INSERÇÃO DE MARCAS D'ÁGUA DIGITAIS USANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Fabio Iguchi

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Sergio Lima Netto, Ph.D.

Prof. Murilo Bresciani de Carvalho, D.Sc.

Prof. Weiler Alves Finamore, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2007

IGUCHI, FABIO

Inserçao de marcas d'água digitais usando recorrência de padroes multiescalas [Rio de Janeiro] 2007

XVII,159 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2007)

Dissertaçao - Universidade Federal do Rio de Janeiro, COPPE

1.Ocultamento de Informaçao

2.Recorrência de Padroes Multiescalas

3.Marcas D'água Digitais 4.Segurança da Informaçao

I.COPPE/UFRJ II.Título (série)

Agradecimentos

Agradeço a Deus pela saúde e disposição para a realização deste trabalho.

À minha namorada Carolina Furukawa que esteve sempre do meu lado me ajudando a superar as dificuldade e sendo tão compreensiva com minhas alterações de humor. À minha família que pode prover tudo que necessitei e por tornar meu trabalho mais fácil e confortável.

A todos os professores e funcionários do Laboratório de Processamento de Sinais que sempre se mostraram disponíveis pra me ajudar em qualquer dificuldade que eu pudesse ter.

Em especial ao Professor Eduardo sempre atento ao meu trabalho, ajudando mesmo quando eu não percebia que precisava.

Aos meus amigos do LPS que sempre me ajudaram em todos os passos no mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

INSERÇÃO DE MARCAS D'ÁGUA DIGITAIS USANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Fabio Iguchi

Março/2007

Orientadores: Eduardo Antônio Barros da Silva

Programa: Engenharia Elétrica

Nesse trabalho, é proposto um método de inserção e decodificação de marcas d'água robustas e frágeis utilizando o algoritmo de codificação MMP (**Multidimensional Multiscale Parser**). Esse algoritmo utiliza uma estrutura em árvore e vários dicionários adaptativos com vetores de tamanhos diferentes na codificação de blocos de pixels. Essa estrutura garante uma enorme capacidade de adaptação a qualquer tipo de imagem.

O método proposto é composto de três algoritmos diferentes: O treinador, o insersor e o decodificador. Modificou-se o algoritmo MMP para ser usado como treinador dos dicionários a partir de várias imagens. Esses dicionários devem ser construídos de forma direcionada dependendo da robustez ou fragilidade da marca. O insersor de marcas d'água aproveita-se da estrutura em árvore utilizada pelo MMP para codificar cada um dos bits da marca d'água a serem inseridos. No final do processo, o resultado é uma imagem levemente alterada para que cada bit da marca corresponda a um bloco da imagem. O decodificador acha a melhor árvore para cada bloco da imagem e define qual bit ela representa restaurando assim a marca inserida.

Foi testada a robustez ou fragilidade do método de inserção de marcas d'água atacando a imagem de diversos modos diferentes.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

DIGITAL WATERMARK CODING USING MULTIDIMENSIONAL MULTISCALE PARSING

Fabio Iguchi

March/2007

Advisors: Eduardo Antônio Barros da Silva

Department: Electrical Engineering

In this work, it is proposed a method of insertion and extraction of robust and fragile watermarks using the algorithm MMP (Multidimensional Multiscale Parser). This algorithm uses of a tree structure and adaptative dictionaries with vectors of different sizes for coding of pixel blocks. This structure guarantees an enormous capacity of adaptation to any type of image.

The proposed watermark method is composed of three different algorithms: The trainer, the coder and the decoder. The MMP algorithm was modified to be used as trainer of a dictionary from some images. This dictionary must be constructed in a specific form depending on the robustness or fragility of the mark. The watermark encoder uses the tree structure of MMP to encode each one of the bits of the watermark to be embedded. In the end of the process, the result is a slightly modified image so that each bit of the mark corresponds to a block of the image. The decoder finds the best tree for each block of the image and defines which bit it represents, thus restoring the inserted watermark.

The robustness or fragility of the watermark method was tested by attacking the image in different ways.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

1.1 Ocultamento de Informações

As técnicas de inserção de marcas d'água digitais fazem parte de um conceito mais abrangente chamado de Ocultamento de Informações ou **Information Hiding** [1].

A proposta desse conceito é de esconder uma informação secreta dentro de uma outra informação que será o "disfarce". Para qualquer observador, a única coisa que deve ser percebida é a informação disfarce que geralmente é um material livre de qualquer suspeita. Por outro lado, a informação secreta é invisível a qualquer observador a não ser que ele saiba como decodificá-la a partir da informação disfarce. A própria existência de uma comunicação secreta deve passar despercebida para qualquer observador.

A idéia anterior é diferente do conceito de criptografia. Nesse último, é óbvio para o observador que existe uma mensagem secreta sendo passada, mas como não possui a chave para decodificação, ele somente enxerga uma seqüência de caracteres aleatórios que não fazem sentido. A mensagem neste caso não está oculta, só está modificada para impedir a leitura por pessoas não autorizadas.

Fica claro então que as técnicas de Ocultamento de Informação e Criptografia se diferenciam muito tanto no seu conceito como na sua implementação.

1.2 Esteganografia e Marcas D'água

Nesse trabalho serão abordados dois conceitos relacionados a Ocultamento de Informações: Esteganografia e Marcas D'água. Apesar de ambas visarem esconder uma informação dentro de outra, elas se diferem na sua aplicação e, portanto, na sua implementação.

Na Esteganografia, o objetivo é esconder uma mensagem secreta dentro de uma informação disfarce. Este último objeto pode estar em qualquer formato: outra mensagem, uma imagem, vídeo ou som. Desde que o método permita, praticamente qualquer dado pode ser usado para abrigar a mensagem oculta. Essa informação disfarce poderá ser observada e analisada por outros sem que a mensagem secreta seja detectada. Ao chegar ao seu destino, o receptor pode usar o método de decodificação apropriado para enxergar o que foi ocultado dentro do disfarce.

O ponto principal da Esteganografia é a invisibilidade à detecção. A técnica deve tentar prever quais métodos serão usados para detectar mensagem e se prevenir contra eles.

Na inserção de Marcas D'água ou **Watermarking**, o objetivo é marcar uma imagem, som ou qualquer dado apropriado com uma assinatura escolhida pelo autor ou proprietário da obra original. A assinatura (chamada, nesse caso, de marca d'água) pode ser qualquer coisa que represente o proprietário como seu nome, um logotipo ou sua voz. O objeto com a marca poderá então ser distribuído através da Internet ou qualquer outro meio que ele deseje. Se por acaso, uma pessoa ou pirata tentar copiar a imagem e registrá-la como sendo sua, a Marca D'água pode ser decodificada da obra do copiador provando a verdadeira propriedade do dado.

Entretanto, como se está trabalhando com imagens e sons, o pirata poderia se prevenir utilizando uma das diversas técnicas de processamento de dados amplamente divulgados na internet, alterando a imagem ou som, destruindo a Marca D'água sem no entanto estragar o trabalho original.

Então, a principal característica da técnica de inserção de Marcas D'água, nesse caso, seria a robustez. Mesmo que o pirata altere a imagem, a Marca D'água deve permanecer intacta ainda provando a propriedade do autor.

Existe também uma situação oposta ainda relacionada às técnicas de Marcas D'água. O autor pode inseri-la em uma imagem com a intenção de detectar

alterações na sua obra, ou seja, qualquer modificação destrói a informação inserida. Nesse caso, temos uma marca frágil.

A esteganografia está diretamente ligada à resistência à detecção enquanto que a inserção de Marcas D'água está ligada diretamente à resistência de (ou ausência) ataques.

Posteriormente, será explicado mais a fundo como esses dois conceitos são aplicados.

1.3 Motivações

A tecnologia avança de modo extraordinário e atitudes que antes pensavam-se impossíveis são uma realidade nos dias de hoje. A distribuição de arquivos de computador através de uma rede mundial, a possibilidade de criar CDs e DVDs no próprio computador pessoal e o armazenamento de arquivos de alguns **Gigabytes** em pequenos **pendrives** de alta velocidade, gera uma série de possibilidades e com isso, uma série de problemas. Tudo isso abriu portas para que pirataria de músicas e vídeos se expandisse.

Nunca se investiu tanto em medidas de segurança de informação. Empresas que desenvolviam programas anti-vírus tiveram que se reinventar e desenvolver atualizações constantemente para atender a alta taxa com que novos vírus são criados. A indústria de **firewall** (bloqueadores de transmissão de dados suspeitos) foi uma das primeiras a se beneficiar com esse avanço repentino.

