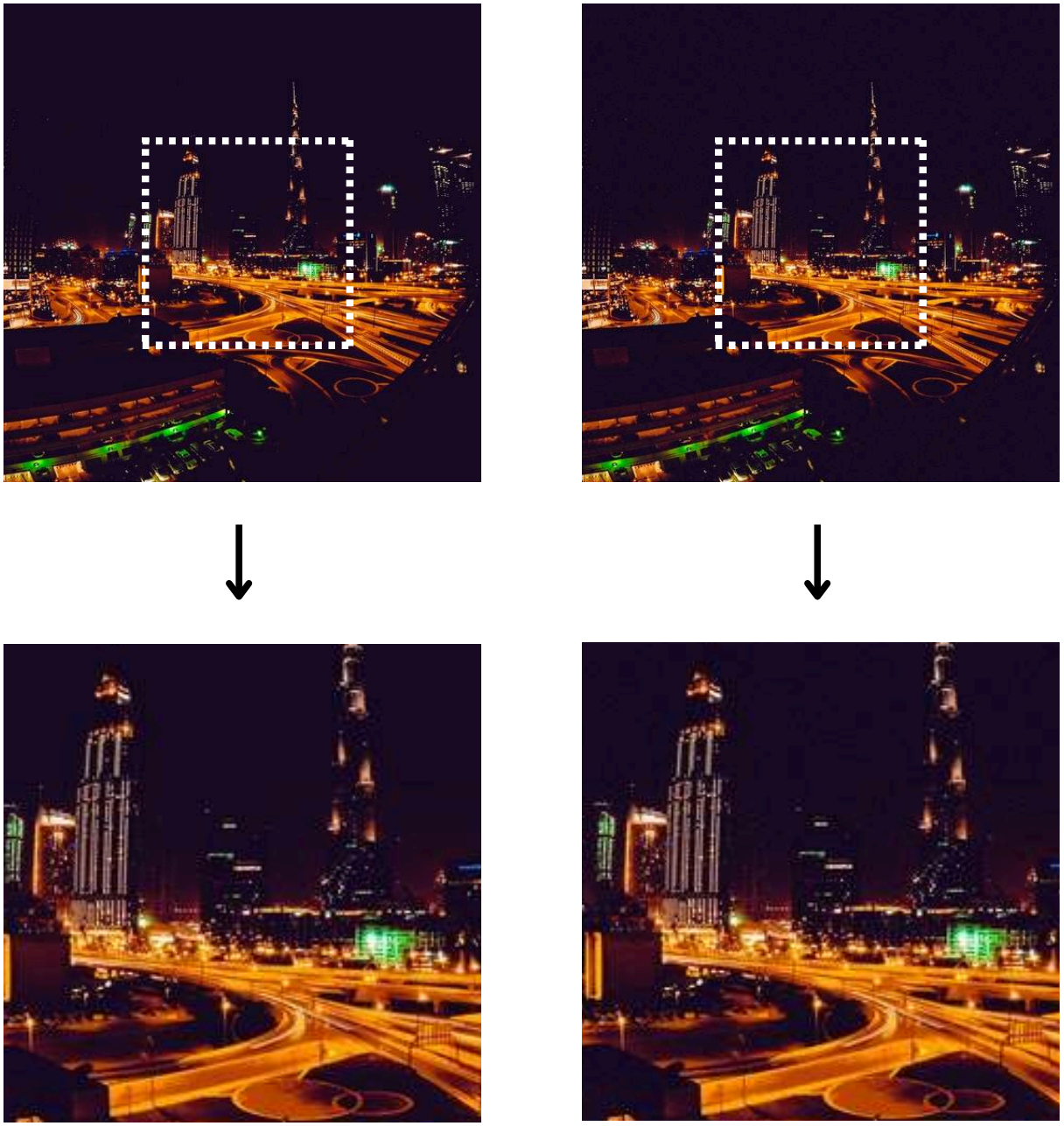
(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.9: PSNR 42,3273 dB. (Imagem utilizada no processo: Woman [1]).



(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.10: PSNR 42.5782 dB. (Imagem utilizada no processo: Anaheim Hills, Jordan Wozniak [1]).



(a) Imagem original **(b)** Imagem marcada: **Método proposto**

Figura A.11: PSNR 41.5481 dB. (Imagem utilizada no processo: Dusit Thani Dubai, Harshil Gudka [1]).

