

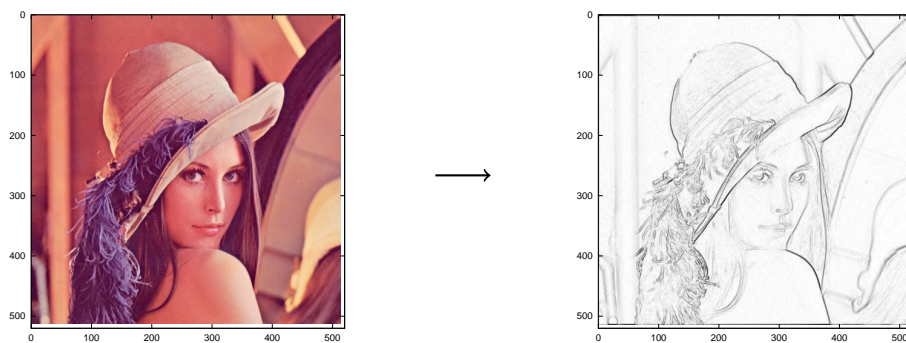
# Enunciado do Trabalho de Arquitectura de Computadores I

Licenciatura em Engenharia Informática

2021-2022

## 1 Objectivo

Pretende-se desenvolver um conjunto de funções em assembly RISC-V para obter os contornos de imagens a cores. Dado um ficheiro com uma imagem no formato RGB, o programa deverá gerar outro ficheiro de imagem, em tons de cinzento, com fundo branco e com traços escuros nos locais onde existem contornos na imagem original.

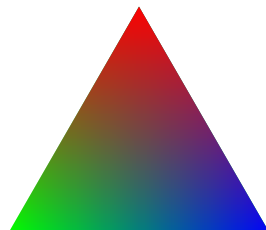


## 2 Teoria

### 2.1 Cores RGB

Embora a luz tenha uma infinidade de cores distintas, o sistema visual humano apenas é sensível a três: vermelho, verde e azul. Por este motivo, as imagens que se destinam a consumo humano usam apenas estas três cores.

A representação de cores em RGB refere-se à representação de uma cor somando as três componentes *Red*, *Green* e *Blue* com intensidades luminosas apropriadas.



Num computador, cada uma das três intensidades é representada por um número inteiro com  $N$  bits. Se  $N = 8$  bits (1 byte), então temos níveis de intensidade de 0 a 255 para cada componente de cor. Como existem  $256^3 = 16777216$  combinações diferentes, temos mais de 16 milhões de cores distintas.

## 2.2 Formato de imagem RGB

Neste trabalho, o ficheiro de imagem original contém uma imagem a cores no formato RGB, em que a intensidade de cada componente de cor é codificada por um byte com valores de 0 a 255. Assim, cada pixel é codificado por três bytes, um byte para cada uma das cores primárias.

Uma imagem completa consiste numa matriz em que cada elemento é um pixel de cor RGB. Uma imagem de dimensão  $m \times n$  corresponde à matriz

$$\begin{bmatrix} (R_{11}G_{11}B_{11}) & (R_{12}G_{12}B_{12}) & \cdots & (R_{1n}G_{1n}B_{1n}) \\ (R_{21}G_{21}B_{21}) & (R_{22}G_{22}B_{22}) & \cdots & (R_{2n}G_{2n}B_{2n}) \\ \vdots & \vdots & & \vdots \\ (R_{m1}G_{m1}B_{m1}) & (R_{m2}G_{m2}B_{m2}) & \cdots & (R_{mn}G_{mn}B_{mn}) \end{bmatrix}.$$

Os pixels da imagem anterior são guardados sequencialmente num ficheiro pela seguinte ordem<sup>1</sup>:

$$\left[ \underbrace{R_{11}G_{11}B_{11} \cdots R_{1n}G_{1n}B_{1n}}_{1^{\text{a}} \text{ linha}} \underbrace{R_{21}G_{21}B_{21} \cdots R_{2n}G_{2n}B_{2n}}_{2^{\text{a}} \text{ linha}} \cdots \underbrace{R_{m1}G_{m1}B_{m1} \cdots R_{mn}G_{mn}B_{mn}}_{m\text{-ésima linha}} \right].$$

O ficheiro RGB não contém informação acerca do tamanho da imagem, nem o número de bits usado para codificar cada cor. Essa informação é perdida na conversão. Para decodificar uma imagem neste formato é necessário conhecer as características da imagem original.

No terminal, o comando **convert**<sup>2</sup> permite converter entre diversos formatos de imagem. Alguns exemplos de conversão entre os formatos JPEG, RGB e GRAY (escala de cinzentos) são os seguintes:

```
convert imagem.jpg imagem.rgb          # JPG -> RGB
convert -size 512x512 -depth 8 imagem.rgb imagem.jpg # RGB -> JPG
convert -size 512x512 -depth 8 imagem.gray imagem.jpg # GRAY -> JPG
```

Os formatos RGB e GRAY não guardam as dimensões da imagem nem o número de bits de cada cor. Para converter a partir destes formatos é necessário passar os parâmetros **-size 512x512 -depth 8** a indicar o tamanho da imagem e o número de bits usados para representar cada componente de cor.