Uma das novas tecnologias em segurança é o Ocultamento de Informação ou **Information Hiding**. Ele é impulsionado por dois dos maiores problemas dessa era de informação digital: direitos autorais e vigilância.

Atualmente, é extremamente fácil fazer cópia de qualquer produto digital como CDs ou DVDs. A indústria de entretenimento teme uma queda nas vendas devido a um crescimento na pirataria de tais produtos. Além disso, métodos de compressão como o MP3 fazem com que seja fácil distribuir arquivos de música pela Internet. Uma maneira de solucionar esse problema pode vir da maneira como esses produtos são vendidos. Afinal de contas, empresas que desenvolvem **softwares** já desistiram a muito tempo de confiar em mecanismos de proteção contra cópia, optando

por outros métodos como **upgrades** (atualizações) constantes, suporte técnico para os clientes que se cadastrarem em suas páginas via **Internet** e processando piratas de larga escala. No caso de música e vídeo, tais ações não se aplicam, mas, ainda assim, espera-se que uma solução técnica para proteção contra cópias ainda possa ser eficiente. Se forem inseridos dados da proteção contra cópia dentro do vídeo e da música de modo que sejam difíceis de apagar, solucionaria uma enorme parte do problema causado pela pirataria.

Na questão da vigilância, muito se discute sobre os perigos do uso de criptografia nas comunicações via **Internet**. Departamentos de inteligência de governos e agências de polícia proclamam que isso dificultaria o uso de escutas ou rastreamento. Eles demandam uma restrição no nível de segurança e força dos algoritmos de encriptação e requerem que cópias de chaves de segurança estejam disponíveis para eles. Por outro lado, as leis de direito do cidadão vêem essas medidas como uma enorme afronta à privacidade. Todas essas visões são um tanto simplistas em relação ao problema em questão. A maior parte das investigações policiais é feita rastreando quem está falando com quem ou rastreando os remetentes e destinatários de arquivos ou objetos que estão sendo investigados. Outra alternativa seria usar qualquer método relacionado à inserção de marcas de direitos autorais para ocultar um dado conteúdo numa outra mensagem. Isso faria com que as mensagens dos criminosos passassem despercebidas pelas autoridades e, mesmo que descobertas, elas não estariam inseridas na categoria de "criptografias não autorizadas pelo governo". Portanto, pelo menos nessa questão, eles não estariam cometendo um crime. Além do fato de o método ser mais difícil de se decodificar pois é menos conhecido pelas autoridades. Entretanto, as técnicas de ocultamento de informação podem ser usadas no rastreamento e não somente na evasão de detecções. Se uma marca for inserida numa imagem, por exemplo, pode-se colocar um software "robô"para rastrear na **Internet** essa dada imagem sendo usada de maneira ilícita. O mesmo vale pra vídeos, música ou textos.

As técnicas de ocultamento de informação também são importantes para privacidade. Existem informações como registros médicos, documentação pessoal e resultados de censos que não podem se tornar públicos. Registros médicos, por exemplo, algumas vezes são codificados de uma maneira sensata onde não se pode

rastrear o paciente. Caso seja feita de maneira errada, essa informação que deveria ser privada pode se tornar pública.

Todas essas questões impulsionam a pesquisa na área de ocultamento de informações. Em apenas 5 anos desde 1995 até 2000, o avanço em tecnologias de Ocultamento de Informação foi comparativamente maior que em 45 anos de pesquisa em criptografia desde 1945 até 1990. Muitos sistemas de Ocultamento de Informação foram propostos e a maioria deles já foi quebrado.

Nesse sentido, a técnica de compressão de imagens MMP (**Multidimensional Multiscle Parser**) é uma área interessante para a pesquisa de Ocultamento de Informações. O MMP tem um desempenho muito eficiente na compressão de imagens comparável aos melhores conhecidos. Ele se utiliza de uma divisão de imagem em blocos de pixels codificando cada um desses blocos seqüencialmente. Cada bloco de pixels é codificado com um dicionário com elementos do mesmo tamanho do bloco. Caso não se consiga decodificar o bloco com um erro abaixo de um determinado valor, o bloco é partido em dois menores e cada um deles é codificado com outro dicionário com elementos do mesmo tamanho deles. Esse processo se repete formando uma estrutura em árvore onde o bloco maior é a raíz e os blocos codificados são as folhas.

O MMP é um método adaptativo. Ele começa com um dicionário genérico e a cada bloco codificado, seu dicionário é renovado se adaptando a imagem codificada. Isso faz com que o MMP tenha um bom desempenho independente do tipo de imagem a ser codificada. Ele funciona bem em imagens comuns (de variações de tonalidade suaves) e em imagens com texto (com variações bruscas de tonalidade). O MMP é um método novo de compressão e por isso não existem pesquisas sobre sua eficácia em métodos de Ocultamento de Informação. Nesse trabalho, será estudado o desempenho do MMP na inserção de Marcas D'água digitais.

1.4 Organização da Tese

O Capítulo 2 apresenta o conceito de Ocultamento de Informação e seus desdobramentos em Esteganografia e Marcas D'águas Digitais. Conceitos básicos relacionados a marcas d'água que serão utilizados nesse trabalho serão apresentados.

O Capítulo 3 apresenta o algoritmo simplificado do MMP (**Multidimensional Multiscale Parser**) e propõe ambientes onde se possa inserir marcas d'água digitais.

O Capítulo 4 descreve o método proposto de inserção de Marcas D'água Robustas utilizando o conceito de Recorrência de Padrões Multiescalas. No final, analisam-se os resultados obtidos.

O Capítulo 5 demonstra como pode-se implementar um insersor de marcas frágeis utilizando os dicionários comuns gerados pelo algoritmo do MMP. No final, demonstram-se os resultados.

Por fim, o Capítulo 6 apresenta as conclusões deste trabalho.

Capítulo 2

Ocultamento de Informação

2.1 Introdução

Este capítulo tem como objetivo apresentar uma introdução sobre Ocultamento de Informação ou **Information Hiding**. Em primeiro lugar, será apresentado um histórico dessas técnicas que podem dar margem a abstração de métodos semelhantes com informações digitais. O histórico tem a intenção de mostrar que se pode elaborar métodos de Ocultamento de Informação de diversos objetos diferente em diversos contextos.

Em seguida, será apresentado o conceito de Esteganografia que dá uma base para o entendimento das técnicas de Marcas D'água que virá em seguida.

2.2 Histórico

A noção de Ocultamento de Informações tem origem antiga. Um dos exemplos mais famosos vem de 440 AC. Histaeus raspava a cabeça do seu escravo mais fiel e tatuava a mensagem no couro cabeludo que, depois que o cabelo crescia, ficava totalmente escondida. As mensagens passadas instigavam uma revolta contra os Persas. Por incrível que pareça, esse método ainda foi usado no começo do século XX por espiões Alemães.

Herotodus relata como Demeratus, um Grego na corte Persa, avisou Esparta de uma invasão iminente por Xerxes, rei da Pérsia. Ele escreveu uma mensagem em uma tábua e a cobriu com cera de vela. Para um observador qualquer, parecia ser

somente uma tábua de mensagens em branco. O objeto chegou a quase enganar o destinatário que a princípio, não suspeitou que pudesse haver uma mensagem secreta em tal lugar.

Um número grande de técnicas de Ocultamento de Informações foram registradas ou inventadas por Eneas, o estrategista, incluindo mensagens escritas nas solas dos pés de mensageiros, mensagens escondidas em brincos de mulheres e recados enviados por pombos correio. Ele também propôs esconder mensagens mudando levemente a altura das letras em uma mensagem ou fazendo pequenos furos acima ou abaixo das letras. Essa última técnica ainda foi usada no século XVII por Wilkins usando tinta invisível ao invés de furos. Posteriormente, essa técnica ainda foi usada durante as duas grandes guerras. Existe uma adaptação mais moderna dessa técnica que é ainda usada nos dias de hoje. Imprimem-se pequenos pontos do tamanho de pixels nas páginas de textos para se codificar informações como datas, identificadores de impressora ou de usuário.

Outro método foi sugerido pro Brewster em 1857. Ele sugeriu ocultar mensagens secretas em espaços "menores que um ponto final ou que um pequeno ponto de tinta". Em 1960, um fotógrafo francês resolveu o problema de criar mensagens minúsculas durante a guerra entre França e Prússia criando microfilmes que eram enviados por pombos correio.

