

# EJERCICIOS. TALLER DE PROCESAMIENTO DE IMÁGENES. VERANO DE LA COMPUTACIÓN 2013

OSCAR DALMAU CEDEÑO  
JOSÉ LUIS MARROQUÍN

## 1. EJERCICIOS

- 1.- Implementar un *código en C* usando las librerías *shell* y *procim* que permita cargar y visualizar una imagen, por ejemplo, la imagen *manzana.bmp*.
- 2.- Sean  $n_f$  y  $n_c$  el número de filas y columnas de una imagen
  - a) Implementa una función que permita construir la imagen  $I$  de dimensiones  $n_f \times n_c$  de modo que

$$I[f][c] = \frac{2\pi a f}{n_f} + \frac{2\pi b c}{n_c} \quad (1)$$

donde  $a$  y  $b$  son parámetros y  $f$ ,  $c$  son el número de fila y columna del pixel respectivamente.

- b) Implementa una función que permita obtener el coseno de una imagen, i.e., void *cosenoimagen(float \*\* Origen, float \*\* Destino, int nf, int nc)*
  - c) Muestra la imagen  $J = \text{cosenoimagen}(I)$  donde la imagen  $I$  se obtiene según la ecuación (1). Muestra varias imágenes para diferentes parámetros  $a$  y  $b$
- 3.- Implementar una función en C que permita calcular la combinación lineal de 2 imágenes de acuerdo a la siguiente expresión (alpha matting)

$$T = \alpha I_1 + (1 - \alpha) I_2, \quad \alpha \in [0, 1] \quad (2)$$

- a) Carga 2 imágenes (por ejemplo: *caballo.bmp* y *manzana.bmp*) y muestra el resultado de la combinación lineal para  $\alpha = 0.2, 0.4, 0.6, 0.8$  en una misma ventana.
- 4.- Carga la imagen *manzana.bmp*,  $I$ . Usando las funciones del *procim.cp* calcula y muestra
  - a) La derivada gaussiana en  $x$  y en  $y$ , denotadas por  $I_x$  e  $I_y$  respectivamente.
  - b) La magnitud del gradiente,  $\|\nabla I\| = \sqrt{I_x^2 + I_y^2}$
  - c) Implementa las siguientes transformaciones

$$F(I) = \frac{\gamma}{\gamma + \max(|I_x|, |I_y|)} \quad (3)$$

$$F(I) = \exp(-\gamma \|\nabla I\|^2) \quad (4)$$

donde  $\gamma > 0$  es un parámetro. Muestras resultados de las transformaciones para algunos parámetros  $\gamma$ .

5.- Realizar los ejercicios anteriores usando el CALIMAN

## 2. CATÁLOGO DE FUNCIONES

### 2.1. Manejo de memoria (procim).

```
float *vectorf(int i0,int i1)
float **matrix(int nrl,int nrh,int ncl,int nch)
void free_vector(float *v,int i0, int i1)
void free_matrix(float **m,int nrl,int nrh,int ncl,int nch)
```

### 2.2. Suavizado de Imágenes (procim).

```
void SuavizaGaussiana(float **dat,float **res,float sigma,int nr,int nc)
void SuavizaCaja(float **dat,float **res,int tam,int nr,int nc)
```

### 2.3. Derivadas (procim).

```
void Dxb(float **f,float **g,int nr,int nc)
void Dyb(float **f,float **g,int nr,int nc)
void DerivaGaussianaX(float **dat,float **res,float sigma,int nr,int nc)
void DerivaGaussianaY(float **dat,float **res,float sigma,int nr,int nc)
void MagnitudGradiente(float **dat,float **res,float sigma,int nr,int nc)
void DerivaGaussiana2X(float **dat,float **res,float sigma,int nr,int nc)
void DerivaGaussiana2Y(float **dat,float **res,float sigma,int nr,int nc)
```

### 2.4. Lectura de imágenes (shell).

```
void DimensionesBmp(char *fname,int *pnr,int *pnc)
void LeeBmp(float **M,char *fname)
```

### 2.5. Muestra imágenes (procim).

```
void GraficaImagen(float **f,int i0,int j0,float fact,float fmin,float fmax)
```

### 2.6. Otras salidas (shell).

```
W.PlotBitmap(float **f,int nr,int nc,int x0,int y0,float zoom)
W.line(int x0,int y0,int x1,int y1)
W.printf(int Px,int Py,char *fmt)
```