FICH, UNL - Departamento de Informática - Ingeniería Informática

Procesamiento Digital de Imágenes

Guía de Trabajos Prácticos 5

Procesamiento y filtrado en el dominio frecuencial

1 Objetivos

- Fijar los conceptos relacionados a la Transformada de Fourier bidimensional mediante su visualización como imagen.
- Comprender el proceso de filtrado en el dominio de la frecuencia.
- Aplicar filtros de acentuado en frecuencia para el mejoramiento de una imagen, combinándolos con otras técnicas de realce.
- Verificar la ventaja computacional de la rutina de FFT.

2 Conceptos e ideas a repasar

- ¿Qué son las imágenes base? ¿De qué tamaño serán éstas si la imagen a analizar es de MxN?
- ¿De qué tamaño es la TDF de una imagen de $M \times N$?
- ¿Cómo es la distribución de la energía en la TDF? ¿Dónde se localiza la información de bajas y de altas frecuencias?
- ¿Qué propiedades tiene la TDF respecto a rotaciones, traslaciones, simetría, etc.?
- ¿Qué información se puede encontrar en el módulo y cuál en la fase?
- La secuencia de operaciones para filtrado en el dominio de la frecuencia es: a) creación del filtro H, b) cálculo de TDF(f), c) producto elemento a elemento TDF(f).*H, d) cálculo de la TDF inversa, e) obtención de la parte real.
- ¿Recuerda cuando y cómo se manifiesta el fenómeno de Gibbs en las imágenes?
- La secuencia de operaciones para el filtrado homomórfico es: $f(x,y) \rightarrow \ln \rightarrow \mathfrak{F} \rightarrow H(u,v) \rightarrow \mathfrak{F}^{-1} \rightarrow \exp \rightarrow g(x,y)$

3 Trabajos Prácticos

Antes de comenzar, le recomendamos estudiar el funcionamiento de las siguientes funciones de openCV (`https://docs.opencv.org/`) y numpy (`https://numpy.org/`):

- `dst = cv.dft(src[, dst[, flags[, nonzeroRows]]])`
  versión numpy: `dst = np.fft.fft2(src)`
- `dst = numpy.fft.fftshift(x, axes=None)`
- `dst = cv.magnitude( x, y[, magnitude])`
- `dst = np.log(x)`
- `dst = numpy.fft.ifftshift(x, axes=None)`
- `dst = cv.idft(src[, dst[, flags[, nonzeroRows]]])`
  versión numpy: `dst = np.fft.fft2(src)`
- `val = cv.getOptimalDFTSize(vecsize)`
- `dst = cv.copyMakeBorder(src, top, bottom, left, right, borderType[, dst[, value]])`
- `dst = cv.normalize(src, dst[, alpha[, beta[, norm_type[, dtype[, mask]]]]])`
- `magnitude, angle = cv.cartToPolar(x, y[, magnitude[, angle[, angleInDegrees]]])`
- `x, y = cv.polarToCart(magnitude, angle[, x[, y[, angleInDegrees]]])`
- `dst = cv.mulSpectrums(a, b, flags[, c[, conjB]])`

* En `OpenCV.org` recomiendan el uso de `cv.dft()` y `cv.idft()` porque son más rápidas que las de Numpy, sin embargo, las de Numpy son más amigables.
¿Se anima a comprobarlo?

Ayuda: vea `t = timeit.default_timer()` de la biblioteca *timeit* (Python)

Ejercicio 1: Conceptos básicos de la TDF 2D

1. Construya imágenes binarias de: una línea horizontal, una línea vertical, un cuadrado centrado, un rectángulo centrado, un círculo.
   ¿Qué espera ver en las TDF de cada una de estás? ¿Cómo estima que estará distribuida la energía?

2. Utilice cada una de las imágenes anteriores para calcular la TDF y visualice.
   ¿Se cumplieron sus pronósticos respecto de sus definiciones?
   Varíe las dimensiones y localización de los objetos en estas imágenes y repita el análisis.

3. Construya una imagen de 512x512, que contenga una línea vertical blanca centrada y de un pixel de ancho sobre un fondo negro.
   Rote la imagen 20 grados y extraiga una sección de 256x256 de la imagen original y de la imagen rotada, de manera que las líneas tengan sus extremos en los bordes superior e inferior, sin márgenes.
   Visualice la TDF de ambas imágenes. Explique, utilizando argumentos intuitivos, por qué las magnitudes de Fourier aparecen como lo hacen en las imágenes, y a qué se deben las diferencias.

   Ayuda: utilice `dst = imutils.rotate(src, angle)` de la biblioteca *imutils* (`https://github.com/jrosebr1/imutils`).

4. Cargue diferentes imágenes y visualice la magnitud de la TDF. Infiera, a grandes rasgos, la correspondencia entre componentes frecuenciales y detalles de las imágenes.

Ejercicio 2: Importancia de la fase

1. Calcule la TDF de una imagen, obtenga magnitud y fase.
   Genere una imagen reconstruida sólo con la magnitud considerando fase cero y genere otra imagen reconstruida usando sólo fase de la imagen considerando magnitud unitaria.
   Visualice las imágenes y saque conclusiones sobre el aporte de ambas componentes a la reconstrucción de la imagen.