Durante a guerra entre Rússia e Japão em 1905, imagens microscópicas eram escondidas em orelhas, narizes ou debaixo de unhas. Finalmente, durante a Primeira Guerra Mundial a idéia de Brewster se tornou realidade. Mensagens mandadas e recebidas por espiões eram reduzidas a medidas menores que um ponto por uma série de reduções fotográficas.

Tintas invisíveis foram extensivamente usadas para esconder mensagens. Elas eram fabricadas de substâncias orgânicas como leite ou urina e sua revelação era feita com calor. Os avanços na área química ajudaram a criar, durante a Primeira Guerra Mundial fórmulas mais eficientes tanto para tinta quanto para o líquido de revelação. Entretanto, o uso da tinta invisível caiu em desuso quando desenvolveram os chamados "reveladores universais"que revelavam onde as fibras de um documento foram molhadas com alguma substância. Esse processo remete a aplicações mais modernas e conhecidas como as marcas d'água em documentos importantes como

notas de dinheiro de modo a impedir falsificações.

Avanços mais modernos nessa área permitem fabricar tintas fluorescentes que respondem a luz ultra-violeta. Esse tipo de tinta é usado em cheques de viagem (**traveler's check**). As copiadoras comuns possuem um alto padrão ultra-violeta em suas luzes. Marcar um cheque com uma tinta que fluoresce sobre esse tipo de luz faz com que a cópia apareça com uma palavra "Inválido"impresso sobre o conteúdo do cheque. Isso torna fácil para qualquer pessoa reconhecê-la como sendo uma falsificação.

Outro exemplo vem da arquitetura. Desde os tempos remotos, os artistas perceberam que esculturas e pinturas tinham uma aparência diferente se for vista de ângulos específicos. Os estudos nessa área produziram regras para perspectiva e anamorfose. Entre os séculos XVI e XVII, imagens anamórficas eram usadas para esconder discursos políticos proibidos ou mensagens hereges. Um dos quadros anamórficos mais intrigantes, o Vexierbild, foi criado em 1530. Quando olhado diretamente, ele mostra uma paisagem um tanto incomum, mas quando vista pelas laterais, ela revela um quadro de um rei famoso.

Dentro da esteganografia, um método bastante utilizado é o acróstico. Como exemplo, existe um poema bastante famoso do escritor Giovanni Boccaccio (1313-1375) chamado **Amorosa visione** que é dito ser o maior acróstico do mundo. Boccaccio primeiramente escreveu três sonetos contendo mais ou menos 1500 letras cada. Posteriormente, escreveu outros poemas onde a letra inicial a cada três tercetos sucessivos correspondiam exatamente as letras dos sonetos originais. Outro famoso exemplo de um acróstico é o livro **Hypnerotomachia Poliphili**, publicado em 1499. Esse livro enigmático tem seu autor desconhecido, mas revela a estória de um amor proibido entre um monge e uma mulher. As primeiras letras de cada um dos trinta e oito capítulos do livro formam a frase "**Poliam frater Franciscus Columna peramavit**". Traduzindo, a frase significa "Irmão Francisco Colonna ama apaixonadamente Polia"onde Colonna era um monge ainda vivo quando o livro fora publicado.

Monges e outras pessoas ligadas a literatura logo expandiram a idéia simples do acróstico para encontrar métodos mais eficientes de ocultar mensagens em textos. Entre os séculos XVI e XVII surgiu uma enorme quantidade de literatura so-

bre esteganografia, muitos dos métodos dependiam bastante das novas tecnologias de codificar informação. Um exemplo é o livro escrito por Gaspar Schott (1608-1666) chamado **Schola Steganographica**. Nesse livro, o autor propõe 40 tabelas, cada uma com 24 palavras (uma para cada letra na época) em quatro idiomas diferentes: Latim, alemão, italiano e francês. Cada letra da mensagem a ser codificada é substituída pela palavra ou frase correspondente na tabela. No final, o texto formado acaba parecendo uma oração religiosa, uma simples carta ou uma espécie de frase mágica. Schott também explica como ocultar mensagens com notas musicais. Cada nota correspondendo a uma letra. John Wilkins mostrou como "dois músicos podem dialogar um com o outro usando música para transmitir suas mensagens como se fossem seus instrumentos de fala".

Pode-se conseguir resultados mais eficientes dessa técnica se a mensagem a ser escondida for espalhada em lugares aleatórios pelo texto. Essa idéia é a base de inúmeras técnicas de esteganografia atualmente. Num protocolo de segurança desenvolvido na China antiga, o remetente e o destinatário possuem "máscaras"de papel que nada mais são do que folhas de papel com furos do tamanho de letras. Essa folha é colocada em cima do papel onde será escrita a mensagem. Em cada um dos furos, em uma seqüência determinada, são escritas os caracteres da mensagem secreta. Após feito isso, a pessoa retira a máscara de papel e escreve uma mensagem disfarce de modo que os caracteres da mensagem secreta se encaixem nela. Quando o destinatário recebe a carta, ele simplesmente coloca a máscara sobre a carta e pode ler diretamente a mensagem que foi escondida. Lembrando que os caracteres chineses representam uma idéia e não somente uma letra como no nosso idioma.

2.3 O Princípio de Kerckho□s

A criptografia foi um tema muito mais discutido do que Ocultamento de Informação durante muito tempo. Por isso, apesar de a esteganografia ser diferente da criptografia, muito do que se desenvolveu da última pode ser usado em técnicas de Ocultamento de Informação. Em 1883, Auguste Kerckhoffs enunciou o primeiro princípio da engenharia criptográfica. Nesse princípio, ele sugere que nunca se presuma que o método de inserção de uma informação seja desconhecido

pelo seu oponente. Em outras palavras, deve-se criar algum dispositivo que garanta que, mesmo que um observador saiba como se insere e decodifica uma mensagem, ele ainda assim não consiga ver a mensagem oculta original.

Kerckhoffs propõe então que se use uma chave na inserção e decodificação das mensagens. Essa chave é um dado secreto que somente o remetente e o destinatário devem saber enquanto que o método de inserção e decodificação podem ser públicos.

Desde então, foi provado que confiar na obscuridade dos métodos de criptografia é realmente uma enorme falha.

Aplicando esse conhecimento em esteganografia, pode-se obter uma tentativa de uma definição para um sistema seguro: Um sistema seguro de esteganografia é aquele onde o oponente pode entender como ele funciona, mas sem a chave ele não terá evidência alguma do que está sendo transmitido ou até se existe alguma comunicação acontecendo. Um dos princípios centrais da esteganografia será o de publicar os processos de codificação a serem usados amplamente, assim como nos algoritmos de processos criptográficos comerciais. Esse princípio tem grande importância em particular na codificação de Marcas D'água. Nesse tipo de codificação, assume-se que o oponente sempre examinará detalhadamente o informação disfarce a procura de uma evidência de ocultamento de informação.

2.4 Aplicações Para as Técnicas de Ocultamento de Informação

Em atividades militares, a descoberta de comunicações secretas pode levar a um ataque imediato do inimigo. Mesmo com a encriptação, a simples detecção do sinal é fatal pois descobre-se não somente a existência de inimigos como também a sua posição. Unindo o conceito de ocultamento de informação com técnicas como modulação em espalhamento de espectro torna-se mais difícil de os sinais serem detectados ou embaralhados pelo inimigo.

Várias técnicas relacionadas a Ocultamento de Informação levam em consideração sistemas com níveis de segurança. Em uma rede de computadores militares existem vários níveis de segurança indo do mais restrito ao mais aberto. Um vírus ou um programa malicioso se propaga dentro do sistema indo de níveis de segurança

inferiores para os superiores. Uma vez que ele alcança seu objetivo, ele tenta passar informações sigilosas para setores de nível de segurança menores. Para isso, ele se utiliza de técnicas de Ocultamento de Informação para esconder informações confidenciais em arquivos comuns de maneira que o sistema deixe ele ultrapassar níveis de segurança.

Existem situações onde se deseja enviar uma mensagem sem que seja possível

Caso algum dos erros seja menor que o crit erio de m axima distor cao, o bloco pode ser codi cado. Caso contr ario, o bloco e dividido em dois menores.

No caso, o bloco ser a dividido em dois blocos 8 4 (8 linhas 4 colunas). Cada um desses novos blocos e comparado com os dicion arios de tamanho 8 4. Caso nenhum dos vetores codi que o bloco com um erro menor que crit erio de m axima distor cao ocorre uma nova divis ao para blocos 4 4. O algoritmo segue dividindo o n umero de colunas ou linhas por dois at e que todos os blocos sejam codi cados.

