

Trabalho Prático 2 – Restauração de Fotos em Escala de Cinza

Objetivo

O objetivo do trabalho é melhorar a aparência de fotografias antigas esmaecidas e danificadas pela passagem do tempo utilizando a linguagem de programação C/C++ e as técnicas de equalização de histograma e filtragem bilateral de imagem.

Contexto

Alguns problemas comuns encontrados em fotografias antigas são: o esmaecimento das tonalidades da foto e a presença de ruído. Ambos problemas podem ser percebidos ao se analisar cuidadosamente a Figura 1.



Figura 1 - Imagem com presença de ruído e tonalidades "apagadas"

Como foi visto em sala de aula, filtros e outras técnicas de processamento digital de imagem podem ser utilizadas para corrigir pequenas imperfeições e realçar características das imagens. Por exemplo, para o caso da Figura 1, uma operação de **equalização do histograma** é suficiente para melhorar o contraste da imagem, tornando as tonalidades mais vívidas e as diferenças de iluminação mais claras, como pode ser verificado na Figura 2.



Figura 2 – Resultado da operação de equalização de histograma aplicada à Figura 1

Ao ampliar a imagem da Figura 2, obtemos a Figura 3, na qual é possível perceber mais claramente a presença de ruído. Nesse caso, a presença de ruído se dá não só por causa da idade da imagem, mas também pela compressão utilizada para seu armazenamento em meio digital. Através da utilização de técnicas mais avançadas de filtragem, é possível diminuir a quantidade de ruído observada sem perder os contornos de alto contraste, como mostrado na Figura 4.



Figura 3 – Parte da imagem da Figura 2 ampliada, com presença considerável de ruído mesmo em regiões de textura suave



Figura 4 – Parte da imagem da Figura 2 após o uso da técnica de filtragem bilateral para diminuição do ruído

Ao comparar as Figuras 3 e 4, nota-se que os contornos das faces e as dobras das vestimentas foram mantidos, o que não aconteceria caso fosse utilizado um filtro de média ou mediana, por exemplo. O filtro de imagem utilizado para obter o efeito mostrado na Figura 4 é chamado de **filtro bilateral**, que possui esse nome pelo fato de aplicar a máscara de filtragem levando em consideração a proximidade dos *pixels* tanto em termos de posição na imagem, quanto na similaridade entre suas cores. Assim como a operação de equalização de histograma, esse trabalho prático prevê que o filtro bilateral será implementado pelo aluno.

Entrada

Nessa atividade, serão utilizadas como entrada apenas imagens *bitmap* de três canais em escala de cinza (cada pixel é representado por 3 três canais com o mesmo valor, correspondendo a um de 256 níveis de cinza). Em complemento a essa especificação, serão fornecidas outras imagens de referência.

A aplicação de um filtro bilateral é semelhante às operações de filtragem realizadas em sala de aula; no entanto, o filtro bilateral possui pelo menos duas grandes diferenças com relação aos filtros implementados até então:

1. Ao invés de uma matriz fixa que define a máscara a ser aplicada à imagem, devem ser utilizadas funções para determinar a contribuição de cada “vizinho” para a cor do *pixel* processado.
2. Como principal característica do filtro bilateral, é necessário incluir também a *cor* dos *pixels* vizinhos como elemento de ponderação da sua contribuição na cor do *pixel* processado.

Assim um filtro bilateral que passa por todos os *pixels* de uma imagem pode ser representado pela expressão:

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(I_{\mathbf{p}} - I_{\mathbf{q}}) I_{\mathbf{q}}$$

Detalhando as variáveis da equação acima temos:

- I - imagem de entrada
- p - *pixel* que está sendo processado
- q - *pixels* contidos na janela de processamento
- $\|p - q\|$ - distância euclidiana entre os pixels p e q
- I_p - intensidade (i.e., valor de cinza) do *pixel* $p = (p_x, p_y) \in I$
- $BF[I]_p$ - intensidade do *pixel* p após o processamento.
- S – “janela” 9x9 de processamento centralizada no *pixel* a ser processado
- W_p é o fator de normalização:

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(I_{\mathbf{p}} - I_{\mathbf{q}})$$

- $G(x)$ é a função Gaussiana¹:

$$G_{\sigma}(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

- σ_s - fator de peso atribuído à proximidade entre os *pixels* da janela e o pixel processado
- σ_r - fator de peso atribuído à proximidade das cores dos *pixels* da janela e o *pixel* processado

O programa entregue deve receber via linha de comando o nome de uma imagem para ser carregada, e aplicar em sequência as operações de equalização do histograma e filtragem bilateral como descrito acima. Para o filtro bilateral devem ser utilizados os parâmetros $\sigma_s = 3$ e $\sigma_r = 8$, e uma janela de filtragem de **9x9** pixels a fim de produzir resultados consistentes para fins de avaliação.

Saída

A saída esperada para os programas desenvolvidos nesta atividade prática é a exibição lado a lado da imagem **antes e depois da aplicação das operações de processamento**. Além disso, **a imagem final gerada depois do processamento (bitmap de três canais em 256 tons de cinza) deverá ser salva na pasta do executável**.

Por exemplo, considerando a Figura 1 como entrada da atividade, o resultado esperado do processamento é próximo ao exibido na Figura 5:



Figura 5 - Resultado da aplicação das operações de equalização de histograma e filtragem bilateral à Figura 1

¹ Para mais informações sobre a função $\exp(x)$ veja <http://www.rocketmime.com/rockets/exp.html>

Qualquer dúvida da especificação do trabalho, pergunte ao professor ou estagiário em docência com antecedência.

Entrega

Até as 16:40 do dia 14/12/2017. O trabalho é individual e deverá ser enviado através do Moodle. A entrega consiste de um arquivo fonte C/C++ que utilize apenas as bibliotecas de OpenGL usadas nas aulas práticas.

O critério de similaridade dos níveis de cinza, entre as imagens (*bitmap*) geradas pelos programas e as imagens esperadas, será considerado para atribuição de nota ao trabalho:

90<% Similaridade<=100 => Nota 10

70<% Similaridade<=90 => Nota 8

40<% Similaridade<=70 => Nota 6

20<% Similaridade<=40 => Nota 4

Exemplo do cálculo de similaridade: A percentagem de similaridade entre imagens A e B, a seguir, seria: $(1 - [(10 + 10 + 0 + 10 + 55 + 0) / (2 \times 3 \times 255)]) \times 100 = 94,4\%$:

A		
10	100	30
0	200	255

B		
20	90	30
10	255	255

Não serão avaliados os códigos:

- Que não gerem a imagem de saída (bitmap de três canais em 256 tons de cinza), que não compilem com o GCC (compilador C usado nas aulas práticas), que usem bibliotecas adicionais, ou que apresentem travamento/*crash* com as entradas definidas nesse documento.
- Códigos similares terão sua nota penalizada em função da percentagem de similaridade. Código com 100% de plágio terá nota zero.

Referências

BMP file format

https://en.wikipedia.org/wiki/BMP_file_format

Filtragem bilateral em computação (Para os fins desse trabalho prático, os slides 1, 2 e 3 são os mais relevantes).

PARIS, Sylvain et al. A gentle introduction to bilateral filtering and its applications. In: ACM SIGGRAPH 2007 courses. ACM, 2007. p. 1. (Notas de aula disponíveis em: https://people.csail.mit.edu/sparis/bf_course/)