A.2 Estatísticas de Extração

Tabela A.1: Resultados para cada conjunto de imagens de teste. Ataques de compressão. (***NC***: NC médio, ***DP$_{nc}$***: desvio padrão NC, ***BER***: BER médio). Marca-d'água: **V1**

Ataques de compressão	**imgs1**	**imgs2**	**imgs3**	**imgs4**
Compressão JPEG (80)	NC: 0.9994 DP_{nc}: 0.0009 BER: 0.0001	NC: 0.9993 DP_{nc}: 0.0037 BER: 0.0001	NC: 0.9997 DP_{nc}: 0.0011 BER: 0.0001	NC: 0.9963 DP_{nc}: 0.0165 BER: 0.0010
Compressão JPEG (50)	NC: 0.9490 DP_{nc}: 0.0209 BER: 0.0139	NC: 0.9214 DP_{nc}: 0.0897 BER: 0.0194	NC: 0.9355 DP_{nc}: 0.0605 BER: 0.0193	NC: 0.9571 DP_{nc}: 0.0472 BER: 0.0119
Compressão JPEG (40)	NC: 0.9098 DP_{nc}: 0.0246 BER: 0.0255	NC: 0.9070 DP_{nc}: 0.0991 BER: 0.0247	NC: 0.8970 DP_{nc}: 0.0617 BER: 0.0312	NC: 0.9365 DP_{nc}: 0.0522 BER: 0.0178
Compressão JPEG (20)	NC: 0.7804 DP_{nc}: 0.0157 BER: 0.0716	NC: 0.7241 DP_{nc}: 0.1189 BER: 0.0901	NC: 0.7563 DP_{nc}: 0.0709 BER: 0.0852	NC: 0.7525 DP_{nc}: 0.0442 BER: 0.0825

Tabela A.2: Resultados para cada conjunto de imagens de teste. Ataques geométricos (**Parte 1**). (***NC***: NC médio, ***DP$_{nc}$***: desvio padrão NC, ***BER***: BER médio). Marca-d'água: **V1**

Ataques geométricos	**imgs1**	**imgs2**	**imgs3**	**imgs4**
Reflexão vertical	NC: 0.9765 DP_{nc}: 0.1331 BER: 0.0158	NC: 0.8756 DP_{nc}: 0.2824 BER: 0.0788	NC: 0.8651 DP_{nc}: 0.2858 BER: 0.0892	NC: 0.8803 DP_{nc}: 0.2746 BER: 0.0796
Reflexão horizontal	NC: 0.9543 DP_{nc}: 0.1800 BER: 0.0309	NC: 0.8791 DP_{nc}: 0.2796 BER: 0.0767	NC: 0.8750 DP_{nc}: 0.2788 BER: 0.0832	NC: 0.8684 DP_{nc}: 0.2855 BER: 0.0874
Translação x(0.4%) y(0.4%)	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9936 DP_{nc}: 0.0104 BER: 0.0014	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9938 DP_{nc}: 0.0215 BER: 0.0017
Translação x(5%) y(7%)	NC: 0.9948 DP_{nc}: 0.0225 BER: 0.0014	NC: 0.9882 DP_{nc}: 0.0192 BER: 0.0028	NC: 0.9778 DP_{nc}: 0.0385 BER: 0.0062	NC: 0.9844 DP_{nc}: 0.0378 BER: 0.0044
Translação x(22%) y(12%)	NC: 0.9764 DP_{nc}: 0.0429 BER: 0.0066	NC: 0.9313 DP_{nc}: 0.0640 BER: 0.0176	NC: 0.9130 DP_{nc}: 0.0806 BER: 0.0263	NC: 0.9741 DP_{nc}: 0.0497 BER: 0.0074
Translação x(2%) y(0.19%)	NC: 0.9998 DP_{nc}: 0.0011 BER: 0.0000	NC: 0.9890 DP_{nc}: 0.0235 BER: 0.0026	NC: 0.9985 DP_{nc}: 0.0051 BER: 0.0004	NC: 0.9958 DP_{nc}: 0.0171 BER: 0.0012

Tabela A.3: Resultados para cada conjunto de imagens de teste. Ataques geométricos (**Parte 2**). (***NC***: NC médio, $\boldsymbol{DP_{nc}}$: desvio padrão NC, ***BER***: BER médio). Marca-d'água: **V1**