E importante ressaltar que n ao existe problema do algoritmo vir dir os blocos inde nidamente pois o dicion ario de vetores 1 1 reproduz todos os vetores existentes no seu espa co. Portanto, qualquer bloco 1 1 pode ser codi cado com erro zero por esse dicion ario.

O algoritmo de divis ao de blocos pode ser visto na gura 3.3.

Figura 3.3: Esquema de compara cao dos blocos sendo codi cados com os dicion arios correspondentes ao seu tamanho. Caso a codi ca cao n ao possa ser feita devido ao erro maior que o crit erio de m axima distor cao, o bloco e dividido em dois.

3.2.3 A Estrutura em Arvore

Observando a forma como os blocos são divididos percebe-se uma estrutura em arvore se formando. Na codificação, é necessário que se represente não somente o vetor que foi usado na codificação mas qual estrutura foi usada. Para isso, utiliza-se um bit de flag em cada nó ou folha para determinar se houve a divisão de bloco ou se foi encontrado um vetor que codifica o bloco.

Define-se, então que, para um determinado bloco, caso todos os erros sejam maiores que o critério de máxima distorção, utiliza-se o bit 0. Caso o bloco possa ser codificado, utiliza-se o bit 1.

A sequência de bits definida por esse esquema deve definir unicamente a arvore sendo codificada. Exemplos de estruturas em arvore e seus mapas de bits são demonstradas na figura 3.4.

Figura 3.4: Exemplos de estruturas em arvore e seus mapas de bits.

A sequência de bits é lida conforme os blocos vão sendo divididos. Por exemplo, no caso da arvore 0011011, os dois primeiros zeros referem-se a raiz e ao nó da esquerda. Como esse último não foi decodificado, ele se refere a dois outros blocos. Ambos foram codificados então obtem-se o terceiro e quarto bit que são iguais a 1. O próximo bloco é o da divisão do bloco raiz que ainda não foi fechado.

Abaixo dele pela direita existem um bloco não codificado que é o quinto bit que é igual a zero e as duas folhas abaixo dele da esquerda e depois pela direita são dos bits 1.

3.2.4 O Algoritmo Adaptativo

Cada vez que o algoritmo consegue codificar um bloco, ele adiciona nos dicionários as expansões e contrações do vetor utilizados. Isso é feito para que a dimensão do vetor adicionado seja igual a do dicionário onde ele está sendo inserido. O algoritmo de escalonamento citado em [17] é dado pelas formulas abaixo.

Para a expansão de um vetor S_{N_0} a um vetor maior S_N utiliza-se o seguinte procedimento:

$$
\begin{aligned}
& N_0 \; ! \quad N; N > N_0 \\
& m_n^0 = \left\lfloor \frac{n \quad (N_0 \quad 1)}{N} \right\rfloor; \\
& m_n^1 = \begin{cases} m_n^0 + 1; & m_n^0 < N_0 \quad 1; \\ m_n^0; & m_n^0 = N_0 \quad 1; \end{cases} \\
& {}_n = n \quad (N_0 \quad 1) \quad N \quad m_n^0; \\
& S_n^s = \left\lfloor \frac{{}_n \quad (S_{m_n^1} \quad S_{m_n^0})}{N} \right\rfloor + S_{m_n^0}; \\
& n = 0; 1; :::; N \quad 1:
\end{aligned}
$$

Onde o termo m_n^0 e m_n^1 são índices e $S_{m_n^0}$ e $S_{m_n^1}$ são os pixels relacionados aos índices.

No caso da contração onde transforma-se o vetor S_{N_0} em um vetor menor S_N:

$$N_0 \; ! \quad N; N < N_0$$

$$m^0_{n;k} = b\frac{n \quad (N_0 \quad 1) + k}{N}c;$$

$$m^0_{n;k} = \begin{cases} m^0_{n;k}; & m^0_{n;k} < N_0; \\ N_0 \quad 1; & m^0_{n;k} = N_0; \end{cases}$$

$$m^1_{n;k} = \begin{cases} m^0_{n;k} + 1; & m^0_{n;k} < N_0 \quad 1; \\ m^0_{n;k}; & m^0_{n;k} = N_0 \quad 1; \end{cases}$$

$$_{n;k} = n \quad (N_0 \quad 1) + k \quad N \quad m^0_{n;k};$$

$$S^s_n = \frac{1}{N_0 + 1} \sum_{k=0}^{N_0} (b\frac{_{n;k}(S_{m^1_{n;k}} \quad S_{m^0_{n;k}})}{N}c + S_{m^0_{n;k}});$$

$$n = 0; 1; :::; N \quad 1:$$

Os procedimentos acima s-ao descritos para casos unidimensionais. No caso de vetores bidimensionais, deve-se aplicar o algoritmo em uma dimens-ao e depois na outra.

Outro ponto importante do algoritmo adaptativo e que al-em da adic-ao das expans-oes e contrac-oes dos vetores encontrados, deve-se tamb-em adicionar as concatenac-oes dos mesmos. Por exemplo, no caso da arvore 011 e dos blocos 8 4 s-ao codi cados, a concatenac-ao dos dois blocos tamb-em deve ser adicionada no dicion-ario 8 8. Al-em disso, as expans-oes e contrac-oes desse vetor 8 8 em todas as dimens-oes s-ao adicionados.

3.2.5 Mem-oria

De acordo com o algoritmo adaptativo e f-acil perceber que, conforme os dicion-arios v-ao sendo renovados, em algum momento a mem-oria pode n-ao ser su ciente para a quantidade de dados sendo inserida. A n-ao ser que sejam cados os espacos na mem-oria dinamicamente, seria melhor haver algum mecanismo de administrac-ao desse recurso.

A cada novo vetor inserido no dicion-ario, e importante que seja dada a ele

uma prioridade maior. Por isso, cada novo vetor inserido e colocado no comeco do dicionario (ndice zero) e todos os outros elementos s-ao deslocados para espacos de menor prioridade (ndices maiores). Esse esquema e visto melhor na gura 3.5.

Figura 3.5: Exemplo esquematico do algoritmo de aumento de prioridade para vetores que codi cam um bloco.

3.3 MMP em Marcas D'agua

A insercao e ciente de uma Marca D'agua depende muito do meio que se escolhe para codi ca-la. Por exemplo, se o meio for robusto a modi cac-oes em relac-ao ao brilho da imagem, ent-ao a marca d'agua ser-a robusta a ataques focados no brilho. Se o meio for muito pouco robusto a diversos tipos de alterac-ao, pode-se utiliza-lo para abrigar marcas d'agua fr-ageis.

O MMP possui uma combinac-ao interessante de estruturas na sua codi cac-ao. Os dicion-arios s-ao bastante redundantes e, ap-os a codi cac-ao da imagem, estar-ao bem direcionados a codi car a mesma. A estrutura em arvore e utilizada na divis-ao dos blocos da imagem em vetores menores. Ela pode ser utilizada para abrigar bits da marca d'agua pois sua estrutura de ne genericamente como bloco ser-a codi-

cado pelos dicionarios sem de nir exatamente quais vetores dos dicionarios est ao sendo utilizados. Com isso, manipulando bem a utiliza c ao do MP nas imagens, pode-se gerar dicionarios direcionados tanto na codi ca o de marcas d'agua robustas como fr ageis.

Captulo 4

Marcas D'agua Robustas Usando Recorrência de Padrões Multiescalas

4.1 Introducao

Esse captulo explicara a base do conceito de insercao de marcas d'agua robustas utilizando o conceito de MMP. O codi cador utilizado pelo compressor pode ser modi cado para gerar um dicionario que ajude na insercao de marcas d'agua robustas. Isso e feito de modo que cada vetor dentro do dicionario seja o mais distante possvel dos outros.

Alem disso, o algoritmo modi cado faz com que as concatenacoes do dicionario imediatamente menor sejam distantes de cada elemento do dicionario atual. O mesmo ocorre pro dicionario imediatamente maior que esse. Cada par de vetores concatenados do dicionario atual deve ser o mais distante possvel do dicionario acima. Isso evita que codi cacoes feitas com um vetor grande sejam, apos um ataque, erroneamente decodi cadas como dois blocos menores. Este conceito sera explicado melhor mais a frente.

O processo todo envolvendo Marcas D'agua robustas e composto por tres algoritmos distintos:

O treinador

O insersor

O extrator

O esquema do processo todo de treinamento do dicionario, insercao e decodi-
cacao esta na figura 4.1.

