

Processamento Digital de Imagens

Métricas de Qualidade

Agenda

- Métricas de Qualidade

Métricas de Qualidade

- Métrica de qualidade é um assunto complicado
- Diversos autores utilizam diversas técnicas para quantificar a qualidade dos seus algoritmos
- As principais técnicas tratam das imagens após o pré-processamento (filtragem)
 - Etapa na qual entra uma imagem e sai uma imagem
 - Geralmente associadas a eliminação de ruídos ou aumento da qualidade

Métricas de Qualidade

- A maneira mais adequada para qualificar como bom ou ruim (na função para o qual foi criado) é por meio de métricas
- Algoritmos de classificação e segmentação avaliam por meio de
 - Falsos positivos
 - Falsos negativos
 - Taxa de acertos (%)
- Algoritmos de filtragem utilizam diversas métricas

Métricas de Qualidade

- Algumas métricas amplamente usadas são
 - MSE – Mean Square Error
 - RMSE – Root-Mean Square Error
 - PSNR – Peak Signal to Noise Error
 - SSIM – Structural Similarity
 - GMSD – Gradient Magnitude Similarity Deviation
- Devemos implementar métricas de qualidade em FPGA?
 - A resposta é NÃO
 - Salvo métricas de erro – parte do próprio algoritmo
 - Pode ser critério de parada de um processamento iterativo

Métricas de Qualidade

- É útil utilizar métricas para
 - Comparar algoritmos
 - Comparar diferentes implementações do mesmo algoritmo
 - Comparar algoritmos com computação aproximada implementada
 - Comparar algoritmos em SW e em HW
- Cada métrica indicará seu resultado avaliando alguma característica da imagem
 - Estrutura espacial, distorção espectral, semelhança (percentual)
 - A medida pode ser %, dB, ângulo, etc...

Métricas de Qualidade - MSE

- O MSE é uma técnica utilizada para identificar distorções espaciais (estruturas - simetria) da imagem após processamento
 - Quanto mais próximo do zero, mais iguais as imagens são

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(x, y) - g(x, y)]^2$$

- M e N são as dimensões das imagens
 - $f(x,y)$ e $g(x,y)$ são as imagens de entrada (f) e saída (g)

```
def mse(imageA, imageB):  
    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)  
    err /= float(imageA.shape[0] * imageA.shape[1])  
  
    return err
```

Métricas de Qualidade - RMSE

- O RMSE é o MSE com a aplicação de uma raiz quadrada
 - Indica o erro padronizado e é geralmente mais utilizado que o MSE – quanto maior pior

$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(i,j) - g(i,j)]^2}$$

- M e N são as dimensões das imagens
 - f(x,y) e g(x,y) são as imagens de entrada (f) e saída (g)

```
def mse(imageA, imageB):  
    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)  
    err /= float(imageA.shape[0] * imageA.shape[1])  
    err = np.sqrt(err)  
    return err
```


Métricas de Qualidade - PSNR

- PSNR é a razão entre a potência máxima de um sinal e a potência máxima do sinal do ruído
 - Medido em dB
 - Quanto maior melhor (considerável bom ≥ 20 dB)

$$PSNR = 20 \log_{10} \frac{L^2 MN}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f(x, y) - g(x, y)]^2} = 20 \log_{10} \frac{L^2}{MSE}$$

- Boa para avaliar filtros de suavização
 - L indica o valor máximo do pixel (ex: 255)

```
def PSNR(original, compressed):  
    mse = np.mean((original - compressed) ** 2)  
    if(mse == 0): #indica que as imagens são totalmente iguais  
        return 100  
    max_pixel = 255.0  
    psnr = 20 * log10(max_pixel / sqrt(mse))  
    return psnr
```

Métricas de Qualidade - SSIM

- O SSIM é uma métrica baseada em recurso do Sistema Visual Humano (HVS) – Hue, Saturation or Intensity Value
 - O SSIM mede a semelhança entre duas imagens, sendo uma melhoria em relação a métodos como MSE e PSNR
- Basicamente, calcula a degradação da imagem para algum processo
 - Vai de -1 a 1, sendo 1 para imagens idênticas

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Métricas de Qualidade - SSIM

- O SSIM

- μ_x e μ_y são as médias de x e y, σ_x^2 e σ_y^2 são a variacão de x e y, σ_{xy} é a covariance de x e y, as restrições $c1 = (K_1L)^2$ e $c2 = (K_2L)^2$
- L = 255 (padrão), já k1 = 0,01 e k2 = 0,03 (padrão)

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

```
#python3 -m pip install SSIM-PIL
```

```
from SSIM_PIL import compare_ssim  
from PIL import Image
```

```
image1 = Image.open(path)  
image2 = Image.open(path)  
value = compare_ssim(image1, image2)  
print(value)
```

Métricas de Qualidade - GMSD

- O GMSD, assim como SSIM, é uma métrica para o HSV
- Avalia a distorção causada pelo processamento entre a imagem de referência e a imagem processada
 - Os m 's referem-se as imagens e c é uma constante > 0
 - Quanto maior o valor melhor a qualidade

The Gradient Magnitude Similarity, $GMS(j) = \frac{2m_r(j)m_d(j) + c}{m_r^2(j) + m_d^2(j) + c}$

$$GMSD = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (GMS(i, j) - GMSM)}$$

Métricas de Qualidade - GMSD

- O GMSD

```
#pip install sporco  
import sporco.metric as sm  
print(sm.gmsd(img_original, img_processed))
```

PS: SPORCO também oferece PSNR, MSE e RMSE