Ataques geométricos	**imgs1**	**imgs2**	**imgs3**	**imgs4**
Rotação (30.0°) com expansão	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9998 DP_{nc}: 0.0023 BER: 0.0001	NC: 0.9997 DP_{nc}: 0.0021 BER: 0.0001	NC: 0.9981 DP_{nc}: 0.0090 BER: 0.0005
Rotação (90.5°) com expansão	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9999 DP_{nc}: 0.0006 BER: 0.0000	NC: 1.0000 DP_{nc}: 0.0003 BER: 0.0000	NC: 0.9999 DP_{nc}: 0.0007 BER: 0.0000
Rotação (58.0°) com recorte	NC: 0.9935 DP_{nc}: 0.0223 BER: 0.0018	NC: 0.9203 DP_{nc}: 0.0601 BER: 0.0201	NC: 0.9718 DP_{nc}: 0.0480 BER: 0.0080	NC: 0.9746 DP_{nc}: 0.0492 BER: 0.0073
Rotação (−215.0°) com recorte	NC: 0.9934 DP_{nc}: 0.0239 BER: 0.0018	NC: 0.9406 DP_{nc}: 0.0522 BER: 0.0148	NC: 0.9711 DP_{nc}: 0.0485 BER: 0.0082	NC: 0.9778 DP_{nc}: 0.0481 BER: 0.0064
Rotação (334.0°) com recorte	NC: 0.9943 DP_{nc}: 0.0218 BER: 0.0015	NC: 0.9543 DP_{nc}: 0.0455 BER: 0.0113	NC: 0.9754 DP_{nc}: 0.0457 BER: 0.0070	NC: 0.9838 DP_{nc}: 0.0402 BER: 0.0046
Rotação (30.0°) com recorte + Escala (0.8)	NC: 0.9687 DP_{nc}: 0.0495 BER: 0.0089	NC: 0.8397 DP_{nc}: 0.1050 BER: 0.0472	NC: 0.9175 DP_{nc}: 0.0915 BER: 0.0261	NC: 0.9430 DP_{nc}: 0.0911 BER: 0.0184
Escala (0.8)	NC: 0.9998 DP_{nc}: 0.0009 BER: 0.0000	NC: 0.9898 DP_{nc}: 0.0166 BER: 0.0023	NC: 0.9962 DP_{nc}: 0.0120 BER: 0.0010	NC: 0.9969 DP_{nc}: 0.0107 BER: 0.0008
Escala (0.9)	NC: 1.0000 DP_{nc}: 0.0002 BER: 0.0000	NC: 0.9982 DP_{nc}: 0.0048 BER: 0.0004	NC: 0.9992 DP_{nc}: 0.0037 BER: 0.0002	NC: 0.9991 DP_{nc}: 0.0048 BER: 0.0002
Escala (1.2)	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 1.0000 DP_{nc}: 0.0003 BER: 0.0000	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9999 DP_{nc}: 0.0005 BER: 0.0000
Escala (1.5)	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9999 DP_{nc}: 0.0011 BER: 0.0000
Escala (2.0)	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 1.0000 DP_{nc}: 0.0003 BER: 0.0000	NC: 1.0000 DP_{nc}: 0.0001 BER: 0.0000	NC: 0.9999 DP_{nc}: 0.0018 BER: 0.0000
Escala (2.5)	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9999 DP_{nc}: 0.0017 BER: 0.0000	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9998 DP_{nc}: 0.0027 BER: 0.0000
Rotação (−50.5°) com recorte	NC: 0.9927 DP_{nc}: 0.0290 BER: 0.0020	NC: 0.9284 DP_{nc}: 0.0576 BER: 0.0181	NC: 0.9694 DP_{nc}: 0.0496 BER: 0.0087	NC: 0.9756 DP_{nc}: 0.0497 BER: 0.0070
Rotação (−90.0°) com recorte	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9768 DP_{nc}: 0.0329 BER: 0.0056	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9925 DP_{nc}: 0.0204 BER: 0.0020
Rotação (−120.1°) com recorte	NC: 0.9932 DP_{nc}: 0.0232 BER: 0.0019	NC: 0.9184 DP_{nc}: 0.0609 BER: 0.0206	NC: 0.9724 DP_{nc}: 0.0478 BER: 0.0079	NC: 0.9733 DP_{nc}: 0.0538 BER: 0.0078

Tabela A.4: Resultados para cada conjunto de imagens de teste. Ataques geométricos (**Parte final**). (***NC***: NC médio, $\boldsymbol{DP_{nc}}$: desvio padrão NC, ***BER***: BER médio). Marca-d'água: **V1**