Figura 4.1: Esquema total do processo de treinamento do dicionario, insercao e decodificacao de marcas dagua robustas.

Terminada o treinamento, espera-se que os dicionarios sejam compostos por diversos vetores distantes entre si. Espera-se tambem que seja muito difcil que uma concatenacao de dois vetores tenha a mesma configuracao de qualquer outro vetor de tamanho maior. Com isso, deseja-se obrigar que, uma vez utilizado um vetor na codificacao, seja muito difcil que se utilize qualquer outro ou quaisquer concatenacoes de vetores.

A insercao dos bits e feita de acordo com a estrutura em arvore mencionada na secao 3.2.3. A estrutura em arvore e mais eficiente do que usar os vetores do dicionario para abrigar os bits, pois a estrutura em arvore diz pouco sobre a configuracao da imagem. Uma mesma arvore pode definir um bloco escuro ou claro. Depende de que vetores ela estara usando. Com isso, pode-se definir quaisquer bits a serem usados numa insercao sem se alterar muito a configuracao dos pixels da imagem.

No final do captulo e feito uma analise dos resultados obtidos.

4.2 O Treinador

O treinador deve ser usado em diversas imagens para gerar um dicionario que seja eficiente quando for utilizado na codificacao de marcas d'agua robustas.

Em diversos pontos do processo de treinamento, o treinador se assemelha muito ao MMP. Ambos utilizam um dicionario inicial e tentam codificar a imagem em blocos atraves da comparacao com vetores. Tambem ocorre a divisao de blocos em ambos os casos.

De fato, os algoritmos sao identicos com excecao do momento em que e adicionado um novo vetor a qualquer um dos dicionarios. No caso do MMP o novo vetor e adicionado e colocado com uma prioridade grande no dicionario. O treinador para marca d'agua robusta atua de maneira diferente.

4.2.1 Limite de Vetores por Dicionario

Como foi visto anteriormente, o MMP se utiliza de dicionarios compostos por vetores de tamanhos diferentes. Se o tamanho dos blocos usados na codificacao da imagem for de 8 × 8 entao sao necessarios 7 dicionarios diferentes para a codificacao indo de 8 × 8 ate 1 × 1.

O treinador tenta afastar os vetores dos dicionarios assim como suas concatenacoes. O motivo e que a distância entre os vetores esta diretamente ligada a robustez do sistema. Quanto mais distantes forem os vetores, mais facil vai ser o erro cometido ao se passar de um vetor ao outro ou a uma concatenacao deles. Por isso, escolher um dicionario com uma quantidade grande de vetores tende a limitar esse afastamento pois o espaco vai estar mais densamente ocupado.

A princpio, tentou-se utilizar uma taxa r constante para os dicionarios para definir qual seria a quantidade de vetores por dicionario de acordo com a seguinte formula:

$$Q = 2^{r \cdot Dim} \tag{4.1}$$

Onde Q e a quantidade de vetores do dicionario sendo examinado, r e a taxa (bits=pixels) e Dim e a dimensao dos vetores do dicionario. Utilizando essa formula, com uma taxa de 1 bit por pixel, chega-se aos seguintes numeros:

Tabela 4.1: Quantidade de vetores por dicionário.

Dimensões	Quantidade de pixels	Quantidade de vetores
d6 2 1	2	4
d5 2 2	4	16
d4 4 2	8	256
d3 4 4	16	65.536
d2 8 4	32	4.294.967.296
d1 8 8	64	aprox. $1,8 \cdot 10^{19}$

Os nomes d1, d2 até d6 foram dados aos dicionários para facilitar a referência. O dicionário d7 de tamanho 1 1 não faz parte desse cálculo pois ele sempre tem 256 vetores com valores indo desde 0 até 256. Estes vetores raramente são alterados.

Seria impossível o processamento de uma base de dados tão grande quanto essa. Decidiu-se então que a quantidade de elementos seria limitada em 8000 vetores. Com isso, do dicionário de vetores 4 4 para cima, o número de elementos será 8000. Desse modo, pode-se alterar a taxa, desde que sempre se respeite esse limite de vetores por dicionário.

É importante também ressaltar que a quantidade de vetores por dicionário não pode ser pequena. É necessário que a marca d'água consiga representar a maior quantidade de imagens possível com baixa distorção. Para isso, seu dicionário precisa ser grande o suficiente para representar de modo eficiente uma grande quantidade de imagens, mas também pequeno o suficiente para não ter muitos vetores redundantes e conseguir assim aumentar sua robustez.

Nesse ponto, surgiu a idéia de se utilizar um dicionário diferente para cada um dos blocos de mesma dimensão conforme a figura 4.2. Por exemplo, na divisão do bloco 8 8 em dois 8 4, cada um dos blocos seria codificado, atualizado e decodificado por um dicionário diferente D2a e D2b. Isso seria feito em todas as dimensões. Então, seriam utilizados quatro dicionários para os blocos 4 4 (D3a,D3b,D3c e D3d), oito para os de 4 2 e assim por diante.

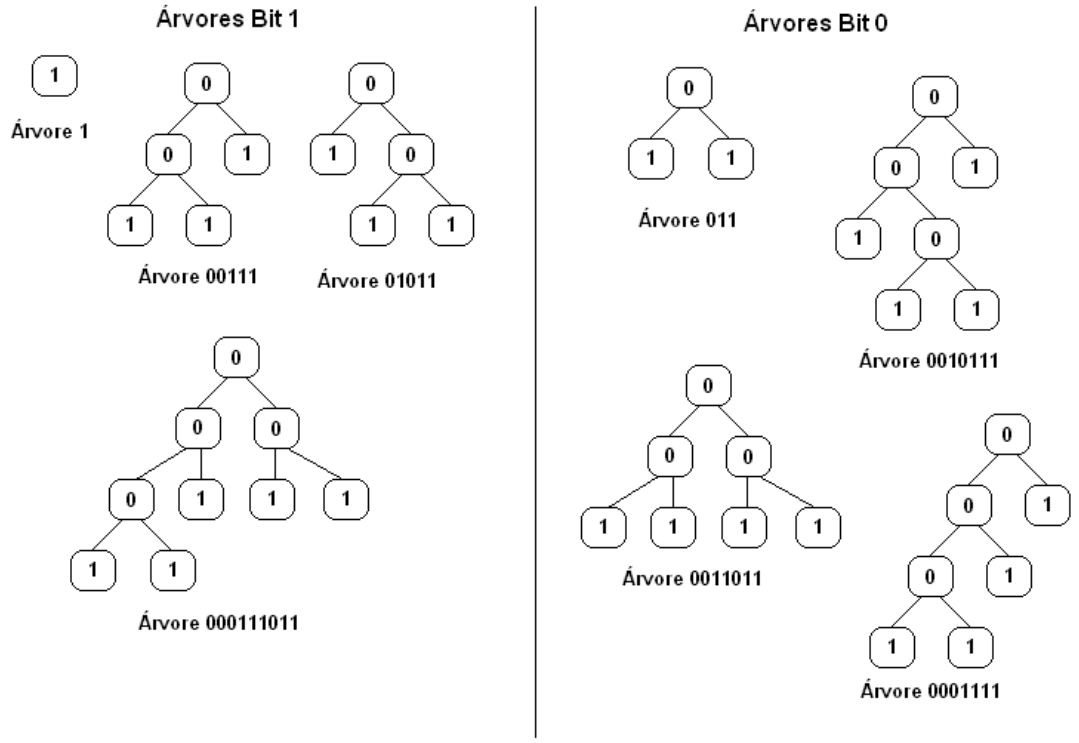

Figura 4.8: Exemplos de árvores que representam bit 0 ou bit 1.

Como o número de árvores é pequeno, seus mapas (em bits) foram armazenados em um arquivo que será lido pelo insersor e decodificador para serem utilizados.

4.3.3 Critério de Máxima Distorção e Uso da Taxa no Cálculo do Erro

Uma vez definido as entradas do processo e os padrões que serão utilizados, o insersor pode começar a atuar.

São lidos os arquivos que definem a marca que será utilizada, os dicionários e os mapas das árvores correspondentes a cada um dos bits. A marca d'água é transformada numa seqüência de bits. A imagem é dividida em blocos do mesmo tamanho que o utilizado no treinamento (no caso, 4 × 4).

O algoritmo tenta codificar cada um dos blocos com os vetores do dicionário. Caso algum dos vetores consiga fazer isso com um erro médio quadrático menor que uma dada tolerância (critério de máxima distorção), o algoritmo escreve o vetor na imagem resultante. Caso contrário, o bloco é partido em dois e cada um

deles é comparado aos dicionários correspondentes. Essa divisão vai sendo feita da mesma forma que o algoritmo do MMP. Caso nenhuma árvore satisfaça a tolerância estabelecida, a árvore com a menor distorção é codificada.

