Pontifícia Universidade Católica
do Rio de Janeiro

Jaisse Grela Escobar

Uma ferramenta para o rastreamento de vídeos e imagens utilizando técnicas de esteganografia

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC-Rio.

Orientador: Prof. Bruno Feijó

Rio de Janeiro
Abril de 2015

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Jaisse Grela Escobar

Uma ferramenta para o rastreamento de vídeos e imagens utilizando técnicas de esteganografia

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Bruno Feijó
Orientador
Departamento de Informática - PUC-Rio

Prof. Hélio Côrtes Vieira Lopes
Departamento de Informática - PUC-Rio

Prof. Raul Queiroz Feitosa
Departamento de Engenharia Elétrica - PUC-Rio

Prof. José Eugênio Leal
Coordenador Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 10 de abril de 2015

Jaisse Grela Escobar

Graduou-se em Ciência da Computação na Universidade de Havana (UH) em 2005. Trabalhou como pesquisador no laboratório VisionLab/ICAD da PUC-Rio. Atuou como desenvolvedor em projetos em parceria com a Petrobras no laboratório Tecgraf.

Ficha Catalográfica

Escobar, Jaisse Grela

Uma ferramenta para o rastreamento de vídeos e imagens utilizando técnicas de esteganografia / Jaisse Grela Escobar ; orientador: Bruno Feijó. – 2015.
57 f. : il. (color.) ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2015.
Inclui bibliografia

1. Informática – Teses. 2. Esteganografia. 3. Esteganografia adaptativa. 4. Marca d'água 5. SURF. 6. Segurança de conteúdo de TV. I. Feijó, Bruno. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Silenciosamente teve que ser....
porque há coisas, que para alcançá-las, precisam estar ocultas.
José Martí.

Agradecimentos

Ao meu orientador, professor Bruno Feijó, pela confiança depositada. Pelo incentivo, seguimento e implicação no meu processo de formatura.

Aos professores membros da Banca, que através de seus conselhos e críticas ajudaram a melhorar o resultado final desta tese.

À minha família que com muito amor e sacrifício fizeram possível minha educação, sempre foram suporte e alentaram minha superação profissional.

Aos professores e funcionários do Departamento de Informática da PUC-Rio por esta oportunidade de aprendizado.

À PUC-Rio e CAPES pelo suporte financeiro recebido durante a pesquisa.

Aos professores da Universidad de Havana (UH) por me dar uma inestimável formação de base para poder realizar esta caminhada.

Resumo

Escobar, Jaisse G.; Feijó, Bruno. **Uma ferramenta para o rastreamento de vídeos e imagens utilizando técnicas de esteganografia.** Rio de Janeiro, 2015. 57p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Na indústria de TV frequentemente ocorrem vazamentos de materiais de filmagem quando estes se distribuem entre os colaboradores de produção, prejudicando grandemente as empresas. Neste trabalho propomos uma ferramenta que, utilizando técnicas de esteganografia adaptativa, permite detectar a fonte do vazamento com um elevado grau de confiança. Um requisito importante é que a informação mascarada no vídeo (ou na imagem) resista a operações de processamento tais como redimensionamento e mudança de resolução. A ideia é usar o algoritmo "*Speeded Up Robust Features*" (SURF), estratégia consagrada, na detecção e descrição de características em imagens para detectar regiões robustas da imagem e inserir nelas uma pequena identificação mascarada. A ferramenta utiliza a transformada "*Haar – Discrete Wavelet Transform*" em duas dimensões, para depois fazer modificações na imagem. Esta dissertação propõe direções iniciais promissoras para a identificação segura de certificados de origem de imagens e vídeos.

Palavras-chave

Esteganografia; Esteganografia Adaptativa; Marca d'água; SURF; Segurança de Conteúdo de TV.

Abstract

Escobar, Jaisse G.; Feijó, Bruno. **A tool for tracking videos and images using steganography techniques.** Rio de Janeiro, 2015. 57p. MSc Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

In the TV industry, leaks of film materials occur frequently when they are distributed among the members of the production team, causing great harm to the companies. In this paper, we propose a tool that allows detecting the source of the leak with a high degree of confidence, using techniques of adaptive steganography. An important requirement is that the information embedded in the video (or image) should resist to processing operations such as resizing and resolution changes. The idea is to use the “Speeded Up Robust Features” (SURF) algorithm, a well-known strategy for detection and description of images features, to detect the robust regions of the image and insert a small masked identification in them. The tool uses the "Haar - Discrete Wavelet Transform" in two dimensions and then modifies the image. This dissertation proposes promising initial directions for secure identification of the certificate of origin of digital images and videos.

Keywords

Steganography; Adaptive Steganography; Watermarking; SURF; TV Content Security

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

1
Introdução

A indústria de TV necessita identificar a origem de vídeos de sua propriedade que foram disponibilizados sem seu consentimento. Acontece que quando um vídeo está pronto para exibição, ou quase pronto, a indústria de TV remete para vários profissionais envolvidos no projeto com o objetivo de consultas ou simplesmente para que eles vejam o trabalho final. Porém durante esse processo ocorre a disponibilização indevida da obra, prejudicando de forma significativa a empresa. Em geral a publicação dos conteúdos ocorre em formatos diferentes do original e em tamanhos menores da imagem ou dos *frames* no caso dos vídeos. Por isso, o alvo é reconhecer a fonte do vazamento em arquivos mesmo com essas duas mudanças, no caso de alterações nas dimensões da imagem ou *frame* o ideal é que o algoritmo seja efetivo em arquivos onde cada uma das dimenções (comprimento e largura) são de até 33% do tamanho original, ou seja, menos de 10% do tamanho original no global.

Uma solução natural para identificar a origem de um vídeo ilegalmente copiado é embarcar um identificador no vídeo original que seja invisível e resistente a alterações no vídeo. Esta solução pode ser realizada através de esteganogramas (*i.e.* mensagens secretas embutidas) nos frames de um vídeo. Esteganografia é a arte de embutir mensagens secretas (esteganogramas) em um portador, que tanto pode ser usada para fins legais (*e.g.* proteção de copyright através de marcas d'água digitais) como ilegais (*e.g.* envio de instruções criminosas).

Nesta dissertação estamos interessados em esteganografia apenas para embutir um identificador secreto e muito simples em vários frames de um vídeo. Entretanto, esteganografia pode usar uma variedade muito grande de portadores (*carriers*), tais como imagens, vídeos, textos e sons; como também a mensagem secreta embutida pode ser de vários tipos (texto em uma imagem, uma imagem em outra imagem, ...). No capítulo 2, apresentamos esteganografia de uma maneira geral. A Figura 1.1 apresenta uma das primeiras imagens esteganográficas alegadamente difundida na internet, onde a mensagem escondida pode ser recuperada através de uma chave secreta (a palavra *abc*, no caso) (Provos 2001). Esteganografia digital já existe há bastante tempo, mas

esta seria a primeira imagem esteganográfica detectada na internet. Entretanto, esta imagem parece ter sido, na realidade, fabricada pela ABC News em uma reportagem sobre esteganografia para mostrar os perigos de mal uso desta técnica por terroristas na internet (Lau 2001).

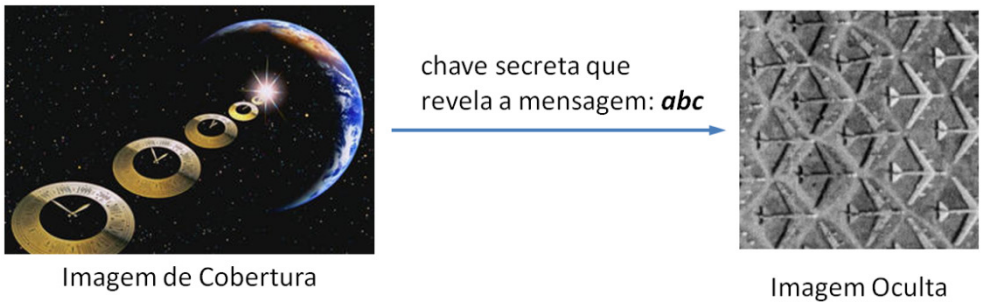


Figura 1.1 Imagem esteganográfica contendo imagem oculta de uma base militar de aviões (Provos, 2001)

1.1. Estado da Arte

Apesar do fato de esteganografia ser conhecida há séculos, apenas recentemente esta técnica tem se proliferado em várias formas e meios, tais como mídia digital, redes de computadores e serviços populares de telecomunicações. O número de aplicativos de esteganografia para esconder dados tem suplantado em muito as ferramentas para detecção e extração de esteganogramas – algo como 1025 aplicativos para esconder mensagens versus 111 ferramentas de detecção e extração, em 2007 (Zielinska et al., 2014).

Existem alguns produtos comerciais, tais como “Outguess” e “F5 Steganography” que podem ser utilizados para inserir códigos secretos em imagens e vídeos. Na época eles implementaram algoritmos que foram considerados seguros, mas agora os métodos de quebra para estes produtos já são bem conhecidos. Outro grande problema é que nenhum deles trata vídeos de alta resolução que são posteriormente modificados para baixa resolução e que garantam um grau de confiança elevado no reconhecimento do certificado digital.

Em matéria de algoritmos vários trabalhos são apresentados. Chen e Lin (2006) propõem novas técnicas de esteganografia que incorpora as mensagens secretas no domínio da frequência. Dependendo das exigências do problema sobre a capacidade de incorporação e qualidade de imagem, o algoritmo proposto é dividido em dois modos e cinco casos. As mensagens secretas estão

embutidas nos coeficientes de alta frequência que resultaram da aplicação da transformada discreta *wavelet*. Coeficientes na sub-região de baixa frequência são preservados inalterados para melhorar a qualidade da imagem. Algumas operações matemáticas básicas são executadas nas mensagens secretas antes da incorporação. Estas operações e uma tabela de mapeamento são projetados para manter as mensagens protegidas de usuários mal-intencionados e, portanto, proporcionar segurança satisfatória. O artigo apresenta o algoritmo com bastante detalhes mas não mostra resultados nenhum para ataques do tipo compressão da imagem, nem redução do tamanho.

Hamid et al. (2012), no artigo entitulado "*A Comparison between Using SIFT and SURF for Characteristic Region Based Image Steganography*", apresentam uma nova técnica de esteganografia baseada na seleção de regiões da imagem para esconder dados. Naquele artigo, os autores comparam duas abordagens promissoras para detectar as regiões robustas da imagem, a "*Scale Invariant Feature Transform*" (SIFT) e a "*Speeded Up Robust Features*" (SURF). A robustez dos dois algoritmos foi testada contra diferentes tipos de ataques. Apresentaram-se resultados em compressão de até 80% em JPGE, com 85% de grau de confiança, mas não foram apresentados resultados com diminuição do tamanho da imagem.