2. Reproduzca el experimento de Openheim utilizando las imágenes `puente.jpg` y `ferrari-c.png`. Visualice y comente los resultados.

Ejercicio 3: Filtros pasa-bajos y pasa-altos

1. Construya un filtro pasa-bajos ideal (círculo de altura 1 sobre una matriz de ceros). Cargue una imagen y fíltrela en el dominio de frecuencias, y recupere la imagen suavizada. Visualice las imágenes y compárelas.
   Repita el ejercicio para diferentes frecuencias de corte y compruebe la aparición del fenómeno de Gibbs.

2. Construya un filtro pasa-bajos tipo Butterworth utilizando la definición en frecuencia. Filtre una imagen, modificando la frecuencia de corte y comprobando el efecto sobre la imagen filtrada. Verifique el efecto del filtro respecto al fenómeno de Gibbs.

3. Define la función de transferencia $h(x, y)$ de un filtro gaussiano pasa-bajos de 35×35, luego obtenga la respuesta en frecuencia aplicando la TDF. Luego, escale ésta al tamaño de la imagen que va a filtrar. (Use `dst = cv.resize()`)
   Calcule y visualice la TDF de la imagen y el producto de transformadas, verificando la acción del filtro. Obtenga la imagen filtrada y compare con la imagen original.

4. Repita el ejercicio anterior pero ahora utilice un filtro gaussiano definido en frecuencia. Compare los resultados.

5. Repita los ejercicios del 1 al 4, para filtros pasa-altos.

Ejercicio 4: Filtrado homomórfico

Un uso extendido del filtrado homomórfico es la corrección de iluminación no uniforme en distintas zonas de la imagen, generalmente con alto contenido de información en la zona de bajo brillo. Por ejemplo, en filmaciones de cámaras de seguridad y fotos con luz de día con sol de frente. En imágenes de este tipo, el filtro homomórfico corrige el contraste en la zona de interés y acentúa los detalles simultáneamente.

1. Genere la función de transferencia H que caracteriza a un filtro homomórfico.

2. Aplique el proceso en las imágenes `'casilla.tif'` y `'reunion.tif'`, con valores apropiados de g_L, g_H, D_0 y orden (prueba y error en cada imagen...).

3. Verifique las bondades del método comparando el resultado anterior con la imagen que se obtiene al ecualizar la imagen original.
   Esta técnica suele ser eficaz combinada con alguna técnica de manipulación de histogramas, por ejemplo ecualización. Ecualice el resultado del filtrado y visualícelo junto a los demás.

Ejercicio 5: Filtros de acentuado en el dominio frecuencial

1. A partir de la definición de una máscara de filtrado pasa-altos en el dominio espacial, obtenga la función de transferencia correspondiente a un filtro de alta potencia según $H_{AP} = (A-1) + H_{PA}$, y a un filtro de énfasis de alta frecuencia según: $H_{EAF} = a + b\ H_{PA}$.

2. Elija apropiadamente los valores de los parámetros A, a y b y aplique los filtros a la imagen `'camaleon.tif'`, visualizando la imagen original junto a su TDF, y la imagen resultante con su TDF.

3. Compare la imagen de alta potencia con la que se obtiene al aplicar el filtro en el dominio espacial.

Ejercicio 6: Ejercicios y aplicaciones

1. El método de la transformada rápida de Fourier es una forma de cálculo simplificada de la TDF, fue diseñado utilizando las propiedades de la transformada y sólo se puede aplicar a imágenes cuyos tamaños sean potencia de 2 (es lo que permite simplificar más aun los cálculos).
   Evalúe el rendimiento de FFT (Fast Fourier Transform) respecto de la TDF.

   - Cargue una imagen y obtenga el tamaño óptimo (para filas y columnas) al cual debe llevar la imagen para el cálculo de la FFT.
   - A partir de la imagen original genere una de tamaño óptimo ($N_{opt} \times M_{opt}$) agregando ceros.
   - Genere una imagen cuyo tamaño en filas y columnas sea 1px menor al tamaño óptimo ($(N_{opt}-1) \times (M_{opt}-1)$) agregando ceros.
   - Calcule la TDF de las 3 imágenes, visualice las magnitudes de las TDFs y saque conclusiones, evalúe el tiempo de cómputo de cada una.
   - *¿Qué efecto numérico (objetivo) y a la vista (subjetivo) produce el agregado de ceros?*

2. Realice una aplicación de preprocesamiento para OCR.
   La imagen de entrada es un texto escaneado (puede utilizar `parrafo0.jpg` y `parrafo1.jpg`), usted debe identificar si el texto está rotado y de ser así, debe corregir la orientación del texto.

   (Opcional): busque otras imágenes en peores condiciones, o genere algunas utilizando escaners o cámaras, y agregue algunas funcionalidades al código que le permita realzar la imagen, utilizando todo lo que ya conoce.