

IEE239 - Procesamiento de Señales e Imágenes Digitales

Laboratorio 4 - Guía Práctica

Primer Semestre 2017

Martes, 6 de junio del 2017

Horario 07M2

- Duración: 2 horas, 30 minutos.
- Está permitido el uso de material adicional.
- La evaluación es **estrictamente** personal.
- **Está terminantemente prohibido copiar código externo (ejemplos de clase, material en línea, etc.)**

1. (*3 puntos*) El ángulo de los gradientes de una imagen es útil para caracterizar su textura. Para demostrar su utilidad, se va a buscar de entre tres imágenes la que más se parece a una imagen base. Para ello se va a usar un histograma del ángulo de los gradientes.
 - a. Cargar la imagen `ladrillos.jpg`¹ (convertirla a `double` y normalizarla dividiendo entre 255) y calcular las derivadas de primer orden $df(x,y)/dx$, $df(x,y)/dy$ de la imagen. Utilizar la función `conv2` con la opción '`same`' (revisar la documentación de la función) y los siguientes filtros espaciales:

$$s_x(x, y) = \begin{pmatrix} 1 & 2 & 1 \\ 0 & \uparrow & 0 \\ -1 & -2 & -1 \end{pmatrix}, s_y(x, y) = \begin{pmatrix} 1 & 0 & -1 \\ 2 & \uparrow & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

Describir gráficamente la imagen y las derivadas de primer orden (usar la función `subplot`). Indicar cual de las dos máscaras $s_x(x, y)$ o $s_y(x, y)$ permiten resaltar los bordes verticales de la imagen.

- b. Calcular el ángulo de las gradientes usando la función `atan2` (revisar la documentación de la función). Determinar el histograma **normalizado** h_0 de la imagen de ángulos obtenidos A . Para ello, primero generar el vector `xbin = linspace(A_{min} , A_{max} , 128)` para generar 128 bins uniformemente espaciados en el rango de valores de ángulos encontrados. Segundo, normalizar la imagen de ángulos A al rango $[0, 1]$. Finalmente, calcular el histograma de A usando la función `imhist(..., 'nbins')` para `nbins = 128`, y normalizarlo (dividir entre la cantidad de elementos de A). Graficar el histograma de ángulos usando la función `bar(xbin, ...)`. Observar los valores predominantes en el histograma y discutir si esto concuerda con lo que se observa en la imagen.
- c. Cargar el archivo `histogramas.mat`². En este archivo está el histograma normalizado de los ángulos del gradiente para las tres imágenes mostradas en la Figura 1 (los histogramas

¹La imagen `.jpg` está almacenada en la carpeta 'laboratorio/lab04'

²El archivo `.mat` está almacenado en la carpeta 'laboratorio/lab04'

están en el mismo orden que las imágenes mostradas). Graficar el histograma h_0 que calculó en el inciso anterior y cada uno de los histogramas cargados (h_1 , h_2 y h_3) usando la función `subplot`. Calcular la diferencia entre histogramas a partir de su distancia Euclidiana

$$d_i = ||h_0 - h_i||_2 = \sqrt{\sum_k |h_0(k) - h_i(k)|^2}, \quad (1)$$

donde $i = 1, 2, 3$, usando la función `norm(v)` donde $v = h_0 - h_i$ es el vector residual, diferencia entre el histograma de la imagen `ladrillos.jpg` y el histograma de una de las imágenes de la Figura 1. Indicar en los comentarios cuál de los histogramas cargados es más parecido (esto es, tiene la menor distancia al histograma h_0) a la imagen `ladrillos.jpg` y discutir cuál podría ser el motivo.

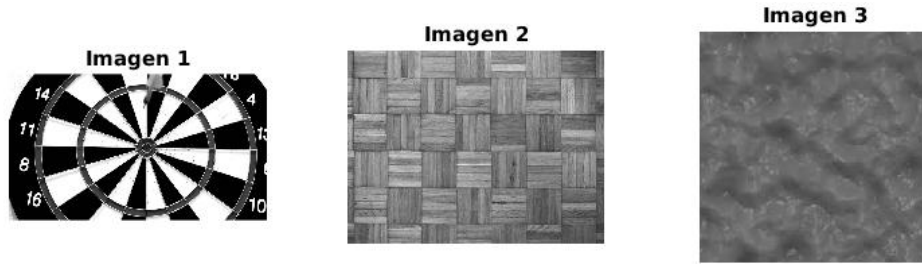


Figura 1: imágenes para comparar con la imagen `ladrillos.jpg`.