No formato RGB cada pixel é representado por 3 bytes. No formato monocromático GRAY cada pixel é codificado por um único byte com a intensidade 0 (preto) até 255 (branco).

<sup>1</sup>A ordenação em que os elementos de cada linha são guardados sequencialmente juntos uns aos outros chama-se *raw major*. A outra possibilidade seria *column major*. [https://en.wikipedia.org/wiki/Row\\_and\\_column-major\\_order](https://en.wikipedia.org/wiki/Row_and_column-major_order)

<sup>2</sup>O comando **convert** é disponibilizado pelo pacote **ImageMagick** em Linux.

## 2.3 Conversão RGB para tons de cinzento

A conversão RGB para tons de cinzento consiste em converter as três componentes RGB para apenas uma componente que codifica a intensidade luminosa total de um pixel.

Sabe-se que a retina do olho tem sensibilidades diferentes para cores diferentes, sendo mais sensível ao verde e menos ao azul. Portanto, se pretendermos que a imagem em tons de cinzento reflecta estas diferenças de sensibilidade, as cores vermelho, verde e azul têm de ter pesos diferentes na intensidade luminosa total de um pixel. A fórmula seguinte é um dos vários *standards* neste tipo de conversão:

$$I = 0.30R + 0.59G + 0.11B.$$

Cada pixel da imagem em tons de cinzento é obtido pela aplicação da fórmula anterior ao pixel correspondente da imagem RGB.

A figura 1(b) mostra a imagem que se obtém com esta conversão.

## 2.4 Operadores Sobel

Um operador Sobel permite medir variações na intensidade dos tons de cinzento numa certa direcção.

Um operador Sobel consiste numa matriz de dimensão  $3 \times 3$  que é convolvida com a imagem original. Para as direcções horizontal e vertical temos os operadores

$$S_h = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{e} \quad S_v = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Estes operadores vão detectar variações na intensidade de cor nas direcções horizontal e vertical, respectivamente.

A aplicação destes operadores é efectuada por uma operação chamada *convolução*. Por exemplo, supondo que a matriz  $A$  é a imagem original em escala de cinzentos, a convolução de  $A$  com o operador  $S_h$  produz a imagem

$$B = S_h * A,$$

onde o símbolo  $*$  denota a operação de convolução (não é uma multiplicação).

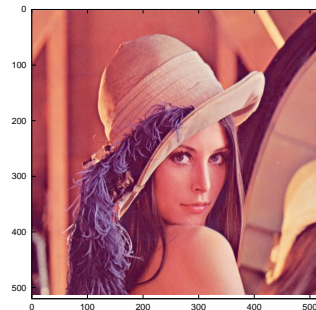
A convolução é calculada da seguinte maneira: percorre-se cada posição  $(i, j)$  da imagem  $B$ , e calcula-se o valor nessa posição com a fórmula

$$B(i, j) = \sum_{p \in \{-1, 0, 1\}} \sum_{q \in \{-1, 0, 1\}} A(i + p, j + q) S_h(2 - p, 2 - q).$$

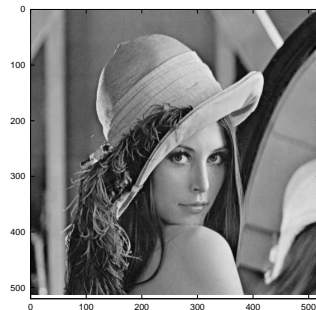
Esta operação consiste essencialmente em sobrepor a matriz  $S_h$  centrada na componente  $(i, j)$  da matriz  $A$ , fazer todos os produtos de elementos de  $S_h$  com os respectivos de  $A$ , e somar tudo obtendo-se no final  $B(i, j)$ . Este procedimento é repetido para calcular todos os elementos de  $B$ .

A convolução não está bem definida junto às margens da imagem  $A$ . Nessas regiões a solução fica ao critério do aluno.

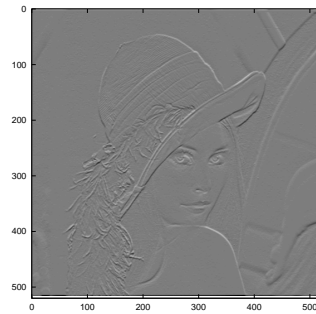
As figuras 1(c) e 1(d) mostram o resultado de aplicação dos operadores Sobel. Nestas imagens, o cinzento corresponde ao valor zero, enquanto as tonalidades claras e escuras correspondem a valores negativos e positivos.



