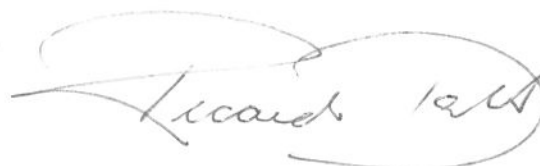


Uma Alternativa Pública para o Método de Marcas d'Água Raster em Mapas Vetoriais (RAWVec)

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Karina Mochetti de Magalhães e aprovada
pela Banca Examinadora.

Campinas, 09 de junho de 2009.

A handwritten signature in black ink, appearing to read "Ricardo Dahab", enclosed within a large, loopy oval flourish.

Prof. Ricardo Dahab (Orientador)
Instituto de Computação - UNICAMP

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Crislene Queiroz Custódio – CRB8 / 7966

Magalhães, Karina Mochetti de

M27a Uma alternativa pública para o método de marcas d'água Raster em mapas vetoriais (RAWVec) / Karina Mochetti de Magalhães -- Campinas, [S.P. : s.n.], 2009.

Orientador : Ricardo Dahab

Dissertação (Mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Marcas d'agua digital. 2. Proteção de dados. 3. Mapas vetoriais.
I. Dahab, Ricardo. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

(cqc/imecc)

Título em inglês: A blind alternative for raster watermerking vector maps (RAWVec).

Palavras-chave em inglês (Keywords): 1. Digital watermarking. 2. Data protection. 3. Vector maps.

Área de concentração: Ciência de Computação

Titulação: Mestre em Ciência de Computação

Banca examinadora: Prof. Dr. Ricardo Dahab (IC-Unicamp)
Prof. Dr. Michel Abdalla (École Normale Supérieure-França)
Prof. Dr. Siome K. Goldenstein (IC-Unicamp)

Data da defesa: 11/03/2009

Programa de Pós-Graduação: Mestrado em Ciência de Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 11 de março de 2009, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Michel Abdalla
École Normal Supérieure - França.



Prof. Dr. Siome Klein Goldenstein
IC / UNICAMP.



Prof. Dr. Ricardo Dahab
IC / UNICAMP.

Uma Alternativa Pública para o Método de Marcas d'Água Raster em Mapas Vetoriais (RAWVec)

Karina Mochetti de Magalhães¹

Março de 2009

Banca Examinadora:

- Prof. Ricardo Dahab (Orientador)
Instituto de Computação - UNICAMP
- Prof. Michel Abdalla
École Normal Supérieure, França
- Prof. Siome K. Goldenstein
Instituto de Computação - UNICAMP
- Prof. Roberto A. Lotufo (Suplente)
Faculdade de Engenharia Elétrica e de Computação - UNICAMP
- Prof. Julio López (Suplente)
Instituto de Computação - UNICAMP

¹Suporte financeiro de: FAPESP (processo 2006/05219-7) 2007-2008.

Resumo

Marcas d'água vêm sendo utilizadas em vários tipos de arquivos de multimídia, como áudio, vídeo e imagens de bitmap, para combater sua cópia e distribuição ilegais. Somente com a popularização dos Sistemas de Informação Geográfica (GIS) e da utilização de dados espaciais na Web, marcas d'água para mapas vetoriais ganharam maior atenção, principalmente porque este tipo de imagem possui produção cara, mas pode ser reproduzido facilmente. Esse trabalho revisa o método RAWVec, que insere uma marca d'água representada por uma imagem raster ou bitmap num mapa vetorial, e apresenta uma alternativa pública para tal método.

Abstract

Watermarks have been used for a long time in several digital media to avoid the illegal copy and distribution of these items. Only with the popularization of Geographic Information Systems (GIS) and the use of spatial data on the Web, watermarks for vector maps gained more attention, mostly because vector maps are expensive to produce, but easy to copy. This work reviews the RAWVec Method that embeds a watermark, in the form of a bitmap image, into digital vector maps; and presents a semi-blind alternative method.

Agradecimentos

Agradeço à minha mãe Cristina por todo amor e dedicação e por me encorajar a estudar Computação.

Ao meu pai Tony por todo amor e por sempre acreditar em mim.

À minha avó Virgínia e ao meu avô Roberto (in memoriam) por todo carinho.

À minha avó Cândida e ao meu avô Heraldo (in memoriam) por sempre incentivarem os meus estudos.

Ao Fabio por ter aguentado comigo cada momento feliz, triste e estressante da faculdade nesses últimos 7 anos.

Ao meu orientador Ricardo Dahab por todo o conhecimento compartilhado.

E a todos os amigos, da computação ou não, que ajudaram direta ou indiretamente na construção desta tese.

”The scientist does not study Nature because it is useful; he studies it because he delights in it, and he delights in it because it is beautiful. If Nature were not beautiful, it would not be worth knowing, and if Nature were not worth knowing, life would not be worth living”

Henri Poincaré

Sumário

Resumo	2
Abstract	3
Agradecimentos	4
1 Introdução	1
1.1 Trabalhos Correlatos	2
1.2 Organização do Texto	4
2 O Método RAWVec	5
2.1 Funções Auxiliares	5
2.2 O Algoritmo de Point Pattern Matching	7
2.3 O Método RAWVec	9
2.3.1 Algoritmo de Inserção	9
2.3.2 Algoritmo de Detecção	11
2.4 O Método SB-RAWVec	13
2.4.1 Criptografia Visual	13
2.4.2 Algoritmo de Inserção	17
2.4.3 Algoritmo de Detecção	19
3 Resultados Experimentais	23
3.1 Análise dos Algoritmos	23
3.1.1 Complexidade	23
3.1.2 Perturbação Inserida no Mapa	25
3.1.3 Métricas para Comparação de Imagens	25
3.2 Resultados Práticos	26
3.2.1 Extração sem Ataques	26
3.2.2 Ataques Geométricos	32
3.2.3 Ataques Criptográficos	46

3.2.4	Ataques de Protocolo	46
3.2.5	Ataques de Remoção	47
4	Conclusão e Trabalhos Futuros	48
4.1	Trabalhos Futuros	48
	Bibliografia	50

Lista de Tabelas

3.1	Complexidade de cada passo dos algoritmos de inserção.	24
3.2	Complexidade de cada passo dos algoritmos de detecção.	24
3.3	Erros absolutos e relativos para os testes realizados com três diferentes mapas vetoriais.	31
3.4	Resultados da execução dos métodos sem nenhum ataque.	31
3.5	Resultados da execução dos métodos com ataques de rotação.	33
3.6	Resultados da execução dos métodos com ataques de translação.	34
3.7	Resultados da execução dos métodos com ataques de escalonamento.	35
3.8	Resultados da execução dos métodos com ataques de cropping.	37
3.9	Resultados da execução dos métodos com ataques de inserção de objetos.	40
3.10	Resultados da execução dos métodos com ataques de alteração na ordem dos objetos.	42
3.11	Resultados da execução dos métodos com ataques de inserção de ruído aleatório.	43
3.12	Resultados da execução do Método RAWVec com ataques combinados.	44
3.13	Resultados da execução do Método SB-RAWVec com ataques combinados.	45

Lista de Figuras

2.1	Exemplo de aplicação da função $w(A)$	5
2.2	Exemplo de aplicação da função $v()$	6
2.3	Um exemplo de <i>Point Pattern Matching</i>	8
2.4	Diagrama do Algoritmo de Inserção da Marca d'Água do Método RAWVec.	10
2.5	Criação da imagem raster E , a partir da imagem raster R redimensionada.	11
2.6	Diagrama do Algoritmo de Detecção da Marca d'Água do Método RAWVec Original.	12
2.7	Composição dos pixels na Criptografia Visual.	14
2.8	Imagem original.	15
2.9	Imagem 1.	15
2.10	Imagem 2.	15
2.11	Sobreposição das imagens 1 e 2.	15
2.12	Imagem original.	16
2.13	Imagem 1.	16
2.14	Imagem 2.	16
2.15	Imagem 3.	16
2.16	Imagem 4.	16
2.17	Sobreposição das imagens 1 e 2.	16
2.18	Sobreposição das imagens 1 e 3.	16
2.19	Diagrama do Algoritmo de Inserção da Marca d'Água do Método SB- RAWVec.	18
2.20	Diagrama do Algoritmo de Detecção da Marca d'Água do Método SB- RAWVec.	21
3.1	O deslocamento máximo de um ponto para os dois métodos apresentados.	25
3.2	Mapa Vetorial Original 1 - 12540 pontos.	27
3.3	Mapa Vetorial 1 marcado pelo Método RAWVec, com $C = 0.027$	27
3.4	Mapa Vetorial 1 marcado pelo Método SB-RAWVec, com $C = 0.027$ e $I = 175$	28
3.5	Mapa Vetorial Original 2 - 5149 pontos.	29

3.6	Mapa Vetorial 2 marcado pelo Método RAWVec, com $C = 0.05$.	29
3.7	Mapa Vetorial 2 marcado pelo Método SB-RAWVec, com $C = 0.05$ e $I = 200$.	29
3.8	Mapa Vetorial Original 3 - 2146 pontos.	30
3.9	Mapa Vetorial 3 marcado pelo Método RAWVec, com $C = 0.042$.	30
3.10	Mapa Vetorial 3 marcado pelo Método SB-RAWVec, com $C = 0.042$ e $I = 100$.	30
3.11	Marca d'água usada na inserção.	31
3.12	Marca d'água usada na inserção.	32
3.13	Marca d'água usada na inserção.	36
3.14	Mapa Vetorial marcado pelo Método RAWVec.	38
3.15	Mapa Vetorial marcado pelo Método RAWVec e atacado com remoção de 75% dos pontos.	38
3.16	Mapa Vetorial marcado pelo Método SB-RAWVec.	39
3.17	Mapa Vetorial marcado pelo Método SB-RAWVec e atacado com remoção de 75% dos pontos.	39
3.18	Marca d'água usada na inserção.	40
3.19	Marca d'água usada na inserção.	41
3.20	Marca d'água usada na inserção.	43
3.21	Marca d'água usada na inserção.	43

Capítulo 1

Introdução

Imagens geralmente são armazenadas no computador como uma matriz, em que cada elemento representa a cor ou intensidade de um pixel. Essas imagens são conhecidas como *bitmap* ou *raster*. *Mapas vetoriais* são imagens representadas por estruturas geométricas, como pontos e círculos, portanto são imagens facilmente escaláveis, que possuem um tamanho menor e são facilmente atualizadas.

Uma *marca d'água* é uma informação adicionada de forma imperceptível num dado digital, geralmente com o objetivo de combater cópia e distribuição ilegais de arquivos multimídia como imagens, vídeos e áudio. Nos últimos anos, no entanto, marcas d'água em mapas vetoriais têm recebido maior atenção, especialmente devido à popularidade dos Sistemas de Informação Geográfica (GIS) e da utilização de dados espaciais na Web.

O objetivo desse trabalho é rever o Método de Marcas d'Água Raster em Mapas Vetoriais (RAWVec) [9], propor uma alternativa pública para tal método e analisar a robustez e eficiência dos dois métodos. A principal vantagem do método RAWVec é a utilização de uma imagem raster ou bitmap como marca d'água ao invés de uma sequência binária. Assim, a verificação da marca d'água é mais completa, e pode ser feita tanto por algoritmos probabilísticos como pelo olho humano.

Um *Método de Marca d'Água* é composto por dois algoritmos, o *algoritmo de inserção* e o *algoritmo de detecção*. Durante o algoritmo de inserção, uma marca d'água deve ser inserida no *dado original*, resultando em um novo *dado marcado* protegido contra cópia. Para testar se um determinado *dado de prova* é uma cópia não autorizada do dado marcado, o algoritmo de detecção deve ser utilizado para extrair uma marca d'água do dado prova, que deve, finalmente, ser comparada com a marca d'água usada originalmente. O algoritmo de detecção geralmente precisa de alguma informação extra, chamada de *chave*.

Marcas d'água podem ser classificadas como *públicas (blind)* ou *privadas (non-blind)*. Métodos de marcas d'água privados utilizam o dado original, que foi marcado durante

o algoritmo de detecção, além da chave e do dado de prova. Já os métodos públicos não utilizam o dado original durante o algoritmo de detecção; assim a detecção pode ser realizada por qualquer um sem que nenhuma informação adicional, além da chave, seja necessária. A maioria dos métodos de marca d'água são públicos, pois eles possuem maior aplicabilidade, enquanto os métodos privados geralmente são mais eficientes.

Além disso, marcas d'água também podem ser classificadas como *semi-públicas* (*semi-blind*). Neste caso, uma versão marcada ou alterada do mapa vetorial original é utilizada durante a detecção. Note que o mapa original é usado, mas não é revelado durante a detecção. Assim, um método semi-público possui a mesma aplicabilidade de um método público.

O método SB-RAWVec é uma alternativa semi-pública para o método RAWVec e utiliza *Criptografia Visual* [11] para dividir a imagem raster utilizada como marca d'água em duas. O objetivo da Criptografia Visual é dividir uma imagem em duas ou mais partes de tal maneira que a imagem só possa ser reconhecida quando as duas partes forem sobrepostas e não é possível conseguir nenhuma informação a respeito da imagem a partir de um subconjunto próprio das partes. Neste trabalho consideramos somente o caso em que imagens são divididas em duas partes. Com a marca d'água dividida em duas, dois mapas marcados são gerados, um para publicação e um que irá substituir o mapa original no algoritmo de detecção.