Jiansheng et al. (2009) apresentam um algoritmo de esteganografia baseado em uma transformada discreta *wavelet* (DWT) na imagem de cobertura e uma marca d'água que é uma imagem com caracteres visíveis. A marca d'água é incorporado na banda de alta frequência de domínio da transformada *wavelet* em alguns blocos ou regiões da imagem de cobertura. Antes de incorporar a marca d'água, a transformada discreta dos cossenos (DCT) é aplicada para melhorar a sua robustez. Os resultados das simulações sugerem que este sistema de marca d'água não só pode manter uma boa qualidade de imagem, como também pode ser robusto contra vários tipos de ataque de sinal – tais como a adição de ruído do tipo sal, compressão de imagem, corte de imagem e rotação. Os resultados apresentados em compressão (sem detalhar em que porcentagem) são aceitáveis, mas não apresentam resultados com diminuição do tamanho da imagem.

A ideia desta dissertação é iniciar a investigação de outras técnicas robustas em imagens que possa ser levada a vídeos, e que permitam tomar partido do tipo de vídeo mais usado por emissoras de televisão.

Esta dissertação não pretende resolver todas as questões, mas procura propor algumas direções consideradas promissoras.

1.2. Contribuições

Neste trabalho, propomos um método baseado em esteganografia adaptativa para assinar imagens e vídeos e reconhecer a assinatura mesmo quando ocorrem vários ataques ao arquivo original. Nosso método baseia-se parcialmente em algoritmos apresentados por Hamid et al. (2012) e Jiansheng et al. (2009).

Nossa principal contribuição é a possibilidade de reconhecer a assinatura ainda que cada uma das dimensões da imagem seja reduzida a 33% do tamanho original. Deve se destacar também que as reduções do tamanho podem ser combinadas com compressão de formato e, mesmo assim, o algoritmo detecta a assinatura satisfatoriamente.

Nossos resultados mostram um bom desempenho, mesmo lidando com imagens cujos tamanhos em bits são reduzidos em até 6%.

1.3. Estrutura da Tese

Esta dissertação está organizada como se segue. O Capítulo 2 (Esteganografia) apresenta uma breve revisão das principais áreas do conhecimento envolvido no desenvolvimento desse trabalho. O Capítulo 3 (Trabalhos relacionados) detalha os dois artigos mais importantes consultados na pesquisa prévia e apresenta as conclusões que chegamos na implementação deles. O Capítulo 4 (Método proposto) apresenta o nosso algoritmo de esteganografia. Em um primeiro momento, descrevemos a proposta de uma maneira geral e, em seguida, detalhamos especificidades do método que incidem diretamente na otimização do processo. No Capítulo 5 (Resultados Experimentais e Discussão) apresentamos os resultados e conclusões acerca da parametrização do algoritmo, baseado em várias métricas e em uma das imagens padrão para algoritmos desse tipo. No final do capítulo, destacamos uma ideia para melhorar o grau de confiança do reconhecimento da assinatura que surgiu como resultado dos experimentos. Por fim, no Capítulo 6 (Conclusões e trabalhos futuros), apresentamos as conclusões finais e as sugestões para trabalhos futuros.

2
Esteganografia

Este capítulo apresenta a área de estudo da esteganografia, a terminologia usada, os requisitos fundamentais dos sistemas esteganográficos, técnicas gerais e linhas de pesquisa.

Esteganografia deriva do grego, onde “estegano” significa esconder, mascarar e “grafia” significa escrita. Logo, as duas palavras combinadas resultam literalmente “escrita encoberta”. Durante toda a história, as pessoas buscam inúmeras formas de esconder informações dentro de outros meios, para, de alguma forma, obter mais privacidade para seus meios de comunicação. A esteganálise, por sua vez, é o inverso da esteganografia – *i.e.* a esteganálise é a arte de detectar mensagens escondidas nos mais diversos meios de comunicação.

Há uma distinção entre a estegnografia e a criptografia que pode ser apresentada de uma maneira superficial, porém simples (Kessler 2001). A primeira é a ciência de esconder mensagens secretas em algum portador, enquanto a segunda é a ciência de escrever códigos secretos de maneira que terceiros nunca os decifrem.

Uma visão recente das tendências em esteganografia pode ser encontrada em Zielinska et al. (2014). Uma referência resumida, porém mais antiga, pode ser encontrada em (Kessler 2004).

2.1.
Terminologia

A seguir, encontram-se alguns dos principais termos utilizados nestas áreas. Estes são ilustrados na Figura 2.1:

- dado embutido ou *embedded data*: é o dado que será enviado de maneira secreta, normalmente em uma mensagem, texto ou figura;
- mensagem de cobertura ou *cover–message*: é a mensagem que servirá para mascarar o dado embutido. Esta mensagem de cobertura pode ser de áudio (*cover-audio*), de texto (*cover-text*) ou uma imagem (cover-image);
- estego-objeto ou *stego-object*: após a inserção do dado embutido na mensagem de cobertura se obtém o estego-objeto;

- estego-chave ou *stego-key*: adicionalmente pode ser usada uma chave para inserir os dados do dado embutido na mensagem de cobertura. A esta chave damos o nome de estego-chave.

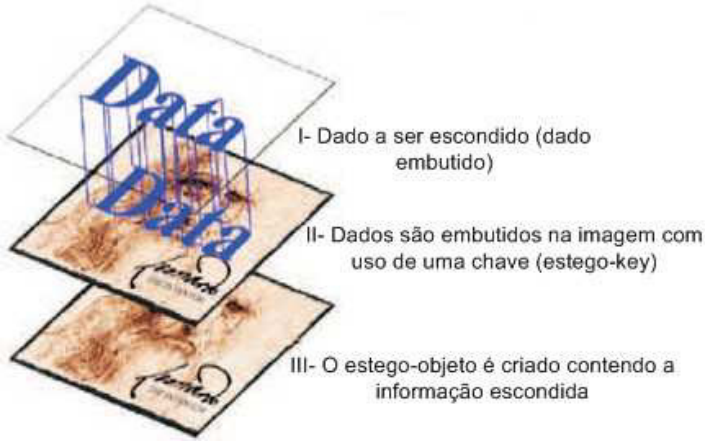


Figura 2.1 Escondendo dados em uma imagem (Petitcolas; Aanderson; Kuhn 1999).

2.2. Requisitos para Sistemas Esteganográficos

Os três requisitos mais importantes que devem ser satisfeitos para qualquer sistema esteganográfico são:

- segurança: a fim de não levantar suspeita, enquanto tenta criar uma blindagem contra um algoritmo de descoberta, o conteúdo escondido deve ser invisível, tanto por inspeção visual quanto por meios estatísticos;
- carga útil: a esteganografia é direcionada à comunicação escondida e, portanto, normalmente exige capacidade de inclusão suficiente. Os requisitos para capacidade significativa de dados e segurança são frequentemente contraditórios. Considerando as características da aplicação específica, um compromisso entre estes requisitos deve sempre ser buscado;
- robustez: embora robustez contra ataques não seja uma prioridade importante, ter a capacidade de resistir à compressão é certamente desejável, pois a maioria das imagens JPEG coloridas são comprimidas antes de serem colocadas *online*.

2.3. Esteganografia em Imagem

As imagens são a mídia de cobertura mais popular para esteganografia e podem ser armazenadas em um formato bitmap direto (como BMP) ou em um formato comprimido (como JPEG). O ocultamento de informações é realizado no domínio espacial ou no domínio de frequência.

Em termos de esquemas de inserção, vários métodos (como substituição, adição e ajuste) podem ser usados. As abordagens mais comuns de inserção de mensagens em imagens abrangem técnicas de inserção no bit menos significativo, técnicas de filtragem e mascaramento, e algoritmos de transformações. Cada uma destas técnicas pode ser aplicada a imagens, com graus variados de sucesso. (Petitcolas; Anderson; Kuhn 1999)

2.4. Esteganografia em Vídeo

A esteganografia em vídeo é muito similar à esteganografia em imagens, exceto pelo fato de que as informações são escondidas em cada frame do arquivo de vídeo. As técnicas em vídeo tiram proveito do vasto domínio de cobertura e da coerência entre frames.

2.5. Esteganografia em Áudio

Esconder dados em sinais de áudio é algo desafiante, pois o sistema auditivo humano (SAH) pode trabalhar em uma faixa muito grande de frequências. O SAH pode captar até um bilhão de potências diferentes de sinais (altura) e até mil frequências de sinais distintas. A sensitividade a ruído também é muito apurada. Uma perturbação em um arquivo de som pode ser detectada tão baixa quanto uma em 10 milhões de partes ou 80 dB em um ambiente comum. Apesar de ser tão poderoso para captar sinais e frequências, o SAH não consegue fazer diferenciação de tudo que recebe. Sendo assim, sons mais altos tendem a mascarar sons mais baixos. Além disso, o SAH não consegue perceber um sinal em fase absoluta, somente em fases relativas. Também existem algumas distorções do ambiente muito comuns que são simplesmente ignoradas pelo ouvido na maioria dos casos.

As técnicas de esteganografia exploram muitas destas “vulnerabilidades” do ouvido humano, porém sempre têm que levar em conta a extrema sensibilidade do SAH.

2.6. Técnicas de Esteganálise

Existem diversas abordagens para detectar a presença de conteúdo escondido em imagens digitais. Estas abordagens podem ser divididas em três tipos (Rocha, 2006): ataques aurais, estruturais e estatísticos.

- ataques aurais: estes ataques consistem em retirar as partes significativas da imagem como um meio de facilitar aos olhos humanos a busca por anomalias na imagem. Um teste comum é mostrar os bits menos significativos da imagem;
- ataques estruturais: a estrutura do arquivo de dados algumas vezes muda assim que outra mensagem é inserida. Nesses casos, um sistema capaz de analisar padrões estruturais seria capaz de descobrir a mensagem escondida. Por exemplo, se mensagens são escondidas em imagens indexadas (baseadas em paletas de cores), pode ser necessário usar diferentes versões de paletas. Este tipo de atitude muda as características estruturais da imagem de cobertura, logo as chances de detecção da presença de uma mensagem escondida aumentam (Wayner, 2002);
- ataques estatísticos: os padrões dos pixels e seus bits menos significativos frequentemente revelam a existência de uma mensagem secreta nos perfis estatísticos. Os novos dados não têm os mesmos perfis esperados. Muitos dos estudos de Matemática e Estatística têm por objetivo classificar se um dado fenômeno ocorre ao acaso. Cientistas usam estas ferramentas para determinar se suas teorias explicam bem tal fenômeno. Estas técnicas estatísticas também podem ser usadas para determinar se uma dada imagem e/ou som possui alguma mensagem escondida. Na maioria das vezes, os dados escondidos são mais aleatórios que os dados que foram substituídos no processo de mascaramento ou inserem padrões que alteram as propriedades estatísticas inerentes do objeto de cobertura.

3
Trabalhos Relacionados

Este capítulo descreve os dois artigos mais relevantes consultados na pesquisa prévia e apresenta as conclusões que chegamos na implementação dessas duas abordagens.

Durante a pesquisa consultamos dois trabalhos que têm uma abordagem sobre técnicas de esteganografia muito próxima dos objetivos da presente dissertação. Esses trabalhos foram de muita utilidade, aportaram uma grande informação e nossa proposta tomou como base vários dos resultados apresentados neles. Por sua importância, descreveremos os artigos enfatizando os pontos que mais nos ofereceram contribuições significativas. Como contribuição elucidamos alguns procedimentos e equações que não ficam claros para o leitor destes artigos.

