

## Aula Prática 2

Docente: Miguel Tavares Coimbra

### 1. Espaço de cor RGB

- Crie uma função que permita visualizar individualmente as componentes R, G e B de uma imagem.
- Sugestões:
  - i. Crie 3 novas imagens usando objectos do tipo *ImagePanel* (ver código de apoio).
  - ii. Crie 3 botões/funções diferentes, um por cada componente do espaço RGB, que actuem sobre a imagem visualizada.

### 2. Espaço de cor HSI

- Crie uma função que permita visualizar individualmente as componentes H, S e I de uma imagem.
- Sugestões:
  - i. Crie 3 novas imagens usando objectos do tipo *ImagePanel* (ver código de apoio).
  - ii. Crie 3 botões/funções diferentes, um por cada componente do espaço HSI, que actuem sobre a imagem visualizada.

$$\text{iii. } H = \begin{cases} \theta & \leftarrow B \leq G \\ 2\pi - \theta & \leftarrow B > G \end{cases} \quad \theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{\left[ (R-G)^2 + (R-B)(G-B) \right]^{1/2}} \right\}$$

$$\text{iv. } S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)]$$

$$\text{v. } I = \frac{1}{3}(R+G+B)$$

- vi. Há valores em que o H e o S estão indefinidos. Porquê?



RGB



H



S



I

### 3. Pseudocor

- Crie uma função que aplica um algoritmo de pseudocor a uma imagem a preto-e-branco. Pode usar a técnica que quiser (intensity slicing, intensity to colour,...).



### 4. Histogramas.

- Crie uma função que calcule o histograma da *Intensity (HSI)* de uma imagem. Este deverá ser visualizado numa nova janela (*Frame*).
- Sugestões:
  - i.  $I = (R+G+B)/3$
  - ii. Trabalhe a função *paint* (*public void paint(Graphics g)*) da nova janela, que deverá desenhar um histograma baseado numa variável interna do tipo *int[256]*.
  - iii. Classe *java.awt.Graphics*: *drawLine*, *drawRect*, etc.
- [Opcional] Crie uma função que calcule os três histogramas de cor (espaço RGB) de uma imagem. Estes deverão ser visualizados numa única janela.

