

Boletín de ejercicios de Visión Artificial

Grado de Ing. Informática

Curso 14/15

Las implementaciones propuestas se realizarán en Matlab (o herramientas similares: Octave). Exceptuando las operaciones básicas de entrada/salida/visualización de imagen (imread, imwrite, imshow...) no se pueden utilizar funciones ya existentes de las librerías de tratamiento de imágenes ni otras relacionadas como convoluciones etc.

1. Histogramas

Implementar el algoritmo de compresión/estiramiento de histograma, que permita introducir los nuevos límites inferior y superior.

```
function outputImage = histStretch(inputImage,minValue,maxValue)
```

2. Interpolación

Implementar una función que permita realizar un zoom 2x a una imagen. Se implementarán dos métodos de interpolación: interpolación por el vecino más cercano e interpolación bilineal.

```
function outputImage = zoomIn2(inputImage,mode)
```

donde mode='bilinear' | 'neighbor'

3. Filtrado espacial: Realce y Suavizado

Implementar el algoritmo High Boost mediante una función que permita especificar el tamaño y el factor de amplificación A.

```
function outputImage = highBoost(inputImage,filterSize,A);
```

Implementar el filtro de orden de medianas. La función permitirá establecer el tamaño del filtro.

```
function outputImage = medianFilter(inputImage,filterSize)
```

4. Operadores morfológicos

Implementar los operadores morfológicos de erosión, dilatación, apertura y cierre para imágenes binarias. Ambas funciones deben permitir especificar el tamaño del elemento estructurante y su forma (cuadrada o cruz).

```
function outputImage = erode(inputImage, strElType, strElSize)
function outputImage = dilate(inputImage, strElType, strElSize)
function outputImage = opening(inputImage, strElType, strElSize)
function outputImage = closing(inputImage, strElType, strElSize)
```

donde strElType = 'square' | 'cross'.

5. Bordes y Esquinas

5.1. Canny

Implementar el detector de bordes de **Canny** mediante una función que permita especificar los umbrales a utilizar en el proceso de histéresis y el valor de σ en el suavizado gaussiano.

```
function outputImage = edgeCanny(inputImage , sigma , tlow , thigh)
```

5.2. Esquinas

Implementar el detector de esquinas basado en **SUSAN**. La función permitirá establecer el radio de la máscara circular (r) y el umbral de diferencia de luminosidad admitida respecto al núcleo de la máscara (t)

```
function outputImage = cornerSusan(inputImage , r , t)
```