Tanto o método RAWVec, quanto o método SB-RAWVec, inserem marcas d'água em mapas vetoriais no domínio espacial, ou seja, o algoritmo de inserção modifica as coordenadas das estruturas presentes no mapa vetorial. Outras maneiras de inserir uma marca d'água em um mapa vetorial incluem transformadas (Transformada Wavelet Discreta ou DWT, Transformada Discreta de Fourier ou DFT e Transformada Discreta de Cosseno ou DCT) e algoritmos derivados de modelos 3D.

Os resultados apresentados nesse trabalho encontram-se publicados em conferências. O Método RAWVec é descrito e analisado em [10]. Este artigo obteve o prêmio de melhor artigo do congresso. O Método SB-RAWVec e sua análise encontram-se em [8].

1.1 Trabalhos Correlatos

Vários métodos de marca d'água em mapas vetoriais podem ser encontrados na literatura; em sua maioria, os métodos são públicos. Abaixo, apresentamos uma breve descrição e avaliação de alguns métodos.

Um método privado, baseado no domínio espacial, pode ser encontrado em [13]. Nele, o mapa vetorial é dividido em blocos e em cada bloco um bit da marca d'água é inserido de forma aditiva. Este método é robusto contra ataques de transformação, remoção de vértices, adição de ruído aleatório, mas é vulnerável a ataques de cropping.

Em [19] um método público, baseado no domínio espacial, é apresentado; nele, um dado multibit é inserido como marca d'água. Tal método só é robusto contra adição de ruído aleatório, sendo vulnerável a ataques de transformação e a adição ou remoção de dados. Apesar disso, o algoritmo possui vantagem de controlar facilmente a distorção inserida no mapa marcado.

Outro método público, baseado no domínio espacial, pode ser encontrado em [5]. O mapa vetorial é dividido em blocos, e para cada bloco é usada uma máscara que possui tamanho arbitrário. Cada máscara é dividida em dois triângulos que são deslocados de acordo com o bit da marca d'água. Apesar de ser robusto contra adição de ruído aleatório e reordenação de objetos, o método é vulnerável contra ataques de transformação e o erro adicionado ao mapa vetorial não pode ser controlado.

Além de métodos baseados no domínio espacial, existem também os métodos baseados em transformadas. Em [20] é apresentado um método público baseado na Transformada Discreta de Cosseno (DCT), em que cada oito vértices são combinados em uma unidade, onde um bit da marca d'água é inserido. Para combinar os vértices em unidades, o método usa a alta correlação entre as coordenadas de um vértice, uma das características mais importantes de mapas vetoriais. Este método é reversível, ou seja, é possível recuperar o mapa vetorial original durante o algoritmo de detecção e extração da marca d'água.

Um método público, baseado na Transformada Wavelet Discreta (DWT), robusto contra ataques de transformação e de adição de ruído aleatório pode ser encontrada em [6], enquanto um exemplo de outro método público, mas baseado na Transformada Discreta de Fourier (DFT), é descrito em [16]. Tal método é robusto contra transformações, reordenação de dados e adição de ruído aleatório. Este tipo de método depende de um grande número de vértices para ser confiável, e sua carga computacional cresce muito rapidamente com o número de vértices.

Em [12] encontramos a descrição de um método privado, baseado em domínio de *mesh-spectrum*. O método original aborda somente imagens em 3D, mas em [14] o método é adaptado e expandido para imagens 2D. Tal método é robusto contra ataques de transformação, adição de ruído aleatório, cropping e adição de vértices.

Baseado em interpolação, o método encontrado em [17] representa a marca d'água por bits e adiciona vértices ao mapa original durante o algoritmo de inserção. Este esquema utiliza propriedades e características especiais de certos tipo de mapas vetoriais, não sendo tão genérico quanto a maioria dos métodos.

O método público descrito em [15] calcula uma representação do mapa vetorial, criando uma matriz adjacente. A partir dessa matriz os pontos são classificados e selecionados em grupos, para então, serem marcados. Este método é robusto contra cropping e adição de ruído aleatório.

As duas alternativas do método RAWVec - original e semi-pública - mostram-se ro-

bustas contra diversos ataques, tais como ataques de transformação, ataques de cropping, ataques de inserção e alteração na ordem de objetos, ataques de inserção de ruído aleatório e, principalmente, ataques combinados. Ambas são baseadas em domínio espacial e permitem parametrização da distorção a ser inserida no mapa vetorial. Além disso, ambas usam imagens raster como marca d'água, permitindo que a verificação seja realizada tanto por um método computacional quanto por um observador humano.

1.2 Organização do Texto

As descrições do Método RAWVec e sua alternativa semi-pública são apresentadas no Capítulo 2, assim como algumas funções e algoritmos utilizados pelos dois métodos. No Capítulo 3, testes realizados com cada método são apresentados, além da análise de ataques comuns na área. Finalmente no Capítulo 4 são apresentadas as conclusões e os trabalhos futuros.

Capítulo 2

O Método RAWVec

Duas variantes do método RAWVec, para marcas d'água raster em mapas vetoriais, são descritos nesse capítulo: a versão original [10] e a semi-pública [8]. Além disso, tanto o Método RAWVec quanto o Método SB-RAWVec utilizam duas funções $w()$ e $v()$ e um algoritmo de *Point Pattern Matching*. Esses conceitos também são descritos nas seções a seguir.

2.1 Funções Auxiliares

A função $w()$ modifica a matriz quadrada $A_{n \times n}$ da seguinte forma: seja A uma matriz quadrada de ordem n ; então $B = w(A)$ tem $b_{ij} = a_{uv}$, onde $u = n - i + 1$ e $v = n - j + 1$.

Definição 2.1.1. *Seja $A = (a_{ij})$ uma matriz quadrada de ordem n . Definimos a função $w()$, tal que $w(A) = (w_{ij}) = (a_{uv})$ é uma matriz quadrada de ordem n , onde $u = n - i + 1$ e $v = n - j + 1$.*

A Figura 2.1 mostra um exemplo de aplicação da função $w()$.

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \Rightarrow w(A) = \begin{pmatrix} 16 & 15 & 14 & 13 \\ 12 & 11 & 10 & 9 \\ 8 & 7 & 6 & 5 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

Figura 2.1: Exemplo de aplicação da função $w(A)$.

É importante observar algumas propriedades importantes da função $w()$:

- (a) Propriedade 1: $w(w(A)) = A$

(b) Propriedade 2: $c * w(A) = w(c * A)$

(c) Propriedade 3: $w(A) + w(B) = w(A + B)$

A marca d'água usada nos dois métodos é uma imagem bitmap, ou imagem raster, que é representada por uma matriz bi-dimensional, cujos elementos armazenam as informações de cor e intensidade de cada pixel. O mapa vetorial também deve ser representado como uma matriz através da função $v()$, descrita a seguir.

A função $v()$ calcula a representação por pontos P de um mapa vetorial M . Mapas vetoriais são formados por estruturas geométricas e, baseado nos pontos dessas estruturas geométricas, é possível criar uma representação por pontos $P = v(M)$. Para tal, cada estrutura deve ser decomposta em pontos segundo o seu tipo:

- Objetos pontuais: Estruturas descritas por pontos, como símbolos e textos. Neste caso as coordenadas de cada ponto devem ser armazenadas.
- Objetos lineares: Estruturas descritas por uma seqüência de pontos, como linhas e polígonos. As coordenadas de cada ponto devem ser armazenadas em seqüência.
- Objetos parametrizados: Estruturas descritas por parâmetros, como círculos e elipses. Neste caso o objeto pode ser segmentado em alguns pontos que são armazenados; ou os pontos principais são armazenados. Se o objeto for um círculo, por exemplo, podem ser armazenados alguns pontos do círculo, ou somente o seu centro.

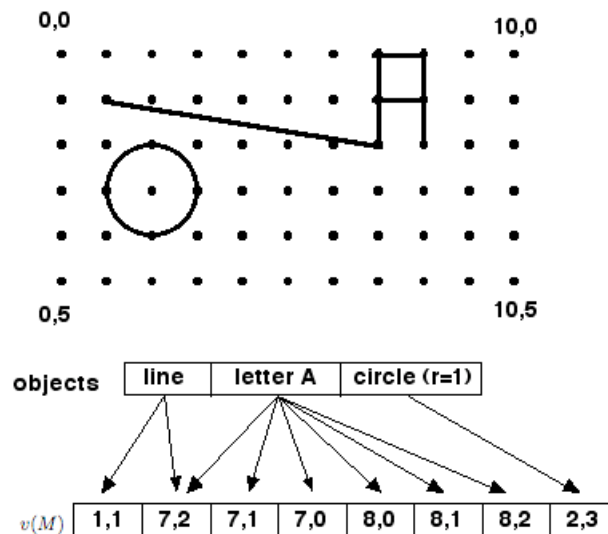


Figura 2.2: Exemplo de aplicação da função $v()$.

Cada coordenada deve possuir uma referência ao seu objeto original, pois somente assim é possível reconstruir cada estrutura do mapa vetorial a partir da representação por pontos. Um exemplo de aplicação da função $v()$ é mostrado na Figura 2.2.

2.2 O Algoritmo de Point Pattern Matching

O i -ésimo ponto em $P = v(M)$ corresponde ao i -ésimo ponto em $Q = v(M')$. Assim, se o mapa vetorial N é realmente o mapa vetorial M' , então o i -ésimo ponto em $v(M)$ deve corresponder ao i -ésimo ponto em $P = v(N)$. Entretanto, o mapa vetorial N pode ser obtido a partir de uma transformação T aplicada em M' e a relação entre os pontos pode não ser tão simples. Um algoritmo de *Point Pattern Matching* acha essa relação entre os pontos e retorna uma transformação T que pode ter sido utilizada em M' para produzir N . Neste trabalho, utilizamos o algoritmo de *Point Pattern Matching* encontrado em [18], que é descrito pelo Algoritmo 1 abaixo.

Algorithm 1 Point Pattern Matching

INPUT: representações por pontos $P = v(N)$ e $Q = v(M')$.

OUTPUT: transformação T e duas representações por pontos L_p e L_q .

1. Obter os k vizinhos mais próximos (KNN) para cada ponto em P e em Q usando a *Triangulação de Delaunay*, como descrito em [3]
 2. $global \leftarrow falso$;
 3. **enquanto** $global = falso$
 - 3.1. **para** cada ponto $p \in P$
 - 3.1.1 Obter o ponto mais próximo $q \in Q$
 - 3.1.2 Achar (a_f, b_f) , os vizinhos mais distantes do KNN, onde $a_f \in KNN(p)$ e $b_f \in KNN(q)$
 - 3.1.3 Calcular a transformação local T que, usada em (p, a_f) , produz (q, b_f)
 - 3.1.4 **se** $T(p, a) = (q, b)$ para pelo menos ρk pares (a, b) **então**
 - 3.1.4.1 **se** T é uma transformação global **então**
 - 3.1.4.1.1 $global \leftarrow verdadeiro$;
 - 3.1.4.1.2 retornar (T) ;
-

O algoritmo de *Point Pattern Matching* acha uma transformação que produz (q, b) a partir de (p, a) , onde $p, a \in P$ e $q, b \in Q$. Tal transformação é testada em um conjunto restrito de pontos; caso ela seja adequada para esse conjunto de pontos, ela é finalmente verificada para todos os pontos. O conjunto menor de pontos consiste no conjunto dos k vizinhos mais próximos (KNN) de cada ponto em P e Q . Portanto, $a \in KNN(p)$ e $b \in KNN(q)$ são os vizinhos mais distantes de p e q , respectivamente. O cálculo

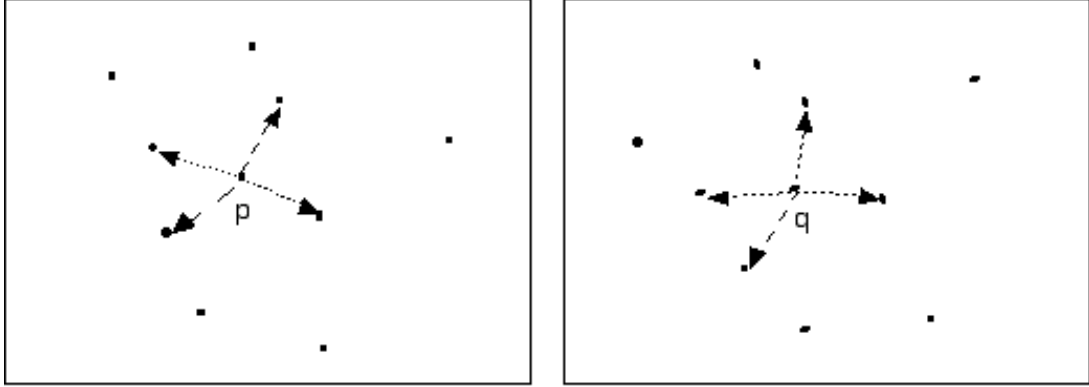


Figura 2.3: Um exemplo de *Point Pattern Matching*.

dos k vizinhos mais próximos de um ponto pode ser realizado usando a *Triangulação de Delaunay*, como descrito em [3].