O algoritmo apresentado por Hamid et al. (2012) oculta dados nas regiões robustas da imagem, utilizando as estratégias SIFT e SURF para detectá-las. A capacidade de ocultação de informação deste algoritmo é relativamente limitada; uma questão que faz com que seja mais adequado para aplicações de proteção de direitos autorais.

O algoritmo apresentado por Jianshen et al. (2009) trata como inserir uma imagem transformada com a DCT, na banda de alta frequência da imagem de cobertura transformada com uma DWT bidimensional (2d-DWT). O artigo em questão apresenta os resultados das simulações e mostra que esse algoritmo é invisível e tem boa robustez para algumas operações comuns de processamento de imagem, ou seja, aumento e compressão de imagem, adição de ruído e corte de imagem.

3.1.
A Comparison between Using SIFT and SURF for Characteristic Region Based Image Steganography

O algoritmo proposto por Hamid et al. (2012) oculta uma mesma imagem em várias regiões robustas da imagem de cobertura. A inserção redundante da

mesma imagem é para aumentar a probabilidade de obtê-la após mudanças na estego-imagem.

As regiões robustas ou pontos-chave robustos são aqueles pontos da imagem que podem resistir a uma ampla gama de operações de processamento de imagens, tais como o redimensionamento e a rotação. Tais regiões robustas podem ser detectadas mesmo quando a imagem sofre ataques de vários tipos diferentes. No artigo (*op. cit.*), propõem-se as abordagens SURF e SIFT para detectar as regiões robustas, e conclue-se que SURF tem resultados melhores. A Figura 3.1 e a Figura 3.2 exibe uma imagem e a detecção dos pontos-chave após a aplicação do detector SURF na imagem.

Figura 3.1 Imagem original

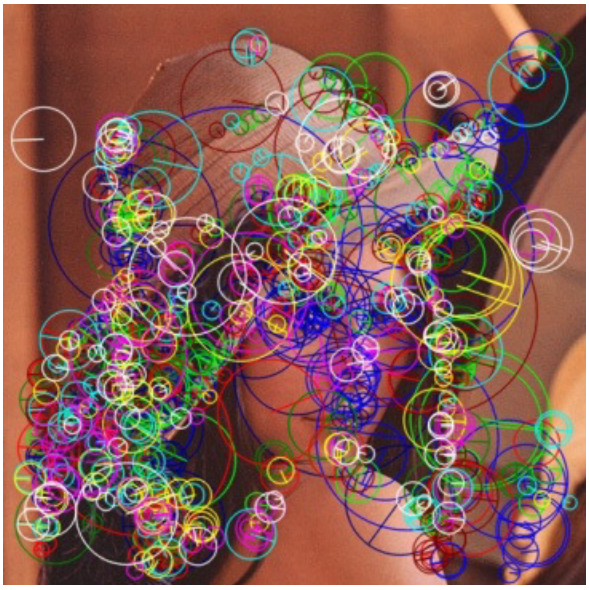


Figura 3.2 Alguns pontos-chaves detectados após a aplicação do SURF.

Alguns pontos são descartados a fim de evitar quaisquer regiões de interseção (aquelas com pontos-chaves muito próximos). Desta maneira, garantimos que as regiões locais são disjuntas, ou seja, as regiões não podem ter interseções porque um dado solaparia os outros.

As regiões robustas onde será inserida a imagem são obtidas a partir da lista do pontos-chave. A Figura 3.3 mostra um exemplo dessas regiões circulares selecionadas. Embora os pontos-chave definam regiões circulares, o algoritmo extrai a região quadrada de menor área que contém a região circular. O tamanho de cada região robusta tem que ser quatro vezes o tamanho da imagem embutida. Na Figura 3.3 mostra possíveis regiões circulares selecionadas pelo algoritmo.

Figura 3.3 Regiões circulares selecionadas pelo algoritmo

Para cada uma das regiões robustas é aplicada uma variante da transformadas 2d–DWT conhecida como *9/7 Biorthogonal Wavelet*. A Figura 3.4 mostra a notação utilizada para ganhar em claridade na descrição dos passos do algoritmo apresentados em Hamid et al. (2012).

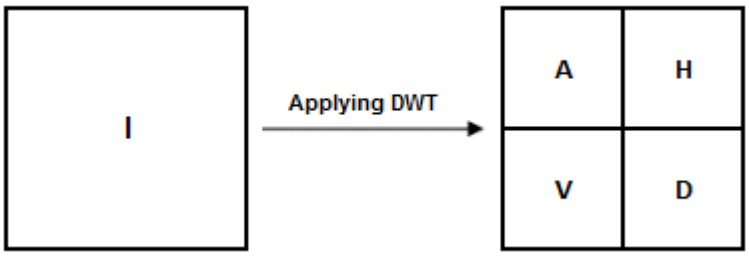


Figura 3.4 Notação utilizada para explicar os passos do algoritmo.

A seguir vem a fase de embarcar os bits da imagem com a identificação.

Algoritmo 3.1 Fase de embarcar os bits da imagem embutida

```
1:  {Sendo W e H o comprimento e a largura da imagem embutida}
2:  Para x = 1 até W
3:      Para y = 1 até H
4:          Se b(x, y) = 1 e D1 = H(x, y) – V(x, y) < T então
5:                  H'(x, y) = H'(x, y) + (T - D1) / 2
6:                  V'(x, y) = V'(x, y) – (T – D1) / 2
7:          Fim Se
8:          Se b(x, y) = 0 e D2 = V(x, y) – H(x, y) < T então
9:              H'(x, y) = H'(x, y) – (T – D2) / 2
10:             V'(x, y) = V'(x, y) + (T – D2) / 2
11:         Fim Se
12:     Fim Para
13: Fim Para
```

Onde b(x, y) representa o pixel (x, y) da imagem embutida, neste caso, de um bit só, H representa a sub-região superior direita da região robusta depois de aplicar o 2d–DWT e V representa a sub-região inferior esquerda. H' representa o valor do pixel da sub-região superior direita da estego-imagem e V' a sub-região inferior esquerda. Finalmente, T é um limiar para controlar a informação de invisibilidade.

Como passo final para inserir o dado, a função inversa da transformada *9/7 Biorthogonal Wavelet* é aplicada, e a região robusta é modificada na imagem de cobertura obtendo-se assim a estego-imagem.

Na extração dos dados, as regiões robustas são selecionadas como no caso do processo de inserção sobre a imagem original. Para cada uma delas aplicamos a mesma variante da transformada 2d–DWT; e os bits da imagem inserida são obtidos mediante o seguinte procedimento:

Algoritmo 3.2 Fase de extração dos bits da imagem de cobertura

```
{Sendo W e H o comprimento e a largura da imagem em cobertura}
Para x = 1 até W
    Para y = 1 até H
        Se H(x, y) > V(x, y) então
            b(x, y) = 1
        Senão
            b(x, y) = 0
        Fim Se
    Fim Para
Fim Para
```

Do passo anterior, várias imagens são obtidas. Estas imagens vão ser diferentes da imagem original embutida produto da compressão própria do formato da imagem de cobertura. Então as imagens obtidas vão ser misturadas em uma imagem só, determinando a cor de cada pixel como sendo aquela cor que mais aparece nos respectivos pixels das imagens.

3.2. A Digital Watermarking Algorithm Based On DCT and DWT

A ideia geral deste algoritmo proposto por Jiansheng et al. (2009) é ocultar partes da imagem em regiões espalhadas na imagem de cobertura. Para melhorar a robustez e o sigilo da imagem a inserir, aplicamos a *Discrete Cosine Transform* (DCT) sobre ela, de maneira que uma imagem desordenada é obtida.

A DCT é uma transformada matemática baseada em cossenos, muito utilizada em processamento digital de imagens e compressão de dados. O valor da função da DCT de um vetor *p* de pixels de comprimento *n* é:

$$G_f = \frac{1}{2} C_f t = 0n - 1 p_t \cos\left(\frac{(2t+1)\pi}{2n}\right) \qquad \text{(Equação 3.1)}$$

$$\text{onde: } C_f = \begin{cases} \frac{1}{\sqrt{2}} f=0 \\ 1 \; f>0 \end{cases} para \; f = 0, 1, \ldots, n-1$$

Em seguida, decompomos a imagem de cobertura em L-níveis usando uma 2d–DWT. Escolher os L-níveis da transformada vai depender dos tamanhos da imagem original e da imagem (marca d'água) a inserir. Quanto maior for o nível da DWT, melhor fica o efeito de incorporar os dados escondidos.

A ideia básica da DWT em imagens é uma decomposição da imagem em sub-imagens de diferentes espaços de domínios e regiões de frequências

independentes. Depois que a imagem original é transformada, ela é decomposta em sub-regiões agrupando as frequências baixas em uma sub-região denotada por L e as frequências altas em outra sub-região denotada por H. De uma transformada em duas dimensões obtemos 4 sub-regiões (também chamadas de sub-bandas): uma sub-região de baixa frequência (denotada por LL) e 3 sub-regiões de alta frequência (denotadas por HL, LH e HH). A sub-região de baixa frequência LL é a parte suave da imagem e, portanto, aparenta ser muito similar à imagem original. As sub-regiões de alta frequência representam detalhes de arestas/texturas (horizontais, verticais e diagonais).

A sub-região LL, por sua vez, também pode ser decomposta em 4 sub-regiões. Desta maneira, a imagem original pode ser descomposta em *n* níveis. A Figura 3.5 mostra como ficariam as sub-regiões após aplicar a transformada em três níveis (Chen e Lin, 2006). Humanos são muito sensíveis a mudanças de regiões suaves de uma imagem, mas não são tão sensíveis a pequenas mudanças de arestas e faixas. Portanto, o sinal de marca d'água deve ser colocado na sub-região de alta frequência de maior índice, que no caso da Figura 3.5 é a sub-região HH_3.

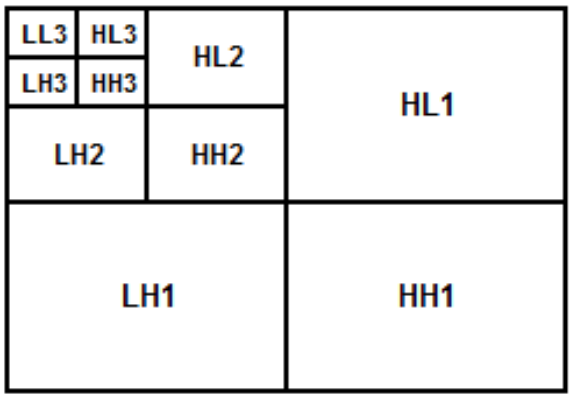


Figura 3.5 Sub-regiões após aplicar a transformada 2d-DWT três vezes

No artigo de Jiansheng et al. (2009), a sub-região de alta frequência HH de maior nível é plotada em sub-blocos B_k de dimensões de 2 * 2. Em seguida, temos que calcular a entropia e o valor ao quadrado de cada sub-bloco B_k. O sub-bloco com pequeno valor de entropia deve ser um bloco suave, e aquele com um valor maior deve ser um bloco de aresta ou um bloco de faixa/textura. O valor ao quadrado é então usado para identificar se o bloco é um bloco de aresta (valor menor) ou um bloco de faixa/textura (valor maior). Escolhendo o valor limite apropriado da entropia e do valor ao quadrado, os blocos de faixa/textura U_k *(k = 1,2, ..., P x Q)* são obtidos.

