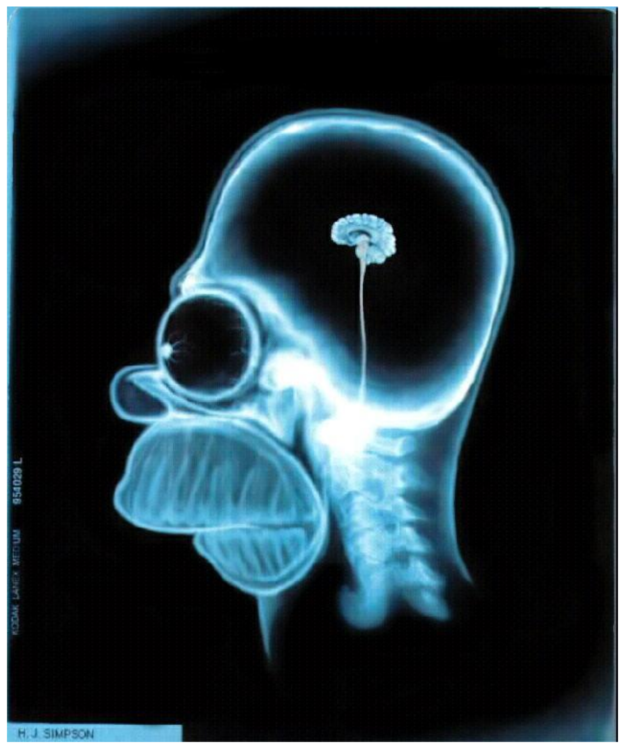
Procesamiento digital de Imágenes

Tomografía

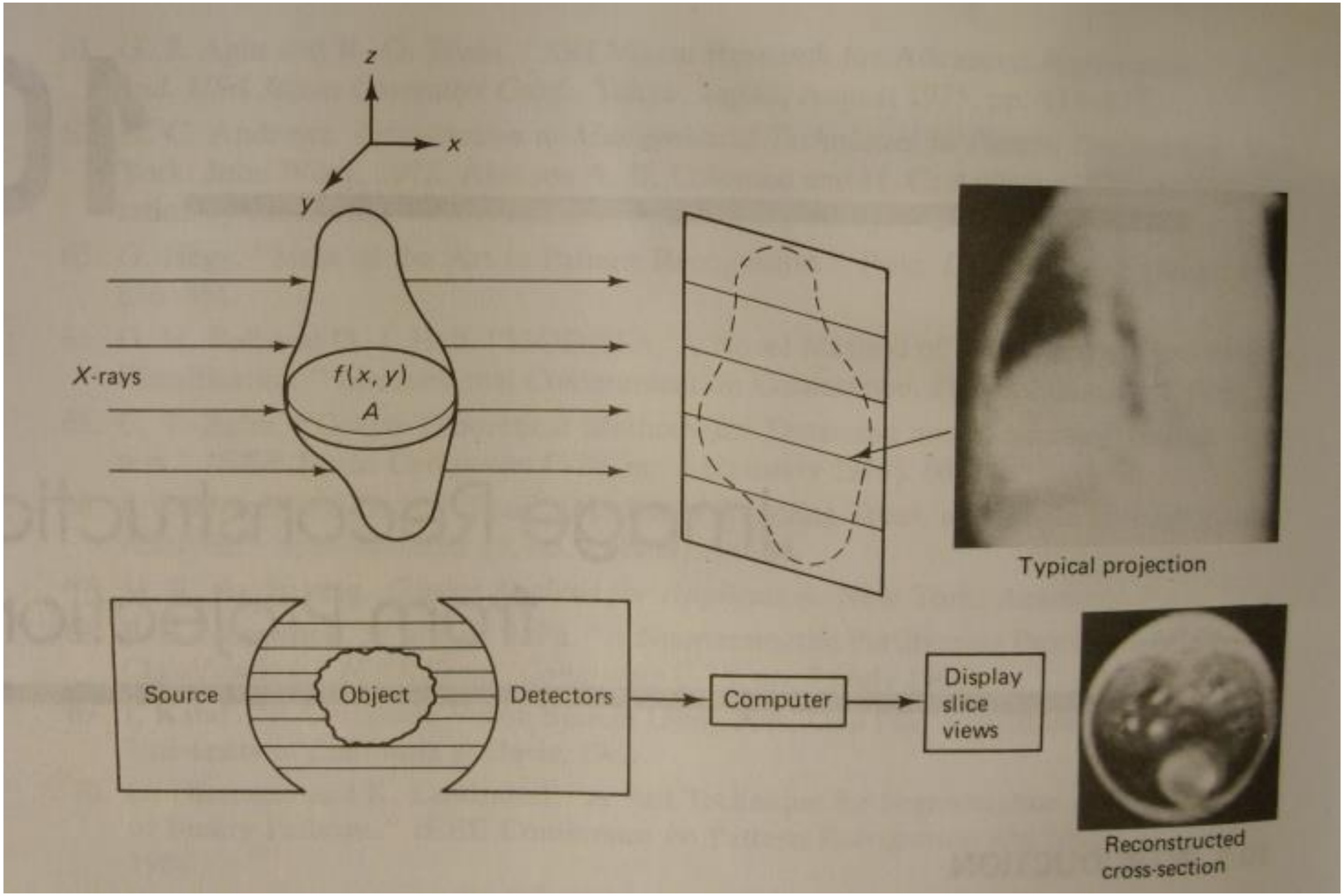


Introducción

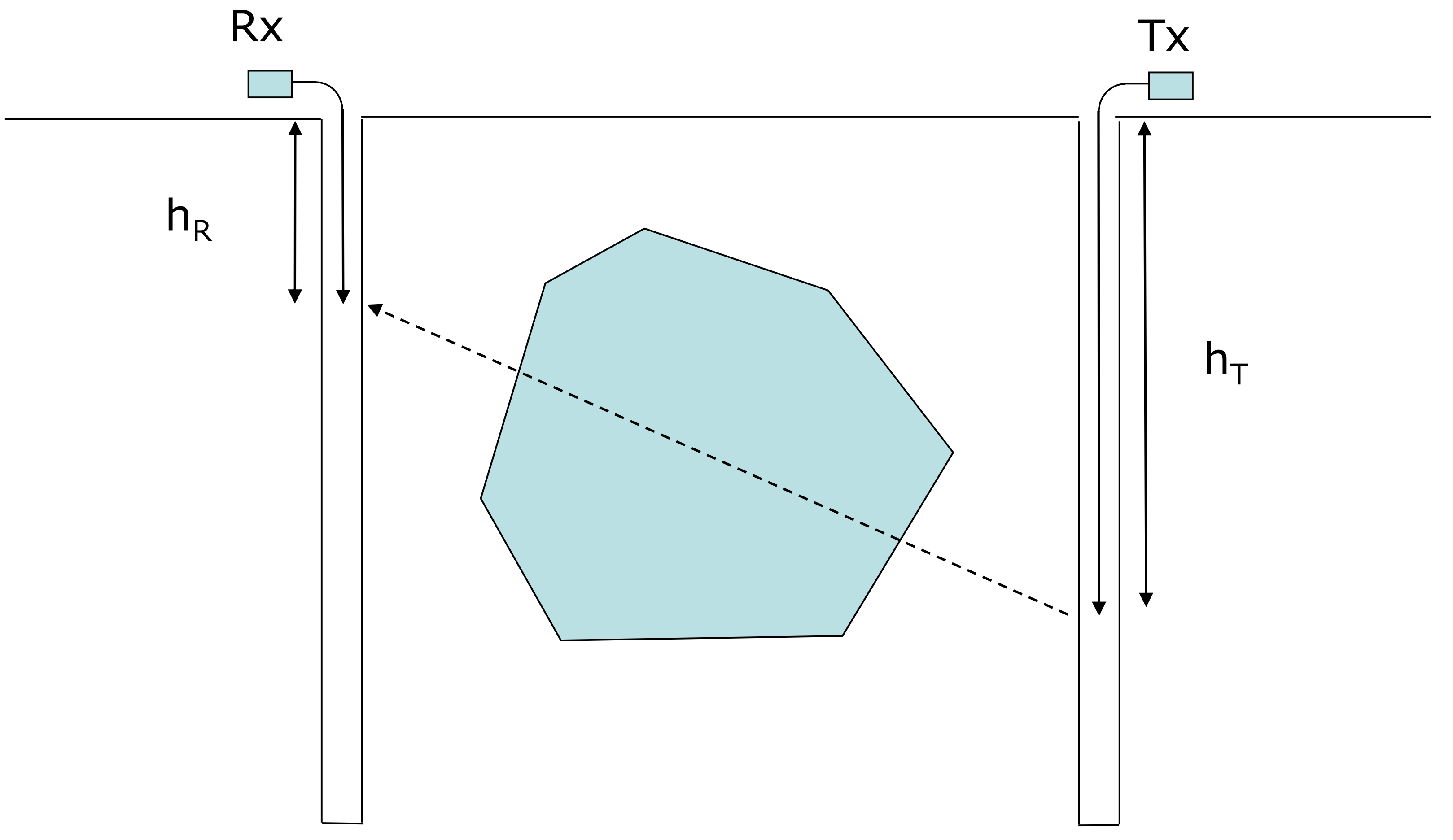
- Uso mas frecuente en medicina
- Acústica
- Microondas
- Geología
- Microscopios electrónicos
- Radiotelescopios
- Rayos-X Positrones Rayos gama