Existe uma questão a ser considerada. No algoritmo descrito até agora, as únicas coisas que importam na decisão da divisão dos blocos em blocos menores são o erro médio quadrático e a tolerância. Usando esses parâmetros, é provável que o algoritmo use árvores maiores, com mais ramos, pois a comparação com blocos menores tende a ter um erro menor que os blocos grandes. Isso é um problema pois, da mesma maneira que a codificação tem um erro menor com blocos pequenos, ela também fica mais sujeita a alterações causadas por ataques. Em outras palavras, árvores grandes são mais frágeis a ataques.

Decidiu-se, então, introduzir a taxa no cálculo do erro. Ela leva em consideração quantos bits seriam gastos na representação da árvore e dos índices dos dicionários correspondentes. A fórmula seguinte calcula a taxa de um dicionário sozinho:

$$\mathbf{r}_{m\times n} = \log_2 \mathbf{Q}_{m\times n} \tag{4.3}$$

Aonde r é a quantidade de bits necessária pra representar um vetor do dicionário $\mathbf{m \times n}$ e $\mathbf{Q}$ é a quantidade de elementos no dicionário.

Além disso, é necessário somar a quantidade de bits usada na representação da árvore em questão.

Por exemplo, caso sejam utilizados as árvores 011, a taxa correspondente é dada pelas fórmulas abaixo:

$$\mathbf{r}_{4\times 2} = \log_2 \mathbf{Q}_{4\times 2} = \log_2 8000 \approx 13 \tag{4.4}$$

Multiplicando por dois (o dicionário é usado duas vezes nessa árvore) e somando com os bits da árvore obtém-se o valor da taxa dessa árvore:

$$\mathbf{R}_{i\times j} = 2 \times 13 + 3 = 29 \tag{4.5}$$

Esse valor deve entrar no cálculo da distorção. Para que se possa regular a importância desse termo no cálculo do erro, será acrescentado um multiplicador $\boldsymbol{\lambda}$. Com isso, obtém-se um novo cálculo pra distorção em cada árvore dado por:

$$\mathbf{D}_{i\times j} = \mathbf{EMQ}_{i\times j} + \boldsymbol{\lambda} \cdot \mathbf{R}_{i\times j} \tag{4.6}$$

Se o valor de $\boldsymbol{\lambda}$ for bem regulado, pode-se controlar o algoritmo de modo que ele tenda a escolher as árvores menores e assim espera-se obter uma robustez maior na marca d'água.

É claro que a noção do critério de máxima distorção ainda vale. Só que, agora, ele também leva em consideração a taxa.

O método de inserção está explicitado no algoritmo A.6.

4.4 O Decodificador

O objetivo do decodificador é descobrir qual a árvore o insersor utilizou na codificação dos bits. Para isso, ele se utilizará de 2 mecanismos:

- A distância entre vetores geradas no treinamento.
- A idéia que um ataque não causará uma distorção muito grande na imagem.
- Uma estatística sobre a redundância dos bits inseridos.

Ele tem como entrada a imagem marcada (e possivelmente atacada) e os dicionários treinados. Ele também tem acesso aos parâmetros iniciais como o tamanho dos blocos e o tamanho da marca d'água em número de bits. No caso da utilização de chaves, ele também terá acesso a elas. Em alguns casos, pode-se até ter acesso a imagem marcada original (sem ataques). Afinal de contas, o proprietário da imagem tem acesso a esse dado.

É possível que, antes de se executar o algoritmo de decodificação em si, seja necessário fazer um pré-processamento na imagem a ser testada. Alguns ataques podem modificar bastante elementos da imagem que podem ser corrigidos com simples alterações como correção de brilho ou da faixa dinâmica de luminância da imagem. Para isso, basta comparar a imagem sendo testada com a imagem marcada original em um algoritmo de comparação dos fatores que se deseja investigar.

O decodificador divide a imagem marcada em blocos de pixels como foi feito no treinador e no insersor. Cada um desses blocos contém um bit da marca d'água

na seqüência. O decodificador decide qual bit está escrito em cada bloco e adiciona o dado num contador. Por exemplo, se ele achar que no primeiro bloco da imagem contém um bit zero, o contador de bits zero pro primeiro bit da marca é adicionado de 1. No final da decodificação, obtém-se uma estatística de cada um dos bits da marca d'água. Em cada um dos bits da marca, aquele que tiver a maior estatística é considerado o bit correto.

A decisão de qual bit será utilizado é feita pela distância entre o bloco sendo decodificado e os elementos dos dicionários. Para cada bloco, tenta-se codificar com todas as árvores relacionadas ao bit 1 e todas do bit 0. A árvore que for mais próxima do bloco sendo codificado indica o bit inserido.

No decodificador, não é necessário considerar a taxa nem o critério de máxima distorção. A taxa interfere no cálculo da árvore mais próxima pois adiciona um erro para árvores maiores. A tolerância não pode ser usada também pois pode permitir que se decida por uma árvore menor que a que foi utilizada na inserção. Portanto, o ponto principal do decodificador é decidir qual a árvore que foi usada no insersor achando a árvore com configuração de bits mais próxima do bloco da imagem.

O método de decodificação de marcas d'água robustas está descrito no algoritmo A.7 e sua continuação em A.8.

4.5 Qualidade de Inserção

Nessa seção será analisada a qualidade das imagens após a inserção da marca tanto na imagem de treino LENA como em imagens que não participaram do treinamento: BOAT (de tamanho 512×512), BABOON(512×512) e AERIAL(256×256). Isso é feito pois o objetivo do método é gerar dicionários que sejam genéricos para qualquer tipo de imagem.

Considerando o conceito básico de marcas d'água, a qualidade visual é importante mas fica em segundo plano se comparada à robustez. Para isso, foram testados os tamanhos de dicionários diferentes mostrados na tabela 4.2.

O dicionário d7 é utilizado no treinamento mas não na inserção ou decodificação e seu tamanho permanece inalterado em 256 elementos.

A marca inserida foi uma palavra de 10 bytes representando os números: 129,

Tabela 4.2: Quantidade de vetores por dicionário no treinamento robusto.

Dimensões	Quantidade de vetores
d7 1 1	256
d6 2 1	128
d5 2 2	256
d4 4 2	512
d3 4 4	8000

1, 2, 3, 4, 5, 6, 7, 8 e 255. A tolerância utilizada na inserção e o valor de foram zero para garantir uma maior qualidade da imagem.

O resultado da inserção na imagem LENA é demonstrada em 4.9.

Figura 4.9: Imagem LENA marcada.

A qualidade da imagem é boa se comparada a imagem original e pode ser utilizada em boa parte das aplicações comerciais. O erro médio quadrático dessa imagem em comparação a LENA original é de 17,637283. Entretanto, percebe-se que há alterações na mesma, principalmente nas partes de mais baixa frequência como a parte do ombro ou o cenário ao fundo. A princípio, o algoritmo é parecido com o MMP e a imagem deveria ter uma qualidade boa. Entretanto, as distorções são

causadas pela utilização de árvores que normalmente não seriam escolhidas numa codificação normal em MMP e por causa de um dicionário reduzido.

Isso demonstra que a quantidade de elementos por dicionário ainda foi pequena apesar do tempo de processamento gasto. Um aumento do número de vetores nos dicionários menores (2 1 ou 2 2) poderia resolver isso mas diminuiria a robustez do método. Outra solução seria aumentar a quantidade de vetores nos dicionários de dimensão maior mas isso aumentaria drasticamente o tempo de processamento.

Foi executada, então, a decodificação da marca d'água a partir dessa imagem sem ataques. O programa resulta em uma marca correta como era de se esperar. Entretanto, existe a possibilidade de algum dos blocos ter sido decodificado errado mesmo sem alterar a estatística dos bits da marca. Para verificar isso, foi implementado um programa que marca em branco os blocos que foram decodificados com bits diferentes da marca original.

O resultado do programa pode ser visto na figura 4.10.

Figura 4.10: Blocos decodificados erroneamente desenhados em branco na imagem marcada.

Existem dois blocos em branco na figura, um no canto superior direito e outro

perto do ombro da modelo. Se a imagem de tamanho 256 256 for considerada uma matriz de blocos 4 4 então ela teria dimensão 64 64 e os blocos codificados erroneamente seriam os $B_{1\ 64}$ e $B_{55\ 45}$. Analisando a configuração de cada um desses blocos, obtêm-se os seguintes dados mostrados na figura 4.11.