Depois da escolha dos blocos de faixa/textura U_k, os valores de coeficiente *wavelet* C_k destes blocos são corrigidos para incorporar a marca d'água através da seguinte equação:

$$C_k' = C_k + a \times V_k, k = 1, 2, \ldots, P \times Q \qquad \text{(Equação 3.1)}$$

onde, C_k representa o valor do coeficiente *wavelet* do sub-bloco U_k, V_k representa a k-ésima componente da sequência V da marca d'água digital unidimensional a inserir, C_k' representa o novo valor do coeficiente *wavelet* do sub-bloco U_k e a representa a profundidade de mascaramento na imagem a inserir.

Depois de embutir o sinal da marca d'água nos blocos U_k, unimos a informação da sub-região de mais baixa frequência com a subregião de alta frequência corrigida. Por fim, a função inversa da 2d–DWT é aplicada e a estego-imagem é, então, obtida.

Na extração dos dados, todos os passos anteriores são repetidos, substituindo, obviamente, só o passo da inserção pela extração. Esse passo consiste em achar a diferença entre os coeficientes dos blocos de faixas/texturas nas duas imagens e dividir pelo mesmo fator a que foi usado no passo anterior. A formulação matemática é a seguinte:

$$V_k = \frac{C_k' - C_k}{a}, k = 1, 2, \ldots, P \times Q \qquad \text{(Equação 3.2)}$$

3.3. Resultados obtidos na implementação

Em um primeiro momento decidimos implementar os algoritmos da seção 3.1 e 3.2 para assim verificar os resultados. A imagem para ser oculta contém uma identificação única junto a uma máscara ou formato que possibilita a detecção da proveniência. Esta é branca e preta porque só precisa de um bit por pixel para assim diminuir o tamanho. A Figura 3.6 mostra uma imagem típica.

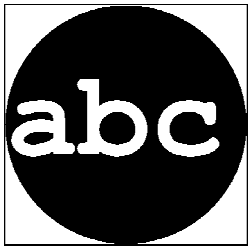


Figura 3.6 Imagem típica para ser embutida na imagem de cobertura

Na implementação do algoritmo de Hamid et al. (2012), foi comprovado que o algoritmo funciona bem para algumas transformações, fundamentalmente

para a mudança de formato e a ampliação da imagem. No entanto, a ampliação da imagem não é interesse do presente trabalho e sim a redução do tamanho.

A imagem inserida ficou maior do que o esperado. Uma imagem típica tem dimensões de *32 x 32 pixels* e precisamos de um ponto robusto com um diâmetro de 64 pixels para ocultá-la. Esse ponto robusto não representa um ponto especialmente forte para transformações do tipo redução do tamanho da imagem.

Outra opção é espalhar a imagem em várias regiões da imagem de cobertura que é uma das ideias do algoritmo descrito em Jiansheng et al. (2009).

Na verdade, um identificador é apenas uma combinação de bits, que nós vamos fixar em 24 bits. Esse valor comparado com o tamanho da imagem, 32 x 32 (2^{10} bits), é bem menor. Então, decidimos fazer várias modificações nos algoritmos originais chegando à nossa proposta.

4
Método Proposto

Este capítulo descreve o algoritmo de esteganografia proposto nesta dissertação. Em um primeiro momento descrevemos a proposta em geral e, em seguida, detalhamos especificidades do mesmo que incidem diretamente na otimização do processo.

Na maior parte das técnicas atuais de esteganografia, a ocultação de informação modifica a mensagem de cobertura, no caso imagem ou vídeo. Isto pode afetar negativamente a qualidade visual e aumentar a possibilidade de perda de dados após os ataques possíveis. Nos casos onde o tamanho do dado embutido é muito menor em relação ao tamanho da mensagem de cobertura é possível aplicar técnicas de esteganografia adaptativa.

A esteganografia adaptativa identifica as áreas de textura ou quase-textura para a incorporação dos dados secretos. Esta estratégia extrai características estatísticas globais da imagem antes de tentar incorporar as informações secretas em regiões específicas da imagem. Estas estatísticas irão ditar onde fazer as alterações. Uma vantagem adicional deste processo é que a seleção de algumas regiões para esconder dados irá minimizar a distorção da estego-imagem.

Uma das aplicações da esteganografia é precisamente o rastreamento de documentos e a certificação digital. Em nosso caso, o alvo é conseguir identificar univocamente cada uma das versões dos vídeos entregues a cada pessoa ou entidade.

4.1.
Inserção dos dados

O objetivo do nosso algoritmo é mascarar em alguns *frames* do vídeo uma mensagem. Essa mensagem é bem pequena em relação à imagem de cobertura, o dado embutido só vai conter apenas três bytes para cada um dos vídeos entregues. A mensagem é uma combinação única de bits que permita que quando seja pego um vídeo vazado, ou seja, disponibilizado indevidamente, possamos identificar a origem.

Em geral, nós vamos mascarar um ou vários bits do dado embutido em algumas regiões da imagem, até que a mensagem toda esteja contida. Vamos repetir esse processo várias vezes redundantemente, para aumentar o grau de confiança na hora da extração do dado.

As regiões vão ser determinadas a partir de uma lista de pontos-chave obtidas através do detector SURF. Em nossa implementação os resultados com o SURF foram melhores do que com o SIFT, confirmando os resultados apresentados em Hamid et al. (2012). Um ponto-chave ou região robusta consiste em: a coordenada centro da região, o raio e o grau de fortaleza. A ideia é obter regiões pequenas, com maior grau de fortaleza. Como nos outros algoritmos consultados, as regiões selecionadas não podem se interceptar. Em geral a quantidade de regiões selecionadas vai estar determinada pela fórmula:

$$Q = \frac{L \times B}{R} \qquad \text{(Equação 4.1)}$$

sendo L a longitude em bits do dado embutido, B a quantidade de bits mascarados por região, e R a quantidade de vezes redundante que a mensagem é mascarada. A Figura 4.1 mostra possíveis regiões circulares selecionadas pelo algoritmo proposto, onde podemos comparar as diferenças com a Figura 3.3. Finalmente, extraímos a região quadrada de menor área que contém a região circular.

Figura 4.1 Regiões locais disjuntas úteis para mascarar dados na imagem

Para cada uma das regiões robustas é aplicada uma variante da transformada 2d–DWT chamada Haar–DWT.

4.1.1. Haar-DWT

Existem vários tipos de DWT, sendo uma delas a Haar-DWT, descrita pelo matemático húngaro Alfred Haar. A Haar-DWT de duas dimensões consiste em duas operações fundamentais: uma operação horizontal e outra vertical (Chen e Lin, 2006).

Em um primeiro passo, os pixels são escaneados da esquerda para a direita na direção horizontal. Calculamos, então, a adição e a subtração para cada par de pixels vizinhos. A adição se conserva na primeira metade da imagem e a subtração na segunda, como mostra a Figura 4.2. Esse processamento se repete para cada fila da imagem. Os pixels que representam a soma são a parte baixa da frequência da imagem original (na figura representado por L), entanto, os pixels que representam o restante são a parte alta da frequência desta (na figura denotado por H).

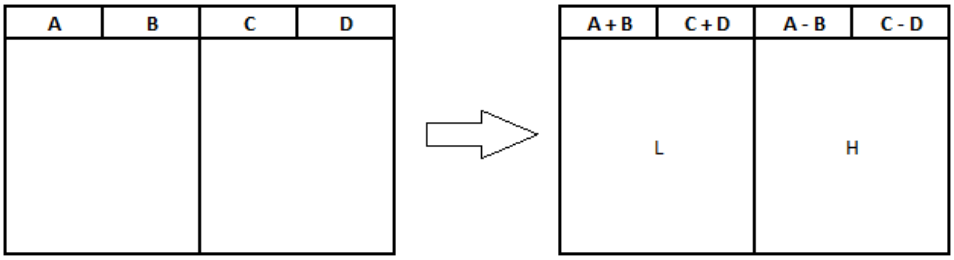


Figura 4.2 Aplicação da operação horizontal na Haar-DWT de duas dimensões

A transformação vertical consiste em escanear os pixels da imagem desde o topo até a base, guardando a soma dos pixels vizinhos no topo da imagem e o restante na parte baixa da imagem como mostra a Figura 4.3. Finalmente, obtemos as quatro sub-regiões mencionadas anteriormente. A sub-região LL, de baixa frequência, vai ser muito parecida com a imagem original.

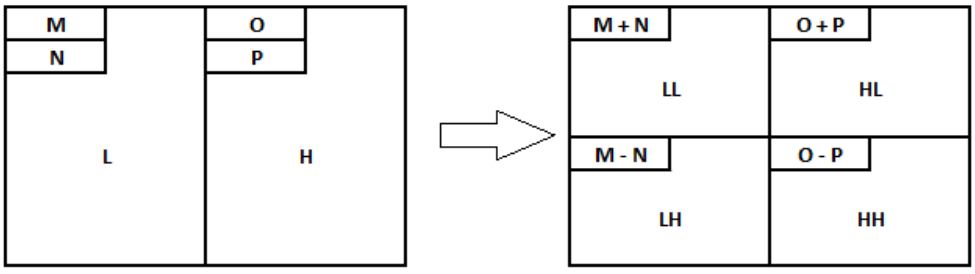


Figura 4.3 Aplicação da operação vertical na Haar-DWT de duas dimensões

O procedimento anterior descreve o Haar–DWT em um nível só. Para o segundo nível temos que aplicar os passos anteriores na sub-região LL; e assim sucessivamente em níveis superiores.

Depois vem a fase de mascarar os bits do dado embutido. As modificações à matriz só vão ser na sub-região HH de maior nível.

Algoritmo 4.1 Fase de mascarar os bits do dado embutido

```
1: {Sendo W e H o comprimento e a largura da sub-região HH de maior nível}
2: Para x = 1 até W
3:     Para y = 1 até H
4:         Se b(x * W + y) = 1 então
5:             HH'(x, y) = HH(x, y) + F1
6:         Fim Se
7:     Fim Para
8: Fim Para
```

Onde b(i) representa o bit *i-ésimo* da mensagem secreta, HH representa o valor do pixel da sub-região inferior direita do maior nível da região robusta depois de aplicar o Haar–DWT em vários níveis. H' representa o valor do pixel da sub-região superior direita da estego-imagem. Finalmente, F_1 é um limiar para controlar a informação de invisibilidade.

Depois é aplicada a função inversa da Haar–DWT. Num primeiro passo o que fizemos foi conservar a adição e a subtração de dois números na matriz; então, para obter os números originais, aplicamos a semi-soma e a semi-diferença desses valores. O primeiro passo da inversa é a transformação vertical (ver Figura 4.4) e o segundo a transformação horizontal (ver Figura 4.5). Se originalmente a Haar–DWT foi aplicada em vários níveis, devemos começar pela sub-matriz de nível máximo e logo repetir até chegar ao primeiro nível.

