

Projeto Final: Relatório Final

Hiding Message Maker (HMM)
Esteganografia

Prof. Dr. Moacir Ponti

Giuliano Lourençon
Vitor Giovani Dellinocente

10295590
9277875

Descrição do problema	3
Métodos	4
LSB	4
XOR-LSB	6
Informações sobre os textos de entrada	7
Testes	8
LSB	8
XOR-LSB	12
Considerações sobre os resultados	15
Conclusão	16

Descrição do problema

No mundo atual, troca de informações se tornou cada mais necessária e cada vez mais surgem novas maneiras seguras de trocar essas mensagens. Uma maneira muito conhecida é a esteganografia, que tem por finalidade esconder uma mensagem numa imagem. Essa técnica é muito utilizada com criptografia para que se torne mais efetiva ainda, lembrando que nesse projeto focamos apenas em esteganografia pois tem maior relação com a matéria. O problema consiste em vários textos, de vários tamanhos que serão escondidos em várias imagens.

Em relação ao texto, temos 4 categorias de tamanho:

- Small (pequeno);
- Medium (médio);
- Large (grande);
- Xlarge (muito grande);

No tópico **Informações sobre os textos de entrada**, temos descreveremos informações mais detalhadas sobre os textos.

Em relação às imagens, utilizamos 4 delas para realizar a esteganografia, são elas:

- Tigre
 - **Nome:** tiger.jpg
 - **Imagem:**
<https://github.com/VitorGDellino/Steganography/tree/master/images/tiger.jpg>
 - **Tamanho:** 7680 x 4320
- Mar
 - **Nome:** sea.jpg
 - **Imagem:**
<https://github.com/VitorGDellino/Steganography/tree/master/images/sea.jpg>

- **Tamanho:** 1250 x 2000
- Cachoeira
 - **Nome:** waterfall.jpg
 - **Imagem:**
<https://github.com/VitorGDellino/Steganography/tree/master/images/waterfall.jpg>
 - **Tamanho:** 3840 x 2160
- Mountain
 -
 - **Nome:** mountain.jpg
 - **Imagem:**
<https://github.com/VitorGDellino/Steganography/tree/master/images/mountain.jpg>
 - **Tamanho:** 4000 x 2662

Lembrando que a fonte das imagens está especificada no README do github, logo abaixo de cada imagem.

Métodos

Para a realização do projeto utilizamos dois métodos, o LSB, que já é um algoritmo muito conhecido e o nosso próprio algoritmo de esteganografia que intitulamos XOR-LSB.

LSB

Utilizamos o método LSB, que consiste em esconder a mensagem utilizando os bits menos significativos. Explicando de maneira aprofundada, utilizamos 3 pixels para esconder uma letra de 8 bits, pois em cada pixel possui 3 canais de cores e escondemos um bit em cada um dos canais totalizando 8 canais necessários, ou seja 3 pixels. Fica mais fácil observar na Figura 1.

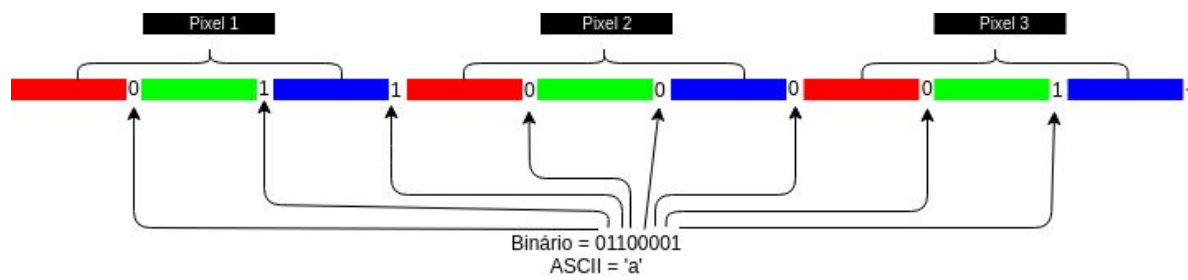


Figura 1 - Esquema de funcionamento do LSB

XOR-LSB

O XOR-LSB tem como base um LSB comum, pois utiliza os bits menos significativos. Porém, nesse caso utilizamos 8 pixels para esconder uma letra de 8 bits pois esse algoritmo consiste em utilizar os 3 canais de cores para esconder um bit da letra. Basicamente o XOR-LSB calcula o XOR entre os bits menos significativos de cada canal de cor (RGB). Fica mais fácil visualizar na Figura 2.