Ataques geométricos	**imgs1**	**imgs2**	**imgs3**	**imgs4**
Rotação (30.0°) com recorte	NC: 0.9931 DP_{nc}: 0.0241 BER: 0.0019	NC: 0.9456 DP_{nc}: 0.0487 BER: 0.0134	NC: 0.9724 DP_{nc}: 0.0475 BER: 0.0078	NC: 0.9811 DP_{nc}: 0.0439 BER: 0.0054
Rotação (90.5°) com recorte	NC: 1.0000 DP_{nc}: 0.0000 BER: 0.0000	NC: 0.9382 DP_{nc}: 0.0584 BER: 0.0156	NC: 0.9996 DP_{nc}: 0.0020 BER: 0.0001	NC: 0.9788 DP_{nc}: 0.0452 BER: 0.0061
Rotação (30.0°) com recorte + Ajuste de iluminação ([0.2, 0.8] -> [0,1])	NC: 0.9771 DP_{nc}: 0.0486 BER: 0.0068	NC: 0.8383 DP_{nc}: 0.1700 BER: 0.0563	NC: 0.8759 DP_{nc}: 0.1623 BER: 0.0513	NC: 0.9174 DP_{nc}: 0.1323 BER: 0.0315
Escala (0.8) + Filtro Gaussiano (3 x 3)	NC: 0.9925 DP_{nc}: 0.0200 BER: 0.0020	NC: 0.9340 DP_{nc}: 0.0532 BER: 0.0163	NC: 0.9679 DP_{nc}: 0.0480 BER: 0.0091	NC: 0.9760 DP_{nc}: 0.0485 BER: 0.0069
Rotação (6.4°) com recorte + Aumento da iluminação (2.0 x)	NC: 0.9537 DP_{nc}: 0.0812 BER: 0.0142	NC: 0.8382 DP_{nc}: 0.2064 BER: 0.0657	NC: 0.9086 DP_{nc}: 0.1505 BER: 0.0378	NC: 0.9261 DP_{nc}: 0.1554 BER: 0.0328
Corte (1/4) + Redução da iluminação (0.5 x)	NC: 0.9960 DP_{nc}: 0.0218 BER: 0.0011	NC: 0.9694 DP_{nc}: 0.0377 BER: 0.0074	NC: 0.9771 DP_{nc}: 0.0462 BER: 0.0066	NC: 0.9885 DP_{nc}: 0.0323 BER: 0.0032
Corte (1/4)	NC: 0.9981 DP_{nc}: 0.0104 BER: 0.0005	NC: 0.9914 DP_{nc}: 0.0176 BER: 0.0020	NC: 0.9910 DP_{nc}: 0.0281 BER: 0.0025	NC: 0.9956 DP_{nc}: 0.0186 BER: 0.0012
Corte (1/3)	NC: 0.9969 DP_{nc}: 0.0165 BER: 0.0008	NC: 0.9791 DP_{nc}: 0.0304 BER: 0.0050	NC: 0.9833 DP_{nc}: 0.0408 BER: 0.0048	NC: 0.9915 DP_{nc}: 0.0275 BER: 0.0024
Corte (1/2)	NC: 0.9938 DP_{nc}: 0.0279 BER: 0.0018	NC: 0.9366 DP_{nc}: 0.0565 BER: 0.0158	NC: 0.9581 DP_{nc}: 0.0622 BER: 0.0123	NC: 0.9791 DP_{nc}: 0.0491 BER: 0.0061

Tabela A.5: Resultados para cada conjunto de imagens de teste. Ataques comuns (**Parte 1**). (***NC***: NC médio, $\boldsymbol{DP_{nc}}$: desvio padrão NC, ***BER***: BER médio). Marca-d'água: **V1**

Ataques comuns	**imgs1**	**imgs2**	**imgs3**	**imgs4**
Ruído 'Salt & Pepper' (0.01)	NC: 0.9887 DP_{nc}: 0.0088 BER: 0.0030	NC: 0.9847 DP_{nc}: 0.0126 BER: 0.0031	NC: 0.9862 DP_{nc}: 0.0093 BER: 0.0036	NC: 0.9858 DP_{nc}: 0.0087 BER: 0.0036
Ruído Gaussiano (0.005)	NC: 0.9594 DP_{nc}: 0.0122 BER: 0.0108	NC: 0.9429 DP_{nc}: 0.0849 BER: 0.0158	NC: 0.9584 DP_{nc}: 0.0283 BER: 0.0114	NC: 0.9468 DP_{nc}: 0.0505 BER: 0.0149