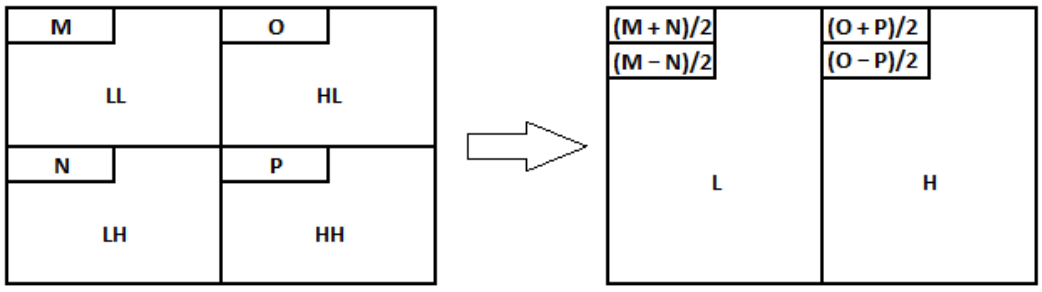


Figura 4.4 Primeiro passo da inversa da 2d-Haar-DWT

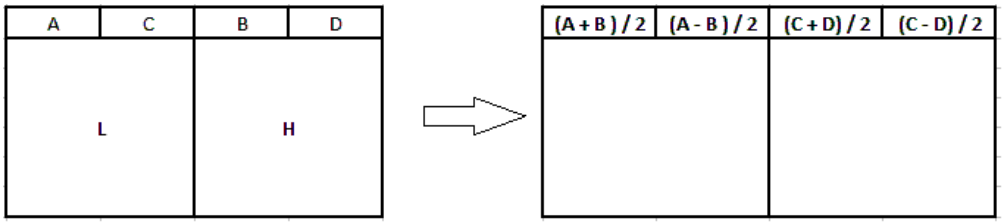


Figura 4.5 Segundo passo da inversa da 2d-Haar-DWT

O último passo no mascaramento é substituir as regiões robustas modificadas pelas correspondentes na imagem de cobertura, obtendo-se assim a estego-imagem.

4.2. Extração dos dados

O algoritmo precisa da imagem de cobertura original. Nessa imagem vão se procurar os pontos-chave pela mesma estratégia do processo de inserção.

Para cada ponto-chave selecionado é extraída à região robusta da imagem de cobertura e da estego-imagem e é aplicada Haar–DWT em vários níveis a ambas. Os bits da mensagem secreta são obtidos mediante o seguinte procedimento:

Algoritmo 4.2 Fase de extração dos bits da imagem de cobertura

```
1:  {Sendo W e H a largura e altura da sub-região HH de maior nível}
2:  Para x = 1 até W
3:      Para y = 1 até H
4:          Se abs(HHS(x, y) - HHC(x, y)) > F2 então
11:                 b(x * W + y) = 1
12:         Senão
13:                 b(x * W + y) = 0
14:         Fim Se
5:      Fim Para
6:  Fim Para
```

Onde b(i) representa o bit *i-ésimo* da mensagem secreta, HHS(x, y) e HHC(x, y) representam o valor do pixel da sub-região inferior direita da região robusta da estego-imagem e a imagem de cobertura respectivamente, depois de aplicar o 2d–Haar–DWT em vários níveis. O F_2 deve ser um valor entre 0 e F_1.

Se a estego-imagem não sofreu modificações, o valor absoluto de (HHS(x, y) - HHC(x, y)) vai ser próximo a 0 ou a F_1, dependendo do valor do bit que se

inseriu. Mas, no caso que a estego-imagem sofra modificações básicas como a mudanças de tamanho, a diferença de casos não vai ficar tão clara. O valor pode ser calcular mediante a fórmula:

$$F_2 = p \times F_1, \qquad com\ 0 < p \leq 1 \qquad \text{(Equação 4.2)}$$

onde p é o grau de confiança para discriminar entre ter inserido um bit ou não.

Obviamente, do passo anterior vão ser extraídas tantas mensagens como foram inseridas. Alguns bits não se recuperam de maneira certa, produto da compressão própria do formato da imagem de cobertura. Então aplica-se um sistema de votação para determinar a mensagem embutida. Os bits dessas mensagens são misturados, determinando como valor final aquele que mais aparece.

4.3. Especificidades do algoritmo

O algoritmo geral tem várias mudanças que são importantes de se destacar. O primeiro é que o tamanho das regiões-robustas é de 2^i x 2^i bits. No nosso processo de seleção não vamos considerar o tamanho dos pontos-chave recuperados através do SURF. Mesmo se o raio do ponto é maior ou menor, nossas regiões serão de raio fixo (2^{i-1}). Para a seleção dos pontos-chave só vamos considerar o grau de fortaleza.

Seguindo os resultados de Hamid, vamos aplicar o número máximo de níveis possíveis à 2d-Haar-DWT, no caso *i* (ver Figura 3.5). Isto resulta em que a sub-região HH_i tem só tamanho de um pixel; portanto, só vai ser possível esconder um bit em cada região robusta. Lembrando da (Equação 4.1, a quantidade de regiões a selecionar neste caso está definida por:

$$Q = L \times R \qquad \text{(Equação 4.3)}$$

Contudo, o algoritmo pode ser otimizado aproveitando várias particularidades desta variante.

A primeira delas é que se o bit do dado embutido correspondente a uma região-robusta tem valor 0, essa região-robusta não vai sofrer modificações no final. Portanto, não é preciso fazer nenhum dos passos anteriores, ou seja, não é preciso aplicar 2d-Haar-DWT nem a inversa dela.

No caso que o bit é 1, também podemos explorar outra particularidade. Temos a seguinte matriz que representa a região-robusta:

$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ onde A, B, C, D representam as sub-matrizes de 2^{i-1} x 2^{i-1}.

Se aplicarmos, na matriz anterior, a 2d-Haar-DWT *i* níveis e, depois, a modificação:

$$HH_i' = HH_i + F_1 \qquad \text{(Equação 4.4)}$$

e, finalmente, aplicarmos a inversa da 2d-Haar-DWT, a matriz final é a seguinte:

$\begin{pmatrix} A+F & B-F \\ C-F & D+F \end{pmatrix}$ onde $F = \frac{F_1 I}{4^i}$ e I a matriz identidade. (Equação 4.5)

A demostração da (Equação 4.4) pode ser feita por indução matemática, como se segue:

Inicialmente temos que, para $i = 1$:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Depois, aplicamos o primeiro passo da 2d-Haar-DWT:

$$\begin{pmatrix} a+b & a-b \\ c+d & c-d \end{pmatrix}$$

Aplicamos, então, o segundo passo da 2d-Haar-DWT:

$$\begin{pmatrix} ((a+b)+(c+d)) & ((a-b)+(c-d)) \\ ((a+b)-(c+d)) & ((a-b)-(c-d)) \end{pmatrix}$$

Em seguida, aplicamos a (Equação 4.4)

$$\begin{pmatrix} ((a+b)+(c+d)) & ((a-b)+(c-d)) \\ ((a+b)-(c+d)) & ((a-b)-(c-d)+F_1) \end{pmatrix}$$

Depois, aplicamos o primeiro passo da inversa da 2d-Haar-DWT:

$$\begin{pmatrix} a+b & a-b+\frac{F}{2} \\ c+d & c-d-\frac{F}{2} \end{pmatrix}$$

e o segundo passo da inversa da 2d-Haar-DWT:

$$\begin{pmatrix} a+\frac{F}{4} & b-\frac{F}{4} \\ c-\frac{F}{4} & d+\frac{F}{4} \end{pmatrix}$$

Para $i = n+1$ temos que:

$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$, onde A, B, C, D representam as sub-matrizes de 2^n x 2^n

$$\begin{pmatrix} \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} & \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \\ \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix} & \begin{pmatrix} d_1 & d_2 \\ d_3 & d_4 \end{pmatrix} \end{pmatrix}$$

onde a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4, d_1, d_2, d_3, d_4 representam as sub-matrizes de 2^{n-1} x 2^{n-1}.

Aplicamos, agora, o primeiro passo da 2d-Haar-DWT em um primeiro nível:

$$\begin{pmatrix} \begin{pmatrix} a_1+b_1 & a_2+b_2 \\ a_3+b_3 & a_4+b_4 \end{pmatrix} & \begin{pmatrix} a_1-b_1 & a_2-b_2 \\ a_3-b_3 & a_4-b_4 \end{pmatrix} \\ \begin{pmatrix} c_1+d_1 & c_2+d_2 \\ c_3+d_3 & c_4+d_4 \end{pmatrix} & \begin{pmatrix} c_1-d_1 & c_2-b_2 \\ c_3-d_3 & a_4-b_4 \end{pmatrix} \end{pmatrix}$$

Aplicamos, em seguida, o segundo passo da 2d-Haar-DWT:

$$\begin{pmatrix} \begin{pmatrix} (a_1+b_1)+(c_1+d_1) & (a_2+b_2)+(c_2+d_2) \\ (a_3+b_3)+(c_3+d_3) & (a_4+b_4)+(c_4+d_4) \end{pmatrix} & \begin{pmatrix} (a_1-b_1)+(c_1-d_1) & (a_2-b_2)+(c_2-d_2) \\ (a_3-b_3)+(c_3-d_3) & (a_4-b_4)+(c_4-d_4) \end{pmatrix} \\ \begin{pmatrix} (a_1+b_1)+(c_1+d_1) & (a_2+b_2)+(c_2+d_2) \\ (a_3+b_3)+(c_3+d_3) & (a_4+b_4)+(c_4+d_4) \end{pmatrix} & \begin{pmatrix} (a_1-b_1)-(c_1-d_1) & (a_2-b_2)-(c_2-d_2) \\ (a_3-b_3)-(c_3-d_3) & (a_4-b_4)-(c_4-d_4) \end{pmatrix} \end{pmatrix}$$

