

Relatório do Trabalho 3

Lucas Veríssimo Botelho - 170109071

Resumo—Nesse trabalho sera realizado experimentos em fotos baseados na alteração do domínio espacial para armazenamento de informações, utilizaremos a transformada de arnold e a transformada discreta de cosseno para tal.Foi chegado ao resultado de que o armazenamento de fotos em outras fotos é algo inteiramente possível e pratico. Foram utilizados a linguagem python em conjunto com as bibliotecas openCV e numpy para a realização deste.

Index Terms—Processamento de imagens, segmentação, morfologia, erosão, dilatação

I. INTRODUÇÃO

[1] Como o uso da internet se tornou predominante na sociedade também era de se esperar que a distribuição de conteúdo fosse através dela. Porém nunca antes o problema de pirataria foi tão serio como atualmente. Assim se criaram técnicas de *watermarking* para deixar a identidade da marca original no conteúdo de um jeito que não possa ser retirada e também não atrapalhe o usuário. Nesse trabalho optamos por esconder essa marca no domínio espacial da foto em vez de pelo da frequência e usamos de base o artigo *Fast and robust image watermarking method in the spatial domain*[1].

Foram utilizadas 2 técnicas para chegar ao resultado final do projeto: transformação de arnold que é um método de encriptação de fotos no qual elas são embaralhadas para esconder sua informação, transformação do cosseno discreto bidimensional que é um método para calcular o valor discreto de cosseno numa matrix de pixels assim podendo alterar seu valor.

Agora explicando como sera organizado o resto do relatório iremos dividir em 4 partes contando com essa, na seção 2 iremos demonstrar a metodologia dos experimentos que foram feitos e como foram feitos, na seção 3 vamos demonstrar os resultados desses experimentos e analisar o porque deles e na seção 4 vamos chegar a conclusão desses experimentos.

II. METODOLOGIA

Aqui iremos dissertar sobre como foi feito os experimentos pedidos nesse trabalho.Foi optado pela utilização da linguagem python em conjunto com as bibliotecas openCV e numpy. Nessa seção iremos dividir em duas partes, a inserção do *watermark* em si e após isso a remoção deste.

A. Inserção do Watermark

Vamos dividir a inserção do *watermark* em 7 passos.

Passo 1:

Primeiro a imagem original é dividida em 3 imagens, cada uma representando uma cor do RGB. No método original esta imagem era dividida em blocos de tamanho $m \times m$ onde o valor de $m=2$ mas aqui optamos pelo valor de $m=1$ pela

avaliação dos resultados.

Passo 2:

A imagem da logo é dividida também em 3 imagens, cada uma representando uma cor do RGB, e após isso é efetuado o método da transformação do arnold em cada uma. A chave de encriptação e deciptação do período de arnold é escolhido de acordo com o tamanho da imagem e da escolha do usuario também. Assim que as 3 imagens estiverem embaralhadas elas são transformadas em 3 arrays de bits distintas.

Passo 3:

Selecionasse 2 blocos subsequentes **A** e **B** da imagem original, na ordem horizontal.

Passo 4:

$$dc_p = \frac{1}{m} \sum_{x=0}^{m-1} \sum_{y=0}^{m-1} f_p(x, y)$$

É calculado o valor DC desses 2 blocos **A** e **B** através da transformada de cosseno discreta, na formula o m seria o tamanho do bloco e $f_p(x,y)$ é o valor de cada pixel dentro do bloco. O resultado dessa formula seria os 2 valores **dc1** e **dc2** correspondentes aos blocos **A** e **B** respectivamente.

Passo 5:

Agora selecionando um único bit da sequencia da logo, o qual chamamos de **w** é calculado os novos valores de **dc1*** e **dc2***.

$$dc_1^* = \begin{cases} avg - 0.5 \times T_i, & \text{if } w = '0' \text{ and } dc_1 - dc_2 > 0 \\ avg + 0.5 \times T_i, & \text{if } w = '1' \text{ and } dc_1 - dc_2 \leq 0 \\ dc_1 & \text{else} \end{cases}$$

$$dc_2^* = \begin{cases} avg + 0.5 \times T_i, & \text{if } w = '0' \text{ and } dc_1 - dc_2 > 0 \\ avg - 0.5 \times T_i, & \text{if } w = '1' \text{ and } dc_1 - dc_2 \leq 0 \\ dc_2 & \text{else} \end{cases}$$

Na formula o valor $avg=(dc_1 + dc_2)/2$ e $T_i=1$.

Passo 6:

Os novos valores **dc1*** e **dc2*** são aplicados no bloco de pixels da imagem original a transformando para os dois novos blocos **A*** e **B*** que juntos guardam a informação de 1 bit.

$$f_p^*(x, y) = f_p(x, y) + \frac{1}{m} \times (dc_p^* - dc_p)$$

A formula calcula o novo valor de cada pixel que será aplicado no novo bloco de pixels.

