



Pontificia Universidad  
**JAVERIANA**  
Bogotá

Pontificia Universidad Javeriana  
Departamento de Ingeniería de Sistemas  
Imágenes Médicas  
Ejercicios: Conceptos de tomografía, 2023-30

## Objetivos

Utilizar MATLAB para interactuar de forma práctica con conceptos básicos de tomografía, como transformada de Radón, sinograma, retroproyección y retroproyección filtrada.

## Desarrollo del ejercicio

El desarrollo del ejercicio práctico consistirá en generar un informe escrito (en formato PDF), el cual debe enviarse a través de la respectiva asignación de BrightSpace. En el informe debe incluirse, como respuesta a cada ejercicio, el código fuente solicitado e impresiones de pantalla del resultado obtenido con ese código. Todos los fragmentos de código utilizados en el ejercicio han sido tomados y adaptados de <https://github.com/SanketD92/CT-Image-Reconstruction/>.

### 1. Creación de imagen sintética

El siguiente código presenta un ejemplo simple de generación de una imagen sintética (*phantom* en inglés) en MATLAB. En este caso, la imagen se compone de círculos de diferentes tamaños e intensidades ubicados en variadas posiciones dentro de una imagen cuadrada:

```
% definicion de los parametros para cada circulo dentro de la imagen
% parametros: centro en x, centro en y, radio, coef. atenuacion (intensidad)
circ = [ 0    0 110  2;
        -65   0  20  1;
         0    0  35  0;
        65 -25  25  4;
        50  50   7  8];

% parametros de la imagen: numero de pixeles, tamano, etc.
nx = 128;
ny = 128;
dx = 2; % resolucion: 2 mm / pixel
x = dx * ([1:nx]'-(nx+1)/2);
y = -dx * ([1:ny]'-(ny+1)/2);
xx = x(:,ones(1,nx));
yy = y(:,ones(1,ny))';
% generar los datos para la imagen sintetica
phantom = zeros(nx,ny);
for ii=1:size(circ,1)
    cx = circ(ii,1);
    cy = circ(ii,2);
    rad = circ(ii,3);
    amp = circ(ii,4);
    t = find( ((xx-cx)/rad).^2 + ((yy-cy)/rad).^2 <= 1 );
    phantom(t) = amp * ones(size(t));
```

```

end
% visualizar la imagen sintetica
figure(1)
    imagesc(x, y, phantom') % Se usa la transpuesta (') y los valores x y y
    colormap('gray')
    axis('square')
    title('Imagen sintetica')
    xlabel('Posicion x')
    ylabel('Posicion y')

```

- **Ejercicio 1**

Ejecute el código para obtener la imagen sintética, e incorpore el resultado obtenido a su reporte.

## 2. Cálculo del sinograma

El siguiente código presenta un ejemplo de generación del sinograma para una imagen (como la sintética producida anteriormente) en MATLAB:

```

% parametros de la geometria
nr = 128; % numero de muestras radiales
dr = 2; % espaciado de los rayos
na = nr*2; % numero de muestras angulares
r = dr * ([1:nr]'- (nr+1)/2); % posiciones de las muestras radiales
angle = [0:(na-1)]'/na * pi; % posiciones de las muestras angulares
% calcular el sinograma para la imagen sintetica
rr = r(:,ones(1,na));
sg1 = zeros(nr, na);
for ii=1:size(circ,1)
    cx = circ(ii,1);
    cy = circ(ii,2);
    rad = circ(ii,3);
    amp = circ(ii,4);
    tau = cx * cos(angle) + cy * sin(angle);
    tau = tau(:,ones(1,nr))';
    t = find( (rr-tau).^2 <= rad.^2 );
    if ii > 1, amp = amp - circ(1,4); end % discos pequenos embebidos
    sg1(t) = sg1(t)+amp*2*sqrt(rad^2-(rr(t)-tau(t)).^2);
end
% visualizar el sinograma
figure(2)
    imagesc(r, angle/pi*180, sg1') % se usa la transpuesta (') y el angulo
    colormap('gray') % se muestra en grados
    title('Sinograma: imagen sintetica')
    xlabel('Posicion (rayos)')
    ylabel('Angulo (vistas)')
% definir una variable general para el sinograma, para posterior ejecucion
sinogram = sg1; % sinograma de la imagen sintetica
% visualizar informacion del sinograma
disp(sprintf('numero de rayos = %g', nr))
disp(sprintf('numero de vistas = %g', na))

```

- **Ejercicio 2**

Ejecute el código para obtener el sinograma de la imagen sintética inicial, e incorpore el resultado obtenido a su reporte.

