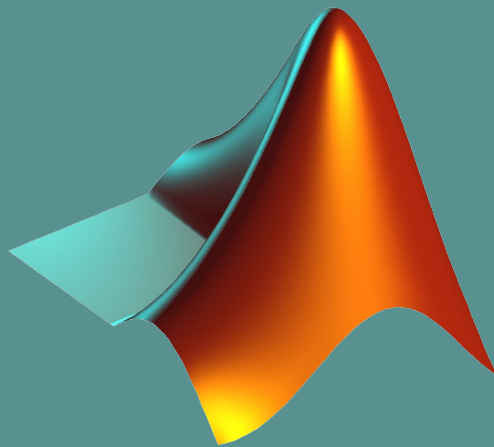


# Incorporação de cor em imagens texturizadas

Por Guilherme Braga e  
Gabriel Preihs



# A idéia principal

A ideia consiste em poder passar uma imagem de um lugar para outro com um custo de memória reduzido, em que, se coleta as principais informações da imagem inicial, por meio de Transformadas Wavelet e uma série de substituições de um processo estabelecido.

Dessa forma, em posse da imagem transformada, o decodificador pode recuperar a imagem original em RGB a partir de uma imagem cinza de apenas 1 canal utilizando o método reverso que foi empregado. Uma aplicação direta disso é poder imprimir uma imagem em preto e branco, e a partir dela apenas, poder recuperar as cores.

# A idéia principal

## Transformada Wavelet

Essa é a principal causa de ser possível realizar as operações, pois ela que é a responsável de capturar as principais características da imagem e deixar tudo guardado em uma imagem só (a partir dos valores de Y retirados da transformação de RGB para YCbCr da imagem original).

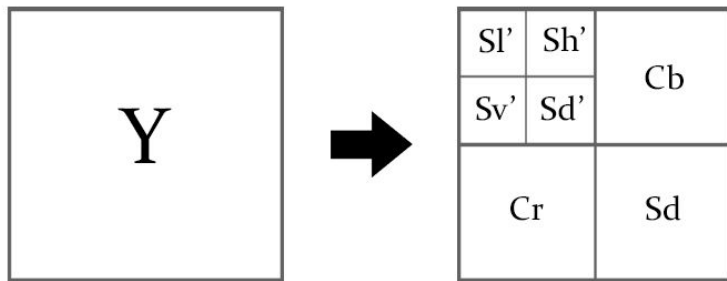
# A idéia principal

No Matlab, a função funciona recebendo a imagem da qual queremos extrair a informação junto com o tipo de onda na qual ela melhor se encaixa, retornando 4 valores da imagem que lhe é passada (matrizes) representando: Coeficiente de Aproximação, Coeficiente de Aproximação Horizontal, Coeficiente de Aproximação Vertical, Coeficiente de Aproximação Diagonal.

# A idéia principal

A partir disso, decompomos a imagem da forma mostrada no seguinte slide, efetuando substituições.

# A idéia principal



# Problemas além do programador

Existe uma série de observações a serem feitas relacionadas ao bom funcionamento do algoritmo que estão relacionadas à distribuição das imagens, como o receptor lida com a imagem texturizada.

# Para recuperar a informação da imagem texturizada:

A partir disso, para decodificação:

Se inicia o processo de captura das informações.

O foco nessa parte é recuperar o Y, Cb e Cr baseados na imagem por meio da aplicação do método de reverso de criação da imagem texturizada, fazendo assim com que seja possível a recuperação de certo nível de fidelidade (como dito antes, a semelhança nem sempre é certa).

Dessa forma, com os dados já recuperados e armazenados separadamente, basta realizar a junção de cada detalhe, para assim se obter a cópia da imagem em RGB.

Exemplos da realização do processo foram dados a seguir:



# Resultados experimentais

Exemplo:



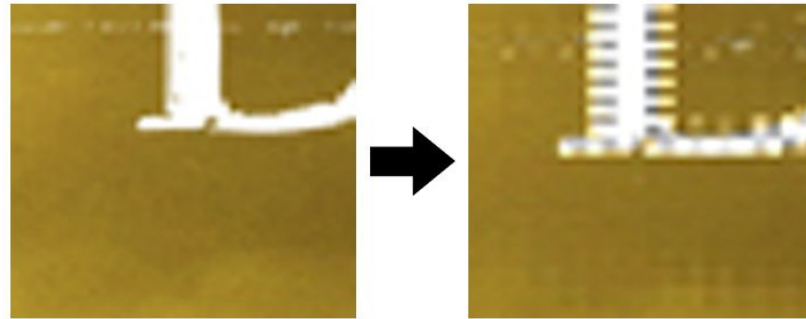
# Resultados experimentais

30% de  
reprovação!



