Gutierrez Castillo, Oscar Joya Venegas, Jehosua Alan Nolasco Fuentes, Juan Carlos

# Practica 3: Convolución lineal y circular utilizando la DFT

# introducción

Se denomina convolución a una función, que de forma lineal y continua, transforma una señal de entrada en una nueva señal de salida.  La función de convolución se expresa por el símbolo (.\*) en Matlab.

La Transformada Discreta de Fourier (DFT, por sus siglas en inglés Discrete Fourier Transform) provee un método para transformar los datos muestreados en el dominio del tiempo a una expresión de estos datos en el dominio de la frecuencia. La inversa de la transformada reinvierte este proceso, convirtiendo los datos en el dominio de la frecuencia en datos en el dominio temporal. Estas transformaciones pueden ser aplicadas en una gran variedad de campos desde la geofísica hasta la astronomía, desde el análisis de señales de sonido hasta el análisis de la concentración de CO2 en la atmósfera. Algunos algoritmos de reconstrucción 3D utilizados en Tomografía Computarizada (CT) hacen uso también de la DFT[1].

La Transformada Rápida de Fourier(Fast Fourier Transform) es una herramienta fundamental en el procesado digital de señales. Su origen es relativamente reciente puesto que fueron J.W.Cooley y J.W Tukey, quienes hacia 1965 abordaron por primera vez el problema de la programación de un algoritmo para el cálculo de series complejas.

    Ante todo debe quedar claro que la FFT no es una nueva transformada sino que se trata de un algoritmo para el cálculo de la Transformada Discreta de Fourier (DFT). Su importancia radica en el hecho que elimina una gran parte de los cálculos repetitivos a que está sometida la DFT, por lo tanto se logra un cálculo más rápido. Además, la FFT generalmente permite una mayor precisión en el cálculo de la DFT disminuyendo los errores de redondeo.

# DESARROLLO

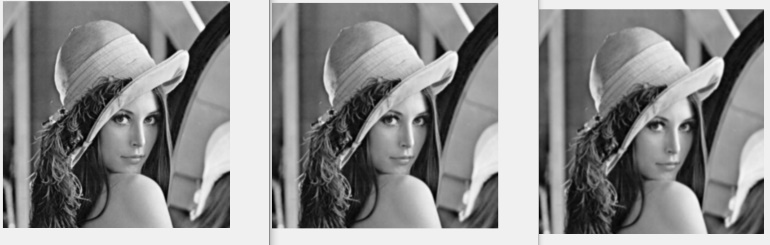
* + 1. Obtener la convolución lineal (comando MATLAB conv2 y argumentos ’full’, ’same’ y ’valid’ de la imagen con un filtro paso bajas (filtro de bloque). Usar 2 o 3 tamaños diferentes de filtros, por ejemplo: 7x7, 9x9 y 11x11. Desplegar las imágenes resultantes.
  1. ‘Full’ 7x7 ,9x9 y 11x11



* 1. ’same’ 7x7 ,9x9 y 11x11

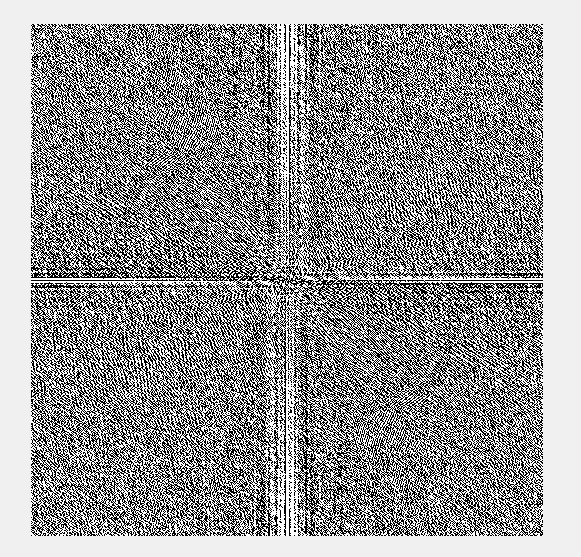


* 1. ’valid’ 7x7 ,9x9 y 11x11



|  |
| --- |
| Codigo:  i=im2double(imread('lenaTest1.jpg'));  i2= binomialCoef(11) .\* binomialCoef(11)';  i3=conv2(i,i2,'valid');  figure();  imshow(i3,[]);  %% Coeficiente binomial  function retval=binomialCoef(n)  l=[];  for i = 0:n  l=[l,binomial(n,i)];  end  retval = l;  end  % Binomial  function retval=binomial(n,k)  retval = fact(n) / (fact(n-k) \* fact(k));  end    function retval = fact (n)  if (n > 0)  retval=n\*fact(n-1);  else  retval=1;  end  end |