### 3. Aplicación de retroproyección simple (laminograma)

El siguiente código presenta un ejemplo de aplicación de la retroproyección simple sobre el sinograma para la generación del laminograma (reconstrucción) de una imagen (como la sintética producida anteriormente) en MATLAB:

```
% retroproyeccion simple
lamin = zeros(nx,ny);
for ia = 1:na
    disp(sprintf('angulo %g de %g', ia, na));
    % extraer angulo actual del sinograma
    projection_ia=sinogram(:,ia);
    % difuminar angulo actual en 128*128
    projection_smear=repmat (projection_ia,1,128);
    % 256 proyecciones corresponden a 180 grados
    % entonces se usa ia*180/256 para el angulo de proyeccion actual
    rot= imrotate(projection_smear', ia*180/256, 'bicubic','crop');
    % lamin necesita ser de 128*128
    % por eso el primer argumento en imrotate debe ser de la misma dimension
    lamin=lamin+rot;
end
% visualizar la imagen reconstruida
figure(3)
imagesc(x, y, lamin);
colormap('gray');
axis('image')
title('Reconstruccion por retroproyeccion simple')
xlabel('mm') %x y y en imagesc nos da limites escalares en mm
ylabel('mm')
```

- **Ejercicio 3**

Ejecute el código para obtener la reconstrucción por retroproyección simple para la imagen sintética original, e incorpore el resultado obtenido a su reporte.

### 4. Filtrado de las proyecciones

El siguiente código presenta un ejemplo de filtrado del sinograma de una imagen (como la sintética producida anteriormente) en MATLAB, utilizando el filtro Ram-Lak. Esto facilitará generar posteriormente una reconstrucción más exacta de la imagen original:

```
% transformar en Fourier el sinograma,
% luego trasladar para centrar las bajas frecuencias
sinogramfiltered=fftshift(fft(sinogram));
% filtrar el sinograma usando un filtro Ram-Lak
a = length(sinogram);
% pendiente -1 hasta 128, luego +1 hasta 256
freqs=linspace(-1, 1, a/2).';
myFilter = abs(freqs);
% asegurarse que el sinograma y el filtro son de las mismas dimensiones
myFilter = repmat(myFilter,1,256);
% multiplicar en frecuencia, luego trasladar, luego transformada inversa
```

```

sinogramfilt=abs(ifft(ifftshift(sinogramfiltered.*myFilter)));
% visualizar el sinograma filtrado en Theta = 45 grados
figure(4)
    % 45 grados es la proyeccion #64 (256*45/180)
    plot(r, sinogram(:,64)./max(sinogram(:,64)), '-','...
        r, sinogramfilt(:,64)./max(sinogramfilt(:,64)),':');
    title('Sinograma filtrado en 45 grados');
    legend('original', 'filtrado');
    xlabel('Posicion (rayos)');
    ylabel('Amplitud');
% visualizar el sinograma completo filtrado
figure(5)
    imagesc(r, angle/pi*180, sinogramfilt');
    colormap('gray');
    axis('image');
    title('Sinograma filtrado')
    xlabel('Posicion (rayos)')
    ylabel('Angulo (vistas)')

```

- **Ejercicio 4**

Ejecute el código para obtener el sinograma filtrado para la imagen sintética original, e incorpore el resultado obtenido a su reporte.

## 5. Aplicación de retroproyección filtrada

El siguiente código presenta un ejemplo de aplicación de la retroproyección filtrada sobre el sinograma (ya filtrado) para la generación de la reconstrucción de una imagen (como la sintética producida anteriormente) en MATLAB:

```

% retroproyeccion filtrada
bpf_recon = zeros(nx,ny);
for ia = 1:na
    disp(sprintf('angulo %g de %g', ia, na));
    bpf_ia=sinogramfilt(:,ia);
    bpf_smear=repmat(bpf_ia,1,128);
    % rotando la proyeccion
    rot1= imrotate(bpf_smear', ia*180/256, 'bicubic','crop');
    bpf_recon=bpf_recon+rot1;
end
% visualizar la imagen reconstruida
figure(6)
    imagesc(x, y, max(bpf_recon,0));
    colormap('gray');
    axis('image')
    title('Reconstruccion por retroproyeccion filtrada')
    xlabel('Posicion x')
    ylabel('Posicion y')

```

- **Ejercicio 5**

Ejecute el código para obtener la reconstrucción por retroproyección filtrada para la imagen sintética original, e incorpore el resultado obtenido a su reporte.