Como $i > 1$, temos que primeiro aplicar a 2d-Haar-DWT na sub-matriz LL $i - 1$ vezes, depois a modificação em HH_i e, finalmente, a inversa da 2d-Haar-DWT. Mas, pela (Equação 4.4), temos que:

$$\begin{pmatrix} \begin{pmatrix} (a_1+b_1)+(c_1+d_1)+E & (a_2+b_2)+(c_2+d_2)-E \\ (a_3+b_3)+(c_3+d_3)-E & (a_4+b_4)+(c_4+d_4)+E \end{pmatrix} & \begin{pmatrix} (a_1-b_1)+(c_1-d_1) & (a_2-b_2)+(c_2-d_2) \\ (a_3-b_3)+(c_3-d_3) & (a_4-b_4)+(c_4-d_4) \end{pmatrix} \\ \begin{pmatrix} (a_1+b_1)+(c_1+d_1) & (a_2+b_2)+(c_2+d_2) \\ (a_3+b_3)+(c_3+d_3) & (a_4+b_4)+(c_4+d_4) \end{pmatrix} & \begin{pmatrix} (a_1-b_1)-(c_1-d_1) & (a_2-b_2)-(c_2-d_2) \\ (a_3-b_3)-(c_3-d_3) & (a_4-b_4)-(c_4-d_4) \end{pmatrix} \end{pmatrix}$$

onde $E = \frac{F_1 I}{4^n}$. A seguir, aplicamos o primeiro passo da inversa da 2d–Haar-DWT:

$$\begin{pmatrix} \begin{pmatrix} (a_1+b_1)+\frac{E}{2} & (a_2+b_2)-\frac{E}{2} \\ (a_3+b_3)+\frac{E}{2} & (a_4+b_4)-\frac{F_1}{2} \end{pmatrix} & \begin{pmatrix} (a_1-b_1) & (a_2-b_2) \\ (a_3-b_3) & (a_4-b_4) \end{pmatrix} \\ \begin{pmatrix} (a_1+b_1)-\frac{E}{2} & (a_2+b_2)+\frac{E}{2} \\ (a_3+b_3)-\frac{E}{2} & (a_4+b_4)+\frac{E}{2} \end{pmatrix} & \begin{pmatrix} (a_1-b_1) & (a_2-b_2) \\ (a_3-b_3) & (a_4-b_4) \end{pmatrix} \end{pmatrix}$$

para, então, aplicarmos o segundo passo da inversa da 2d-Haar-DWT e obtermos:

$$\begin{pmatrix} \begin{pmatrix} a_1+\frac{E}{4} & a_2+\frac{E}{4} \\ a_3+\frac{E}{4} & a_4+\frac{E_1}{4} \end{pmatrix} & \begin{pmatrix} b_1-\frac{E}{4} & b_2-\frac{E}{4} \\ b_3-\frac{E_1}{4} & b_4-\frac{E}{4} \end{pmatrix} \\ \begin{pmatrix} c_1-\frac{E}{4} & c_2-\frac{E_1}{4} \\ c_3-\frac{E}{4} & c_4-\frac{E}{4} \end{pmatrix} & \begin{pmatrix} d_1+\frac{E}{4} & d_2+\frac{E}{4} \\ d_3+\frac{E}{4} & d_4+\frac{E}{4} \end{pmatrix} \end{pmatrix}$$, ou seja,

$$\begin{pmatrix} A+\frac{E}{4} & B-\frac{E}{4} \\ C-\frac{E}{4} & D+\frac{E}{4} \end{pmatrix}$$, sendo que $\frac{E}{4}=\frac{F_1 I}{4(4^n)}=\frac{F_1 I}{4^{n+1}}$ □

q.e.d.

Pela (Equação 4.4) temos uma ideia mais clara de quanto o valor de F_1 vai modificar a região robusta. Entretanto, essa particularidade é muito conveniente para otimizar o algoritmo, porque podemos modificar a matriz diretamente sem a necessidade de aplicar a 2d-Haar-DWT nem sua inversa.

Nós podemos estimar a ordem de modificação da imagem de cobertura pelo algoritmo, para determinar quantas vezes vamos repetir a mensagem e o tamanho da região-robusta.

Comparando com o algoritmo descrito em Hamid et al. (2012), em uma imagem de cobertura de 512 x 512, a menor imagem com a identificação que nós conseguimos guardar é de 32 x 32, precisando para isso regiões robustas de 64 x 64 e repetindo-as umas 7 vezes. O grau de modificação é:

$$G = \frac{Q \times R^2}{M \times N} \qquad \text{(Equação 4.6)}$$

sendo Q a quantidade de vezes que vamos repetir o dado embutido, R tamanho da região robusta, M e N o comprimento e a largura da imagem de cobertura. Por exemplo:

$G = (7 * 64^2) / (512^2) = 7 * 2^{12} / 2^{18} = 7 / 2^6 \approx 0{,}10975$

Ou seja, a nossa implementação do algoritmo descrito em Hamid et al. (2012) modifica menos de 11% dos pixels da imagem de cobertura. Lembrar que o artigo propõe repetir de 5 a 10 vezes a mensagem em regiões robustas de 128 x 128 bits, o que aumentaria consideravelmente a quantidade de pixels da imagem de cobertura modificados.

Na nossa proposta o grau de modificação é:

$$G = \frac{Q \times L \times R^2}{M \times N} \qquad \text{(Equação 4.7)}$$

sendo Q a quantidade de vezes que vamos repetir o dado embutido, L a longitude em bits do dado embutido, R tamanho da região robusta, M e N o comprimento e a largura da imagem de cobertura.

Em nossos experimentos Q tomou valores entre 3 e 7 e o grau de confiança na obtenção do dado embutido foi elevado. Repetir mais do que essa quantidade, ao contrário do que se pode acreditar, influi negativamente nos resultados; porque o algoritmo teria que escolher uma quantidade maior de pontos-chave e esses não têm o grau de fortaleza ideal para esconder o dado.

O tamanho do dado embutido é 24 bits de longitude, mas em geral a quantidade de bits 1 não é maior de 16 (2^4).

Tomando como base Q e G do algoritmo anterior, podemos calcular o tamanho das regiões robustas para manter o mesmo grau de modificação:

$$G_1 = \frac{Q_1 \times {R_1}^2}{M_1 \times N_1} \qquad \text{pela (Equação 4.6)}$$

$$G_2 = \frac{Q_2 \times L_2 \times {R_2}^2}{M_2 \times N_2} \qquad \text{pela (Equação 4.7)}$$

Temos que $G_1 = G_2$, $Q_1 = Q_2$, $M_1 = M_2 > 0$ e $N_1 = N_2 > 0$, então

$${R_1}^2 = L_2 \times {R_2}^2$$

$$R_2 = \sqrt{\frac{R_1}{L_2}} = \sqrt{\frac{64^2}{2^4}} = 2^4$$

Isso geraria como resultado que, se queremos manter o grau de modificação da matriz da proposta de Hamid et al. (2012) e repetir o dado até 7 vezes, podemos utilizar regiões de 16 x 16 pixels.

5
Resultados Experimentais e Discussão

Este capítulo apresenta os resultados e conclusões acerca da parametrização do algoritmo, baseado em imagens e métricas padrão para algoritmos de esteganografia.

Com o propósito de avaliar o algoritmo, são aplicadas modificações na estego-imagem, depois é extraído o dado embutido e é comparado com o dado original calculando o *Bit Error Rate (BER)* definido pela seguinte fórmula:

$$BER = \left(\frac{quantidade\ de\ bits\ errados}{quantidade\ de\ bits\ escondidos}\right) \times 100 \qquad \text{(Equação 5.1)}$$

Também é feito o cálculo depois de aplicado o sistema de votação para obter a razão da quantidade de bits certos do dado embutido, denotado por ADR:

$$ADR = \left(\frac{quantidade\ de\ bits\ certos\ do\ dado\ embutido}{quantidade\ de\ bits\ do\ dado\ embutido}\right) \times 100 \quad \text{(Equação 5.2)}$$

Outra medida usada nestes experimentos é quanto se afeta a qualidade visual da imagem de cobertura. Neste caso temos a formula *Peak Signal to Noise Ratio (PSNR)*:

$$PSNR\ (I, I_S) = 10 \log_{10} \frac{MAX_I^2}{1/mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i,j) - I_S(i,j)\|^2} \qquad \text{(Equação 5.3)}$$

onde *I* é a imagem de cobertura I_S a estego-imagem e MAX_I é o máximo valor possível dos pixels de *I*.

Vamos tomar a imagem “Lena em tom cinza” (Figura 5.1) para mostrar o processo de seleção dos parâmetros. Esta imagem é parte do conjunto de imagens padrão usadas para avaliar este tipo de algoritmo. Os ataques que vamos mostrar são a mudança no formato e a diminuição do tamanho da imagem com compressão no formato JPGE.

O algoritmo tem vários parâmetros que nós podemos modificar para obter melhores resultados. Estes parâmetros são os seguintes:

- F: fator de visibilidade, o valor que vai aumentar o diminuir cada pixel, F = 3, 4;
- M: o tamanho da região robusta, M = 8, 16;

- Q: a quantidade de vezes que vamos esconder o dado, Q = 5, 7.

Os maiores valores de todos os parâmetros vão aumentar a efetividade das medidas BER e ADR, mas vão afetar negativamente o PSNR. A ideia é achar uma boa relação entre eles.

Figura 5.1 Uma das imagens utilizada para avaliar os algoritmos de esteganografia

5.1. Experimentos

A seguir apresentamos execuções do algoritmo com parâmetros diferentes e o que nós concluímos de cada execução.

5.1.1. Experimento 1

M = 8 Q = 5 F = 3

PSNR:56.070

Formato: JPGE sem compressão

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	95.83%	9.17%
50.00%	87.50%	21.67%
40.00%	75.00%	34.17%
33.33%	70.83%	40.83%

Formato: BMP, PNG, TIF

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	95.83%	9.17%
50.00%	95.83%	16.67%
40.00%	75.00%	30.83%
33.33%	79.17%	35.00%

Tabela 5.1 Resultados do algoritmo com M = 8 Q = 5 F = 3

Dos primeiros valores apresentados na Tabela 5.1 temos que o ataque de mudar o formato da imagem de JPGE para BMP, PNG, TIF não piora o resultado do algoritmo. Embora não se coloque esse tipo de ataque nos seguintes experimentos, nós sempre fizemos os testes e comprovamos essa conclusão. Também, temos que com esses parâmetros não alcançamos o alvo de extrair, com alto grau de certeza, o dado embutido de imagens com redução do tamanho.

5.1.2. Experimento 2

M = 8 Q = 7 F = 3

PSNR: 54.931

Formato: JPGE sem compressão

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	100.00%	8.93%
50.00%	87.50%	22.62%
40.00%	79.17%	35.12%
33.33%	70.83%	39.29%

Tabela 5.2 Resultados do algoritmo com M = 8 Q = 7 F = 3

Os resultados da Tabela 5.2 exibem como o aumento do Q influi positivamente nos resultados. Ao comparar com a Tabela 5.1 comprovamos que embora em alguns casos o BER aumentasse, o sistema de votação melhorou a solução. O que quer dizer que, em alguns casos, o algoritmo erra mais; porém os erros são mais espalhados, ou seja, são mais distribuídos.

5.1.3. Experimento 3

M = 8 Q = 7 F = 4

PSNR: 52.69

Formato: JPGE sem compressão

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	100.00%	4.76%
50.00%	100.00%	17.86%
40.00%	91.67%	25.60%
33.33%	87.50%	32.14%

Formato: JPGE, compressão 90%

Zoom	ADR	BER	Tam bits
100.00%	100.00%	0.00%	76.23%
75.00%	100.00%	5.95%	38.89%
50.00%	100.00%	19.05%	18.89%
40.00%	83.33%	29.76%	14.38%
33.33%	70.83%	35.71%	10.30%

Tabela 5.3 Resultados do algoritmo com M = 8 Q = 7 F = 4

Da Tabela 5.3 nós confirmamos que aumentando o fator de visibilidade alcançamos bons resultados em JPGE sem compressão, levando o ADR até 87.50% em imagens com um terço do tamanho original e mantendo o PSNR. Em imagens em JPGE com 90% de compressão, o ADR, embora melhore o resultado do experimento anterior, cai bastante; mas temos que considerar que essas imagens são menos de 15% do tamanho em bytes da imagem original, o que implica em uma redução considerável da informação.