Tabela A.6: Resultados para cada conjunto de imagens de teste. Ataques comuns (**Parte final**). (***NC***: NC médio, $\boldsymbol{DP_{nc}}$: desvio padrão NC, ***BER***: BER médio). Marca-d'água: **V1**

Ataques comuns	**imgs1**	**imgs2**	**imgs3**	**imgs4**
Filtro Mediano (3 x 3)	NC: 0.9548 DP_{nc}: 0.0602 BER: 0.0132	NC: 0.7522 DP_{nc}: 0.1933 BER: 0.0974	NC: 0.8463 DP_{nc}: 0.1464 BER: 0.0577	NC: 0.9089 DP_{nc}: 0.1292 BER: 0.0328
Ajuste de iluminação ([0.2, 0.8] -> [0,1])	NC: 0.9912 DP_{nc}: 0.0372 BER: 0.0026	NC: 0.9599 DP_{nc}: 0.0810 BER: 0.0106	NC: 0.9602 DP_{nc}: 0.0801 BER: 0.0131	NC: 0.9819 DP_{nc}: 0.0500 BER: 0.0054
Ajuste de iluminação ([0,1] -> [0.2, 0.8])	NC: 0.9975 DP_{nc}: 0.0133 BER: 0.0007	NC: 0.9793 DP_{nc}: 0.0272 BER: 0.0050	NC: 0.9831 DP_{nc}: 0.0365 BER: 0.0047	NC: 0.9913 DP_{nc}: 0.0275 BER: 0.0024
Aumento do contraste (2.0 x)	NC: 0.9949 DP_{nc}: 0.0235 BER: 0.0014	NC: 0.9443 DP_{nc}: 0.0577 BER: 0.0135	NC: 0.9639 DP_{nc}: 0.0583 BER: 0.0106	NC: 0.9750 DP_{nc}: 0.0538 BER: 0.0073
Redução do contraste (0.5 x)	NC: 0.9979 DP_{nc}: 0.0114 BER: 0.0005	NC: 0.9818 DP_{nc}: 0.0236 BER: 0.0043	NC: 0.9856 DP_{nc}: 0.0300 BER: 0.0039	NC: 0.9925 DP_{nc}: 0.0225 BER: 0.0020
Filtro de Média (3 x 3)	NC: 0.9271 DP_{nc}: 0.0580 BER: 0.0211	NC: 0.8041 DP_{nc}: 0.0770 BER: 0.0548	NC: 0.8832 DP_{nc}: 0.0808 BER: 0.0360	NC: 0.9124 DP_{nc}: 0.0891 BER: 0.0267
Filtro Gaussiano (3 x 3)	NC: 0.9999 DP_{nc}: 0.0006 BER: 0.0000	NC: 0.9991 DP_{nc}: 0.0034 BER: 0.0002	NC: 0.9989 DP_{nc}: 0.0051 BER: 0.0003	NC: 0.9991 DP_{nc}: 0.0059 BER: 0.0002
Compressão JPEG (80) + Ruído 'Salt & Pepper' (0.01)	NC: 0.9252 DP_{nc}: 0.0161 BER: 0.0209	NC: 0.9071 DP_{nc}: 0.0504 BER: 0.0215	NC: 0.9200 DP_{nc}: 0.0297 BER: 0.0226	NC: 0.9072 DP_{nc}: 0.0394 BER: 0.0260
Filtro Gaussiano (3 x 3) + Redução do contraste (0.5 x)	NC: 0.9983 DP_{nc}: 0.0091 BER: 0.0004	NC: 0.9849 DP_{nc}: 0.0188 BER: 0.0035	NC: 0.9887 DP_{nc}: 0.0213 BER: 0.0030	NC: 0.9943 DP_{nc}: 0.0158 BER: 0.0015
Filtro de Média (3 x 3) + Redução do contraste (0.5 x)	NC: 0.8788 DP_{nc}: 0.0471 BER: 0.0361	NC: 0.7622 DP_{nc}: 0.0619 BER: 0.0684	NC: 0.8363 DP_{nc}: 0.0683 BER: 0.0515	NC: 0.8675 DP_{nc}: 0.0807 BER: 0.0405
Filtro gaussiano (3 x 3) + Ruído Salt & Pepper 0.01 + Redução do contraste (0.5 x)	NC: 0.9314 DP_{nc}: 0.0308 BER: 0.0191	NC: 0.8499 DP_{nc}: 0.0516 BER: 0.0385	NC: 0.8888 DP_{nc}: 0.0545 BER: 0.0329	NC: 0.9098 DP_{nc}: 0.0514 BER: 0.0256