## 6. Transformada de Radón y reconstrucción de la imagen

El siguiente código presenta un ejemplo simple de utilización de la transformada de Radón para generar el sinograma de una imagen (como la sintética producida anteriormente), y la utilización de la transformada inversa de Radón con un filtro en frecuencia para generar la reconstrucción de la imagen en MATLAB:

```
% angulos para la proyeccion
theta=0:180;
% obtener el sinograma aplicando la transformada de Radon
[R,rad_angles]=radon(phantom,theta);
% visualizar el sinograma obtenido
figure(7)
    imagesc(rad_angles,theta,R');
    colormap('gray');
    title('Sinograma generado con la transformada de Radon')
    xlabel('Posicion')
    ylabel('Angulo')
% reconstruir la imagen con la transformada inversa, aplicando un filtro
RamLak_filtered=iradon(R, theta, 'linear','Ram-Lak', 1.0, size(phantom,1));
RamLak_filtered=imrotate(RamLak_filtered,90);
% visualizar la imagen reconstruida
figure(8)
    imagesc(RamLak_filtered);
    colormap('gray');
    title('Reconstruccion con la trans. inversa de Radon y el filtro Ram-Lak')
    xlabel('Posicion x')
    ylabel('Posicion y')
```

- **Ejercicio 6**  
Ejecute el código para obtener la reconstrucción por retroproyección filtrada para la imagen sintética original usando la transformada de Radón en MATLAB, e incorpore el resultado obtenido a su reporte.
- **Ejercicio 7**  
Genere diferentes reconstrucciones por retroproyección filtrada para la imagen sintética original usando la transformada de Radón en MATLAB, esta vez variando el filtro utilizado al momento de la reconstrucción. Las opciones de filtro a probar (en vez de 'Ram-Lak') son: 'Shepp-Logan', 'Hamming' y 'Hann'. Analice en su reporte las posibles diferencias que puedan identificarse del uso de estos filtros.
- **Ejercicio 8**  
Replique el proceso completo (sinograma – ejercicio 2, retroproyección simple – ejercicio 3, sinograma filtrado – ejercicio 4, retroproyección filtrada – ejercicio 5, reconstrucción con la transformada de Radón – ejercicio 6) para una nueva imagen sintética con al menos 7 círculos de diferentes tamaños, posiciones e intensidades (ejercicio 1). Identifique claramente el filtro utilizado al momento de aplicar la transformada inversa de Radón.
- **Ejercicio 9**  
Replique el proceso completo (sinograma – ejercicio 6, retroproyección simple – ejercicio 3, sinograma filtrado – ejercicio 4, retroproyección filtrada – ejercicio 5) para una imagen de su elección, garantizando que la información del sinograma se construye utilizando la transformada de

Radón (ya no manualmente como en el ejercicio 2), y que las variables de geometría para el proceso de retroproyección se ajustan de acuerdo a las características de la imagen seleccionada. Se sugiere trabajar con imágenes de tamaño cuadrado e intensidades en escala de grises (1 solo canal) para facilidad del ejercicio. Como fuente de consulta adicional, se puede revisar el código en [https://github.com/SanketD92/CT-Image-Reconstruction/blob/master/src/Sanket\\_CT\\_Project.m](https://github.com/SanketD92/CT-Image-Reconstruction/blob/master/src/Sanket_CT_Project.m), a partir de la línea 269.

### **Entrega del ejercicio**

La entrega del ejercicio práctico consistirá únicamente en el archivo de reporte en formato PDF, nombrado con los apellidos de los integrantes del grupo. Este archivo deberá enviarse a través de la correspondiente asignación en BrightSpace antes de la medianoche del lunes 4 de septiembre de 2023. El envío del archivo comprimido en otro formato diferente a los especificados resultará en una calificación de (0.0/5.0) para el taller.

La escala de evaluación es la siguiente, para cada ejercicio individual:

- **Excelente (5.0/5.0):** El estudiante propone un código que realiza todos los pasos sugeridos y presenta evidencias de los resultados obtenidos.
- **Bueno (3.5/5.0):** El estudiante propone un código que realiza la mayoría de los pasos sugeridos, y presenta al menos una evidencia de los resultados obtenidos.
- **Aceptable (2.5/5.0):** El estudiante propone un código que realiza solo algunos de los pasos sugeridos, y presenta evidencia(s) de los resultados obtenidos, no todos correctos.
- **Malo (1.0/5.0):** El código propuesto por el estudiante no es correcto en su sintaxis (no puede interpretarse en el entorno de ejecución adecuadamente).
- **No entregó (0.0/5.0):** El estudiante no entrega los resultados solicitados.