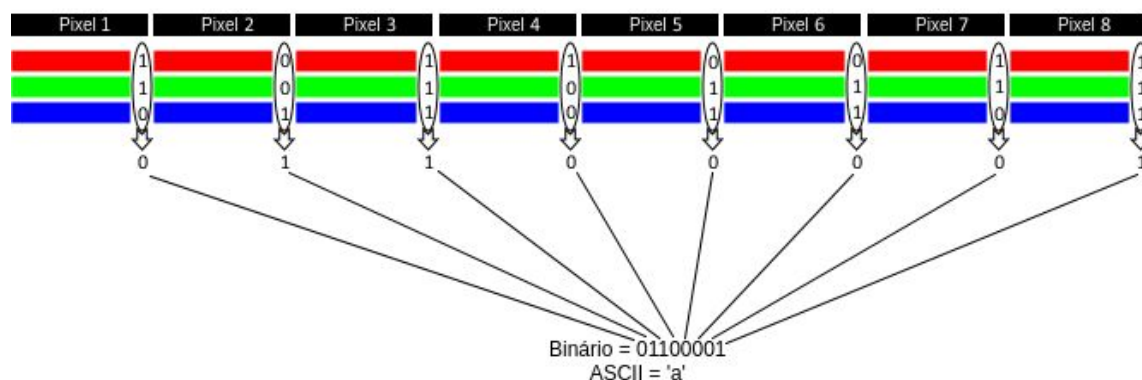


Figura 2 - Esquema de funcionamento do XOR-LSB

Informações sobre os textos de entrada

Texto pequeno:

- Número de caracteres: 2909
- Tamanho: 2.9 kB
- Texto:

<https://github.com/VitorGDellino/Steganography/blob/master/txt/test-small>

Texto médio:

- Número de caracteres: 5900
- Tamanho: 5.9 kB
- Texto:

<https://github.com/VitorGDellino/Steganography/blob/master/txt/test-medium>

Texto grande:

- Número de caracteres: 36234
- Tamanho: 36.4 kB
- Texto:

<https://github.com/VitorGDellino/Steganography/blob/master/txt/test-large>

Texto muito grande:

- Número de caracteres: 263951
- Tamanho: 264.9 kB
- Texto:

<https://github.com/VitorGDellino/Steganography/blob/master/txt/test-xlarge>

Os textos foram gerados usando um site gerador de textos aleatórios, que pode ser encontrado em <https://www.lipsum.com/>.

Testes

LSB

Nome: sea.jpg

Imagem:

<https://github.com/VitorGDellino/Steganography/tree/master/images/sea.jpg>

Tamanho: 1250 x 2000

Antes: 377.2 kB **Depois:** 3.3 MB

- Teste com sea.jpg e texto pequeno
 - Erro entre imagens - 0.03885
 - Tempo de execução do algoritmo - 0.22s
 - Tempo total de execução - 8.64s

- Teste com sea.jpg e texto médio
 - Erro entre imagens - 0.05528
 - Tempo de execução do algoritmo - 0.43s
 - Tempo total de execução - 8.97s

- Teste com sea.jpg e texto grande
 - Erro entre imagens - 0.13714
 - Tempo de execução do algoritmo - 2.62s
 - Tempo total de execução - 10.85s

- Teste com sea.jpg e texto extra grande
 - Erro entre imagens - 0.37045
 - Tempo de execução do algoritmo - 18.86s
 - Tempo total de execução - 25.89s

Nome: tiger.jpg

Imagem:

<https://github.com/VitorGDellino/Steganography/tree/master/images/tiger.jpg>

Tamanho: 7680 x 4320

Antes: 1 MB **Depois:** 8.1 MB

- Teste com tiger.jpg e texto pequeno
 - Erro entre imagens - 0.01084
 - Tempo de execução do algoritmo - 0.19s
 - Tempo total de execução - 52.88s