O próximo passo consiste em achar uma transformação local T que, ao ser aplicada em a , produz b , e que ao ser aplicada em p , produz q ; ou seja, $T(a) = b$ e $T(p) = q$, onde:

$$T = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + s \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \quad (2.1)$$

onde

$$s = \frac{|\vec{qb}|}{|\vec{pa}|}, \quad (2.2)$$

$$\theta = \text{ângulo entre } \vec{pa} \text{ e } \vec{qb}, \quad (2.3)$$

e

$$t_x = q_x - p_x s \cos \theta + p_y s \sin \theta \quad \text{e} \quad t_y = q_y - p_x s \sin \theta - p_y s \cos \theta. \quad (2.4)$$

Devemos, então, verificar se tal transformação produz um subconjunto de $KNN(q)$, quando usada em um subconjunto de $KNN(p)$, com pelo menos ρk pontos. Esses dois subconjuntos são duas representações por pontos, chamadas L_a e L_b , respectivamente.

Agora, é necessário verificar se a transformação local T é uma transformação global, calculando a transformação T' e aplicando-a em cada ponto $q \in Q$. Se $T'(q)$ produz um ponto $p \in P$, então p e q devem ser incluídos nas duas listas de pontos L_p e L_q , respectivamente.

$$T' = \begin{pmatrix} t_x \\ t_y \\ s \cos \theta \\ s \sin \theta \end{pmatrix} = \frac{1}{\mu_d} \begin{pmatrix} \mu_a & 0 & -\mu_{ax} & \mu_{ay} \\ 0 & \mu_a & -\mu_{ay} & \mu_{ax} \\ -\mu_{ax} & -\mu_{ay} & l & 0 \\ \mu_{ay} & -\mu_{ax} & 0 & l \end{pmatrix} \begin{pmatrix} \mu_{bx} \\ \mu_{by} \\ \mu_{a+b} \\ \mu_{a-b} \end{pmatrix}, \quad (2.5)$$

onde

$$\mu_{ax} = \sum_{i=1}^l L_{ax_i} \quad \text{e} \quad \mu_{ay} = \sum_{i=1}^l L_{ay_i}, \quad (2.6)$$

$$\mu_{bx} = \sum_{i=1}^l L_{bx_i} \quad \text{e} \quad \mu_{by} = \sum_{i=1}^l L_{by_i}, \quad (2.7)$$

$$\mu_{a+b} = \sum_{i=1}^l L_{ax_i} L_{bx_i} + L_{ay_i} L_{by_i}, \quad (2.8)$$

$$\mu_{a-b} = \sum_{i=1}^l L_{ax_i} L_{by_i} - L_{ay_i} L_{bx_i}, \quad (2.9)$$

$$\mu_a = \sum_{i=1}^l L_{ax_i}^2 L_{ay_i}^2, \quad (2.10)$$

e

$$\mu_d = l\mu_a - \mu_{ax}^2 - \mu_{ay}^2. \quad (2.11)$$

Se L_p e L_q possuem pelo menos ρt pontos (com t sendo o tamanho de P e Q), a transformação T é global; senão, outro par (p, q) e outra transformação local T devem ser encontrados. Se todos os pontos em P e Q forem usados e nenhuma transformação global foi encontrada, então N não foi produzido a partir de M' e, portanto, não representa um mapa marcado.

2.3 O Método RAWVec

Esta seção descreve o Método RAWVec [10]. O algoritmo de inserção marca o mapa vetorial com a imagem raster deslocando as coordenadas de suas estruturas geométricas. O algoritmo de detecção usa o mapa vetorial original para extrair a imagem raster e compará-la com a marca d'água original.

2.3.1 Algoritmo de Inserção

O Algoritmo 2 abaixo insere uma imagem raster R no mapa vetorial M produzindo o mapa vetorial marcado M' . Para isso uma constante real positiva C é usada; seu principal objetivo é controlar o deslocamento máximo de um pixel, diminuindo, assim, a perturbação inserida no mapa vetorial gerado. Tal fato será discutido em mais detalhes na Seção 3.1.2. A Figura 2.4 representa um diagrama mostrando os cinco passos desse algoritmo.

Algorithm 2 Algoritmo de Inserção da Marca d'Água do Método RAWVec

INPUT: a imagem raster R , o mapa vetorial M e a constante C .

OUTPUT: o mapa vetorial marcado M' .

1. Calcular a representação por pontos $P \leftarrow v(M)$ do mapa vetorial M .
 2. Construir as matrizes A_x e A_y a partir da representação por pontos P .
 3. Redimensionar a imagem raster R , criando a imagem raster E .
 4. Calcular $B_x \leftarrow CE + A_x$; $B_y \leftarrow Cw(E) + A_y$, onde C é uma constante real.
 5. Construir o mapa vetorial marcado M' a partir das matrizes B_x e B_y .
-

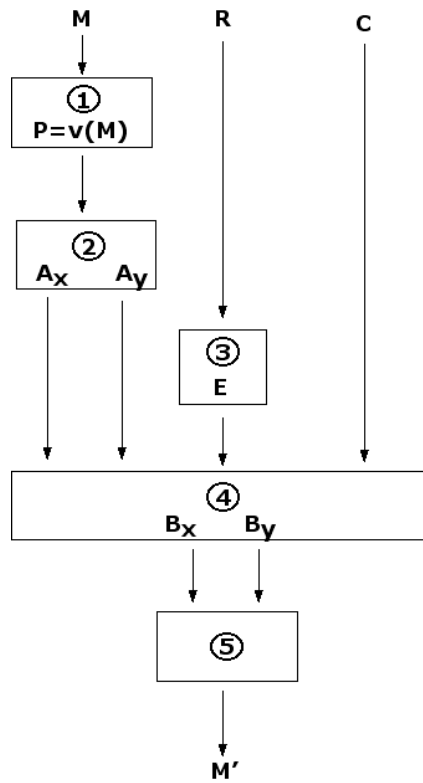


Figura 2.4: Diagrama do Algoritmo de Inserção da Marca d'Água do Método RAWVec.

Os dois primeiros passos do algoritmo criam duas matrizes – A_x para os valores da coordenada x e A_y para os valores da coordenada y – a partir da representação por pontos $P = v(M)$, criada como descrito na Seção 2.1. Seja t o número de pontos armazenados em P , n a ordem das matrizes e $p_i = (x_i, y_i)$ os pontos armazenados em P ; então

$$n = \lfloor \sqrt{t} \rfloor, \quad (2.12)$$

$$(A_x)_{ij} = x_{n(i-1)+j} \quad \text{e} \quad (A_y)_{ij} = y_{n(i-1)+j} \quad (2.13)$$

O terceiro passo redimensiona a imagem raster R , criando uma nova imagem raster E que possui $n \times n$ pixels, como ilustra a Figura 2.5 abaixo.

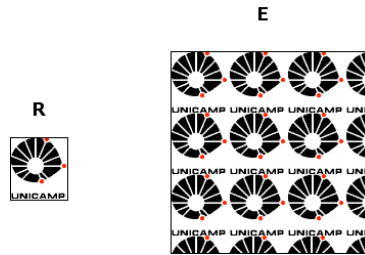


Figura 2.5: Criação da imagem raster E , a partir da imagem raster R redimensionada.

Recomenda-se que a imagem R tenha dimensões inferiores a $n \times n$, para que a marca d'água possa ser replicada. Dessa forma, cria-se redundância sobre cada pixel da marca, o que é fundamental para a restauração dos trechos mais prejudicados por ataques (um pixel pode ter sido alterado em uma das cópias, mas ainda assim manter seu valor original em outras).

Usando as matrizes A_x e A_y , que representam o mapa vetorial M , e a matriz E , que representa a imagem raster, é possível inserir E no mapa vetorial M . Assim, no quarto passo, as duas matrizes B_x e B_y são calculadas da seguinte maneira:

$$B_x = CE + A_x, \quad \text{e} \quad B_y = Cw(E) + A_y. \quad (2.14)$$

Finalmente, o mapa vetorial marcado M' é construído a partir das matrizes B_x e B_y e da estrutura criada pela função $v()$, descrita na Seção 2.1.

2.3.2 Algoritmo de Detecção

O algoritmo de detecção consiste na extração e comparação da marca d'água extraída S com a marca d'água original R . A marca d'água S é extraída do mapa vetorial de prova N , ou seja, o mapa vetorial que será testado, usando a constante C , a marca d'água original R e o mapa vetorial original M . Portanto, esse método não é um método público. Os passos básicos desse algoritmo estão descritos no Algoritmo 3 abaixo e no diagrama da Figura 2.6.

Algorithm 3 Algoritmo de Detecção da Marca d'Água do Método RAWVec Original

INPUT: a imagem raster R , o mapa vetorial M , a constante C e o mapa vetorial a ser testado N .

OUTPUT: a marca d'água S .

1. Inserir a marca d'água R no mapa vetorial M usando o algoritmo descrito na seção 2.3.1, retornando o mapa vetorial marcado M' .
 2. Calcular a representação por pontos $P \leftarrow v(M')$ e a representação por pontos $Q \leftarrow v(N)$.
 3. Comparar as representações por pontos P e Q usando um algoritmo de *Point Pattern Matching*, retornando a transformação T , onde $Q = T(P)$.
 4. Construir as matrizes A_x e A_y a partir de $v(M)$, e B_x e B_y a partir de $T^{-1}(Q)$.
 5. Calcular $D_x \leftarrow \frac{B_x - A_x}{C}$; $D_y \leftarrow \frac{w(B_y) - w(A_y)}{C}$.
 6. Calcular $D \leftarrow \frac{D_x + D_y}{2}$.
 7. Redimensionar a marca d'água D para o tamanho da marca d'água original R , retornando a marca d'água S .
-

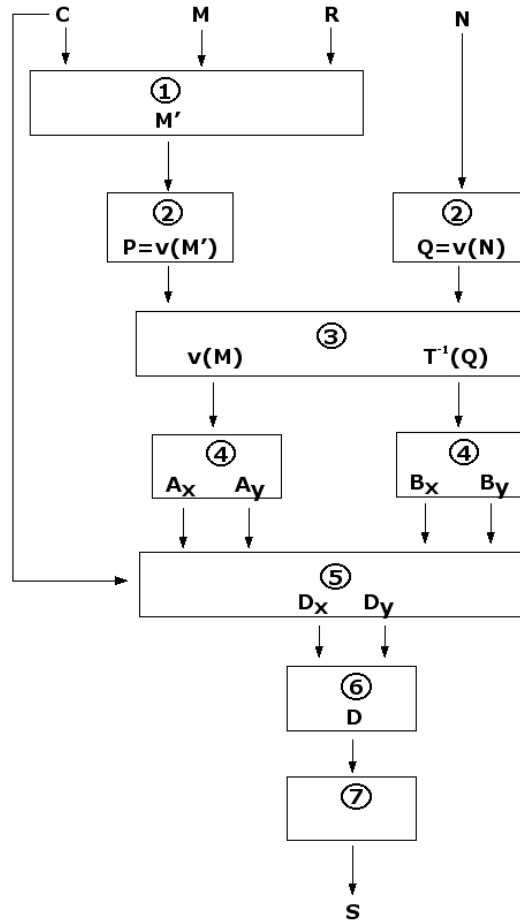


Figura 2.6: Diagrama do Algoritmo de Detecção da Marca d'Água do Método RAWVec Original.

O primeiro passo do algoritmo de detecção é usar o Algoritmo 2 para inserir a marca d'água R no mapa vetorial M , resultando no mapa vetorial marcado M' . Esse mapa vetorial marcado deve ser comparado com o mapa vetorial de prova N , usando o Algoritmo de *Point Pattern Matching* [18], descrito na Seção 2.2. Assim, qualquer transformação T usada em um ataque será encontrada e removida do mapa vetorial N .

Desta forma, podemos, agora, construir quatro matrizes: A_x e A_y a partir de $v(M)$ e B_x e B_y a partir de $T^{-1}(N)$ ¹. As matrizes A_x e A_y representam o mapa vetorial original e as matrizes B_x e B_y representam o mapa vetorial de prova que pode corresponder ao mapa vetorial M' . Assim, a marca d'água presente no mapa vetorial N deve ser calculada e comparada com a imagem usada originalmente. A marca d'água D é calculada usando as quatro matrizes A_x , A_y , B_x e B_y , a constante C e a função w da seguinte maneira:

$$D = \frac{D_x + D_y}{2}, \quad (2.15)$$

onde

$$D_x = \frac{B_x - A_x}{C} \quad \text{e} \quad D_y = \frac{w(B_y) - w(A_y)}{C}. \quad (2.16)$$

Por fim, a marca d'água D , de tamanho $n \times n$, deve ser redimensionada para o tamanho da marca d'água original R , produzindo a marca d'água S . As marcas d'água S e R podem ser comparadas usando um algoritmo probabilístico e/ou a avaliação do ponto de vista do olho humano. De acordo com o resultado dessa comparação podemos concluir se o mapa vetorial de prova N corresponde ao mapa vetorial marcado M' atacado. Detalhes desta comparação podem ser encontrados na Seção 3.1.3.

2.4 O Método SB-RAWVec

Esta seção apresenta as modificações feitas no Método RAWVec, resultando num método semi-público alternativo. Neste novo método, dois mapas vetoriais marcados são criados, um deve ser publicado como mapa vetorial marcado, chamado de *mapa vetorial marcado para a publicação*, o outro deve ser usado somente durante o algoritmo de detecção, chamado de *mapa vetorial marcado para a detecção*. O algoritmo de detecção não utiliza o mapa vetorial original e uma nova constante I é acrescentada para ajudar no controle do deslocamento máximo de um pixel.

