universidade de aveiro

Departamento de Eletrónica, Telecomunicações e Informática

Aula 2

Guião prático 1

Curso	[8240] MI em Engenharia de Computadores e Telemática
Disciplina	[41270] Visão por Computador
Ano letivo	2016/2017

Alunos	[68021] Gabriel Vieira
	[38569] Miguel Azevedo
Docente	António Neves

Aveiro, 28 de Setembro de 2016

Conteúdo

1 Introdução

Neste relatório, pretendemos explicar tudo o que foi feito na aula 2, em relação ao guião prático da mesma. Em cada secção, estará a explicação de cada exercício e os resultados dos mesmos, através de imagens.

2 Instalação do OpenCV

Para a realização dos exercícios propostos, foi necessária a instalação das bibliotecas do OpenCV 3. Para a sua instalação, foi necessário recorrer a alguns tutoriais, no qual abaixo se encontram alguns links de instalação:

- *opencv.org/downloads.html*
- *https://github.com/jayrambhia/Install-OpenCV*

3 Exercício 4

Neste exercício, o objectivo era carregar, modificar e guardar uma imagem. A imagem a carregar foi `lena.jpg` e esta sofreu uma modificação, na qual se decidiu transformá-la em tons de cinzento.

Para tal, usou-se a biblioteca cvtColor em que o primeiro argumento era a matriz que iria conter a imagem transformada, o segundo argumento o nome da imagem gerada da transformação e por fim como último argumento a macro correspondente à transformação.

Abaixo é mostrada a imagem resultante da transformação:

Figura 1: Imagem *lena.jpg* em tons de cinzento

4 Exercício 5

Neste exercício é prosoto fazer a adição de duas imagens usando o opencv. As imagens que foram adicionadas foi a `lena.jpg` e a do exercício anterior `Gray_Image.jpg`.

Para isso, foi necessário em primeiro compreender como se conseguia produzir a soma dessas duas imagens. Mas antes de começarmos a explicar o prodecimento, é importante referir que ambas as imagens têm de ter tamanho e formato igual.

Foi usado o operador *"linear blend"* que por sua vez usa um valor alfa. Este valor alfa vai ser o responsável por determinar o "grau da mistura"das duas imagens. Ou seja, neste caso, como temos uma imagem a cores e a mesma mas em tons de cinzento, podemos facilmente observar este fenómeno. A figura abaixo mostra a adição das duas imagens referidas anteriormente com um valor de alfa igual a 0.5:

Figura 2: Imagem resultante da adição das imagens

5 Exercício 6

A implementação deste exercício consiste na aplicação da função de transformação $g(x) = \alpha f(x) + \beta$ em que os parametros α e β estao associados ao contraste e brilho da imagem resultante. Abaixo encontra-se o resultado da aplicação desta transformação na imagem `lena.jpg` com os parâmetros $\alpha = 1.5$ e $\beta = 50$:

Figura 3: Imagem resultante da aplicação da função de transformação descrita.

6 Exercício 7

Este exercício consiste na implementação do exercício anterior usando métodos de *scanning* mais eficientes do que o anteriormente implementado. Foram implementadas duas novas versões do algorítmo desenvolvido no exercício anterior usando dois dos métodos descritos em [1]. Ambos os métodos usados pressupõem a utilização de uma *Lookup Table* que é inicializada com o resultado da aplicação da função de transformação em questão em todo o *color space* de um canal, sendo esta inicialização feita por aritmética de ponteiros. A diferença entre estes métodos de *scanning* reside na maneira como são manipuladas as matrises durante a aplicação da função, no primeiro método os acessos são feitos de forma explicita usando aritmética de ponteiros, no segundo é usada a função `LUT()` do OpenCV [2].
Na seguinte tabela podemos observar os tempos de execução das três implementações aplicando a transformação à imagem `lena.jpg` com os com os parâmetros $\alpha = 1.5$ e $\beta = 50$.

Implementação	T. Exe (ms)
Original	20.75
Lookup Table + arit. ponteiros	2.90
Lookup Table + `LUT()`	0.58

Estes resultados são facilmente explicados pela redução do número de operações aritméticas proporcionado pelo uso de *Lookup Tables*, no entanto, analizando os resultados obtidos utilizando a função `LUT()` para aplicação da transformação, pode concluir-se que o método de acesso à memória é igualmente relevante neste contexto.

Referências

[1] How to scan images, lookup tables and time measurement with OpenCV, `http://docs.opencv.org/2.4/doc/tutorials/core/how_to_scan_images/how_to_scan_images.html`

[2] Documentação do OpenCV, função textttLUT(), `http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html#lut`