- Teste com tiger.jpg e texto médio
 - Erro entre imagens - 0.01550
 - Tempo de execução do algoritmo - 0.38s
 - Tempo total de execução - 54.66s

- Teste com tiger.jpg e texto grande
 - Erro entre imagens - 0.03749
 - Tempo de execução do algoritmo - 2.36s
 - Tempo total de execução - 56.12s

- Teste com tiger.jpg e texto extra grande
 - Erro entre imagens - 0.10204
 - Tempo de execução do algoritmo - 16.67s
 - Tempo total de execução - 79.32s

Nome: waterfall.jpg

Imagem:

<https://github.com/VitorGDellino/Steganography/tree/master/images/waterfall.jpg>

Tamanho: 3840 x 2160

Antes: 3.4 MB **Depois:** 15.9 MB

- Teste com waterfall.jpg e texto pequeno
 - Erro entre imagens - 0.02169

- Tempo de execução do algoritmo - 0.18s
- Tempo total de execução - 11.38s

- Teste com waterfall.jpg e texto médio
 - Erro entre imagens - 0.03087
 - Tempo de execução do algoritmo - 0.37s
 - Tempo total de execução - 11.59s

- Teste com waterfall.jpg e texto grande
 - Erro entre imagens - 0.07668
 - Tempo de execução do algoritmo - 2.31s
 - Tempo total de execução - 13.64s

- Teste com waterfall.jpg e texto extra grande
 - Erro entre imagens - 0.20679
 - Tempo de execução do algoritmo - 16.04s
 - Tempo total de execução - 27.46s

Nome: mountain.jpg

Imagem:

<https://github.com/VitorGDellino/Steganography/tree/master/images/mountain.jpg>

Tamanho: 4000 x 2662

Antes: 3.6 MB **Depois:** 11.9 MB

- Teste com mountain.jpg e texto pequeno
 - Erro entre imagens - 0.01889
 - Tempo de execução do algoritmo - 0.22s
 - Tempo total de execução - 56.97s

- Teste com mountain.jpg e texto médio
 - Erro entre imagens - 0.02711
 - Tempo de execução do algoritmo - 0.46s
 - Tempo total de execução - 57.34s

- Teste com mountain.jpg e texto grande
 - Erro entre imagens - 0.06739
 - Tempo de execução do algoritmo - 2.82s
 - Tempo total de execução - 59.85s

- Teste com mountain.jpg e texto extra grande
 - Erro entre imagens - 0.18195
 - Tempo de execução do algoritmo - 20.51s
 - Tempo total de execução - 77.21s

XOR-LSB

Nome: sea.jpg

Imagem:

<https://github.com/VitorGDellino/Steganography/tree/master/images/sea.jpg>

Tamanho: 1250 x 2000

Antes: 377.2 kB **Depois:** 3.3 MB

- Teste com sea.jpg e texto pequeno
 - Média dos erros entre imagens - 0.36324
 - Tempo de execução do algoritmo - 0.48s
 - Tempo total de execução - 5.25s

- Teste com sea.jpg e texto médio
 - Média dos erros entre imagens - 0.49827
 - Tempo de execução do algoritmo - 0.95s
 - Tempo total de execução - 5.76s

- Teste com sea.jpg e texto grande
 - Média dos erros entre imagens - 1.21211
 - Tempo de execução do algoritmo - 5.61s
 - Tempo total de execução - 10.90s

- Teste com sea.jpg e texto extra grande
 - O texto não coube na imagem

Nome: tiger.jpg

Imagem:

<https://github.com/VitorGDellino/Steganography/tree/master/images/tiger.jpg>

Tamanho: 7680 x 4320

Antes: 1 MB **Depois:** 8.1 MB

- Teste com tiger.jpg e texto pequeno
 - Média dos erros entre imagens - 0.08224
 - Tempo de execução do algoritmo - 0.45s
 - Tempo total de execução - 31.19s

- Teste com tiger.jpg e texto médio
 - Média dos erros entre imagens - 0.02933
 - Tempo de execução do algoritmo - 0.85s
 - Tempo total de execução - 31.48s