2.4.1 Criptografia Visual

Criptografia Visual [11] é uma técnica de compartilhamento de imagens que consiste em dividir uma imagem em diversas partes, de tal forma que só seja possível visualizar a

¹ $T^{-1}(N)$ significa o conjunto de pontos de P que são pré-imagens de pontos de Q por T .

imagem original sobrepondo todas as partes. Em particular, dada uma imagem em preto e branco, é possível dividi-la em duas imagens que, ao serem sobrepostas, revelam a imagem original, mas que sozinhas não ofereçam qualquer tipo de informação sobre o todo. Para isso, cada pixel da imagem original é dividido em dois subpixels de acordo com a Figura 2.7.















imagem original	imagem 1	imagem 2	imagens 1 e 2 sobrepostas
			
			
			
			

Figura 2.7: Composição dos pixels na Criptografia Visual.

Dessa forma, a sobreposição é realizada utilizando-se a operação lógica OR, considerando que 1 representa a cor preta e 0 representa a cor branca. Os pixels pretos da imagem original estarão intactos, já os pixels brancos serão representados por dois subpixels, um branco e um preto, na imagem final.

A técnica de Criptografia Visual pode ser expandida para imagens em níveis de cinza como em [2] e [7], ou para imagens coloridas como em [23] e [4].

Dois exemplos de Criptografia Visual, extraídos de [1], são apresentados abaixo. A Figura 2.8 é a imagem original que foi dividida em duas partes, representadas pelas Figuras 2.9 e 2.10. A Figura 2.11 mostra o resultado da sobreposição das duas partes.



Figura 2.8: Imagem original.

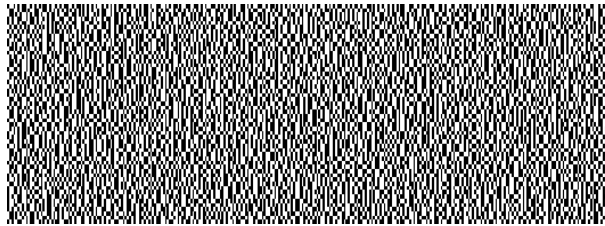


Figura 2.9: Imagem 1.

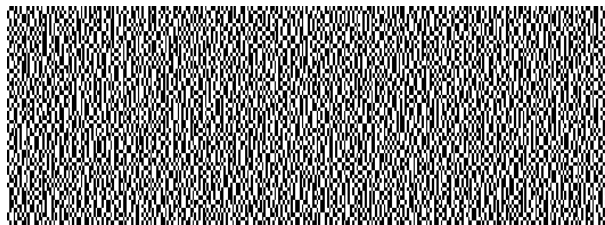


Figura 2.10: Imagem 2.



Figura 2.11: Sobreposição das imagens 1 e 2.

Também é possível dividir a imagem original em mais de duas partes, de tal maneira que ela seja reconstruída somente sobrepondo duas partes seqüenciais, ou seja, as imagens 1 e 2, ou as imagens 2 e 3, por exemplo. Esta técnica de Criptografia Visual é exemplificada nas figuras a seguir.



Figura 2.12: Imagem original.

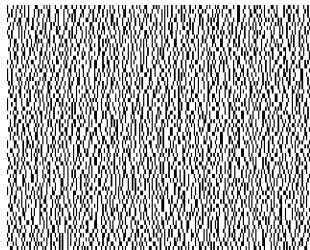


Figura 2.13: Imagem 1.

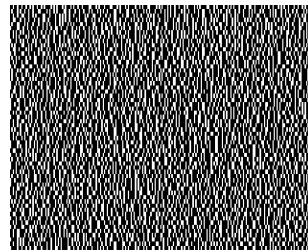


Figura 2.14: Imagem 2.

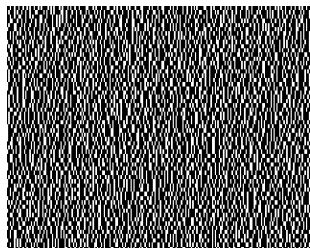


Figura 2.15: Imagem 3.

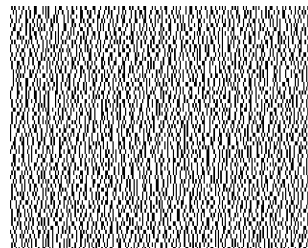


Figura 2.16: Imagem 4.

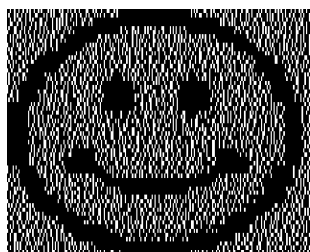


Figura 2.17: Sobreposição das imagens 1 e 2.

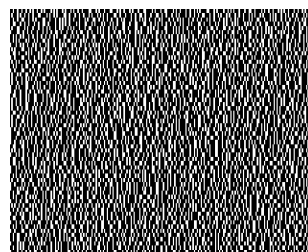


Figura 2.18: Sobreposição das imagens 1 e 3.

A Figura 2.12 mostra a imagem original, que foi dividida em quatro partes, representadas pelas Figuras 2.13, 2.14, 2.15 e 2.16. A imagem original só pode ser obtida a partir da sobreposição das imagens 1 e 2, 2 e 3 ou 3 e 4. A Figura 2.17 mostra que a sobreposição das partes 1 e 2 realmente reconstrói a imagem original, enquanto que a Figura 2.18 mostra que nenhuma informação é obtida a partir da sobreposição das partes 1 e 3.

O Método SB-RAWVec possui um passo baseado em Criptografia Visual, em que a marca d'água é dividida em duas partes e a partir delas, dois mapas vetoriais marcados são produzidos. Diferentemente da técnica de Criptografia Visual, que gera duas imagens com valores positivos, o passo do método semi-público gera duas partes, mas apenas uma delas possui valores positivos; a outra não possuirá qualquer delimitação de valores, portanto as intensidades dos pixels poderão ser negativas, por exemplo, fazendo com que ela não possa ser visualizada como a imagem original.

2.4.2 Algoritmo de Inserção

O Algoritmo 4 mostra os passos do algoritmo de inserção da marca d'água do novo método. A maior diferença deste algoritmo com o Algoritmo 2 do método original é o passo 4, no qual a imagem raster é dividida em duas matrizes, resultando em dois mapas vetoriais marcados. A Figura 2.19 mostra um diagrama desse algoritmo.

Algorithm 4 Algoritmo de Inserção da Marca d'Água do Método SB-RAWVec

INPUT: a imagem raster R , o mapa vetorial M e as duas constantes C e I .

OUTPUT: o mapa vetorial marcado para publicação M_1 e o mapa vetorial marcado para detecção M_2 .

1. Calcular a representação por pontos $P \leftarrow v(M)$ do mapa vetorial M .
 2. Construir as matrizes A_x e A_y a partir da representação por pontos P .
 3. Redimensionar a imagem raster R , criando a imagem raster E .
 4. Decompor a matriz E em duas matrizes aleatórias E_1 e E_2 onde $E = E_1 + E_2$, e $0 \leq z \leq I$ para cada z pertencente a E_1 .
 5. Calcular $B_{1x} \leftarrow A_x + CE_1$; $B_{1y} \leftarrow A_y + Cw(E_1)$, onde C é uma constante real.
 6. Calcular $B_{2x} \leftarrow A_x - CE_2$; $B_{2y} \leftarrow A_y - Cw(E_2)$, onde C é uma constante real.
 7. Construir o mapa vetorial marcado para publicação M_1 usando as matrizes B_{1x} e B_{1y} .
 8. Construir o mapa vetorial marcado para detecção M_2 usando as matrizes B_{2x} e B_{2y} .
-

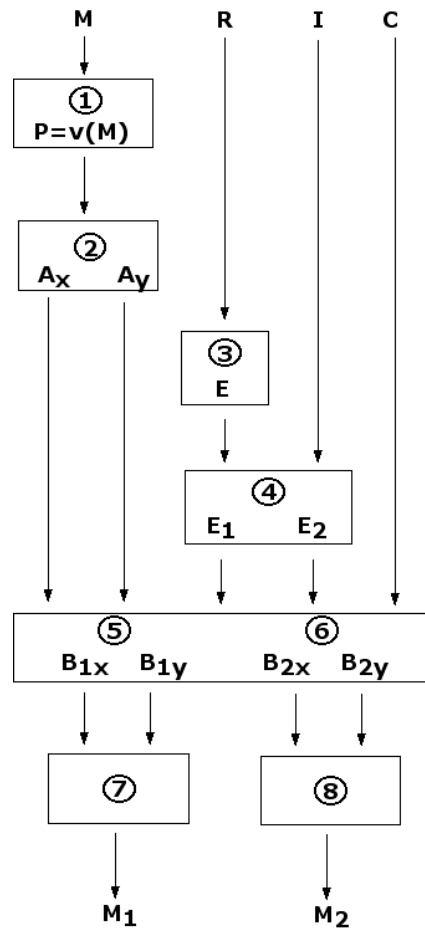


Figura 2.19: Diagrama do Algoritmo de Inserção da Marca d'Água do Método SB-RAWVec.

Os três primeiros passos são idênticos ao Algoritmo 2 do Método RAWVec Original descrito na seção 2.3.1. A representação por pontos P é criada a partir do mapa vetorial M , assim como as duas matrizes A_x e A_y , que representam as coordenadas x e y , respectivamente. Essas duas matrizes devem ser quadradas e devem ter o mesmo tamanho. No próximo passo, a imagem raster R é redimensionada gerando uma matriz quadrada E com o mesmo tamanho de A_x e A_y .

As matrizes A_x e A_y representam as posições dos objetos no mapa vetorial, e a matriz E representa a cor de cada pixel na imagem raster.

O passo 4 representa a criação de duas matrizes E_1 e E_2 , usando a constante inteira positiva I e a matriz E . Este passo é baseado em Criptografia Visual [11], em que uma imagem é dividida em duas, podendo ser recriada somente pela superposição das duas partes. No caso do nosso algoritmo, a imagem pode ser reconstruída somente se as duas partes forem adicionadas. Note que uma das imagens, a imagem E_2 , não pode ser

visualizada como a imagem original já que ela pode conter elementos negativos.

Com as matrizes A_x , A_y e E_1 é possível obter as duas matrizes B_{1x} e B_{1y} e, portanto, construir o mapa vetorial marcado para publicação M_1 :

$$B_{1x} = A_x + CE_1, \quad \text{e} \quad B_{1y} = A_y + Cw(E_1). \quad (2.17)$$

E, com as matrizes A_x , A_y e E_2 é possível obter outras duas matrizes B_{2x} e B_{2y} e, portanto, construir o mapa vetorial marcado para detecção M_2 :

$$B_{2x} = A_x - CE_2, \quad \text{e} \quad B_{2y} = A_y - Cw(E_2). \quad (2.18)$$

Note que a soma usada no cálculo do mapa vetorial marcado para publicação M_1 é substituída por uma subtração no cálculo do mapa vetorial marcado para detecção M_2 .

A perturbação inserida no mapa vetorial M_1 pode ser controlada pelas duas constantes C e I , forçando, assim, a perturbação inserida no mapa M_2 ; portanto, o mapa M_2 só pode ser usado pelo algoritmo de detecção.

2.4.3 Algoritmo de Detecção

Apesar de não usar o mapa vetorial original M , o algoritmo de detecção do Método SB-RAWVec usa um algoritmo de *Point Pattern Matching* para detectar e remover ataques. Isto é possível devido à existência das informações do mapa original escondidas no mapa vetorial marcado para a detecção M_2 . O Algoritmo 5 e o diagrama da Figura 2.20 mostram os passos principais desse algoritmo.

Os dois primeiros passos são simples; eles criam as representações por pontos e as respectivas matrizes para os dois mapa vetoriais: B_{2x} e B_{2y} para o mapa vetorial marcado para detecção M_2 e B_x e B_y para o mapa vetorial a ser testado N . A primeira grande diferença entre este algoritmo de detecção e o Algoritmo 3 está no passo 4, no qual a informação escondida no mapa vetorial de detecção é obtida.

Algorithm 5 Algoritmo de Detecção da Marca d'Água do Método SB-RAWVec

INPUT: a imagem raster R , o mapa vetorial marcado para detecção M_2 , a constante C e o mapa vetorial a ser testado N .

OUTPUT: a marca d'água S .