Passo 7:

Os passos 3 ao 6 são repetidos ate que a sequencia de bits da logo se acabe para as 3 imagens RGB.

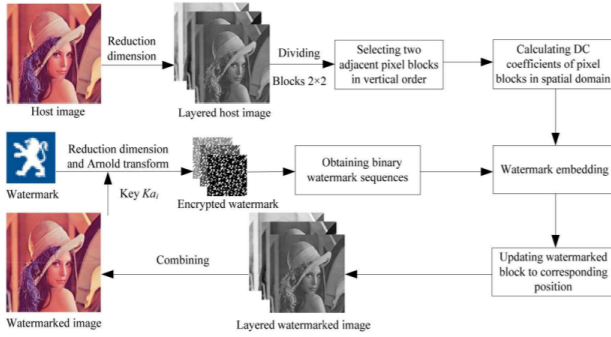


Fig. 1. Diagrama da inserção da watermark

B. Remoção do Watermark

Vamos dividir a remoção do *watermark* em 7 passos.

Passo 1:

Primeiro a imagem com a *watermark* é dividida em 3 imagens, cada uma representando uma cor do RGB. **Passo 2:** Seleccionasse 2 blocos subsequentes **A*** e **B*** da imagem com a *watermark*, na ordem horizontal.

Passo 3:

É calculado o valor DC desses 2 blocos **A*** e **B*** através da transformada de cosseno discreta. Resultando nos 2 valores **dc1*** e **dc2***

$$dc_p^* = \frac{1}{m} \sum_{x=0}^{m-1} \sum_{y=0}^{m-1} f_p^*(x, y)$$

A formula é calculada tal qual a formula de inserção.

Passo 4:

Agora com os dois valores **dc1*** e **dc2*** é possível escolher se o valor binário **w*** desses 2 blocos é 1 ou 0 sendo que se **dc1*** for maior que **dc2*** o valor é 1 e se não for o valor é 0.

$$w^* = \begin{cases} '1', & \text{if } dc_1^* - dc_2^* > 0 \\ '0' & \text{else} \end{cases}$$

Passo 5:

Os passos 2 a 4 são repetidos ate que se complete a sequencia de bits original da logo.

Passo 6:

A transformação inversa de arnold é aplicada na logo com a chave escolhida no inicio de forma que ela retorne a forma original.

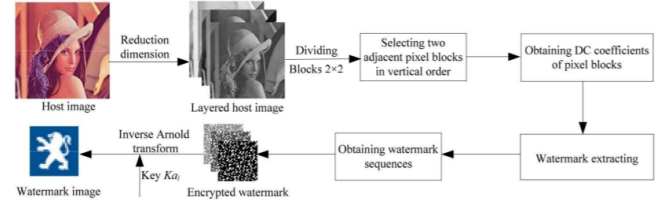


Fig. 2. Diagrama da remoção da watermark

III. RESULTADOS

A. Imagens utilizadas no experimento

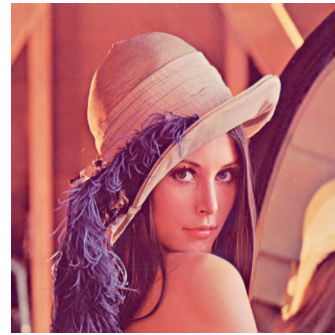


Fig. 3. Foto lena 512x512 utilizada no experimento



Fig. 4. Logo da peugot 128x128 utilizada de exemplo de logo do experimento

A fim de realizar o experimento de forma parecida com o metodo original foram escolhidas essas duas fotos para sua realização, a primeira sendo a foto classica lena de processamento de imagens e a segunda uma foto da logo da Peugeot retirada da internet já que a original não foi encontrada. As duas imagens foram escaladas para o tamanho necessario para o experimento, foi demonstrado que com os métodos utilizados uma logo maior que 1/4 da imagem original não poderia ser inserido nesta.

