

## Trabalho 2

Implemente sua atividade sozinho sem compartilhar. Evite olhar código de seus colegas, ou buscar na Internet por código pronto.

### Transformação dos níveis de intensidade

Nesse trabalho você deverá codificar um programa capaz de realizar transformações de intensidade na imagem usando operações diversas. Você deverá implementar usando a linguagem: C/C++ com `OpenCV`. Leia com calma as instruções de cada etapa. Na seção 3 estão as instruções e formatos aceitáveis para submissão do código.

### 1 Tarefa

Arquivos: `transf.cpp`

O programa deverá:

1. **Receber os parâmetros:**
  - a) nome da imagem de entrada
  - b) nome da imagem de saída
  - c) opção
    - 1 – negativo,
    - 2 – contraste,
    - 3 – logaritmo,
    - 4 – potência.
  - d) 1º parâmetro: constante  $c$  se escolhida opção 3 (opcional),  $\gamma$  para a opção 4 (obrigatório para a opção 4)
  - e) 2º parâmetro: constante  $c$  se escolhida opção 4 (opcional).
2. **Carregar para a memória principal** a imagem de entrada;
3. **Alocar na memória principal** o espaço necessário para armazenar a imagem de saída;
4. **Realizar a transformação** dos pixels;
5. **Exibir uma janela** com a imagem de saída;
6. **Exibir os histogramas** das imagens de entrada e saída (após transformação), para comparação;
7. **Gravar no disco** a imagem de saída;

### 1.1 Operação Negativo

Inverte o intervalo de níveis de cinza da imagem.

$$T(r) = 255 - r$$

### 1.2 Operação Ajuste de Contraste

$$T(r) = (r - \min_r(f)) \left( \frac{255}{\max_r(f) - \min_r(f)} \right),$$

onde  $\max_r(f)$  é o maior valor de intensidade presente na imagem, e  $\min_r(f)$  é o menor valor de intensidade presente na imagem  $f$ .

### 1.3 Operação Logarítmica

Utilizada para realçar valores baixos da imagem (pixels escuros), é obtida pela equação:

$$T(r) = c \log(1 + |r|),$$

onde  $c$  pode ser passada pelo usuário (opcionalmente), ou calculada por padrão:

$$c = \frac{255}{\log(1 + R)},$$

sendo  $R$  o valor máximo de intensidade encontrado na imagem.

### 1.4 Operação de Potência

Utilizada para realçar valores altos da imagem (pixels claros), é obtida pela equação:

$$T(r) = cr^\gamma,$$

onde  $c$  pode ser passada pelo usuário (opcionalmente), ou definida por padrão como  $c = 1$ . Valores de  $\gamma$  comuns são entre 0.5 e 1.5.

### DICA DE IMPLEMENTAÇÃO

Não se esqueça de fazer as conversões adequadas para realizar as operações. Uma função útil é a `saturate_cast<>()`<sup>1</sup>, que pode ser usada na conversão para `uchar` antes de armazenar o valor na matriz da imagem na forma:

```
img.at<uchar>(x,y) = saturate_cast<uchar>(valor);
```

---

<sup>1</sup>ver documentação da função em <http://goo.gl/rLQBtc>

## 1.5 Análise de Histograma

Utilizado para analisar o conteúdo da imagem, será exibido utilizando 8 intervalos de 32 intensidades de cinza cada (ou seja o primeiro será 0 a 31, o segundo 32 a 63, e assim por diante até 255). Para cada intervalo montar um gráfico de barras usando caracteres ASCII, sendo que o intervalo de maior valor deverá ter 20 caracteres, e os demais seguir proporcionalmente. Por exemplo, para uma imagem com o seguinte conteúdo:

```
0, 10, 15, 35, 10, 10, 175, 200
0, 10, 15, 35, 10, 10, 175, 200
0, 10, 15, 35, 10, 10, 175, 200
0, 10, 15, 35, 10, 10, 175, 200
0, 10, 15, 35, 10, 10, 175, 200
0, 10, 15, 35, 10, 10, 175, 200
0, 10, 15, 35, 10, 10, 175, 200
0, 10, 15, 35, 10, 10, 175, 200
0, 10, 15, 35, 10, 10, 175, 200
```

O histograma será:

```
| *****
| **
|
|
|
|
|
| ****
|
```

## 2 Requisitos

Implementar as funções

- `g = iminvert(f)`, gera o negativo da imagem;
- `g = imcontrast(f)`, gera uma imagem com contraste ajustado;
- `g = imlog(f, c)`, que retorne a imagem  $g$  processada pela função logarítmica, a partir dos parâmetros: imagem de entrada  $f$  e constante  $c$ ;
- `g = imgamma(f, c, y)`, que retorne a imagem  $g$  processada pela operação de potência, a partir dos parâmetros: imagem de entrada  $f$ , constante  $c$  e  $\gamma = y$ .
- `imhistogram(f)`, apresenta na tela a frequência de cada tom de cinza presente na imagem, em forma de gráfico, conforme descrito anteriormente.

### 3 Instruções e Informações Importantes:

- Trabalho individual
- Limite para a entrega: 13/03

**Submissão** Utilizar o Tidia-AE (seção Atividades) para submeter um arquivo compactado, contendo uma pasta com o número USP do autor e a implementação separada em duas sub-pastas:

- OpenCV (incluir arquivos fonte `.cc`, `.cpp`, etc. e `Makefile`)
- O trabalho deverá compilar com `Makefile` para Linux conforme ensinado em aula

Na raiz, incluir um arquivo texto (`README`) com: o nome e número USP do autor do trabalho, e o número do trabalho (1, 2, 3, ...)

**Critérios** O projeto será avaliado levando em consideração:

1. Negativo
2. Operação logarítmica
3. Operação de potência
4. Histograma
5. Interface (mensagens de ajuda, erro), carregamento exibição e gravação da imagem de saída.

É **imprescindível** comentar o código, documentando cada função implementada, os trechos importantes, não esquecendo de inserir em cada arquivo fonte seu nome e número USP.

Dúvidas conceituais deverão ser colocadas nos horários de atendimento ou por e-mail com a estagiária PAE ou monitor com o assunto [PDI T2] **dúvida**, especificando o problema, e anexando o código quando for necessário.

A detecção de cópia de parte ou de todo código-fonte, de qualquer origem, implicará reprovação direta no trabalho. Partes do código cujas **ideias** foram desenvolvidas em colaboração com outro(s) aluno(s) devem ser devidamente documentadas em comentários no referido trecho. O que **NÃO** autoriza a cópia de trechos de código. Portanto, compartilhem ideias, soluções, modos de resolver o problema, mas **não o código**. Qualquer dúvida entrem em contato com o professor.