Fundamentos

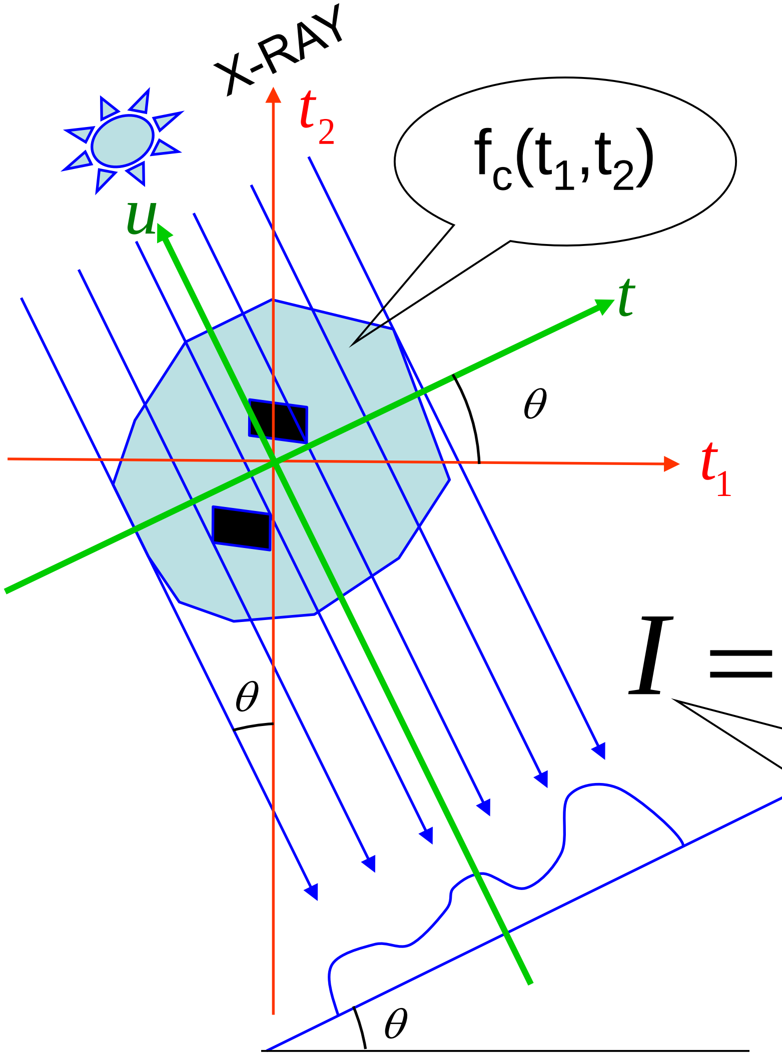
- Concepto de proyección



Aplicación en Geología



Concepto de proyección



Densidad de un objeto Bidimensional

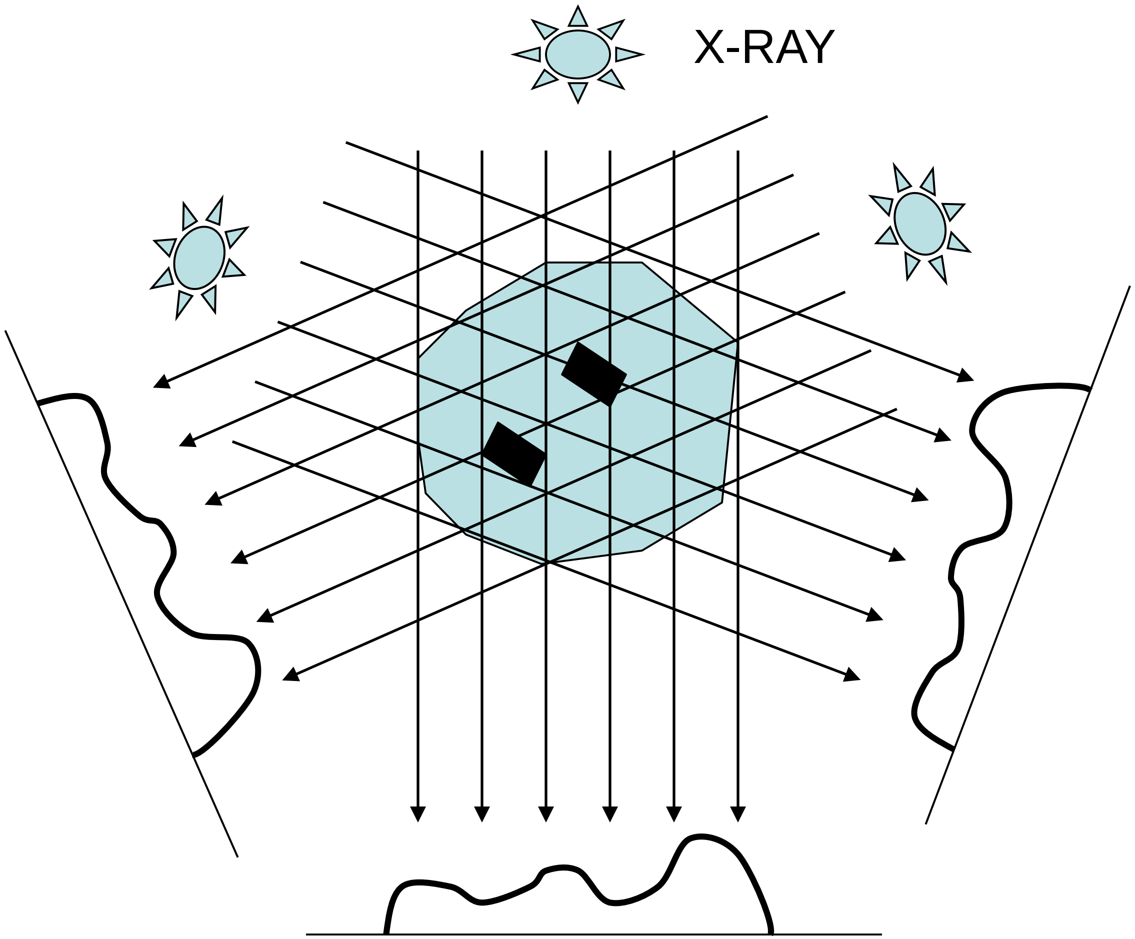
$$I = I_o e^{\left[-\int_L f_c(t_1,t_2)\,du\right]}$$

θ = Angulo de la proyección
t = Ubicación del detector

Intensidad atenuada de un rayo a medida que se propaga a través del objeto a lo largo de una línea L(θ,t)

