

Trabalho 2: realce e superresolução

Desenvolva seu código sem olhar o de colegas. Plágio não será tolerado.

Nesse trabalho você deverá implementar métodos de realce e superresolução utilizando múltiplas imagens.

Você deverá implementar usando `python 3` com as bibliotecas `numpy` e `imageio`.

Leia com calma as instruções de cada etapa.

Seu programa deverá permitir receber o nome base de um conjunto de quatro imagens (1, 2, 3 e 4), a partir das quais você deverá implementar um método de equalização de histogramas em cada imagem, e posteriormente obter uma imagem de maior resolução a partir das imagens de entrada.

1. Receber via teclado:

- a) nome do arquivo base para as imagens de baixa resolução: `imglow`;
- b) nome do arquivo com imagem de alta resolução: `imghigh`;
- c) método de realce, com opções: 0, 1, 2 e 3;
- d) parâmetro do método de realce 3 (default = 1).

As 4 imagens de baixa resolução serão lidas no formato png:

`[imglow]1.png`, `[imglow]2.png`, `[imglow]3.png` e `[imglow]4.png`, a qual iremos referir por L_1 , L_2 , L_3 e L_4 .

A imagem de alta resolução, também no formato png: `[imghigh].png`, ou H

***Opção 0: sem realce** : não utilizar realce nas imagens L , e proceder diretamente para a superresolução.

Equalização de histogramas

Você deverá implementar dois métodos de equalização de histogramas para as imagens de entrada, de baixa resolução, L .

O primeiro método deverá equalizar cada imagem de entrada utilizando como função de transferência o histograma acumulado de cada imagem. O segundo deverá equalizar as imagens utilizando como função de transferência o histograma acumulado de todas as imagens.

***Opção 1: função de transferência individual** : Calcular os histogramas acumulados $ha_i(L_i)$, e então utilizar esse histograma acumulado como função de transferência para equalizar cada imagem.

***Opção 2: função de transferência conjunta** : Calcular um único histograma acumulado com base nas imagens de entrada $i = 1, \dots, 4$, i.e. $ha(L_i)$, e então utilizar esse histograma acumulado como função de transferência para equalizar todas as imagens L_i .

Note que essas opções não usam o último parâmetro de entrada.

Ajuste Gamma

***Opção 3: função de ajuste Gamma** : implementar o ajuste Gamma sobre as imagens L , utilizando a seguinte equação:

$$L'(x, y) = \left\lfloor 255 \cdot \left((L(x, y)/255.0)^{1/\gamma} \right) \right\rfloor,$$

Onde L' indica a imagem processada, e o valor γ é lido a partir do último parâmetro, lido da entrada.

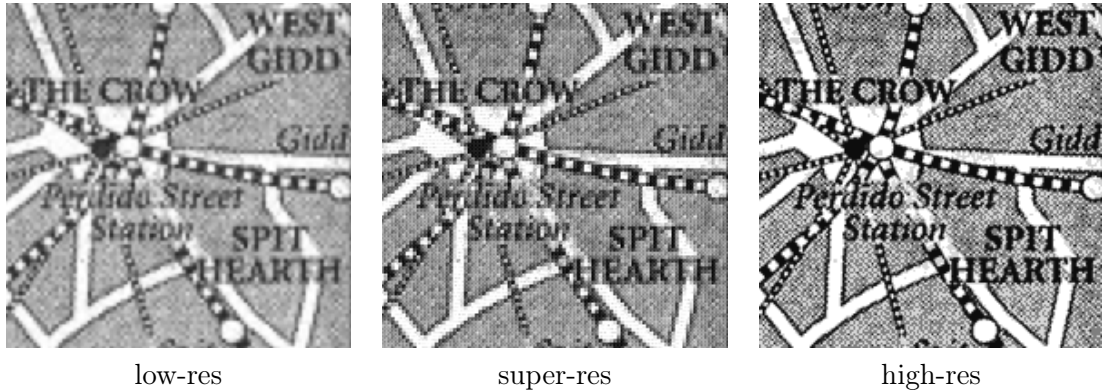
Superresolução

Assumimos que cada uma das imagens de entrada L , de baixa resolução (já equalizadas), possui características que permitam combiná-las em uma imagem de maior resolução. Para isso você deverá compor esses pixels, organizando-os de maneira a obter uma matriz com o dobro do tamanho da resolução das imagens L , e obtendo a imagem de alta resolução estimada \hat{H} . Por exemplo:

$$L_1 = \begin{bmatrix} 100 & 101 \\ 110 & 111 \end{bmatrix}, L_2 = \begin{bmatrix} 200 & 201 \\ 210 & 211 \end{bmatrix}, L_3 = \begin{bmatrix} 300 & 301 \\ 310 & 311 \end{bmatrix}, L_4 = \begin{bmatrix} 400 & 401 \\ 410 & 411 \end{bmatrix}$$
$$\hat{H} = \begin{bmatrix} 100 & 300 & 101 & 301 \\ 200 & 400 & 201 & 401 \\ 110 & 310 & 111 & 311 \\ 210 & 410 & 211 & 411 \end{bmatrix}$$

Esse método é bastante simples, mas a depender do caso pode funcionar muito bem. Note que assumimos sempre que a resolução de H será o dobro da resolução das imagens L , e também que toda imagem L terá a mesma resolução.

Como referência, veja três recortes de uma imagem de baixa resolução, a reconstrução de super-resolução, e a imagem original em alta resolução.



Comparação com imagem de referência

Após gerar a imagem \hat{H} , comparar com a imagem H utilizando o RMSE (raiz do erro médio quadrático), e exibir com 4 casas decimais:

$$RMSE = \sqrt{\frac{1}{NM} \sum_i \sum_j (H(i, j) - \hat{H}(i, j))^2},$$

onde $N \times M$ é a resolução da imagem H .

Qualidade da imagem gerada

O método sugerido nesse trabalho é bastante simples, e será exigida ao menos a implementação desse método. Mas você poderá ainda adicionar outros métodos de processamento ou então estratégias de composição de pixel alternativas, para tentar melhorar os resultados.

Entrada e saída

Exemplo de entrada: imagens de baixa resolução `boat1.png`, `boat2.png`, `boat3.png`, `boat4.png`, de alta resolução `boathigh.png` e método de realce 2, equalização de histograma usando função conjunta. Note que o último parâmetro é ignorado pois é usado só no método de realce 3.

```
map
maphigh
2
1
```

Exemplos de saída: apenas o valor do RMSE em formato float

```
10.1864
```

Submissão e instruções

No sistema Run.Codes deve incluir apenas o arquivo `.py`

1. **É obrigatório comentar seu código.** Como cabeçalho insira seu nome, número USP, código da disciplina, ano/semestre e o título do trabalho. Haverá desconto na nota caso esse cabeçalho esteja faltando, além de descontos para falta de comentários.
2. **É obrigatório organizar seu código em funções.** Utilize uma função para realizar a equalização do histograma, e uma para realizar a superresolução. Outras auxiliares também podem ser criadas.