2. (4 puntos) En ocasiones se quiere realizar la alineación de dos imágenes, donde una de ellas I_2 está desalineada con respecto a una imagen base I_1 debido al efecto de alguna transformación espacial. Para realizar la alineación, es necesario estimar los parámetros ξ que definen la transformación $T(I, \xi)$. Por ejemplo, si la transformación es una rotación $T_r(I, \alpha)$ tendrá asociada un ángulo α mientras que si es una transformación por shear vertical $T_s(I, s_v)$ tendrá asociada el parámetro s_v , como se muestra en la Figura 2.



Figura 2: image ejemplo I (izquierda) y transformaciones de rotación $T_r(I, \alpha)$ (centro) y *shear* vertical $T_s(I, s_v)$ (derecha).

Un método para estimar los parámetros de una transformación es por búsqueda exhaustiva en la cual se prueban varios parámetros $\xi_i, i = 1, \dots, m$ y se calcula la diferencia entre imagen original y transformada como:

$$\text{error}_i = \sqrt{\sum_x \sum_y |T(I_1(u, v), \xi_i) - I_2(x, y)|^2}, \quad (2)$$

para luego elegir el parámetro ξ_i que produzca la menor diferencia entre imágenes, es decir, la transformación $T(I_1, \xi_i)$ que es más cercana a I_2 . Para ello primero se vectorizan las imágenes y luego se calcula la distancia Euclidiana entre las imágenes usando la función `norm(v)` cuya entrada es el vector v de diferencias entre las imágenes vectorizadas.

- a. (1 punto) Considerar las imágenes `parque1.jpg`³ como I_1 (la imagen base) y `parque2.jpg`⁴ como I_2 (la imagen corrompida con una rotación y un *shear* vertical). El objetivo es estimar los valores α y s_y correspondientes al ángulo de rotación y valor del *shear* vertical. Cargar ambas imágenes usando `imread` (convertirlas a `double` y normalizarlas dividiendo entre 255) y rotar I_1 en ángulos de -6° , -4° , -2° , 2° , 4° y 6° usando la función `imrotate(..., 'crop')` (revisar la documentación). Describir gráficamente las 6 transformaciones de la imagen base usando `subplot`.
 - b. (1 punto) Calcular la diferencia entre cada una de las imágenes rotadas e I_2 usando la función `norm` descrita previamente. Indicar en los comentarios la menor de estas normas y el ángulo de rotación α_{min} que corresponde a esta norma. Graficar en una nueva figura las imágenes I_1 , I_2 , y la imagen I_3 que corresponde a I_1 rotada con el valor de α_{min} hallado.
 - c. (1 punto) Ahora, aplicar valores de *shear* de -0.03 , 0.01 , 0.03 y 0.05 a la imagen I_3 usando la función `imshear`⁵ disponible en la intranet. Esta función toma como primer argumento la imagen a la que se le aplicará el *shear* vertical y como segundo argumento el valor del *shear*. Realizar la misma comparación con I_2 usando usando la función `norm`. Indicar en los comentarios la menor de estas normas y el valor de *shear* s_v que corresponde a esta norma.
 - d. (1 punto) Usando los valores que se estimaron para α_{min} y s_v indicar como se podría corregir la imagen corrompida I_2 para que esté alineada con I_1 . Graficar la imagen corregida I_4 y comprobar sus resultados comparándola con la imagen base I_1 .
3. (3 puntos) Muchas veces, el éxito de los algoritmos de procesamiento de imágenes y visión por computadora depende de la calidad inicial de las imágenes de entrada y, por ende, de la etapa de pre-procesamiento (filtrado, eliminación de ruido, resaltado de bordes, etc.). En esta pregunta, se verá como distintos filtros de pre-procesamiento afectan el desempeño de un algoritmo posterior, el cual identifica y cuenta el número de objetos en una imagen. Las imágenes a usar se muestran en la Figura 3. Se observa que la imagen posee 10 objetos. La segunda imagen (N1) ha sido corrompida con ruido sal y pimienta y la tercera imagen (N2) con ruido aditivo gaussiano. Sea I la imagen a pre-procesar, se va a comparar tres opciones de pre-procesamiento:
- i. Filtro de mediana, el cual deberá implementar usando la función `medfilt2` con máscara 3×3 y condiciones de frontera 'symmetric' (revisar la documentación).
 - ii. Filtro de media, el cual deberá implementar usando la función `conv2(..., 'same')` con el filtro espacial dado por