Concepto de proyección

•



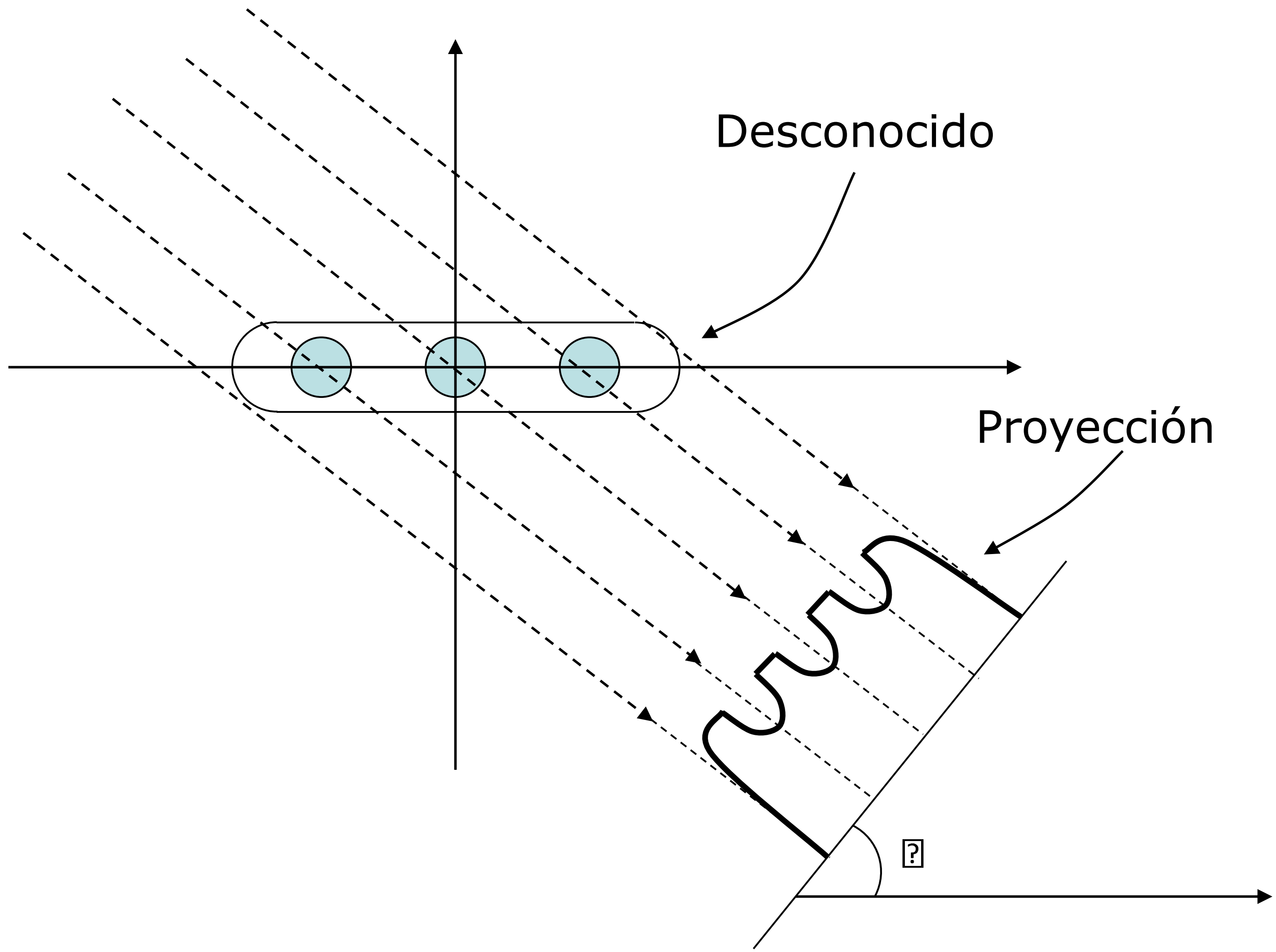
Fundamentos

Proyección:

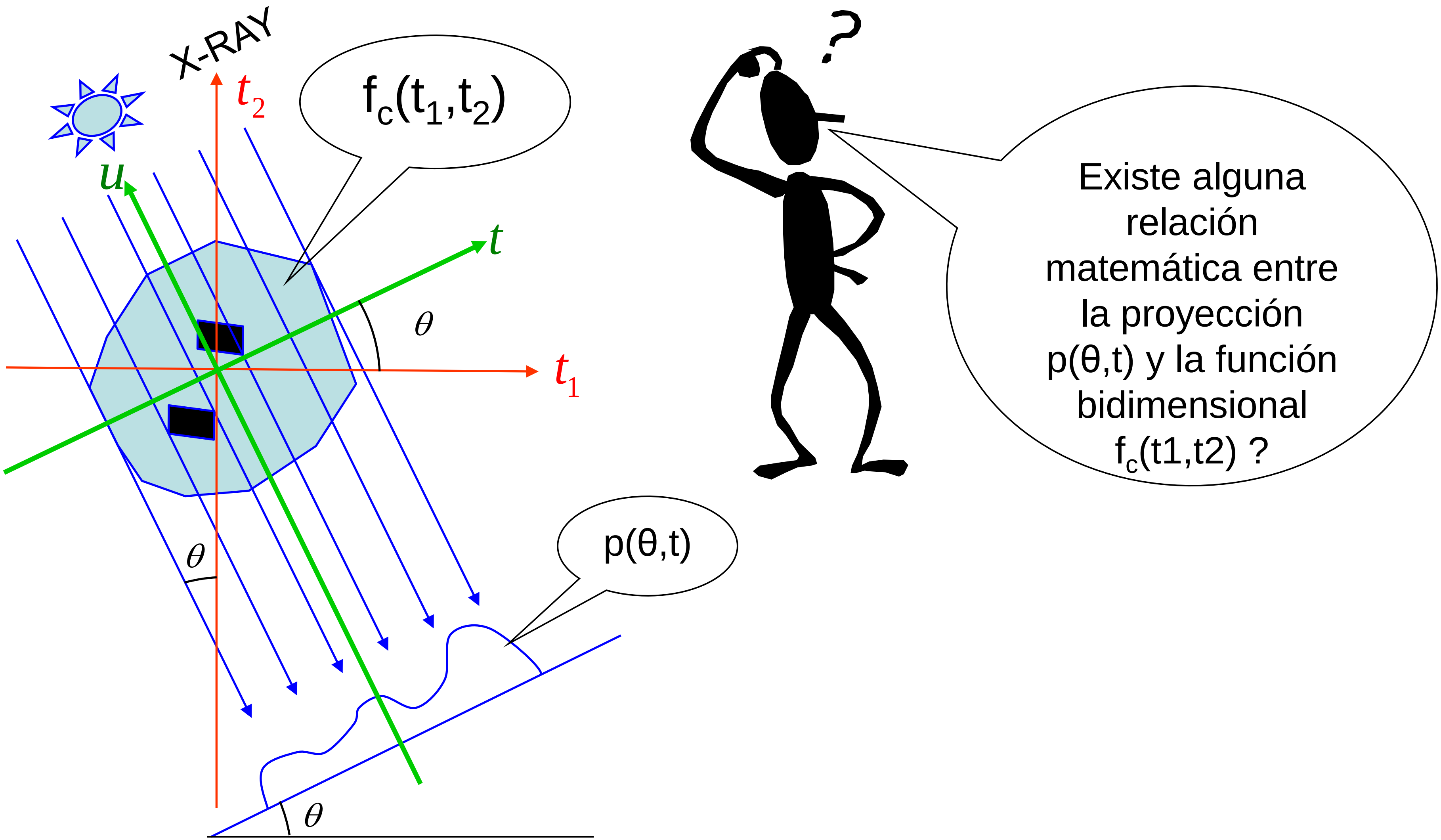
Es la operación matemática cuyo resultado es similar a la operación física de tomar una foto irradiada por un haz colimado de rayos X

Objetivo: Reconstruir el objeto en 2D a partir de proyecciones en diferentes ángulos