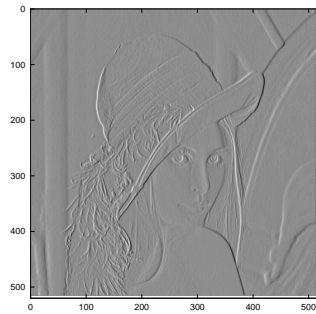
(a) Imagem “Lena” original, tamanho  $512 \times 512$ .



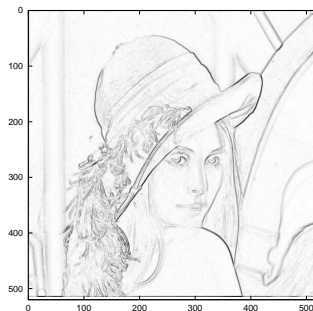
(b) Imagem convertida para escala de cinzentos.



(c) Imagem filtrada pelo operador Sobel ver-



(d) Imagem filtrada pelo operador Sobel horizontal.



(e) Contornos obtidos combinando os resultados dos operadores Sobel horizontal e vertical.

Figure 1: Imagem de teste “Lena” e passos intermédios para a detecção de contornos.

## 2.5 Contornos

Os contornos são obtidos combinando os resultados da aplicação dos operadores Sobel  $S_h$  e  $S_v$  à imagem  $A$ . Suponha-se que se obteve as matrizes

$$B_h = \frac{1}{4}|S_h * A| \quad \text{e} \quad B_v = \frac{1}{4}|S_v * A|,$$

onde as barras  $|\cdot|$  denotam o valor absoluto aplicado a cada elemento da matriz. Então a matriz  $C$  com os contornos pode ser obtida somando as matrizes  $B_h$  e  $B_v$  anteriores:

$$C = \frac{B_h + B_v}{2}.$$

A imagem que se obtém tem intensidade baixa (cor escura) quando não há variações de contorno e alta (cor clara) nos contornos. Para obter um resultado como o da figura 1(e) é necessário calcular a imagem complementar

$$D(i, j) = 255 - C(i, j).$$

A figura 1(e) mostra o resultado final obtido.

## 3 Implementação

A implementação da detecção de contornos em assembly RISC-V deve ser estruturada em várias funções separadas. As funções a desenvolver são as seguintes:

- **read\_rgb\_image** lê um ficheiro com uma imagem no formato RGB para um array em memória. A função tem como parâmetros uma string com o nome do ficheiro a ler e o endereço de um buffer onde a imagem deverá ser escrita.
- **write\_gray\_image** escreve uma imagem em formato GRAY num ficheiro. A função tem como parâmetros o nome de um ficheiro, um buffer com a imagem e o comprimento do buffer.
- **rgb\_to\_gray** converte uma imagem a cores RGB para uma imagem em tons de cinzento GRAY. A função tem como parâmetros um buffer com a imagem RGB, um buffer onde deve ser colocada a imagem em formato GRAY e o tamanho.
- **convolution** calcula a convolução de uma imagem  $A$  com um operador Sobel (matriz  $3 \times 3$ ) e coloca o resultado numa matriz  $B$ . A função tem como parâmetros um buffer com a matriz  $A$ , um buffer com um dos operadores Sobel e um buffer que vai conter a imagem filtrada  $B$ . A função pode assumir internamente uma dimensão de imagem fixa ou ter a dimensão como parâmetros.
- **contour** calcula a imagem final combinando as duas imagens convolvidas. A função tem como parâmetros dois buffers com as imagens a combinar e um buffer que vai conter o resultado.

As funções podem assumir internamente um tamanho de imagem fixo, ou receber o tamanho nos parâmetros.

- Comece com imagens de muito baixa dimensão para testar o código, por exemplo  $5 \times 5$ . Quando funcionar correctamente numa imagem pequena, experimente com imagens maiores.
- Comente o código como no seguinte exemplo:

```
#####
# Funcao: media
# Descricao: Esta funcao calcula a media de 2 numeros.
# Argumentos:
#   a0 - primeiro numero
#   a1 - segundo numero
# Retorna:
#   a0 - media
#####
media:
    add a0, a0, a1
    srai a0, a0, 1
    ret
```

- Escreva o relatório em  $\text{\LaTeX}$ . Entregue apenas o PDF final.

## 4 Submissão do trabalho

- O trabalho deverá ser realizado em **grupos de 2 alunos**.
- O trabalho deve ser submetido no moodle num único ficheiro comprimido (rar, zip, tgz, etc) com nome formado pelos números dos alunos por ordem, por exemplo 11111-22222.zip. Apenas um dos alunos do grupo deve submeter.
- O trabalho é longo. Não deixe para a última semana...
- Qualquer situação de fraude implica a reprovação à unidade curricular e será reportada para instauração de procedimento disciplinar, conforme estipulado no regulamento académico.

Bom trabalho! :)

Miguel Barão