1. Calcular a representação por pontos $P \leftarrow v(M_2)$ do mapa vetorial marcado para detecção M_2 e a representação por pontos $Q \leftarrow v(N)$ do mapa vetorial a ser testado N .
 2. Construir as matrizes B_{2x} e B_{2y} a partir de P , e B_x e B_y a partir de Q .
 3. Redimensionar a imagem raster R , criando a matriz E de mesmo tamanho que a matriz B_{2x} .
 4. Calcular $B'_{2x} \leftarrow B_{2x} + CE$ e $B'_{2y} \leftarrow B_{2y} + Cw(E)$
 5. Comparar a representação por pontos obtida através das matrizes B'_{2x} e B'_{2y} com a representação por pontos obtida através das matrizes B_x e B_y usando um algoritmo de *Point Pattern Matching*.
 6. Remover a transformação retornada pelo algoritmo de *Point Pattern Matching* das matrizes B_x e B_y , criando as matrizes B'_x e B'_y .
 7. Calcular $D_x \leftarrow \frac{B'_x - B_{2x}}{C}$; $D_y \leftarrow \frac{w(B'_y) - w(B_{2y})}{C}$.
 8. Calcular $D \leftarrow \frac{D_x + D_y}{2}$.
 9. Redimensionar a marca d'água D para o tamanho da marca d'água original R , retornando a marca d'água S .
-

Do algoritmo de inserção descrito na Seção 2.4.2 temos que $B_{2x} = A_x - CE_2$, $B_{2y} = A_y - Cw(E_2)$ e $E = E_1 + E_2$; assim, é possível extrair a informação necessária para o Algoritmo de *Point Pattern Matching* da seguinte maneira:

$$\begin{aligned}
 B'_{2x} &= B_{2x} + CE & B'_{2y} &= B_{2x} + Cw(E) \\
 &= A_x - CE_2 + CE & &= A_y - Cw(E_2) + Cw(E) \\
 &= A_x + C(E - E_2) & &= A_y + Cw(E - E_2) \\
 &= A_x + CE_1 & &= A_y + Cw(E_1) \\
 &= B_{1x} & &= B_{1y}
 \end{aligned}$$

Se o mapa vetorial de prova N for uma versão alterada do mapa vetorial marcado para publicação M_1 , será possível compará-lo com M_1 usando um algoritmo de *Point Pattern Matching*. Tal algoritmo achará qualquer transformação e/ou modificação que mapeie N em M_1 , comparando as matrizes B'_{2x} e B'_{2y} com as matrizes B_x and B_y .

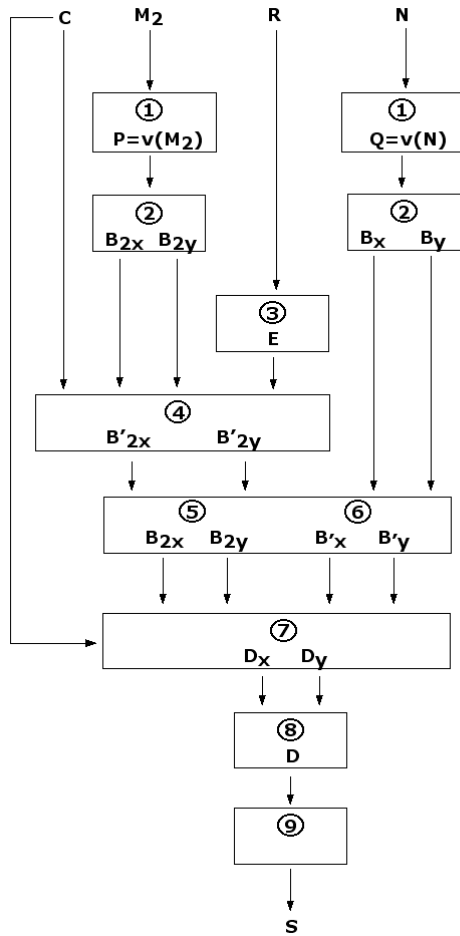


Figura 2.20: Diagrama do Algoritmo de Detecção da Marca d'Água do Método SB-RAWVec.

O passo 6 remove a transformação e modificações encontradas pelo algoritmo de *Point Pattern Matching* das matrizes B_x e B_y , resultando nas duas novas matrizes B'_x e B'_y . Agora é possível calcular a marca d'água extraída e verificar se o mapa de prova N foi roubado:

$$D = \frac{D_x + D_y}{2}, \quad (2.19)$$

onde

$$D_x = \frac{B'_x - B_{2x}}{C} \quad \text{e} \quad D_y = \frac{w(B'_y) - w(B_{2y})}{C}. \quad (2.20)$$

Se o mapa vetorial de prova N for um versão modificada do mapa vetorial marcado para publicação M_1 , as matrizes B'_x e B'_y serão equivalentes às matrizes B_{1x} e B_{1y} ; assim:

$$\begin{aligned}
D_x &= \frac{B'_x - B_{2x}}{C} \\
&= \frac{B_{1x} - B_{2x}}{C} \\
&= \frac{(A_x + CE_1) - (A_x - CE_2)}{C} \\
&= \frac{CE_1 + CE_2}{C} \\
&= E_1 + E_2 \\
&= E
\end{aligned}$$

O cálculo de D_y é mais complexo pois a função $w()$ é usada; assim as três propriedades descritas na Seção 2.1 são essenciais:

$$\begin{aligned}
D_y &= \frac{w(B'_y) - w(B_{2y})}{C} \\
&= \frac{w(B_{1y}) - w(B_{2y})}{C} \\
&= \frac{w(A_y + Cw(E_1)) - w(A_y - Cw(E_2))}{C} \\
&= \frac{(w(A_y) + w(Cw(E_1))) - (w(A_y) - w(Cw(E_2)))}{C} \\
&= \frac{Cw(w(E_1)) + Cw(w(E_2))}{C} \\
&= E_1 + E_2 \\
&= E
\end{aligned}$$

A matriz D deve ser redimensionada para o tamanho da marca d'água original, criando a marca d'água S . As duas imagens, R , representando a marca d'água original e S , representando a marca d'água extraída, podem ser comparadas utilizando a avaliação do ponto de vista do olho humano e/ou um algoritmo probabilístico como o coeficiente de correlação de Pearson [22]. Tal fato será analisado com mais detalhes na Seção 3.1.3.

Capítulo 3

Resultados Experimentais

Neste capítulo são apresentados as análises teóricas dos métodos apresentados no Capítulo 2, assim como descrições dos ataques estudados e da execução e robustez dos métodos com relação a esses ataques.

3.1 Análise dos Algoritmos

Nessa seção, os algoritmos apresentados são analisados quanto a sua complexidade e a qualidade do mapa vetorial marcado gerado. Além disso, são apresentadas as métricas para comparação de imagens usadas nas marcas d'água nos Algoritmos de Detecção 3 e 5.

3.1.1 Complexidade

A Tabela 3.1 apresenta a complexidade para cada passo dos Algoritmos de Inserção 2 e 4. Como podemos ver, todos os passos de ambos os algoritmos possuem complexidade no máximo $O(t)$, pois da equação 2.12 temos que $n^2 \leq t$, portanto $O(n^2) \leq O(t)$. Assim cada passo só acrescenta uma constante na complexidade geral do algoritmo.

Passo	Método RAWVec	Método SB-RAWVec
1	$O(t)$	$O(t)$
2	$O(t)$	$O(t)$
3	$O(n^2)$	$O(n^2)$
4	$O(n^2)$	$O(n^2)$
5	$O(n^2)$	$O(n^2)$
6	—	$O(n^2)$
7	—	$O(n^2)$
8	—	$O(n^2)$
Total	$O(t)$	$O(t)$

Tabela 3.1: Complexidade de cada passo dos algoritmos de inserção.

A complexidade de cada passo dos Algoritmos de Detecção 3 e 5 são mostradas na Tabela 3.2. Tais algoritmos são mais complexos, pois devem detectar e remover ataques usando um algoritmo de *Point Pattern Matching*. O Algoritmo 1 descreve o algoritmo de *Point Pattern Matching* usado nos métodos apresentados; tal algoritmo possui complexidade $O(t(\log t)^{3/2})$, como visto em [18].

Assim, todos os passos dos algoritmos de detecção possuem complexidade $O(t)$ no máximo, exceto o algoritmo de *Point Pattern Matching*. Logo, ambos algoritmos possuem a mesma complexidade: $O(t(\log t)^{3/2})$.

Passo	Método RAWVec	Método SB-RAWVec
1	$O(t)$	$O(2t)$
2	$O(2t)$	$O(t)$
3	$O(t(\log t)^{3/2})$	$O(n^2)$
4	$O(t)$	$O(n^2)$
5	$O(n^2)$	$O(t(\log t)^{3/2})$
6	$O(n^2)$	$O(n^2)$
7	$O(n^2)$	$O(n^2)$
8	—	$O(n^2)$
9	—	$O(n^2)$
Total	$O(t(\log t)^{3/2})$	$O(t(\log t)^{3/2})$

Tabela 3.2: Complexidade de cada passo dos algoritmos de detecção.

3.1.2 Perturbação Inserida no Mapa

É muito importante analisar a qualidade do mapa vetorial marcado gerado pelos Algoritmos de Inserção 2 e 4, isto é, o deslocamento máximo de um ponto no mapa vetorial marcado não pode ser maior do que o erro máximo tolerado por ele.

A Figura 3.1 mostra o deslocamento máximo de um ponto para cada um dos métodos. No Método RAWVec, tal deslocamento pode ser controlado somente pela constante C e por alguma manipulação na intensidade dos pixels da imagem raster (como normalização, por exemplo). O deslocamento máximo de um ponto é dado por $\pm CR_{max}\sqrt{2}$, onde R_{max} é o valor da maior intensidade encontrada na imagem R .

Por outro lado, no Método SB-RAWVec, essa propriedade é controlada por duas constantes, C e I , e nenhuma mudança na intensidade dos pixels da imagem raster é necessária. O máximo deslocamento de um ponto é dado por $\pm CI\sqrt{2}$. Note que este controle só pode ser utilizado no mapa vetorial marcado para publicação M_1 ; o erro inserido no mapa vetorial marcado para detecção M_2 não pode ser controlado, assim, tal mapa não pode ser publicado, pois pode ser visualmente degradado pela marca d'água.

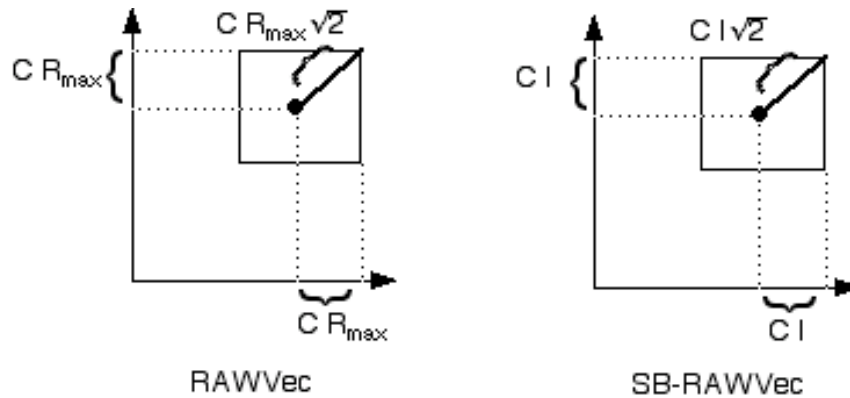


Figura 3.1: O deslocamento máximo de um ponto para os dois métodos apresentados.

Assim, a escolha dos valores das constantes C e I deve ser realizada baseando-se no erro máximo tolerado pelo mapa vetorial. Se este erro for e , então C e I devem ser os maiores valores positivos não nulos tais que $CI\sqrt{2} < e$, onde $0 < I < 255$. Note que quanto maiores forem os valores de C e I , mais robusto será o método, mas o erro inserido no mapa será maior.

3.1.3 Métricas para Comparação de Imagens

A comparação entre a marca d'água original R e a marca d'água extraída S usa dois coeficientes: r , o coeficiente de correlação de Pearson [22] e h , o coeficiente de qualidade.

O coeficiente de qualidade h é baseado na observação humana. Ele varia de 0 a 5 e representa a resposta de 5 pessoas para a pergunta: "Você diria com certa segurança que a imagem S é a mesma imagem R após sofrer algum processo de degradação?". Cada resposta negativa é representada por 0 e cada resposta positiva, por 1; h é o valor da soma de todas as respostas.

O coeficiente de correlação de Pearson r é baseado nas intensidades dos pixels e é calculado da seguinte forma: Seja I_{m_R} e I_{m_S} as intensidade médias das imagens R e S , respectivamente, e $I_{r_{ij}}$ e $I_{s_{ij}}$ as intensidades dos pixels $r_{ij} \in R$ e $s_{ij} \in S$, respectivamente. Então,

$$r = \frac{\sum_i \sum_j (I_{r_{ij}} - I_{m_R})(I_{s_{ij}} - I_{m_S})}{\sqrt{(\sum_i \sum_j (I_{r_{ij}} - I_{m_R})^2)(\sum_i \sum_j (I_{s_{ij}} - I_{m_S})^2)}}. \quad (3.1)$$

Note que o coeficiente de qualidade h varia de 0 a 5, enquanto o coeficiente de Pearson r varia de 0 a 1.

3.2 Resultados Práticos

Nessa seção são apresentados alguns resultados da execução dos métodos e a análise de diversos ataques encontrados na literatura. Além disso, uma execução sem nenhum ataque também é mostrada.

As seções 3.2.1 e 3.2.2 apresentam diversos testes e execuções dos dois métodos para ataques comumente encontrados em marcas d'água para mapas vetoriais, enquanto as seções 3.2.3, 3.2.4 e 3.2.5 mostram uma análise teórica dos métodos para ataques gerais para marcas d'água, que não tem recebido a devida atenção pelos pesquisadores da área de mapas vetoriais. A classificação e definição dos ataques aqui apresentados são as mesmas encontradas em [21].

3.2.1 Extração sem Ataques

Esta seção demonstra a extração de marcas d'água no caso trivial, ou seja, sem ataques, para os dois métodos apresentados. Foram realizados três testes para diferentes tamanhos de mapa vetorial.