Fundamentos



Projection Slice Theorem



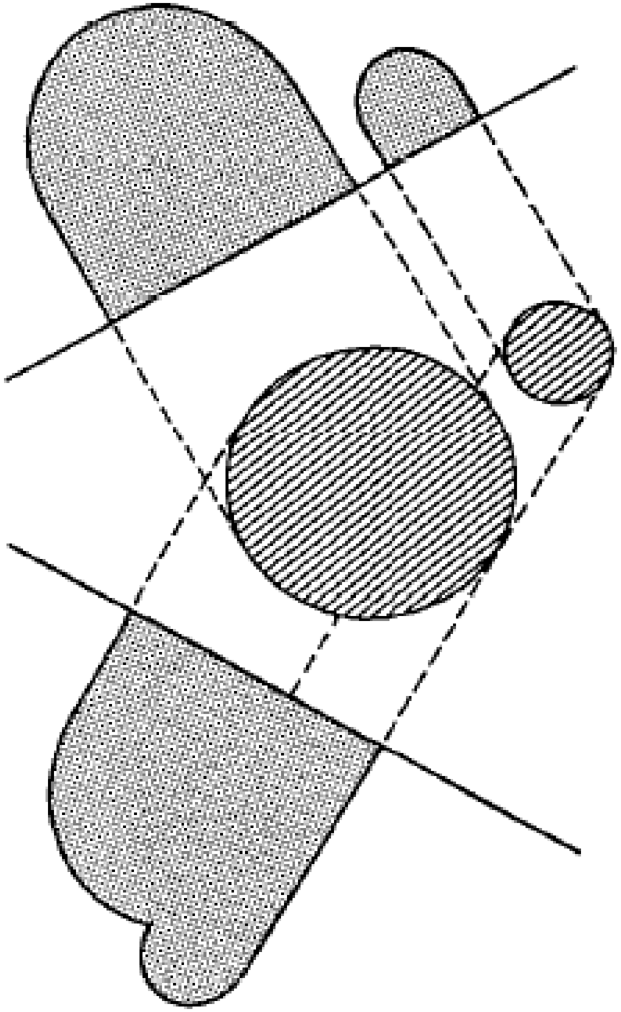
Projection Slice Theorem

$$I = I_o e^{\left[-\int_L f_c(t_1, t_2)\, du \right]}$$

$$\ln \frac{I_o}{I} = \int_L f_c(t_1, t_2)\, du = p(\theta, t)$$

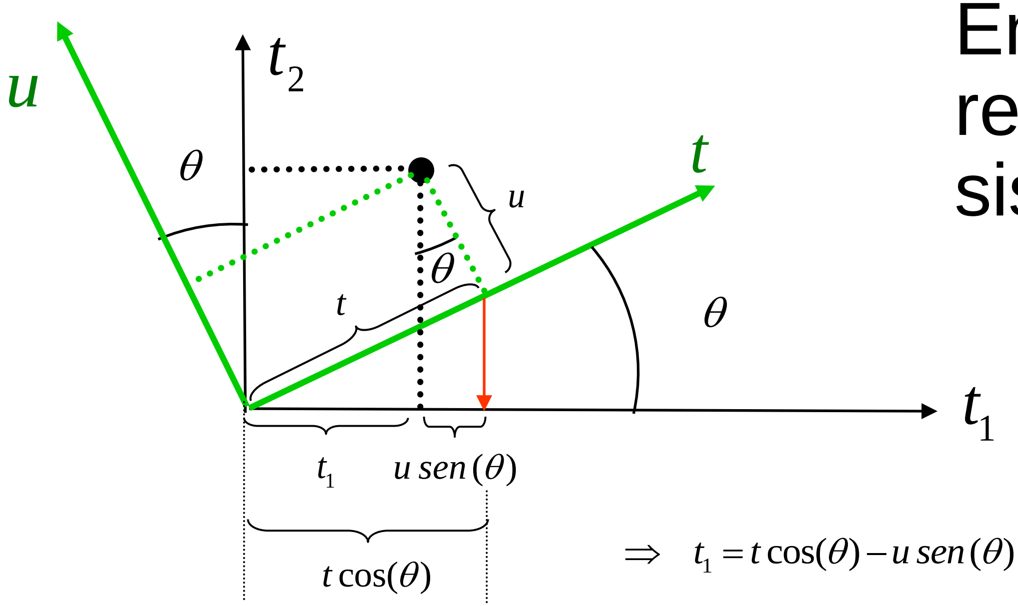
Projection Slice Theorem

Ejemplo



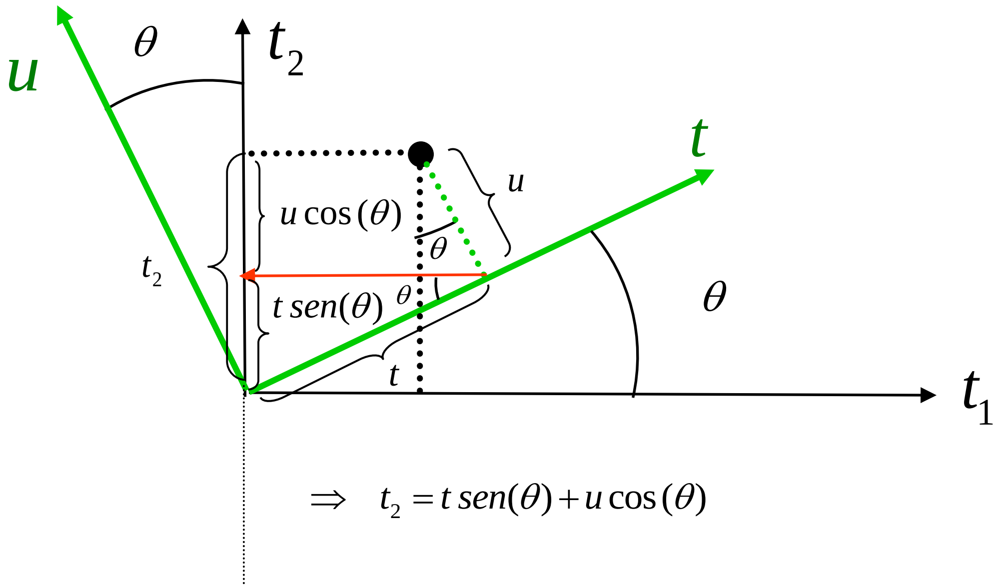
Projection Slice Theorem

Encontramos ahora la relación entre ambos sistemas coordenados.



$$\Rightarrow \quad t_1 = t\cos(\theta) - u\,sen(\theta)$$

Projection Slice Theorem



Resumiendo

$$t_1 = t\cos(\theta) - u\,sen(\theta)$$

$$t_2 = t\,sen(\theta) + u\cos(\theta)$$

Projection Slice Theorem

$$p(\theta,t) = \int_L f_c(t_1,t_2)\Big|_{\substack{t_1 = t\cos(\theta) - u\,sen(\theta) \\ t_2 = t\,sen(\theta) + u\cos(\theta)}} du$$

De aquí en mas nuestro objetivo será encontrar una relación entre la TF 2D de $f_c(t_1,t_2)$ y la TF 1D de $p(\theta,t)$.