$$h_1(x, y) = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

³La imagen .jpg está almacenada en la carpeta 'laboratorio/lab04'

⁴La imagen .jpg está almacenada en la carpeta 'laboratorio/lab04'

⁵La función .m está almacenada en la carpeta 'laboratorio/lab04'

iii. Filtro de resaltador de bordes usando el Laplaciano, el cual deberá implementar usando

$$h_2(x, y) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

y luego calculando la imagen con bordes resaltados dado por $I_p(x, y) = I(x, y) - k \times (I(x, y) * h_2(x, y))$, y parámetro $k = 0.5$.

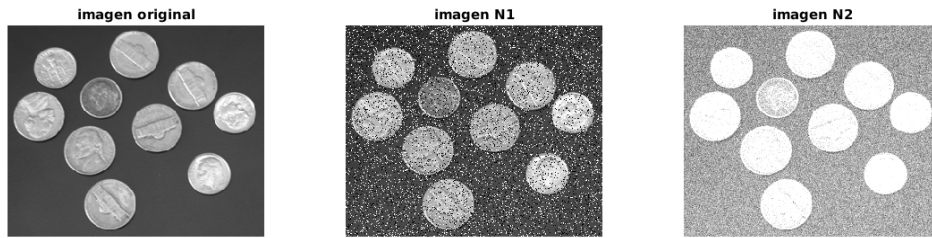


Figura 3: imágenes original (izquierda), y corrompidas con ruido sal y pimienta (centro) y ruido gaussiano (derecha).

- a. (1 punto) Cargar la imagen `imageN1.jpg`⁶ corrompida con ruido sal y pimienta. Realizar los tres pre-procesamientos mencionados. Describir gráficamente las imágenes obtenidas usando la función `subplot`. Aplicar la función `contar_elementos`⁷ sobre la imagen N1 y cada una de las imágenes pre-procesadas. La rutina `contar_elementos(Im)` tiene como entrada la imagen `Im` y como salida el número de objetos detectados en dicha imagen. Discutir cuál método de pre-procesamiento mejoró más el desempeño del algoritmo de cuenta de objetos. ¿Coincide con lo que se espera teóricamente?
- b. (1 punto) Cargar la imagen `imageN2.jpg`⁸ corrompida con ruido aditivo gaussiano. Realizar los tres pre-procesamientos mencionados. Describir gráficamente las imágenes obtenidas usando la función `subplot`. Aplicar la función `contar_elementos` sobre la imagen N2 y cada una de las imágenes pre-procesadas. Discutir cuál método de preprocesoamiento mejoró más el desempeño del algoritmo de cuenta de objetos. ¿Coincide con lo que se espera teóricamente?
- c. (1 punto) Cargar nuevamente la imagen `imageN1.jpg` corrompida con ruido sal y pimienta. Realizar el pre-procesamiento con filtro de mediana con máscaras 5×5 , 7×7 , 9×9 y 11×11 . Describir gráficamente las imágenes obtenidas usando la función `subplot`. Aplicar la función `contar_elementos` sobre cada una de las imágenes pre-procesadas e indicar los valores de cantidad de objetos obtenidos en cada caso. De acuerdo a lo observado indicar las ventajas y desventajas observadas asociadas a un filtro de mediana.

⁶La imagen .jpg está almacenada en la carpeta 'laboratorio/lab04'

⁷La función .m está almacenada en la carpeta 'laboratorio/lab04'

⁸La imagen .jpg está almacenada en la carpeta 'laboratorio/lab04'