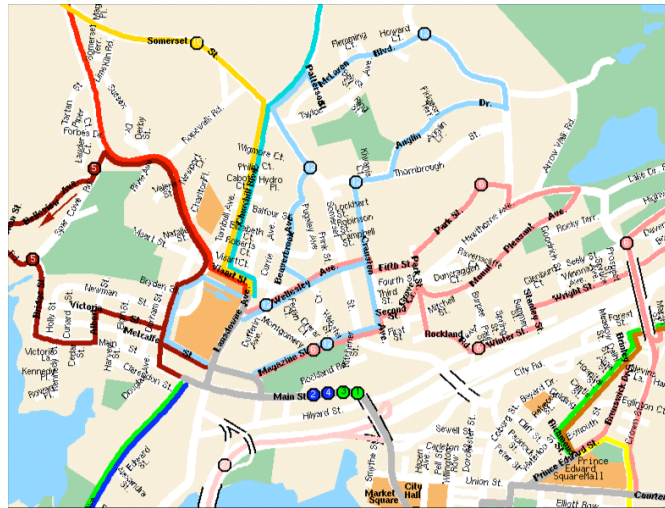


Figura 3.2: Mapa Vetorial Original 1 - 12540 pontos.

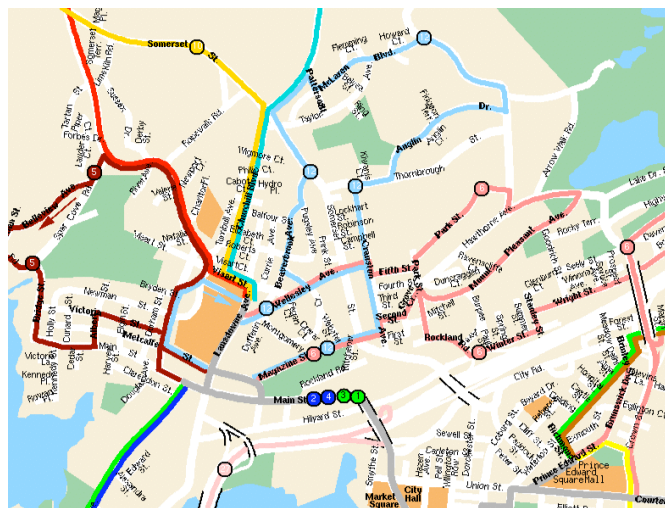


Figura 3.3: Mapa Vetorial 1 marcado pelo Método RAWVec, com $C = 0.027$.

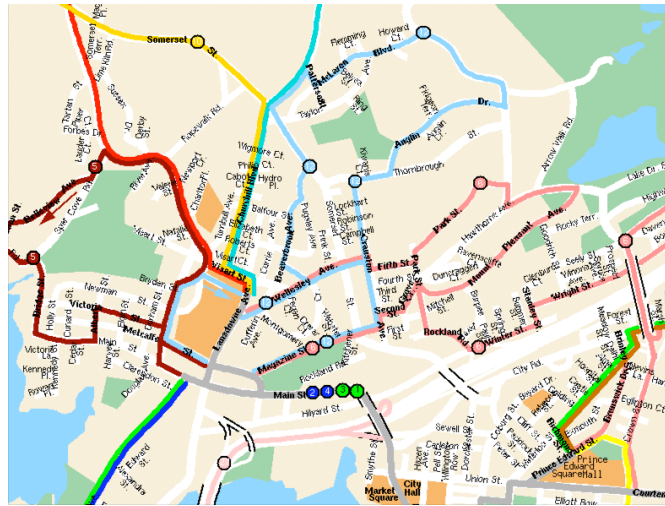


Figura 3.4: Mapa Vetorial 1 marcado pelo Método SB-RAWVec, com $C = 0.027$ e $I = 175$.

A Uno hand

Drawn by Roland Rosenfeld

Figura 3.5: Mapa Vetorial Original 2 - 5149 pontos.

A Uno hand

Drawn by Roland RosenfeldFigura 3.6: Mapa Vetorial 2 marcado pelo Método RAWVec, com $C = 0.05$.

A Uno hand

Drawn by Roland RosenfeldFigura 3.7: Mapa Vetorial 2 marcado pelo Método SB-RAWVec, com $C = 0.05$ e $I = 200$.

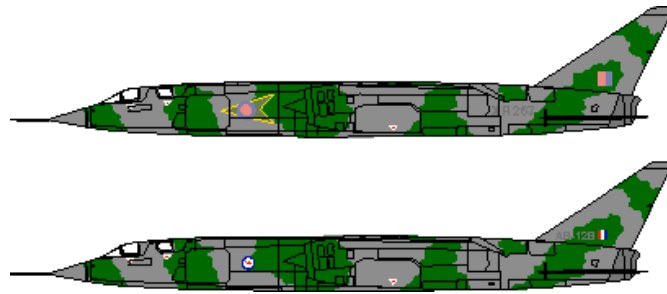


Figura 3.8: Mapa Vetorial Original 3 - 2146 pontos.

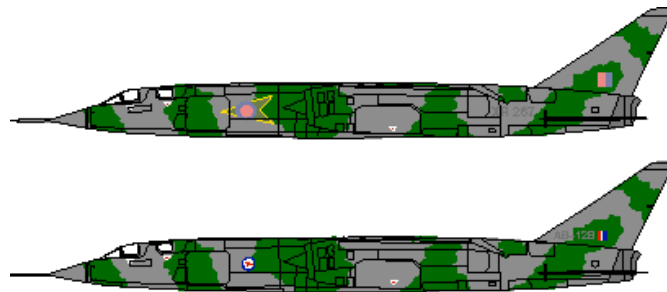


Figura 3.9: Mapa Vetorial 3 marcado pelo Método RAWVec, com $C = 0.042$.

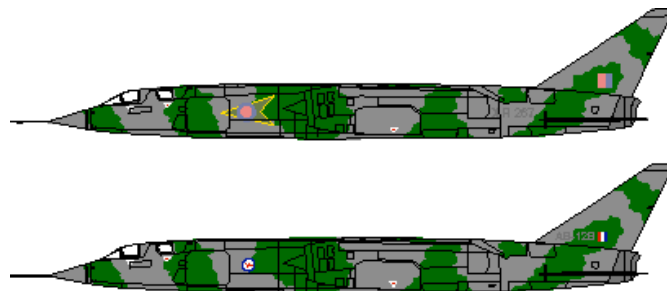


Figura 3.10: Mapa Vetorial 3 marcado pelo Método SB-RAWVec, com $C = 0.042$ e $I = 100$.

A tabela 3.3 mostra os erros absoluto e relativo em cada um dos mapas marcados mostrados anteriormente. Os erros para todos os mapas são muito baixos, mas o Método SB-RAWVec possui erros ainda mais baixos, principalmente porque utiliza duas constantes – C e I – para controlar o deslocamento máximo de um ponto.

Seja p_o um ponto do mapa vetorial original e p_m o ponto do mapa vetorial marcado respectivo; os erros absoluto e relativo são calculados da seguinte forma:

$$Err_{abs} = |p_o - p_m|,$$

$$Err_{rel} = (Err_{abs}/p_o) * 100.$$

Mapa Vetorial	Erro Absoluto (cm) - RAWVec	Erro Relativo (%) - RAWVec	Erro Absoluto (cm) - SB-RAWVec	Erro Relativo (%) - SB-RAWVec
1	0.0097	0.9828	0.0050	0.5017
2	0.0169	0.8402	0.0104	0.5204
3	0.0158	0.3511	0.0042	0.0932

Tabela 3.3: Erros absolutos e relativos para os testes realizados com três diferentes mapas vetoriais.



Figura 3.11: Marca d'água usada na inserção.

Mapa Vetorial	Marca d'Água Método RAWVec	r (%)	h	Marca d'Água Método SB-RAWVec	r (%)	h
1		99.9227	5		99.1574	5
2		99.9656	5		99.644	5
3		99.8941	5		99.4688	5

Tabela 3.4: Resultados da execução dos métodos sem nenhum ataque.

Os valores de r e h para todos os testes mostram que os resultados no caso trivial são muito satisfatórios.

3.2.2 Ataques Geométricos

Ataques Geométricos são os mais utilizados para avaliar marcas d'água em mapas vectoriais. Eles constituem-se de ataques de transformação, como rotação e translação, de cropping, de inserção de objetos, de inserção de ruído aleatório e outros. A avaliação desses ataques para os dois métodos apresentados é realizada nas seções a seguir. Para todos os testes foram utilizados os mapas encontrados nas figuras 3.2, 3.3 e 3.4.

Ataques de Transformação

Os principais ataques de transformação consistem nas três transformações afim básicas: rotação, translação e escalonamento. Na rotação todos os objetos da figura são rotacionados em torno de um ponto, normalmente a origem; na translação, todos os objetos da figura são deslocados mantendo a mesma distância entre eles e o mesmo tamanho; e no escalonamento todos os objetos da figura aumentam ou diminuem de tamanho.

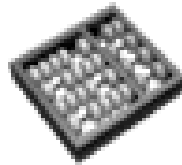


Figura 3.12: Marca d'água usada na inserção.

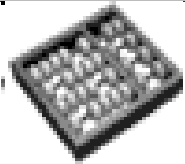
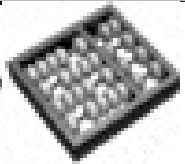
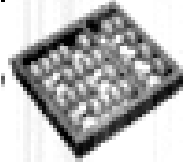
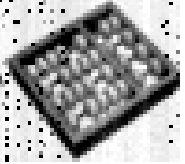
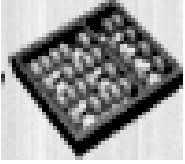
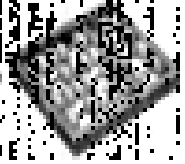
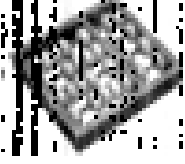
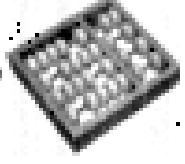
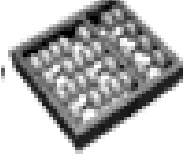
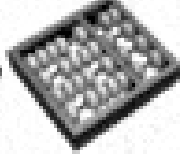
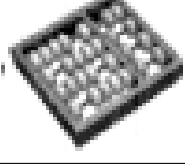
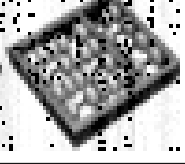
Ângulo (°)	Marca d'Água Método RAWVec	r (%)	h	Marca d'Água Método SB-RAWVec	r (%)	h
0		99.8186	5		99.8996	5
30		99.1666	5		91.2851	5
45		99.3918	5		58.6373	2
60		61.0368	2		98.0505	5
90		99.8054	5		99.8775	5
142		99.6007	5		84.5865	3

Tabela 3.5: Resultados da execução dos métodos com ataques de rotação.

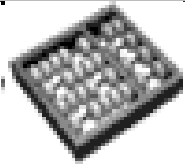
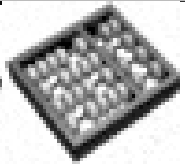
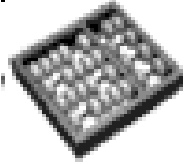

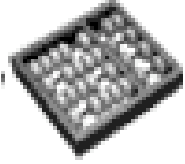
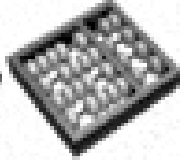
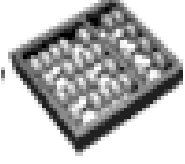
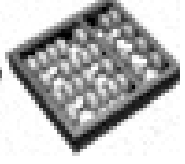
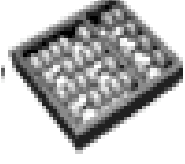
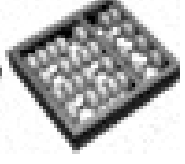
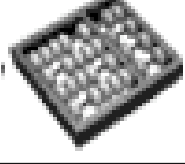
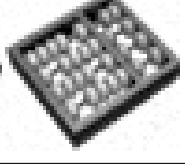
Desloc. (mm)	Marca d'Água Método RAWVec	r (%)	h	Marca d'Água Método SB-RAWVec	r (%)	h
0		99.8186	5		99.8996	5
23		99.8186	5		99.8996	5
25		99.8186	5		99.8996	5
50		99.8186	5		99.8996	5
75		99.8186	5		99.8996	5
100		99.8186	5		99.8996	5

Tabela 3.6: Resultados da execução dos métodos com ataques de translação.

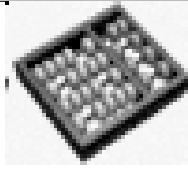
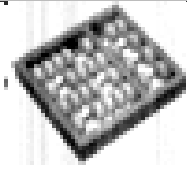
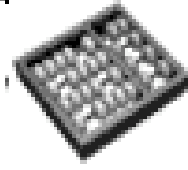
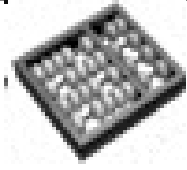
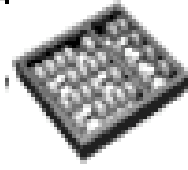
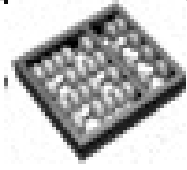

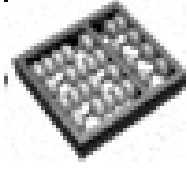

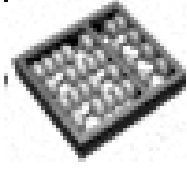
Fator	Marca d'Água Método RAWVec	r (%)	h	Marca d'Água Método SB-RAWVec	r (%)	h
1/2		99.9442	4		98.7657	5
1		99.8186	5		99.8996	5
5		99.8186	5		99.9009	5
7		99.8186	5		99.9006	5
10		99.8186	5		99.9009	5

Tabela 3.7: Resultados da execução dos métodos com ataques de escalonamento.