Esto se puede hacer mediante lo que se conoce como Projection Slice Theorem

Projection Slice Theorem

Empecemos por encontrar la TF de fc(t_1,t_2)

$$F_c(\Omega_1,\Omega_2) = 2D\ CTFT\left\{ f_c(t_1,t_2)\right\}$$

$$F_c(\Omega_1,\Omega_2) = \int_{t_1=-\infty}^{+\infty}\int_{t_2=-\infty}^{+\infty} f_c(t_1,t_2)\, e^{-j\Omega_1 t_1} e^{-j\Omega_2 t_2}\, dt_1 dt_2$$

A su vez la 2D CIFTde $F_c(\Omega_1\Omega_2)$ será:

$$f_c(t_1,t_2) = \frac{1}{4\pi^2}\int_{\Omega_1=-\infty}^{+\infty}\int_{\Omega_2=-\infty}^{+\infty} F_c(\Omega_1,\Omega_2)\, e^{j\Omega_1 t_1} e^{j\Omega_2 t_2}\, d\Omega_1 d\Omega_2$$

Projection Slice Theorem

Por otro lado la TF 1D de p(θ,t) es:

$$P_\theta(\Omega) = 1D \ CTFT\left\{ p(\theta,t) \right\}$$

$$P_\theta(\Omega) = \int_{t=-\infty}^{+\infty} p(\theta,t)e^{-j\Omega t}dt$$

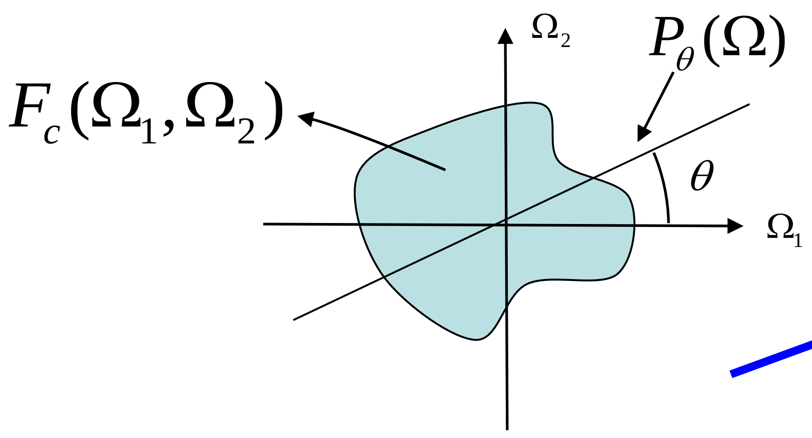
Projection Slice Theorem

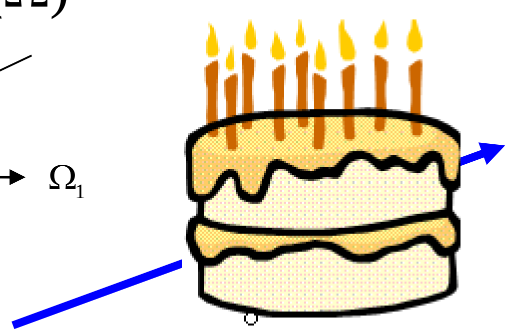
Ahora el Projection Slice Theorem establece que:

$$P_\theta(\Omega) = F_c(\Omega_1, \Omega_2)\Big|_{\substack{\Omega_1=\Omega\cos\theta \\ \Omega_2=\Omega sen\theta}}$$

$$\boxed{P_\theta(\Omega) = F_c(\Omega\cos\theta, \Omega\, sen\,\theta)}$$

Slice of Fc





Projection Slice Theorem

Este teorema es de gran utilidad ya que si tomamos múltiples proyecciones a diferentes ángulos podemos reconstruir la transformada de fourier de fc(t1,t2).Luego aplicando la Transfomada inversa obtenemos el objeto original.

$$P_\theta(\Omega) \rightarrow F_c(\Omega\cos\theta, \Omega\, sen\,\theta) \rightarrow f_c(t_1, t_2)$$