Figura 4.11: Blocos decodificados erroneamente sem ataques.

O treinamento feito com as imagens é eficiente pra maioria dos casos mas ele não garante que todos os seus vetores sejam distantes de suas concatenações. Como foi visto nas seções 4.2.2 e 4.2.3 em primeiro lugar, o algoritmo encontra os vetores que estão mais próximos dentro do dicionário de mesma dimensão que o vetor a ser inserido. Somente depois de selecionar esses vetores ele tenta compará-los com as concatenações de vetores menores e faz a concatenação dos mesmos pra compará-los aos vetores maiores.

O que pode ocorrer com esse algoritmo é que existam dois vetores de mesma dimensão que tenham a mesma configuração de um vetor maior só que nunca serão eliminados pelo algoritmo. Para que isso aconteça, basta que ambos estejam distantes de todos os outros vetores e que nenhum novo vetor esteja próximo a eles.

Isso pode ser explicado melhor observando a figura 4.12. Pela configuração mostrada, é de se esperar que novos vetores tenham uma grande probabilidade de surgirem no meio do espaço mostrado e não próximo dos vetores A ou B. E, por causa da distância dos dois, é improvável que eles sejam escolhidos para serem retirados do dicionário, a não ser que surja um novo vetor próximo deste.

Figura 4.12: Problema causados por vetores distantes que não são retirados dos dicionários pelo treinamento.

Esse é um problema que pode acontecer com o treinador mas de pouco impacto pois a frequência que ele ocorre é baixa e a marca d'água é inserida de forma redundante e compensa esse problema.

O padrão de qualidade percebido nos casos anteriores pode ser visto também nas imagens que não participaram do treinamento.

A figura 4.13 mostra a inserção da marca na imagem BOAT com erro de máxima distorção e iguais a zero. A qualidade da imagem é boa mesmo utilizando um dicionário que não foi treinado com ela. O erro médio quadrático dessa imagem em comparação a imagem BOAT original é de 19,345932.

Figura 4.13: Imagem BOAT marcada com crit erio de m axima dist orc ao e iguais a zero.

Utilizando uma dist orc ao e um fator mais altos a imagem tende a perder a qualidade como pode ser visto na gura 4.14. Nesse caso, foram xados o valor 10 tanto na dist orc ao quanto no fator . O erro m edio quadr atico aumenta para 78,300079 nesse caso.

Figura 4.14: Imagem BOAT marcada com critério de máxima distorção e $\boldsymbol{\lambda}$ iguais a 10.

Na figura 4.15 é demonstrado o resultado na imagem BABOON com distorção e $\boldsymbol{\lambda}$ iguais a zero. O erro médio quadrático desta imagem em relação à original é de 24,528889.

Figura 4.15: Imagem BABOON marcada com crit erio de m axima dist orc ao e iguais a zero.

Na gura 4.16e demonstrado o resultado na imagem BABOON com dist orc ao e iguais a 10. Nesse caso, a imagem possui poucas areas de baixa frequencia e a dist orc ao causada pelo aumento dos valores utilizados na sint ese tem pouco efeito. Consegue-se perceber uma alterac ao maior somente perto do canto. O valor do erro m edio quadr atico em relac ao a imagem original e 117,846650.

estiver codificado, mede-se a distorção em relação ao bloco da imagem original. Os blocos são então ordenados numa sequência de prioridades. Os blocos que tiverem erro abaixo do critério de máxima distorção são colocados com erro decrescente em sequência. Os que tiverem um erro maior que o critério são colocados em sequência crescente após os blocos anteriores. Isso é feito de forma que blocos que estiverem abaixo e próximos do critério de máxima distorção tenham prioridade sobre os outros. Essa ordenação de blocos é feita conforme mostra a figura 5.4.

Figura 5.4: Ordenação de blocos por prioridade em relação ao critério de máxima distorção.

Entretanto, ainda existe um problema. Pode acontecer de existir uma árvore com poucos nós que tenham a mesma configuração de blocos que a árvore que foi escolhida. Isso é demonstrado na figura 5.5. Isso se deve ao fato do dicionário utilizado ser muito redundante. Concatenações de blocos são inseridos nos dicionários e talvez aconteça de um bloco que foi dividido pela árvore escolhida ter sua concatenação em um dicionário de dimensões maiores. Isso é um problema pois o decodificador irá optar em utilizar o bloco concatenado e não dividido como o insersor tenta usar.

Figura 5.5: Possvel problema na escolha das arvores pelo metodo de divisao em arvores proposto.

Esse mesmo problema nao acontece com a primeira arvore que e encontrada (antes que fosse necessario gerar varias outras arvores com quantidades de zeros diferentes). Se existisse um bloco com dimensoes maiores que esse e tivesse a mesma con guracao do bloco sendo codi cado, entao a arvore nao o particionaria. Por exemplo, se um bloco 8 8 tivesse a mesma con guracao dos dois blocos 4 4 codi cados pela arvore 011 entao o algoritmo nem particionaria o bloco, pois a arvore 1 ja iria satisfazer o criterio de maxima distorcao.

Ja com as novas arvores esse problema existe pois elas foram criadas de forma diferente da arvore original. O mapa foi gerado sem que os blocos de maior tamanho tivessem sido testados. E necessario entao testar as arvores com uma parte do decodi cador para ver se existe ou nao uma arvore menor com a mesma con guracao da arvore gerada.

O teste e feito da seguinte maneira. O bloco 8 8 ja codi cado pela arvore gerada e inserido no algoritmo. Quando o bit do mapa da arvore for 0, o bloco atual e comparado com todos os vetores do dicionario de mesmo tamanho. Se algum deles tiver erro zero em relacao ao bloco testado, entao a arvore atual nao sera decodi cada corretamente e deve ser descartada. Caso nenhum vetor tenha erro zero entao o algoritmo pode continuar seu teste e o bloco atual e dividido em dois. Quando o bit do mapa for igual a 1 o bloco testado tera erro zero pois foi codi cado com um vetor do dicionario mas esse teste nao e necessario sendo ele proprio.

Uma vez terminado o teste acima para todos as arvores produzidas, se sobrar

alguma delas, o algoritmo codi ca o bloco com a arvore no topo da lista pois e a que tem maior prioridade.

5.3.2 Os Três Modos de Operacao

O insersor pode agir em três modos diferentes:

1. Insersor com dicionario nao-treinado adaptativo.
2. Insersor com dicionario treinado xo.
3. Insersor com dicionario treinado adaptativo.

No insersor com dicionario nao-treinado xo, e utilizado o dicionario inicial pra treinamentos. Cada um dos dicionarios de dimensao diferente e composto por 256 vetores diferentes. Cada vetor e uma matriz de pixels com mesmo valor indo desde o valor zero ate o 256. Essa matriz e inserida no insersor como se fosse um dicionario treinado. Durante a insercao, quando o bloco for codi cado, os dicionarios sao atualizados de forma idêntica ao treinador. Cada bloco codi cado e expandido e contrado para ser inserido nos dicionarios. O mesmo ocorre para as concatenacoes dos blocos.

E claro que o mesmo deve ser feito no decodi cador pois em cada bloco sendo decodi cado, ele deve ter o mesmo dicionario atualizado que foi utilizado pelo insersor.

O modo com dicionario xo e idêntico ao utilizado pelo insersor de marcas dagua robustas. Um dicionario treinado com varias imagens e dado como entrada no insersor e este permanece inalterado durante todo o processo.

O dicionario treinado adaptativo e uma juncao dos dois modos anteriores. Ele utiliza um dicionario ja treinado mas o atualiza conforme a marca e inserida na imagem.

Posteriormente, cada um desses modos serao analisados.

5.3.3 Chaves

No caso de marcas frageis, nao e interessante que se insira a marca de modo redundante. Esse metodo e utilizado para aumentar a robustez e teria o efeito oposto

ao desejado nesse caso.

Por isso, normalmente se tem uma marca que ocupa pouco espaco em relacao ao tamanho da imagem. Se a marca for inserida em blocos consecutivos, todo o processo terminaria muito antes da imagem ser razoavelmente coberta.

Uma maneira de resolver isso seria espalhar a marca pela imagem utilizando o conceito de chaves. Como foi visto na secao 2.10 as chaves podem ser utilizadas de diversas formas. Uma delas define em que posicoes da imagem foram inseridos os dados da marca d'agua. Nesse trabalho, os blocos foram marcados em sequência mas saltando-se um numero aleatorio de blocos de pixels em cada marcacao. Com isso, a marca fica espalhada pela imagem por interferência da chave utilizada.