# Resultados experimentais

Perda geral:



# Resultados experimentais

Exemplo:



# Resultados experimentais

Exemplo:



# Resultados experimentais

Original:





# Resultados experimentais

Versão  
final:



# Resultados experimentais

Original:

Original Image





# Resultados experimentais

imagem final



Versão  
final:

# Resultados experimentais

Original:



# Resultados experimentais

imagem final



versão  
final:

# Resultados experimentais

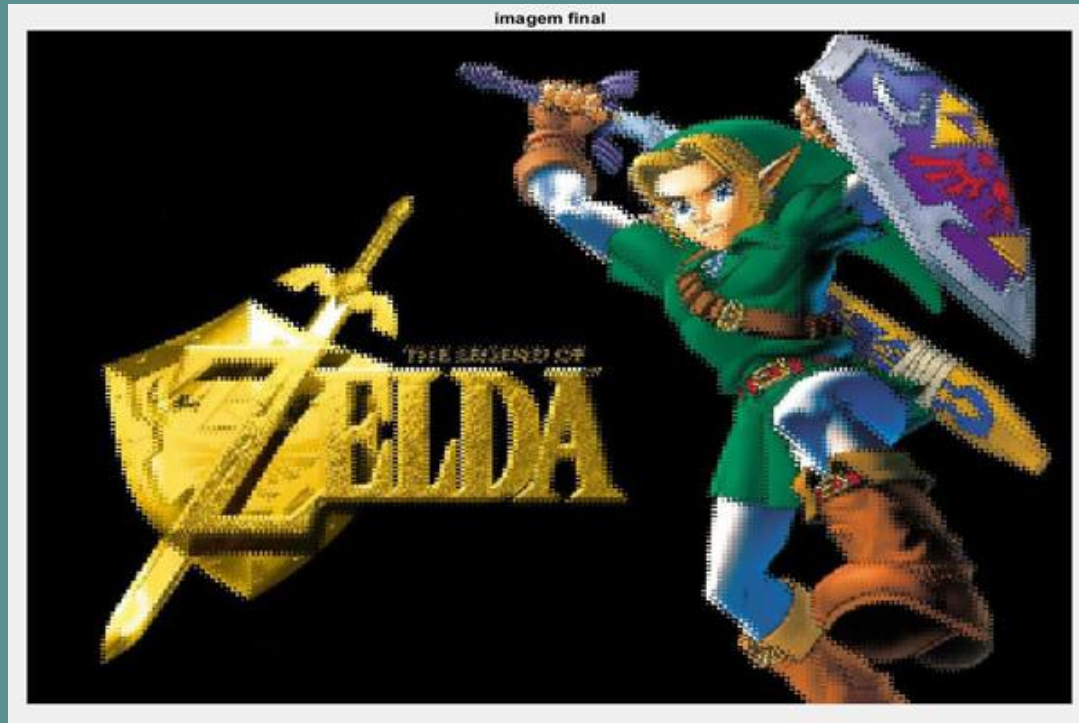


Exemplo:



# Resultados experimentais

26% de  
reprovação:



# Resultados experimentais

42% de  
reprovação:



# Resultados experimentais



Exemplo:

# Resultados experimentais

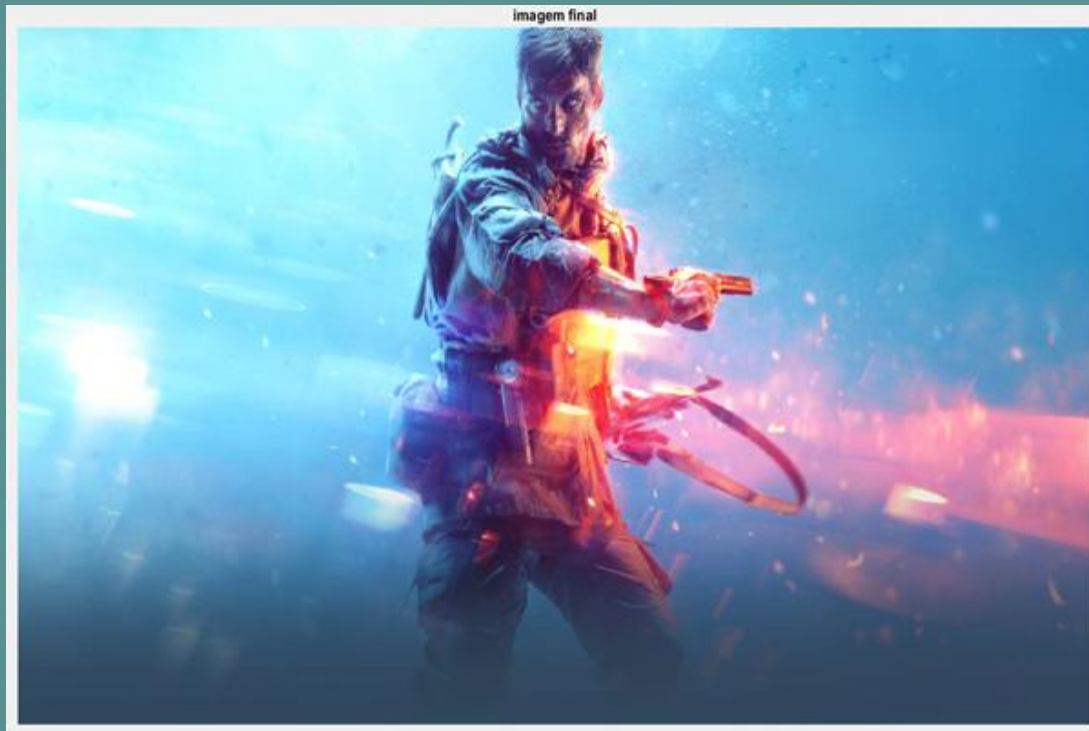
82% de  
reprovação!





# Resultados experimentais

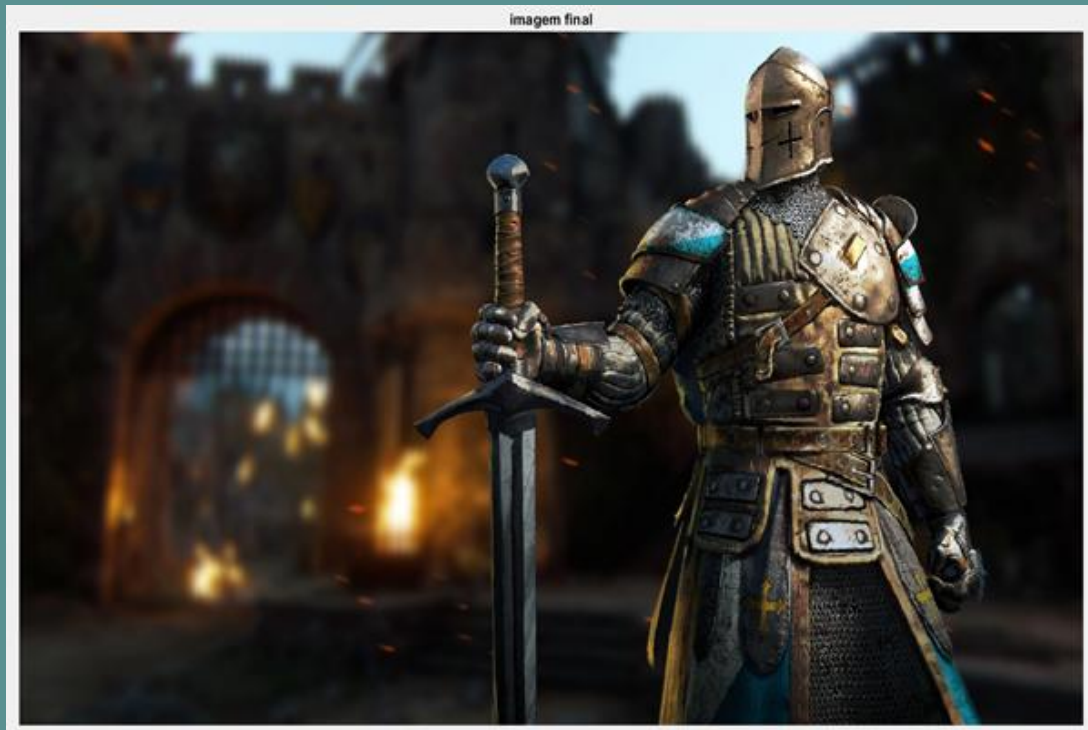
Aprovação  
de 100%!



Em 4k!