B. Transformação de Arnold

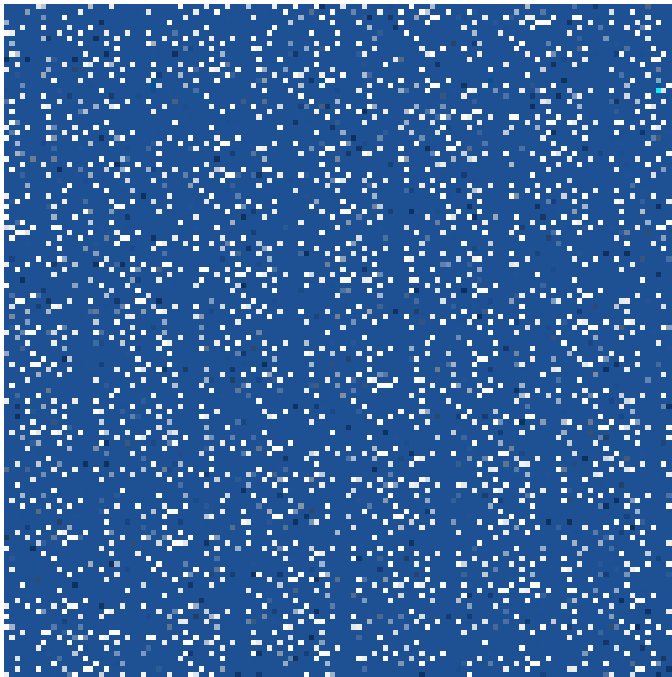


Fig. 5. Logo após executada a transformação de arnold

Como foi optado pelo tamanho de imagem 128 então o período de arnold escolhido foi o de 96. A imagem original passou por 48 transformação de arnold na parte de encriptação resultado na imagem acima.

C. Inserção da watermark

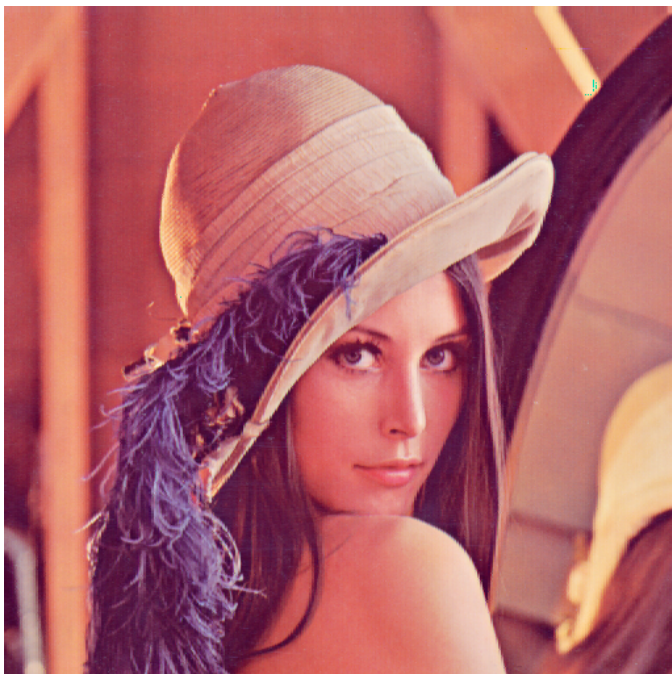


Fig. 6. Resultado da inserção da watermark na imagem de Lena 512x512



Fig. 7. Comparação da imagem original a esquerda com a imagem com watermark a direita

A inserção da watermark utilizando blocos 1x1 se mostrou bastante eficiente. Aqui é demonstrado claramente que não há uma grande ruptura visual da imagem original. Graças a escolha do bloco 1x1 foi possível armazenar uma logo de tamanho 128x128, algo que com o bloco 2x2 só era possível se chegar até uma logo de tamanho 64x64. Com a possibilidade de se armazenar mais informações visuais foi optado por essa escolha.

D. Remoção da watermark



Fig. 8. Logo original 128x128



Fig. 9. Logo após a remoção da watermark e deciptação pela transformada inversa de arnold

No final a imagem resultante da logo foi quase igual a imagem original, não perdendo nenhuma qualidade e demonstrando a viabilidade do experimento para futuros usos.

E. Transformação de arnold inversa

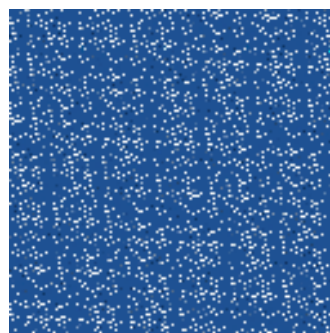


Fig. 10. Resultado da aplicação do bottom-hat da imagem morftest.png



Fig. 11. Resultado da inversão da imagem pós bottom-hat

Após a remoção da logo da imagem com *watermark* foi aplicado a transformação inversa de arnold nos 3 layers da imagem RGB a fim de transforma-la novamente na logo original, já que o período de encriptação era 96 e foi passado por 48 transformadas no começo, aqui foi aplicada mais 48 transformadas a fim de retornar ao numero completo do período decriptando a imagem por completo.

IV. CONCLUSÃO

Nesse projeto foi possível entender os benefícios práticos da alteração de imagens no domínio espacial para armazenamento de informações importantes. Seja watermark ou ate alguma informação secreta esse ferramente poderia ter bons usos no mundo atual. Além disso o aprendizado pratico de encriptação de imagens é algo que poderia ser usado para armazenamento de fotos sensíveis de forma eficiente e pratica. Com o conjunto do experimento completo seria ótimo a utilização deste para proteção contra a pirataria e outros problemas que assolam o meio digital atualmente.

REFERENCES

- [1] Z. Yuan, Q. Su, D. Liu, X. Zhang, and T. Yao, "Fast and robust image watermarking method in the spatial domain," *IET Image Processing*, vol. 14, no. 15, p. 3829–3838, 2020.