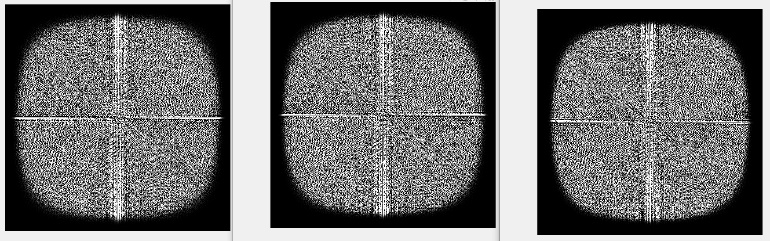
* + 1. Obtener la DFT de la imagen original y desplegarla de manera amplificada utilizando el logaritmo del módulo de la DFT para dicha amplificación. Cambiar el eje de coordenadas (comando MATLAB fftshift) y nuevamente amplificar



Codigo:

|  |
| --- |
| I=imread('lenaTest1.jpg');  I = fft2(I);  I=fftshift(I);  figure();  imshow(I) |

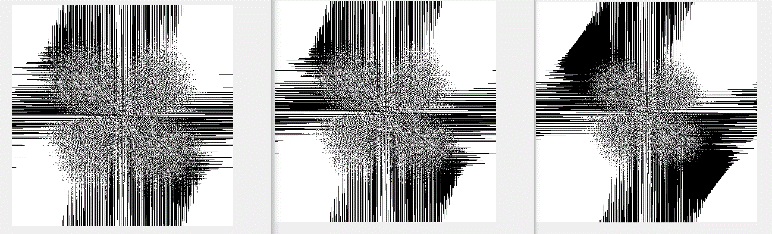
* + 1. Obtener la convolución circular (⊗) de la imagen con el filtro paso bajas a través de la DFT. Usar también diferentes tamaños de filtros. Desplegar las imágenes resultantes.
  1. 7x7 ,9x9 y 11x11



Codigo:

|  |
| --- |
| i=im2double(imread('lenaTest1.jpg'));  i2= binomialCoef(6) .\* binomialCoef(6)';    i3=imfilter(i,i2,'circular','conv');    I = fft2(i3);  I=fftshift(I);  figure();  imshow(I)  %% Coeficiente binomial  function retval=binomialCoef(n)  l=[];  for i = 0:n  l=[l,binomial(n,i)];  end  retval = l;  end  % Binomial  function retval=binomial(n,k)  retval = fact(n) / (fact(n-k) \* fact(k));  end    function retval = fact (n)  if (n > 0)  retval=n\*fact(n-1);  else  retval=1;  end  end |

* + 1. Obtener la convolución lineal (∗) de la imagen con el filtro paso bajas a través de la DFT (comandos MATLAB fft2 y ifft2). Usar también diferentes tamaños de filtros. Desplegar las imágenes resultantes. (Recordar que f(x)⊗ h(x) ←→F F(k)×H(k) y ∗ 6= ⊗)
  1. 7x7 ,9x9 y 11x11



Código:

|  |
| --- |
| i=im2double(imread('lenaTest1.jpg'));  i2= binomialCoef(11) .\* binomialCoef(11)';    i3=imfilter(i,i2,'conv');  I = fft2(i3);  I=fftshift(I);  figure();  imshow(I)  %% Coeficiente binomial  function retval=binomialCoef(n)  l=[];  for i = 0:n  l=[l,binomial(n,i)];  end  retval = l;  end  % Binomial  function retval=binomial(n,k)  retval = fact(n) / (fact(n-k) \* fact(k));  end    function retval = fact (n)  if (n > 0)  retval=n\*fact(n-1);  else  retval=1;  end  end |

# CONCLUSIÓN

--------------------------------------------------------

# Referencias

[1] D. Donnelly and B. Rust, “The Fast Fourier Transform for Experimentalists Part I: Concepts”, Computing in Science & Eng., vol. 7, no. 2, 2005, pp. 80-88.

[2] Levine, M.D., Vision in man and Machine, MCGraw-Hill,1985