# Resultados experimentais

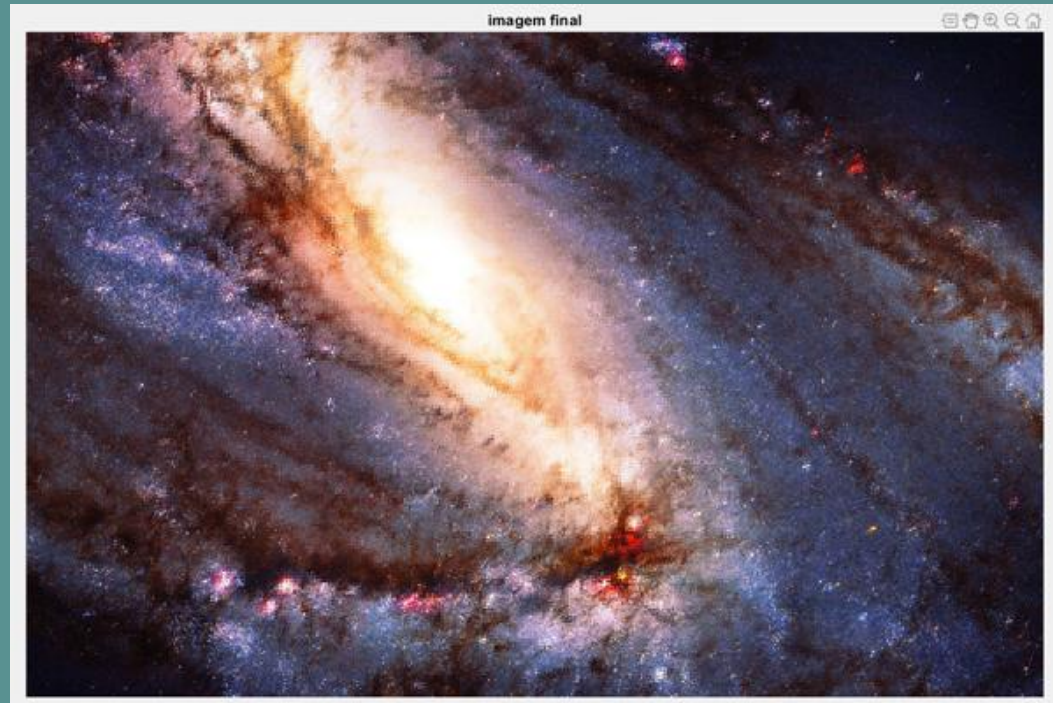
94% de  
aprovação!



Em 8k!

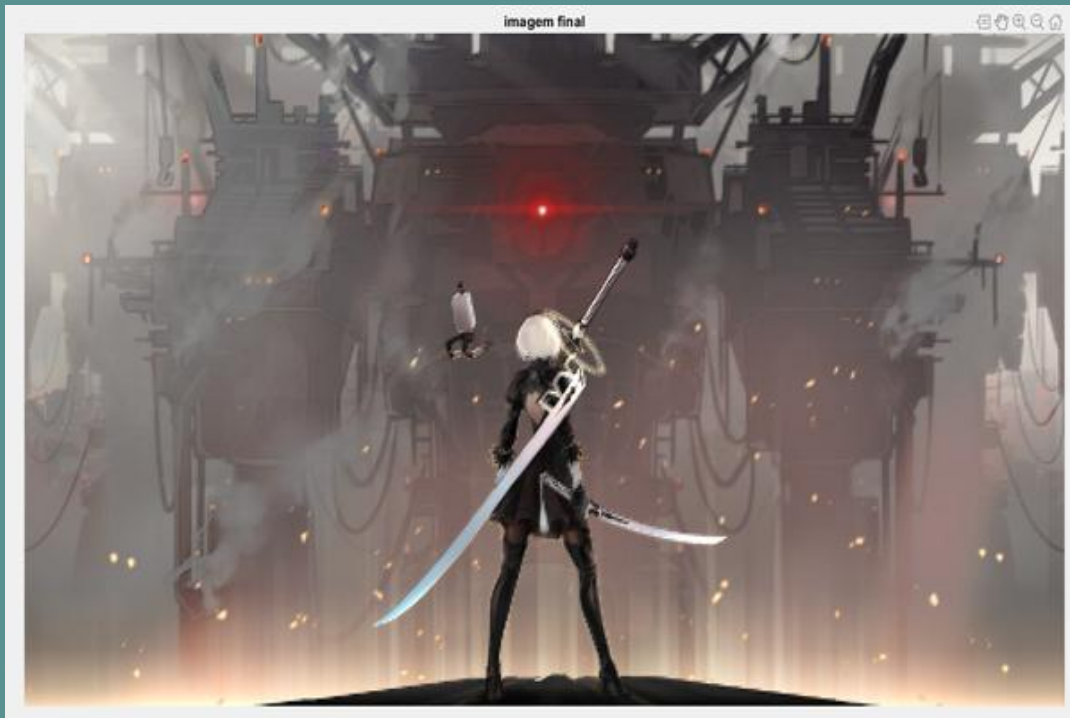
# Resultados experimentais

86% de  
aprovação!