As Tabelas 3.5, 3.6 e 3.7 mostram os testes feitos com ataques de transformação. Os resultados para os ataques de rotação mostram que a marca d'água é reconhecível em grande parte dos ângulos testados. Diferente dos ataques de transformação, em que a imagem resultante é independente do valor do deslocamento, os ataques de rotação dependem do ângulo, pois o erro inserido na marca d'água extraída é baseado no seno e no cosseno do ângulo utilizado, que são números reais.

Assim como a transformação, o escalonamento também produz ótimos resultados, para um fator de escalonamento maior que 1. Para valores menores, ou seja, para a redução do tamanho dos objetos, a marca d'água extraída pode ter uma qualidade um pouco inferior. Isso ocorre pois, assim como os senos e cossenos, os fatores de escalonamento são representados por números reais e os arquivos utilizados no teste como mapa vetoriais representam cada ponto como inteiros. Assim, durante o ataque, uma multiplicação do valor do ponto por um número real deve ser realizada, resultando em um arredondamento que pode propagar um erro na marca d'água extraída durante a detecção. Ainda assim, as imagens resultantes são reconhecíveis.

Ataques de Cropping

Ataques de cropping consistem em remover um conjunto de pontos e objetos do mapa. Para uma grande porcentagem de pontos removidos, é mais difícil reconhecer a marca d'água extraída, mas o mapa vetorial também é degradado, ou seja, não pode ser reutilizado.



Figura 3.13: Marca d'água usada na inserção.



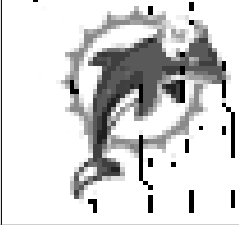







Pontos Removidos (%)	Marca d'Água Método RAWVec	r (%)	h	Marca d'Água Método SB-RAWVec	r (%)	h
0		99.8364	5		99.8991	5
25		85.4109	5		85.5124	5
37		72.0721	5		72.147	5
50		53.8829	4		54.0675	4
75		22.6851	0		23.1251	0

Tabela 3.8: Resultados da execução dos métodos com ataques de cropping.

Os testes representados na Tabela 3.8 mostram que a qualidade da marca d'água extraída depois de um ataque de cropping é totalmente dependente do número de pontos que foram removidos do mapa vetorial. Quando mais de 75% dos pontos são removidos, a imagem resultante é praticamente irreconhecível, mas o mapa vetorial atacado também é irreconhecível, não podendo mais ser utilizado e perdendo seu valor. As Figuras 3.14 e 3.16 mostram os mapas vetoriais das Figuras 3.15 e 3.17, respectivamente, depois de um ataque de cropping com 75% dos pontos removidos aleatoriamente. Assim, podemos considerar o resultado para esse ataque satisfatório.

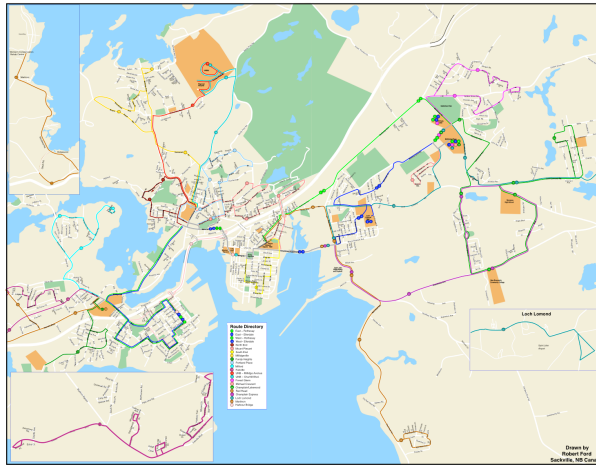


Figura 3.14: Mapa Vetorial marcado pelo Método RAWVec.

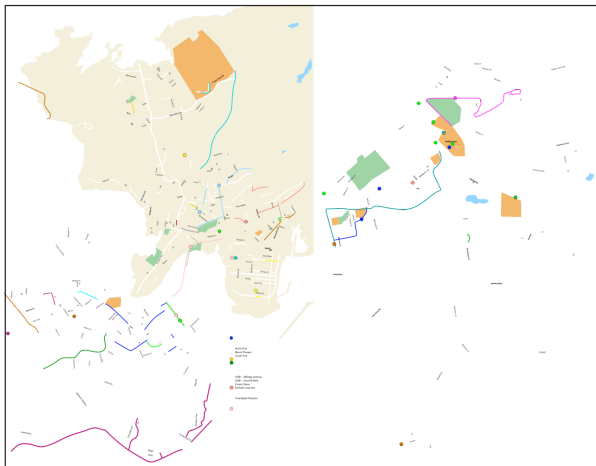


Figura 3.15: Mapa Vetorial marcado pelo Método RAWVec e atacado com remoção de 75% dos pontos.

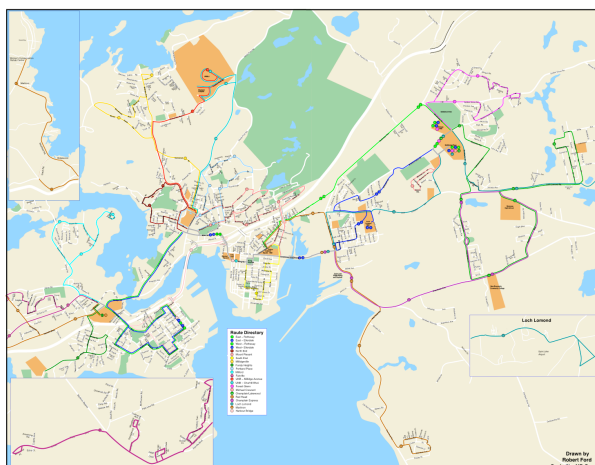


Figura 3.16: Mapa Vetorial marcado pelo Método SB-RAWVec.

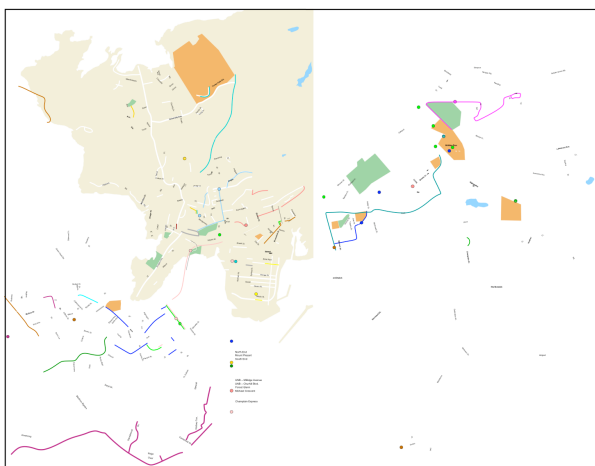


Figura 3.17: Mapa Vetorial marcado pelo Método SB-RAWVec e atacado com remoção de 75% dos pontos.

Ataques de Inserção de Objetos

Objetos pequenos ou até mesmo transparentes podem ser inseridos no mapa vetorial marcado para dificultar a detecção da marca d'água sem alterar o mapa visualmente. O algoritmo de *Point Pattern Matching* detecta tais objetos e os descarta durante o algoritmo de detecção. Nos testes realizados apresentados na Tabela 3.9, o número de objetos inseridos é proporcional ao número de objetos presentes no mapa vetorial.

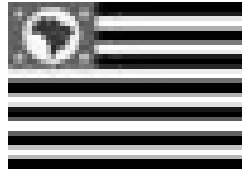


Figura 3.18: Marca d'água usada na inserção.

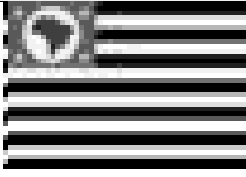
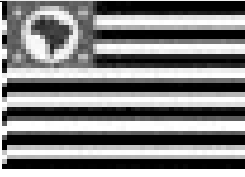
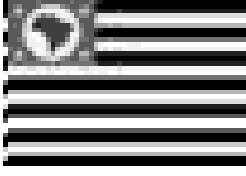
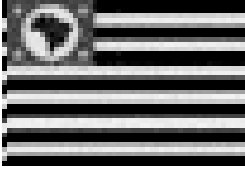

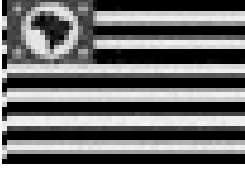
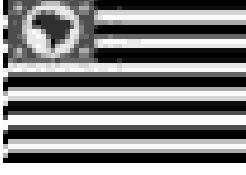
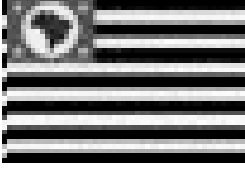
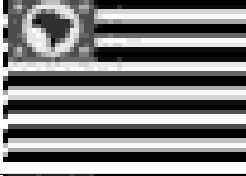
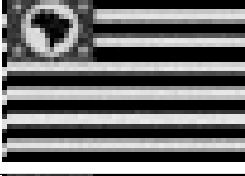
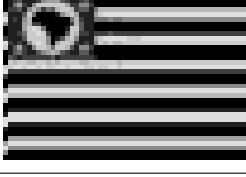
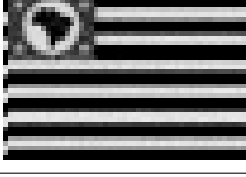
Objetos Inseridos (%)	Marca d'Água Método RAWVec	r (%)	h	Marca d'Água Método SB-RAWVec	r (%)	h
0		99.898	5		99.956	5
25		99.898	5		99.6569	5
50		99.7876	5		99.6569	5
69		99.7876	5		99.8791	5
75		99.7876	5		99.3519	5
100		98.6324	5		99.3519	5

Tabela 3.9: Resultados da execução dos métodos com ataques de inserção de objetos.

A Tabela 3.9 mostra que a qualidade da marca d'água inserida é praticamente a mesma para todos os testes realizados, independentemente do número de objetos inseridos. Esse resultado é bastante satisfatório, já que pode ser possível inserir objetos com transparência nos mapas vetoriais, o que não reduzirá o valor ou a qualidade do mapa marcado. Portanto, os testes mostram que os dois métodos propostos são muito resistentes a ataques de inserção de objetos.

Ataques de Alteração na Ordem dos Objetos

Ataques de alteração na ordem dos objetos mantêm o mapa vetorial visualmente intacto, pois os mesmos objetos são desenhados, mas podem destruir a marca d'água extraída. Assim, é importante que o algoritmo de *Point Pattern Matching* encontre a verdadeira ordem dos objetos para que a marca d'água extraída possa ser reconhecida.



Figura 3.19: Marca d'água usada na inserção.













Objetos Reordenados (%)	Marca d'Água Método RAWVec	r (%)	h	Marca d'Água Método SB-RAWVec	r (%)	h
0		99.8615	5		99.9004	5
13		99.6049	5		97.329	5
25		99.7135	5		95.3429	5
50		99.6234	5		93.2986	5
75		99.7507	5		93.1569	5
100		99.7449	5		96.0793	5

Tabela 3.10: Resultados da execução dos métodos com ataques de alteração na ordem dos objetos.

Da mesma maneira que os ataques de inserção, os ataques de alteração na ordem dos objetos são facilmente percebidos e removidos pelos dois métodos, resultando em imagens de ótima qualidade, independente do número de objetos reordenados, como mostram os resultados da Tabela 3.10. Esse resultado também é satisfatório, pois os mapas vetoriais resultantes de ataques de alteração na ordem dos objetos também mantém o seu valor e sua qualidade originais.

Ataques de Inserção de Ruído Aleatório

Ataques de inserção de ruído aleatório consistem em adicionar um valor aleatório x , onde $0 \leq x \leq max$ a uma porcentagem dos pontos do mapa vetorial. Assim como os ataques de cropping, quanto maior a porcentagem dos pontos afetados, mais difícil é detectar a marca d'água, mas mais degradado o mapa vetorial atacado será. O valor de max utilizado nos testes abaixo é 5 mm.



Figura 3.20: Marca d'água usada na inserção.

Pontos Afetados (%)	Marca d'Água Método RAWVec	r (%)	h	Marca d'Água Método SB-RAWVec	r (%)	h
0		99.9992	5		99.9984	5
10		99.5075	5		99.4892	5
25		97.3663	5		97.3134	5
50		89.07	5		89.0013	5
75		70.073	0		70.1707	0
100		6.10068	0		6.10059	0

Tabela 3.11: Resultados da execução dos métodos com ataques de inserção de ruído aleatório.

O ataque de inserção de ruído aleatório afeta a imagem resultante conforme um grande número de pontos é afetado. Neste caso, o mapa vetorial atacado também possui um grande número de pontos deslocados. Assim como nos ataques de cropping, a marca d'água extraída possui boa qualidade até que 75% dos pontos sejam afetados.

Ataques Combinados

Ataques combinados reúnem os vários tipos de ataques descritos anteriormente nesta seção. Assim, é mais difícil detectar e retirar tal ataque. As Tabelas 3.12 e 3.13 mostram alguns testes realizados e seus respectivos resultados. T representa o deslocamento usado na translação, R , o ângulo usado na rotação, E o fator de escalonamento, C a porcentagem de pontos removidos no ataque de cropping, O a porcentagem de objetos reordenados, N a porcentagem de pontos em que o ruído aleatório de amplitude máxima max é inserido.