5.3.4 Algoritmo de Insercao da Marca D'água Fragil

O metodo de insercao de marca d'agua fragil esta descrito no algoritmo A.9.

A funcao "de nearvore" mencionada no algoritmo A.9 e explicado no algoritmo A.10 e continua em A.11.

5.4 O Decodificador

O decodificador tem como entrada a imagem marcada e possivelmente alterada, os mesmos dicionarios utilizados pelo insersor e uma possivel chave. Ele divide a imagem em blocos com mesma dimensao que o insersor e define quais serao utilizados na decodificacao com a chave.

Cada bloco e entao comparado com os dicionarios de maneira semelhante ao algoritmo MMP e somente considera o bloco bom pra ser decodificado caso ele encontre um vetor que tenha erro zero em relacao a ele.

Conforme o algoritmo divide os blocos, ele escreve o mapa nesta arvore. Esse mapa e utilizado no final para definir qual bit e decodificado do bloco de acordo com o numero de zeros em sua estrutura.

Quando um bloco e decodificado, caso o insersor tenha usado um dicionario adaptativo, ele tambem deve atualizar esses dados.

E de se esperar que, uma imagem que tenha sido modificada antes da decodificacao, dê como resultado uma marca diferente da inserida pois o dicionario foi

gerado sem o afastamento feito no treinador de marcas d'agua robustas. Em outras palavras, e facil se alterar um bloco de modo que o decodificador acabe escolhendo uma arvore diferente da que foi utilizada.

5.4.1 Algoritmo de Decodificacao da Marca D'agua Fragil

A diferenca desse pro decodificador de marcas d'agua robustas e que esse so decodifica blocos cujos vetores tenham distorcao nula em relacao a eles. Portanto, a parte diferente e relativa a funcao de divisao do bloco em arvores. A funcao esta descrito no algoritmo A.12 e continua em A.13.

5.5 Qualidade de Insercao

Como foi dito anteriormente, o insersor pode atuar de 3 maneiras diferentes: Usando o dicionario inicial nao treinado num insersor adaptativo, usando um dicionario treinado num insersor nao-adaptativo e usando um dicionario treinado num insersor adaptativo.

5.5.1 Dicionario Nao Treinado Adaptativo

O primeiro a ser testado foi o dicionario nao treinado adaptativo. Como no caso do treinamento, o criterio de maxima distorcao usado foi zero para que os novos vetores inseridos nos dicionarios fiquem proximos uns dos outros. O tempo de processamento da insercao foi relativamente alto. Isso se deve ao fato da maioria dos blocos nao poder ser codificada com os vetores originais do dicionario. Com isso, era necessario que a arvore fosse muito dividida ate que se encontrasse um bloco que a codificasse com erro zero.

Entretanto, isso e um fator bom pois arvores maiores sao mesmo robustas e o objetivo e criar arvores frageis.

A figura 5.6 mostra o resultado da insercao da marca na imagem LENA.

Figura 5.6: Imagem LENA marcada com dicionario nao treinado adaptativo.

A marca inserida altera imperceptivelmente a imagem. Seu erro medio quadratico em comparacao a imagem original e de 0,000534. Isso e esperado pelo algoritmo MMP mostrado no captulo 3. O algoritmo de insercao utilizado se baseia no MMP diferindo somente quando o bit a ser inserido e diferente da arvore escolhida. Entretanto, mesmo com a arvore diferente, a codificacao ainda se baseia no MMP. Por isso, a codificacao de imagens tende a ser muito proxima da original caso o criterio de maxima distorcao seja definido com um valor baixo.

Na figura 5.7 mostra a imagem BABOON marcada com o mesmo procedimento. O erro medio quadratico dessa imagem se comparada com a original e de 0,000130.

Figura 5.7: Imagem BABOON marcada com dicionário não treinado adaptativo.

A figura 5.8 mostra a imagem BOAT marcada com o dicionário não treinando adaptativo. Nesse caso, a imagem tem um erro médio quadrático em relação a imagem original de 0,000153.

Figura 5.8: Imagem BOAT marcada com dicionário não treinado adaptativo.

A figura 5.9 mostra a imagem AERIAL marcada com o dicionário não treinando adaptativo. O erro desta imagem fica em 0,000626 se comparada a imagem original.

Figura 5.9: Imagem LENA marcada com dicionário não treinado adaptativo.

Foram testadas duas outras imagens nesse modo de inserção: A imagem COUPLE 5.10 e a imagem ELAINE5.11. No caso da imagem COUPLE, a inserção foi possível e o resultado foi bastante satisfatório pois a imagem é idêntica a original.

Entretanto, no caso da imagem ELAINE, não foi possível inserir a marca. Isso se deve ao fato de algum bloco codificado no início da imagem ter adicionado vetores no dicionário que atrapalharam a inserção dos bits em blocos posteriores.

Figura 5.10: Imagem COUPLE original.

Figura 5.11: Imagem ELAINE original.

Pode-se dizer que a inserção foi bem sucedida na maioria dos casos. Entretanto, existe a possibilidade de vetores adicionados no dicionário durante a inserção atrapalharem a codificação de blocos posteriores da mesma imagem.

5.5.2 Dicionário Treinado Não Adaptativo

No caso do dicionário treinado não adaptativo, foram usadas as seguintes imagens de treino: LENA, BABOON, BOAT e AERIAL. Com isso, pretende-se criar um dicionário genérico que consiga representar a maior variedade de imagens possível.

Nesse caso, foram encontrados problemas na inserção. Pelo fato do dicionário ser redundante, existe a possibilidade do extrator encontrar uma árvore diferente do insersor como foi visto anteriormente em 5.3.1.

Nesse caso, uma alternativa caso a árvore encontrada seja grande, é aumentar o critério de máxima distorção. Com isso, espera-se que o inserção tente escolher uma árvore menor que tenha o bit correto ou que, ao gerar outras árvores, ele encontre uma que não seja decodificada errada. Entretanto, caso a árvore seja pequena pode acontecer da imagem não poder ser marcada.

Nos testes com as imagens foi impossível marcar a imagem LENA pois um dos seus blocos acusou erro na inserção como explicado anteriormente e sua árvore era a menor possível (com o mapa 1). Entretanto, nas outras imagens foi possível a inserção regulando a distorção máxima. Na tabela 5.1 são mostrados os critérios necessários para inserção com dicionários treinados não-adaptativos.

Tabela 5.1: Critério de máxima distorção necessário para inserção da marca com dicionário treinado não adaptativo.

	Critério de máxima distorção
LENA	Não foi possível a inserção
AERIAL	50
BABOON	10
BOAT	20

Existe o problema das distorções altas produzirem uma robustez maior indesejada. Isso será testado posteriormente com os ataques.

As imagens que foram marcadas apresentaram uma qualidade boa quanto as mostradas anteriormente.

Entretanto, para as imagens COUPLE e ELAINE que não participaram do treinamento dos dicionários a inserção por esse modo não foi possível. Os dicionários treinados, mesmo com o modo adaptativo que será visto depois, geram erros de inserção nos primeiros blocos da imagem.

Pode-se definir então que o treinamento a priori de dicionários tende a deixar o dicionário suscetível a erros na inserção para imagens que não participaram do treinamento e principalmente para outras imagens fora do treinamento.

5.5.3 Dicionário Treinado Adaptativo

Novamente, o teste com as imagens gerou erros de inserção mas, nesse caso, conseguiu-se contorná-los manipulando o critério de máxima distorção.

O resultado pode ser visto na tabela 5.2.

Tabela 5.2: Critério de máxima distorção necessário para inserção da marca com dicionário treinado adaptativo.

	Critério de máxima distorção
LENA	10
AERIAL	10
BABOON	0
BOAT	10

O resultado visual das imagens é comparável aos anteriores.

5.5.4 Dicionário Não Treinado Fixo

Apesar de não participar do conjunto dos modos de operação do inserção frágil, foram testados dicionários não treinados fixos nas imagens que não participaram do treinamento com o objetivo de averiguar se a adaptabilidade e o treinamento dos dicionários realmente é o que atrapalha a inserção.

Em ambos os casos, as marcas d'água foram inseridas com sucesso mesmo com utilização de árvores grandes que garantem uma maior fragilidade ao método.

5.6 Resposta a Alterações na Imagem

Nessa seção serão testados os ataques de diminuição do sinal utilizados em