$$\theta = \theta_1, \theta_2, \theta_3 \cdots$$

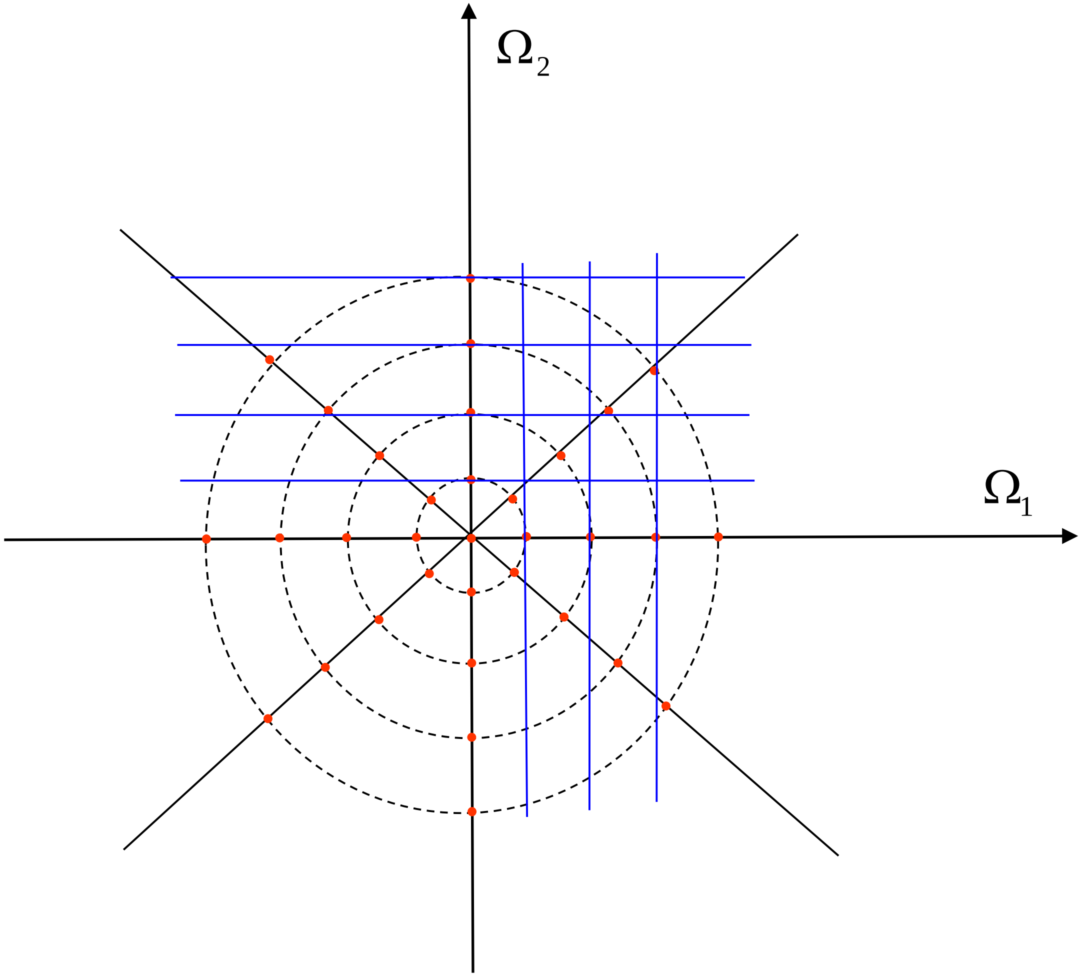
Técnicas de reconstruccion

1. Simple : - Nearest Neighbor (interpolacion de orden cero)
   - Interpolación de primer orden.
2. Formula de inversión de Radon
3. Técnicas Iterativas

Reconstrucción de la Transformada

Polar Sampling: Ej:. DFT de 9 puntos en cada direccion , 8 proyecciones

Las muestras están igualmente espaciadas



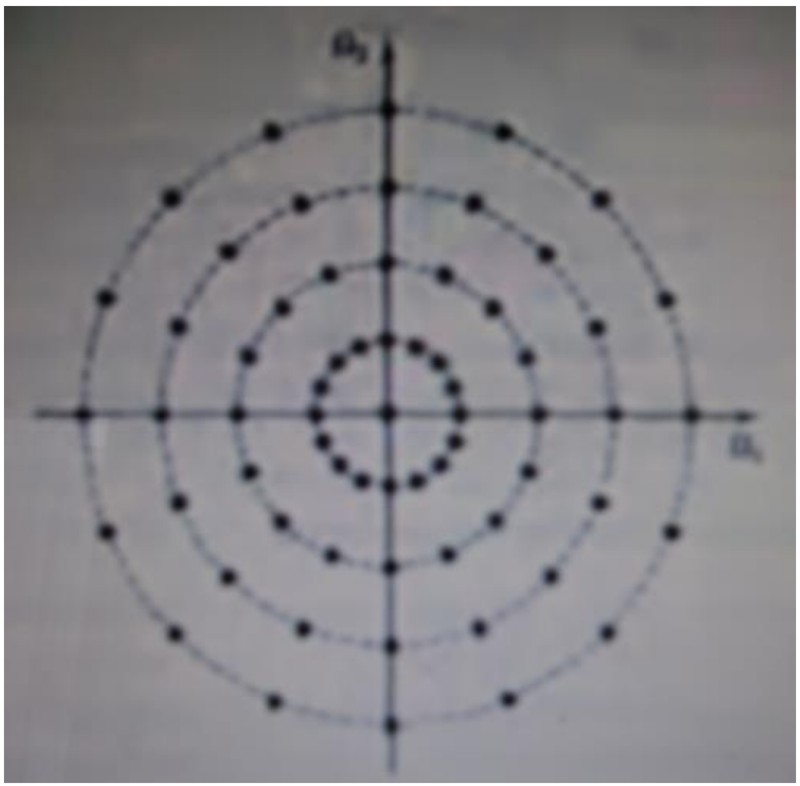
Debemos encontrar los valores de la transformada en La grilla cartesiana a partir de los valores de la misma en forma polar (puntos rojos). La solución es Interpolar. Los métodos mas simples son:

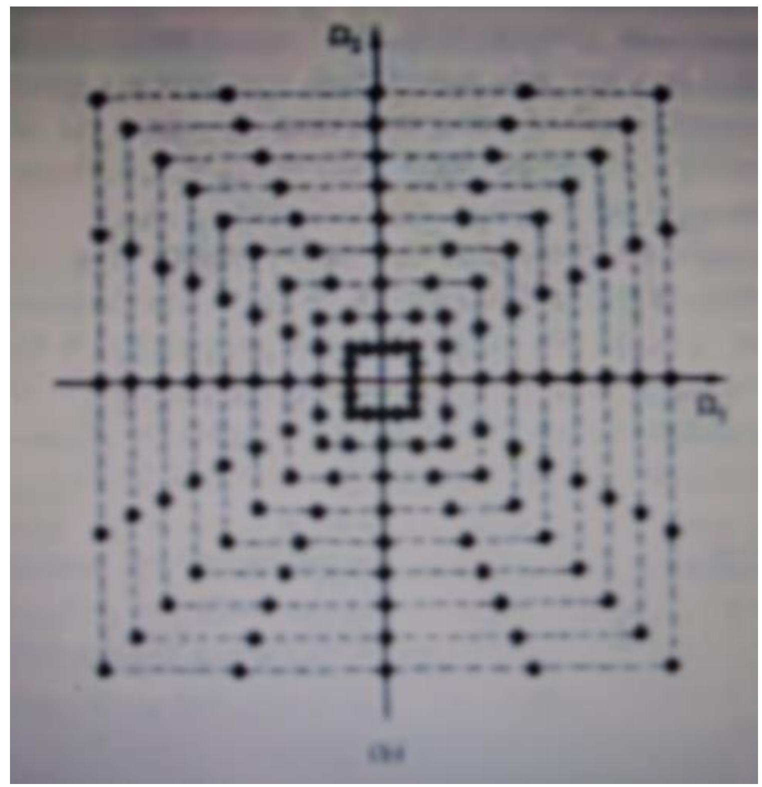
Zeroth order → Nearest Neighbor

First order → Weighed Sum of neighbor samples (Average)

Reconstrucción de la Transformada

Si modificamos la distancia entre las muestras en función del ángulo (obtenemos cuadrados concéntricos).