5.1.4. Experimento 4

M = 16 Q = 7 F = 3

PSNR: 49.512

Formato: JPGE sem compressão

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	100.00%	0.00%
50.00%	100.00%	2.98%
40.00%	100.00%	11.91%
33.33%	100.00%	18.45%

Formato: JPGE, compressão 90%

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	100.00%	0.00%
50.00%	100.00%	4.76%
40.00%	100.00%	11.31%
33.33%	95.83%	19.05%

Formato: JPGE, compressão 80%

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	100.00%	1.19%
50.00%	100.00%	5.95%
40.00%	100.00%	13.10%
33.33%	95.83%	23.81%

Formato: JPGE, compressão 70%

Zoom	ADR	BER	Tam.bits
100.00%	100.00%	0.00%	35.84%
75.00%	100.00%	0.60%	19.21%
50.00%	100.00%	7.74%	9.77%
40.00%	100.00%	17.86%	7.52%
33.33%	95.83%	30.36%	5.57%

Tabela 5.4 Resultados do algoritmo com M = 16 Q = 7 F = 3

Neste experimento, embora o PSNR diminua, os resultados alcançados foram bem próximo aos nossos objetivos. O algoritmo só errou em um bit dos 24 da mensagem escondida originalmente em imagens com 6% do tamanho (em bits) original. Dos valores mostrados na Tabela 5.4 nós podemos perceber quão útil resulta a estratégia de votação porque mesmo o algoritmo errando 30% dos bits escondidos o resultado final é quase perfeito.

Embora o PSNR seja menor do que o resultado anterior, a qualidade das imagens a olho nu é, ao contrário do que se pode imaginar, superior. Ou seja, a estego-imagem deste experimento a olho nu tem menos variações que a estego-imagem do experimento anterior. É evidente que uma quantidade maior de pixels são modificados porque para cada bit se modifica uma região maior; porém o fator de invisibilidade desse experimento é menor e menos perceptível.

5.1.5. Experimento 5

M = 16 Q = 7 F = 4

PSNR: 47.137

Formato: JPGE, compressão 80%

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	100.00%	0.00%
50.00%	100.00%	1.19%
40.00%	100.00%	4.17%
33.33%	100.00%	15.48%

Formato: JPGE, compressão 70%

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	100.00%	0.00%
50.00%	100.00%	2.38%
40.00%	100.00%	6.55%
33.33%	95.83%	17.86%

Tabela 5.5 Resultados do algoritmo com M = 16 Q = 7 F = 4

Nesse experimento temos que o BER caiu bastante, ou seja, a quantidade de erros na extração dos bits escondidos diminuiu quase a metade em alguns casos. E, após a aplicação do sistema de votação, o algoritmo só errou no caso mais demandante. Mas, o PSNR em um resultado lógico se viu afetado, e a qualidade da imagem visual também, como podemos corroborar na Figura 5.2.

Embora a qualidade da estego-imagem tenha sido inferior, o resultado foi significativo. Nós temos dois cenários: certificação digital de imagens e de vídeos. Nos vídeos nós afetamos poucos *frames* e podemos afirmar que é uma quantidade insignificante em proporção à quantidade total do vídeo. Embora que, olhando com atenção na Figura 5.2, para perceber as mudanças, nós precisamos de um tempo para isso – tempo este que é muito maior do que temos quando a imagem fica exposta num vídeo. Ou seja, dá para concluir que,

em imagens, nós devemos parametrizar o algoritmo com valores próximos aos da Tabela 5.4; mas, nos vídeos, o algoritmo pode ser parametrizado como na Tabela 5.5.

Figura 5.2 Estego-imagem do algoritmo com M = 16 Q = 7 F = 4

5.2. Melhoras no reconhecimento da assinatura

Para aumentar o grau de certeza poderíamos afetar a imagem com outros valores dos parâmetros ou usar outras estratégias que não implicam em mudanças na imagem.

Se tomarmos um conjunto de assinaturas muito próximas, onde a distância considerada é a diferença dos bits entre elas, na hora da extração do dado, nós poderíamos errar o certificado digital com maior probabilidade.

Dado que nos experimentos a estratégia erra pouco, no máximo em um bit, a ferramenta pode usar um conjunto de assinaturas para certificar arquivos onde cada uma se diferencia das outras em pelo menos cinco bits. Assim reduzimos o risco de que uma assinatura possa ser confundida com outra. Nesse caso, uma vez extraído o dado, a ferramenta compararia com cada uma das assinaturas usadas e sugeriria graus de certeza para cada uma delas.

Por exemplo, o conjunto de assinaturas pode ser:

{S1 = [-101, 65, 78], S2 = [78, -101, 65], S3 = [65, 78, -101]},

onde as representações em bits delas são:

S1 = 10011011 01000001 01001110,

S2 = 01001110 10011011 01000001,

S3 = 01000001 01001110 10011011.

Nas representações acima, cada uma das assinaturas difere da outra em 14 bits. Então, certificamos uma imagem em três versões diferentes, depois pegamos uma dessas imagens e extraímos o dado embutido e, por fim, comparamos com todas as assinaturas. O resultado está mostrado no experimento a seguir.

5.2.1. Experimento 6

M = 16 Q = 7 F = 3

Formato: JPGE, compressão 80%

	S1		S2		S3	
Zoom	ADR	BER	ADR	BER	ADR	BER
100.00%	100.00%	0.00%	58.33%	41.67%	54.17%	64.09%
75.00%	100.00%	1.19%	57.14%	41.67%	55.36%	64.09%
50.00%	100.00%	5.95%	58.33%	41.67%	54.17%	64.09%
40.00%	100.00%	13.10%	54.76%	41.67%	52.98%	64.09%
33.33%	95.83%	23.81%	51.19%	45.83%	55.36%	64.48%

Tabela 5.6 Resultados do algoritmo com M = 16 Q = 7 F = 4

Podemos perceber que, mesmo quando o conjunto de assinaturas é uma combinação de valores iguais, a diferença da posição dos bits é bem marcada e, na hora de extrair os dados, o algoritmo pode selecionar com uma grande diferença entre uma e outra assinatura. Na Tabela 5.6 podemos corroborar que, no pior caso, com uma redução do tamanho original da imagem a 33.33%, a assinatura S1 é 95.83% similar à extraída da estego-imagem; sendo S3 apenas 55.36% e S2, pior ainda, em 51.19%.

Mesmo com a restrição mencionada anteriormente e acrescentando outras como que cada uma das assinaturas tenha pelo menos um bit a 1 em cada byte, e que a quantidade de bits a 1 não seja maior do que 16, o conjunto é grande suficiente para nossos objetivos; é possível construir um conjunto com quase de 12 mil assinaturas. Também, se se puder estimar a quantidade das assinaturas necessárias, podemos aproveitar isso para construir um conjunto onde a diferença entre elas seja o maior possível.

6
Conclusões e Trabalhos Futuros

Atualmente, embora haja muitas pesquisas na área das técnicas de esteganografia, o problema de mascarar uma informação sobre outro meio tornando-a “invisível” é um tópico complexo e em aberto. As técnicas exploram especificidades dos meios e dos problemas para tentar atingir o alvo satisfatoriamente.

Nossa proposta explora o fato de que para identificar a origem dos arquivos não é preciso inserir neles um certificado digital grande; pelo contrário, podemos atingir o objetivo com uma assinatura de apenas alguns poucos bits. Durante a pesquisa, experimentos realizados com outras assinaturas de maior tamanho (incluindo o caso de uma imagem de pequenas dimensões) mostraram uma qualidade inferior na estego-imagem e maior perda de informação no dado extraido.

Outra característica importante explorada é a possibilidade real de ter a imagem de cobertura na hora de extrair o dado. Neste caso é possível comparar as duas imagens e determinar as diferenças entre elas, assim não precisamos colocar informação extra na estego-imagem para a extração do certificado digital.

As duas características relatadas acima são imprescindíveis na estratégia utilizada pelo algoritmo desenvolvido nesta dissertação

Este capítulo apresenta as conclusões finais, enfatizando as principais contribuições, e sugere alguns trabalhos futuros.

6.1.
Conclusões Finais e Principais Contribuições

Nossa principal contribuição é apresentar um método capaz de identificar o certificado digital inserido previamente em uma imagem, mesmo com as dimensões reduzidas a 33% e com compressão de formato. Com a nossa metodologia e a nossa ferramenta, mostramos ser possível identificar a assinatura em imagens cujo tamanho em bits seja menor que 6% do original.

Nos experimentos apresentados no capítulo 5, mostramos como o algoritmo pode ser adaptado/parametrizado em função dos objetivos que queremos alcançar. Uma redução da imagem em bits de 10% do original pode ser um cenário extremo, onde a qualidade da imagem diminui muito com relação ao original. Se o algoritmo for utilizado em um cenário menos agressivo, quanto à qualidade da imagem, ele pode ser parametrizado para melhorar o PSNR – o que é diretamente proporcional à visibilidade da assinatura. No caso dos vídeos, aproveitando que o tempo que uma imagem fica exposta é relativamente pequeno, podemos deixar a assinatura mais "visível" para aumentar a possibilidade de detectá-la.

Nos resultados apresentados em Hamid et al. (2012) (Tabela 6.1), foi considerada a compressão de formato, mas não a compressão das dimensões da imagem. Comparando com nossos valores (Tabela 6.2), temos que o método proposto oferece vantagens. A qualidade da imagem melhora de um PSNR: 45.28 a 49.512. Assim mesmo, temos que a recuperação do dado embutido foi superior, chegando a ser de 100%. Só no caso com compressão das dimensões de 33.33% a nossa proposta foi inferior em ADR (95.83%), mas o BER ainda foi melhor.

PSNR: 45.28

Formato: JPGE, compressão 80%

Zoom	ADR	BER
100.00%	99.83%	39.84%

Tabela 6.1 Resultados de Hamid et al. (2012)

PSNR: 49.512

Formato: JPGE, compressão 80%

Zoom	ADR	BER
100.00%	100.00%	0.00%
75.00%	100.00%	1.19%
50.00%	100.00%	5.95%
40.00%	100.00%	13.10%
33.33%	95.83%	23.81%

Tabela 6.2 Resultados da nossa proposta

Como uma contribuição prática, desenvolvemos uma ferramenta que permite ao usuário a certificação digital de imagens e a extração do certificado digital dos arquivos previamente firmados, mostrando o grau de confiança na detecção da origem.

6.2. Trabalhos Futuros

Dificuldades com a biblioteca de manipulação de imagens e vídeos têm impossibilitado a conclusão da ferramenta para vídeos. Um próximo passo importante é a finalização da implementação para estes arquivos.

Em um primeiro momento o foco principal do nosso trabalho foi pesquisar sobre algoritmos que pudessem tratar imagens com alta compressão, tanto nas dimensões quanto em compressão de formato, com alto grau de sucesso, para depois levar à estratégia a vídeos – visto que um vídeo é a composição de uma grande quantidade de *frames* (imagens). Porém, um trabalho futuro deve focar em explorar particularidades dos vídeos, como o tempo. Hoje a estratégia consiste em esconder a assinatura completa em uma imagem, mas seria interessante ver como se comporta o algoritmo escondendo a assinatura parcialmente em várias imagens consecutivas como ilustra a Figura 6.1. Escolher todos os pontos robustos necessários para embutir uma assinatura dentro de uma única imagem implica na escolha de um número considerável de pontos – o que ocasiona detrimento do grau de fortaleza de alguns deles. Se, pelo contrário, reduzimos a quantidade de pontos por imagem, aumenta a robustez dos mesmos e a perda de informação pode ser menor.