- Teste com tiger.jpg e texto grande
 - Média dos erros entre imagens - 0.02742
 - Tempo de execução do algoritmo - 5.03s
 - Tempo total de execução - 37.85s

- Teste com tiger.jpg e texto extra grande
 - Média dos erros entre imagens - 0.51445
 - Tempo de execução do algoritmo - 37.44s
 - Tempo total de execução - 85.78s

Nome: waterfall.jpg

Imagem:

<https://github.com/VitorGDellino/Steganography/tree/master/images/waterfall.jpg>

Tamanho: 3840 x 2160

Antes: 3.4 MB **Depois:** 15.9 MB

- Teste com waterfall.jpg e texto pequeno
 - Média dos erros entre imagens - 0.39255
 - Tempo de execução do algoritmo - 0.41s
 - Tempo total de execução - 6.88s

- Teste com waterfall.jpg e texto médio
 - Média dos erros entre imagens - 0.55333

- Tempo de execução do algoritmo - 0.87s
- Tempo total de execução - 7.47s
- Teste com waterfall.jpg e texto grande
 - Média dos erros entre imagens - 1.36065
 - Tempo de execução do algoritmo - 4.92s
 - Tempo total de execução - 12.79s
- Teste com waterfall.jpg e texto extra grande
 - O texto não coube na imagem

Nome: mountain.jpg

Imagem:

<https://github.com/VitorGDellino/Steganography/tree/master/images/mountain.jpg>

Tamanho: 4000 x 2662

Antes: 3.6 MB **Depois:** 11.9 MB

- Teste com mountain.jpg e texto pequeno
 - Média dos erros entre imagens - 0.07159
 - Tempo de execução do algoritmo - 0.50s
 - Tempo total de execução - 33.46s
- Teste com mountain.jpg e texto médio
 - Média dos erros entre imagens - 0.100000
 - Tempo de execução do algoritmo - 0.99s
 - Tempo total de execução - 33.88s
- Teste com mountain.jpg e texto grande
 - Média dos erros entre imagens - 0.24673
 - Tempo de execução do algoritmo - 5.98s
 - Tempo total de execução - 39.55s
- Teste com mountain.jpg e texto extra grande

- O texto não coube na imagem

Considerações sobre os resultados

Em todos os casos, os textos não tiveram perdas e conseguimos recuperá-los, isso graças a compressão PNG que não possui perdas. Caso necessário, neste link <https://github.com/VitorGDellino/Steganography/tree/master/txt> pode-se ver o texto obtido de dentro da imagem, lembrando que os nomes dos texto de saída são, “txt-tamanho-nome_da_imagem-out”, por exemplo, “txt-small-tiger-out”. Ainda em relação a compressão, por não ter perdas, o tamanho da imagem (em relação a memória) cresceu muito.

No link

<https://github.com/VitorGDellino/Steganography/tree/master/images/comparations> é possível observar a diferença (visualmente e por diferença entre os pixels) entre cada imagem gerada nos testes para com as imagens originais.

Comparando os dois algoritmos vemos que o tempo de execução da esteganografia foi menor no LSB comum, porém o, achamos intrigante o fato de no tempo total, o XOR-LSB terminou em um tempo de execução menor. cremos que isso aconteceu devido aos testes terem sido realizados em máquinas diferentes. Além disso o número de caracteres que podemos esconder é maior no LSB comum, uma vez que ele precisa apenas de 3 pixels para realizar isso, já no XOR-LSB precisa-se de 8. Um outro importante é que o XOR-LSB possui um RSME maior que o LSB comum. Em alguns casos de teste notamos uma demora, porém isso foi causado devido ao tempo da compressão PNG que é consideravelmente lenta, esses resultados foram observados em ambas baterias de teste.

Conclusão

Em muito podemos melhorar nosso código, tanto na eficiência do código como na qualidade da esteganografia implementando novos algoritmos que são mais eficientes do que o LSB. Em adição com esse projeto, aprendemos como a esteganografia é utilizada juntamente com a área de segurança. Os resultados em si foram muito satisfatórios, pois o encode e o decode das mensagens foram executados com sucesso.