Figura 3.21: Marca d'água usada na inserção.


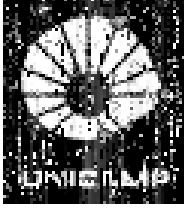


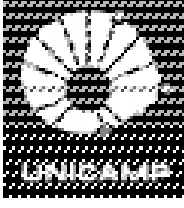

R (°)	T (mm)	E	C (%)	O (%)	N (%)	max (mm)	Marca d'Água	r (%)	h
247	47	6	0	0	0	0		99.8997	5
169	63	2	15	0	0	0		84.7755	4
31	77	6	0	15	0	0		84.7323	4
20	9	5	22	36	0	0		63.3263	3
90	34	9	0	0	1	8		75.3584	5
10	15	2	11	37	7	3		98.1838	5

Tabela 3.12: Resultados da execução do Método RAWVec com ataques combinados.


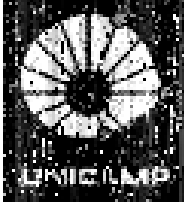


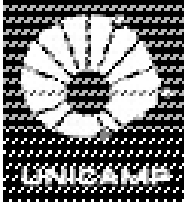
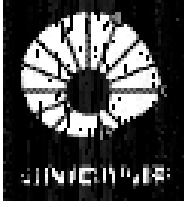
R (°)	T (mm)	E	C (%)	O (%)	N (%)	max (mm)	Marca d'Água	r (%)	h
247	47	6	0	0	0	0		99.8997	5
169	63	2	15	0	0	0		85.6516	3
31	77	6	0	15	0	0		87.3984	5
20	9	5	22	36	0	0		68.2268	3
90	34	9	0	0	1	8		75.2726	5
10	15	2	11	37	7	3		89.6466	4

Tabela 3.13: Resultados da execução do Método SB-RAWVec com ataques combinados.

As Tabelas 3.12 e 3.13 mostram resultados para diversos ataques combinados. Para testes somente com ataques geométricos, os dois métodos possuem resultados muito bons, com marcas d'água extraídas de ótima qualidade. O principal fator que determina a qualidade da imagem resultante é a porcentagem dos pontos afetados pelo ataque de cropping, pois esse ataque não apenas remove informações da marca d'água, como também dificulta a detecção e remoção dos outros ataques envolvidos. Ainda assim, vários ataques são representados nas tabelas, mostrando imagens satisfatórias para diversas combinações de ataques.

3.2.3 Ataques Criptográficos

Ataques criptográficos são ataques que tentam remover a marca d'água ou inserir uma marca d'água enganosa para quebrar a segurança do método. Exemplos de ataques deste tipo são o Ataque de Força Bruta e o Ataque do Oráculo.

Um Ataque de Força Bruta nos Métodos RAWVec e SB-RAWVec teria como principal objetivo conseguir o mapa vetorial original a partir do mapa vetorial marcado. A principal razão que torna esse ataque muito improvável é a dificuldade em determinar se um mapa vetorial dado é original ou não, pois somente o aspecto visual não é suficiente. Além disso, como a marca d'água não é removida do mapa vetorial marcado durante o algoritmo de detecção, e sim detectada, não é possível conseguir o mapa vetorial original, tornando o Ataque de Força Bruta mais difícil.

Outro ataque conhecido é o Ataque do Oráculo. Este ataque utiliza as ferramentas que o projetista forneceu, como algoritmos e informações públicas, pra tentar extrair informações secretas. A informação secreta no Método RAWVec é o mapa vetorial original, mas este é revelado durante o algoritmo de detecção. No Método SB-RAWVec, as informações secretas são o mapa original, o valor de I e as duas partes da marca d'água. Em nenhum momento é possível obter algumas das duas partes da marca d'água, nem durante o algoritmo de detecção. Note que se o algoritmo de detecção for rodado com o mapa vetorial original como entrada é possível obter alguma das duas partes, mas neste caso a informação mais importante, o mapa vetorial original, já teria sido revelada.

3.2.4 Ataques de Protocolo

Ataques de Protocolo focam na aplicação de marcas d'água em geral, sem o objetivo de remover ou destruir a marca d'água. Um tipo de Ataque de Protocolo consiste em tentar detectar uma marca d'água diferente da que foi inserida, mudando assim a autoria do dado. Os Métodos RAWVec e SB-RAWVec são robustos a esse tipo de ataque principalmente porque a marca d'água utilizada é uma imagem raster e não uma sequência aleatória de

bits. Assim, é muito mais difícil para um atacante modificar o mapa marcado de forma a retirar uma marca d'água diferente, pois esta deve ser visualmente reconhecível.

Outro tipo de Ataque de Protocolo são os Ataques de Cópia, em que a marca d'água é estimada a partir do dado marcado e inserida em um outro dado. Este tipo de ataque só é válido quando a marca d'água é privada, no caso do Método RAWVec, a marca d'água é pública e utilizada durante o algoritmo de detecção. Para o Método SB-RAWVec, as duas partes criadas a partir da marca d'água durante a inserção (como descrito em 2.4.2) são privadas. É difícil estimar alguma destas partes, pois nenhuma informação a respeito delas é conhecida, já que o valor da constante I é privado e não é utilizado durante o algoritmo de detecção.

3.2.5 Ataques de Remoção

Ataques com o objetivo de remover a marca d'água completamente sem utilizar a chave ou qualquer outro tipo de informação são chamados de ataques de remoção. Tais ataques incluem os ataques de colisão, que estimam o dado original a partir de uma média. Para isso é necessário que um mesmo mapa seja marcado com pelo menos 10 marcas d'água diferentes, gerando 10 diferentes mapas marcados. Assim, é recomendado que, ao usar os Métodos RAWVec ou SB-RAWVec, o mesmo mapa vetorial não seja marcado diversas vezes com marcas d'água diferentes.

Capítulo 4

Conclusão e Trabalhos Futuros

Neste trabalho o Método de Marcas d'Água Raster em Mapas Vetoriais (RAWVec) é revisto e analisado. Este método insere uma marca d'água, representada por uma imagem raster, num mapa vetorial, modificando as coordenadas dos objetos do mapa. O algoritmo de detecção extrai a marca d'água do mapa vetorial utilizando o mapa original. A marca d'água extraída pode ser comparada com a original através de um algoritmo probabilístico e/ou a avaliação do ponto de vista do olho humano, aumentando a eficácia do método. Ataques podem alterar ou remover pixels da marca d'água, mas o olho humano pode facilmente reconhecer a imagem.

Uma alternativa semi-pública para o Método RAWVec é apresentada e analisada. Esse novo método é tão eficiente e tão robusto quanto o original, já que também utiliza um algoritmo de *Point Pattern Matching* e redundância para detectar e remover ataques.

Além disso, ele possui as mesmas qualidades do método original, como o uso de uma imagem raster como marca d'água, mas com a vantagem de ser semi-público. Métodos públicos e semi-públicos de marca d'água são mais práticos, pois o mapa original pode não estar disponível por uma série de motivos, limitando a aplicação de métodos privados. Outra vantagem do novo método é um melhor controle do deslocamento máximo de um pixel, ou seja, melhor controle da qualidade do mapa vetorial marcado.

4.1 Trabalhos Futuros

Os métodos apresentados neste trabalho utilizam imagens raster em níveis de cinza como marca d'água. Seria interessante modificar o método para que imagens raster coloridas, descritas como três matrizes (RGB), também possam ser utilizadas como marca d'água. Além disso, tantos os mapas vetoriais, como a marca d'água são imagens em duas dimensões; assim, expandir o método para imagens descritas em três dimensões também pode ser interessante.

Tanto o Método RAWVec, como a sua alternativa semi-pública são métodos que inserem marcas d'água em mapas vetoriais no domínio espacial, ou seja, o algoritmo de inserção modifica as coordenadas das estruturas presentes no mapa vetorial e o algoritmo de detecção compara as coordenadas da estrutura do mapa marcado com o mapa de prova. Assim, o método é vulnerável a um ataque que modifique as estruturas do mapa, mantendo sua forma, mas alterando completamente suas coordenadas. Portanto, poderíamos procurar alternativas para o método que sejam ainda mais robustas e consigam detectar e remover esse tipo de ataque.

O Método SB-RAWVec é baseado em um algoritmo de Criptografia Visual. Outras alternativas públicas, baseadas em criptografia de chave pública, podem ser estudadas. Poderíamos, por exemplo, usar chaves públicas e privadas para esconder as informações do mapa original e só obtê-las durante o algoritmo de detecção, sem que o mapa original precise ser divulgado.

O Método SB-RAWVec gera dois mapas marcados, um para publicação e um para detecção. Só é possível controlar e verificar a quantidade de ruído inserido no mapa de publicação. Seria importante verificar o ruído que será inserido no mapa marcado para detecção, pois se o mapa vetorial marcado para detecção for útil, ele pode gerar novos ataques. Também pode ser interessante analisar se a autoria desse mapa pode ser verificada.

Referências Bibliográficas

- [1] Doug stinson’s visual cryptography page. <http://www.cacr.math.uwaterloo.ca/dstinson/visual.html>.
- [2] Carlo Blundo, Alfredo De Santis, and Moni Naor. Visual cryptography for grey level images. *Inf. Process. Lett.*, 75(6):255–259, 2000.
- [3] Jean-Daniel Boissonnat and Monique Teillaud. On the randomized construction of the delaunay tree. *Theor. Comput. Sci.*, 112(2):339–354, 1993.
- [4] Stelvio Cimato, Roberto De Prisco, and Alfredo De Santis. Optimal colored threshold visual cryptography schemes. *Des. Codes Cryptography*, 35(3):311–335, 2005.
- [5] Hwan Kang, Kab Kim, and Jong Choi. A vector watermarking using the generalized square mask. volume 00, page 0234, Los Alamitos, CA, USA, 2001. IEEE Computer Society.
- [6] Yuanyuan Li and Luping Xu. A blind watermarking of vector graphics images. *iccima*, 00:424, 2003.
- [7] Chang-Chou Lin and Wen-Hsiang Tsai. Visual cryptography for gray-level images by dithering techniques. *Pattern Recogn. Lett.*, 24(1-3):349–358, 2003.
- [8] Karina M. Magalhães and Ricardo Dahab. SB-RAWVec - A semi-blind watermarking method for vector maps. In *ICC 2009 Communication and Information Systems Security Symposium*, Dresden, Germany, 2009.
- [9] Douglas Aurélio Marques. Marcas d’Água visuais em mapas vetoriais. Master’s thesis, UNICAMP, 2005.
- [10] Douglas Aurélio Marques, Karina M. Magalhães, and Ricardo Dahab. RAWVec – A method for watermarking vector maps. In *SBSeg 2007: Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 2007.

- [11] Moni Naor and Adi Shamir. Visual cryptography. *Lecture Notes in Computer Science*, 950:1–12, 1995.
- [12] Ryutarou Ohbuchi, Shigeo Takahashi, Takahiko Miyazawa, and Akio Mukaiyama. Watermarking 3d polygonal meshes in the mesh spectral domain. In *GRIN'01: No description on Graphics interface 2001*, pages 9–17, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.
- [13] Ryutarou Ohbuchi, Hiroo Ueda, and Shuh Endoh. Robust watermarking of vector digital maps. In *Proc. IEEE Conference on Multimedia and Expo 2002*, 2002.
- [14] Ryutarou Ohbuchi, Hiroo Ueda, and Shuh Endoh. Watermarking 2d vector maps in the mesh-spectral domain. In *SMI '03: Proceedings of the Shape Modeling International 2003*, page 216, Washington, DC, USA, 2003. IEEE Computer Society.
- [15] Yu-Chi Pu, Wei-Chang Du, and I-Chang Jou. Toward blind robust watermarking of vector maps. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 930–933, Washington, DC, USA, 2006. IEEE Computer Society.
- [16] V. Solachidis, N. Nikolaidis, and I. Pitas. Fourier descriptors watermarking of vector graphics images. In *ICIP00*, pages Vol III: 9–12, 2000.
- [17] Henry Sonnet, Tobias Isenberg, Jana Dittmann, and Thomas Strothotte. Illustration watermarks for vector graphics. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 73, Washington, DC, USA, 2003. IEEE Computer Society.
- [18] P. van Wamelen, Z. Li, and S. Iyengar. A fast expected time algorithm for the point pattern matching problem. Technical report, Louisiana State University, Dept. of Mathematics, 1999.
- [19] Michael Voigt and Christoph Busch. Watermarking 2d-vector data for geographical information systems. In *Proc. SPIE, Security and watermarking of Multimedia Content*, pages 621–628, 2002.
- [20] Michael Voigt, Bian Yang, and Christoph Busch. Reversible watermarking of 2d-vector data. In *MM&Sec '04: Proceedings of the 2004 workshop on Multimedia and security*, pages 160–165, New York, NY, USA, 2004. ACM Press.

- [21] Sviatoslav Voloshynovskiy, Shelby Pereira, Thierry Pun, J. Eggers, and J. Su. Attacks on digital watermarks: Classification, estimation-based attacks and benchmarks. *IEEE Communications Magazine (Special Issue on Digital watermarking for copyright protection: a communications perspective)*, 39(8):118–127, 2001.
- [22] Eugene K. Yen and Johnston G. Roger. The ineffectiveness of the correlation coefficient for image comparisons.
- [23] HOU Young-Chang. Visual cryptography for color images. *Pattern recognition*, 36(6):1619–1629, 2003.