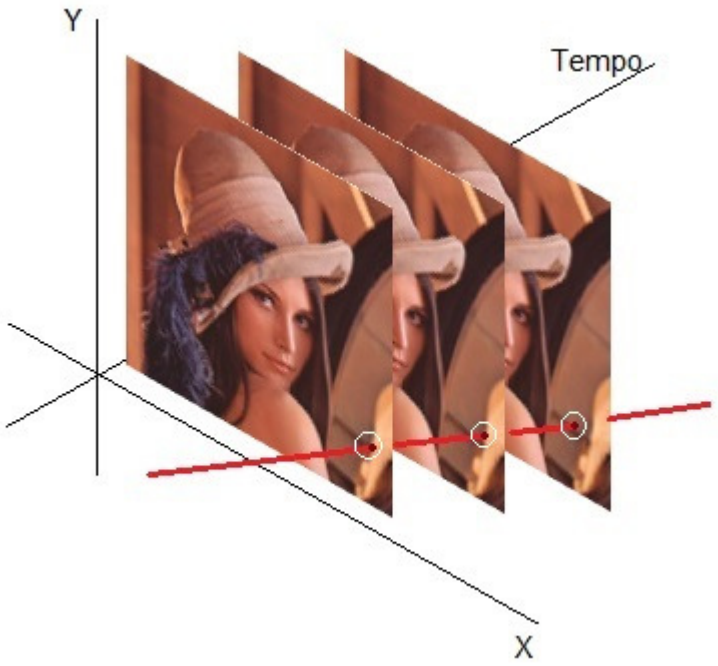


Figura 6.1 Estratégia para esconder a assinatura em vídeos

Referências Bibliográficas

Aniche, M.; 2012; Test-Driven Development: Teste e Design no Mundo Real; *Casa do Código*

Chen, P.; Lin, H.; 2006; A DWT Based Approach for Image Steganography; *International Journal of Applied Science and Engineering, 4, 3: 275-290*; Disponível em: < http://www.cyut.edu.tw/~ijase/2006/4-3(Microsoft%20Word%20-%20010-009-6).pdf > [acessado em 23/Mar/15]

Hamid, N.; Yahya, A.; Ahmad, R. B.; Al-Qershi, O. M.; 2012 A Comparison between Using SIFT and SURF for Characteristic Region Based Image Steganography; *International Journal of Computer Science Issues (IJCSI); Vol. 9 Issue 3, p110*. Disponível em: <ijcsi.org/papers/IJCSI-9-3-3-110-116.pdf> [acessado em 23/Mar/15]

Jiansheng, M.; Sukang, L.; Xiaomei, T.; 2009 A Digital Watermarking Algorithm Based on DCT and DWT; *Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA'09)*. Disponível em: http://www.academypublisher.com/proc/wisa09/papers/wisa09p104.pdf [acessado em 23/Mar/15].

Kessler, G. C. 2001. Steganography: hiding data within data. Disponível em http://www.garykessler.net/library/steganography.html. [acessado em 23/Mar/15].

Kessler, G. C. 2004. An overview of steganography for the computer forensics examiner. Forensic Science Communications, FBI, July 2004, Vol. 6, No. 3. Disponível em http://www.fbi.gov/about-us/lab/forensic-science-communications /fsc/july2004/research/2004_03_research01.htm. [acessado em 23/Mar/15].

Lau, S. 2001. An analysis of terrorist groups' potential use of electronic steganography. SANS Institute, Dec 17, 2001. Disponível em: https://cyber-defense.sans.org/resources/papers/gsec/analysis-terrorist-groups-potential-steganography-102684 [acessado em 23/Mar/15]

Outguess. 2004. Disponível em: <http://www.outguess.org> [acessado em 23/Mar/15]

Pagani, E. J.; Gaspar, W. B; Neves, C. V. A, 2007 Esteganografia e suas Aplicações. Livro de minicursos do SBSEG.

Petitcolas F. A. P.; Aanderson, R. J.; Kuhn, M. G. 1999. Information hiding — A survey. *Proceedings of the IEEE, v. 87, n. 7, p. 1062–1078*.

Provos, N. First steganographic image in the wild. University of Michigan, USA. Disponível em: <http://www.citi.umich.edu/u/provos/stego/abc.html> [acessado em 23/Mar/15].

Rocha, A. de R.; 2006. Randomização Progressiva para Esteganálise. *Dissertação (Mestrado). Universidade Estadual de Campinas, Campinas, Brasil*.

Wayner, P; 2002. Disappearing Cryptography: Information Hiding: Steganography and Watermarking (2nd Edition). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558607692.

Westfeld, A. 2001. F5—a steganographic algorithm: High capacity despite better steganalysis. *4th International Workshop on Information Hiding.* Disponível em: <https://code.google.com/p/f5-steganography/> [acessado em 23/Mar/15]

Zielinska, E., Mazurczyk, W. and Szczypiorski, K. 2014. Trends in Steganography, Comm. *ACM, Vol. 57, No. 3, P. 86-95*. Disponível em <http://cacm.acm.org/magazines/2014/3/172511-trends-in-steganography/ fulltext.> [acessado em 23/Mar/15]

A
Apêndice

Esta seção apresenta os requisitos, casos de usos, a modelagem, a arquitetura e outros detalhes técnicos relativos ao desenvolvimento da ferramenta.

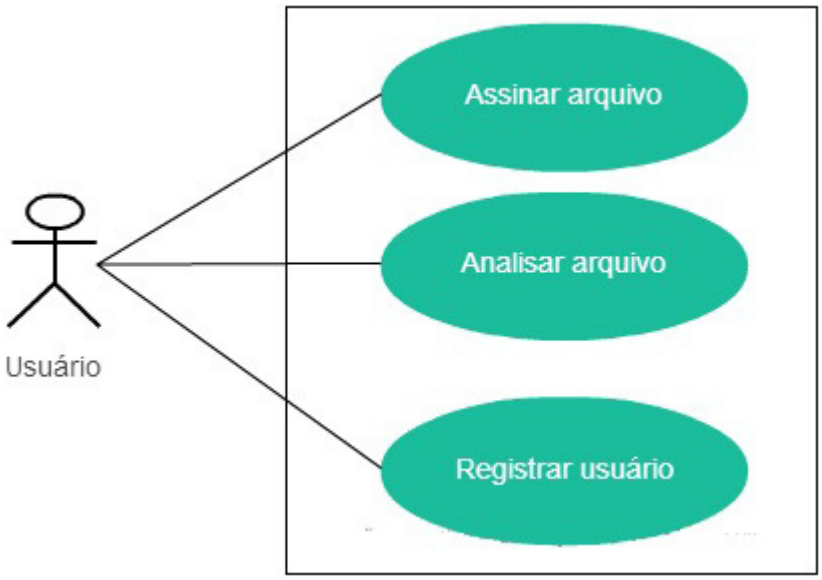
O sistema é desenvolvido em Java (JavaSe-1.7) usando o Eclipse, seguindo uma das práticas ágeis Test-Driven Development (TDD). A ideia do Test-Driven Development, ou, Desenvolvimento Guiado por Testes, é: que o desenvolvedor deve começar a implementação pelo teste e deve, o tempo todo, fazer de tudo para que o seu código fique simples e com qualidade (Aniche, 2012).

A.1.
Requisitos

Os requisitos se resumem em oferecer uma ferramenta que permita o seguinte:

- A certificação digital de imagens e vídeos.
- A extração do certificado digital de vídeos previamente firmados, mostrando o grau de confiança na detecção da origem.

A.2.
Diagrama de Casos de Uso



A.3.
Descrição de Casos de Uso

ASSINAR ARQUIVO - CASO DE USO

NOME
Assinar arquivo.

DESCRIÇÂO SUCINTA
O usuário vai assinar um arquivo, imagem ou vídeo, para um usuário registrado.

ATORES
1. Usuário

PRÉ-CONDIÇÕES
1. Ter registrado o usuário para quem vai ser assinado o arquivo

FLUXO BÁSICO
1. O usuário seleciona “Assinar Arquivo”
2. O sistema mostra tela correspondente
3. O usuário quer procurar o arquivo, ou seja, a imagem ou o vídeo que vai ser assinado
4. O sistema mostra janela com facilidades para procurar imagens ou vídeos
5. O usuário seleciona o arquivo
6. O usuário indica o usuário registrado.
7. O usuário seleciona ação “Assinar”.
8. O sistema executa o processo de assinar o arquivo
9. O sistema mostra o arquivo assinado
10. O usuário seleciona ação “Salvar”.
11. O sistema mostra janela para que usuário selecionar onde o arquivo assinado vai ser salvo.
12. O usuário seleciona a pasta e nome do arquivo.
13. O sistema salva o arquivo.
14. O caso de uso é encerrado

ANALISAR ARQUIVO - CASO DE USO

NOME
Analisar arquivo

DESCRIÇÂO SUCINTA
O usuário quer analisar um arquivo para conhecer para quem foi assinado. O sistema vai mostrar para quem foi assinado se tiver mais de 85% de grau de certeza, caso contrário vai mostrar a mensagem de que não foi possível identificar a assinatura.

ATORES
1. Usuário

PRÉ-CONDIÇÕES
1. Ter o arquivo original, o arquivo utilizado no caso de uso “Assinar arquivo” passo 3

FLUXO BÁSICO
1. O usuário seleciona “Analisar Arquivo”
2. O sistema mostra tela correspondente
3. O usuário quer procurar o arquivo a analisar
4. O sistema mostra janela com facilidades para procurar imagens ou vídeos
5. O usuário seleciona o arquivo
6. O usuário quer procurar o arquivo original
7. O sistema mostra janela com facilidades para procurar imagens ou vídeos
8. O usuário seleciona o arquivo
9. O usuário seleciona ação “Extrair”.
10. O sistema executa o processo de extrair dado do arquivo
11. O sistema mostra mensagem identificando para quem foi assinado se tiver mais de 85% de grau de certeza, caso contrário vai mostrar a mensagem de que não foi possível identificar a assinatura
12. O caso de uso é encerrado

FLUXO ALTERNATIVOS
(A1) Alternativa ao Passo 8 - A imagem não tem dado escondido
8.a O sistema detecta que o arquivo a analisar é muito diferente do que o arquivo original.
8.b O sistema informa ao usuário para que este possa procurar outro arquivo, o sistema retorna ao Passo 3

REGISTRAR USUÁRIO - CASO DE USO

NOME
Registrar usuário.

DESCRIÇÂO SUCINTA
O usuário vai registrar um usuário para quem pode ser assinado um arquivo.

ATORES
1. Usuário

PRÉ-CONDIÇÕES
1. Nenhuma

FLUXO BÁSICO
1. O usuário seleciona “Registrar usuário”
2. O sistema mostra tela correspondente
3. O usuário preenche o campo “nome” do usuário que quer registrar.
4. O sistema mostra se o nome é único no sistema, caso contrário não aparecerá ativa a ação “Registrar”
5. O usuário seleciona ação “Registrar”.
6. O sistema executa o processo de registrar usuário.
7. O caso de uso é encerrado

A.4. Diagrama de Classes