# Resultados experimentais

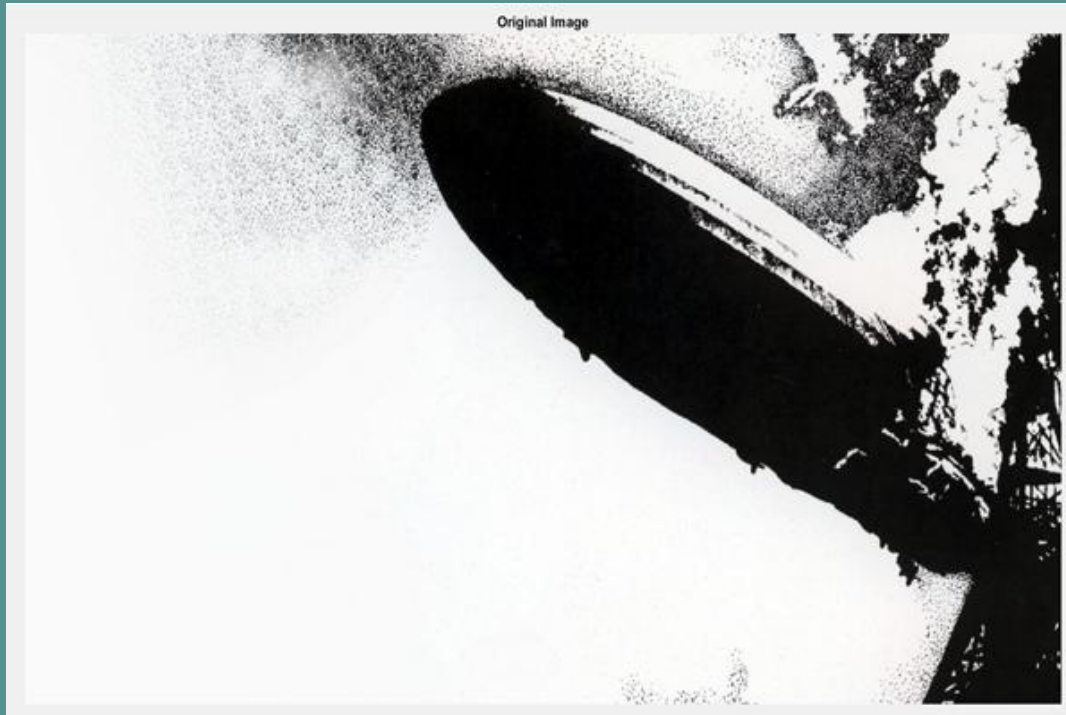
86% de  
aprovação!



1920x1080!

# Resultados experimentais

Original:



1920x1080!

# Resultados experimentais

66% de  
aprovação!



# Resultados experimentais

>Relatamos que tamanhos grandes de imagem diminuem perdas, mas essa definição pode ser perdida dependendo da distribuição da imagem texturizada.

>Ao se distribuir a imagem em um papel para que o receptor da mensagem a decodifique, erros que estão além do programador podem ocorrer, se acrescenta novas variáveis ao processo.

>No final, o funcionamento pode depender da aplicação.

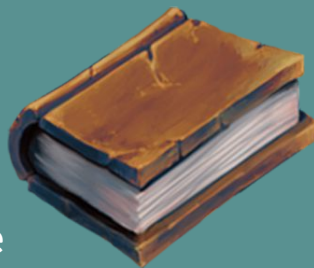


# Aceitação de resoluções

>4% dos usuários tendem a usar apenas 4k.

>40% dos usuários tendem a usar resoluções maiores que Full HD, mas estão satisfeitos com esta resolução de necessário.

>24% dos usuários não se importam caso a informação que se deseja passar é mantida, não sabem a resolução de preferência ou apenas não se importam no geral.





# Conclusões

Foi possível perceber assim que, quanto maior o número de pixels na imagem, mais bem detalhado fica o seu resultado final, com muito menos borramentos perceptíveis.

Como se pode notar, as críticas negativas das imagens finais acaba sempre sendo atribuídas para as imagens de menor qualidade e menor número de pixels.

Além disso, até mesmo as imagens mais “borradas” acabam tendo um certo tipo de aceitação, nem todos se baseiam somente em imagens acima do HD.

Sendo assim, se mostra possível a aplicação do método no cotidiano atual para um número considerável de casos.

**Fim dos slides!**