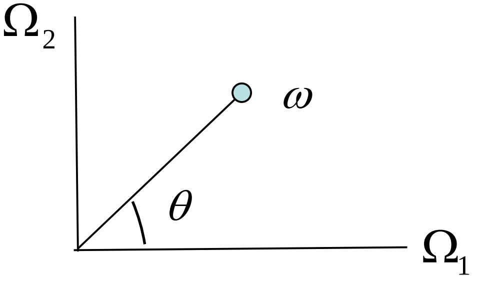
Formula de inversion de Radon

Partiendo de la IDFT 2D de Fc(Ω_1, Ω_2)

$$f_c(t_1,t_2)=\frac{1}{4\pi^2}\int_{\Omega_1=-\infty}^{+\infty}\int_{\Omega_2=-\infty}^{+\infty}F_c(\Omega_1,\Omega_2)\,e^{j\Omega_1 t_1}e^{j\Omega_2 t_2}\,d\Omega_1 d\Omega_2$$

Vamos a convertirla a coordenadas polares esto es: (Ω_1, Ω_2)→ (ω, θ).

Para realizar la conversión deberemos hacer uso del jacobiano



Formula de inversion de Radon

Recordando que :

$$si \quad x = g(u,v) \quad y \quad y = h(u,v)\; entonces:$$

$$\iint f(x,y)\,dxdy = \iint f(g(u,v),h(u,v))\,J\,dudv \quad donde$$

$$J = \frac{\partial(x,y)}{\partial(u,v)} = \begin{vmatrix} \dfrac{\partial(x)}{\partial(u)} & \dfrac{\partial(x)}{\partial(v)} \\ \dfrac{\partial(y)}{\partial(u)} & \dfrac{\partial(y)}{\partial(v)} \end{vmatrix} = \frac{\partial(x)}{\partial(u)}\frac{\partial(y)}{\partial(v)} - \frac{\partial(y)}{\partial(u)}\frac{\partial(x)}{\partial(v)}$$

Formula de inversion de Radon

$$si \quad \Omega_1 = g(\omega,\theta) = \omega \cos\theta \quad y \quad \Omega_2 = h(\omega,\theta) = \omega\, sen\,\theta$$

$$entonces:$$

$$J = \frac{\partial(x,y)}{\partial(u,v)} = \begin{vmatrix} \cos\theta & -\omega sen\,\theta \\ sen\,\theta & \omega\cos\theta \end{vmatrix} = \omega$$

Entonces fc(t_1, t_2) nos queda

$$f_c(t_1,t_2) = \frac{1}{4\pi^2}\int_0^{\pi}\int_{-\infty}^{+\infty} F_c(\omega\cos\theta, \omega\, sen\,\theta)\, e^{j\omega(t_1\cos\theta + t_2\, sen\,\theta)}\left|\omega\right| \; d\omega\, d\theta$$

Formula de inversion de Radon

Observemos que:

$$f_c(t_1,t_2)=\frac{1}{4\pi^2}\int_0^{\pi}\int_{-\infty}^{+\infty}\underbrace{F_c(\omega\cos\theta,\omega\, sen\,\theta)}_{P_\theta(\omega)}e^{j\omega(t_1\cos\theta+t_2\, sen\,\theta)}\left|\omega\right|\; d\omega\, d\theta$$

$$f_c(t_1,t_2)=\frac{1}{4\pi^2}\int_0^{\pi}\underbrace{\int_{-\infty}^{+\infty}P_\theta(\omega)\,e^{j\omega(t_1\cos\theta+t_2\, sen\,\theta)}\left|\omega\right|\; d\omega}_{I}\, d\theta$$

Formula de inversion de Radon

Observemos que:

$$f_c(t_1,t_2)=\frac{1}{4\pi^2}\underbrace{\int_0^{\pi}\int_{-\infty}^{+\infty}P_\theta(\omega)\,e^{j\omega(t_1\cos\theta+t_2\,sen\,\theta)}\left|\omega\right|\;d\omega\,d\theta}_{I}$$

Es una IDFT

$$I=\int_{-\infty}^{+\infty}G_\theta(\omega)\,e^{j\omega t}d\omega \quad donde:$$

$$G_\theta(\omega)=P_\theta(\omega)\left|\omega\right| \quad IDFT\left[G_\theta(\omega)\right]=g_\theta(t)$$

$$y\;\; t=t_1\cos\theta+t_2\,sen\,\theta$$

Formula de inversion de Radon

En definitiva nos queda que: $I = g_\theta(t)\Big|_{t=t_1\cos\theta + t_2 sen\,\theta}$

$$I = g_\theta(t_1\cos\theta + t_2\,sen\,\theta)$$

Reemplazando en fc(t_1, t_2) nos queda:

$$\boxed{f_c(t_1,t_2) = \frac{1}{4\pi^2}\int_0^\pi g_\theta(t_1\cos\theta + t_2\,sen\,\theta)\,d\theta}$$

Formula de inversion de Radon

Interpretacion

$$f_c(t_1,t_2) = \frac{1}{4\pi^2}\int_0^{\pi} g_\theta(t_1\cos\theta + t_2\, sen\,\theta)\, d\theta$$

$$G_\theta(\omega) = P_\theta(\omega)\left|\omega\right| \quad \Rightarrow g_\theta(t) = p(\theta,t)\otimes IDFT\left\{\left|\omega\right|\right\}$$

$$y \;\; t = t_1\cos\theta + t_2\, sen\,\theta$$

Formula de inversion de Radon

Resulta que:

$$g_\theta(t) = p(\theta,t) \otimes IDFT\left\{|\omega|\right\} = \frac{d}{dt}\int_{-\infty}^{+\infty} \frac{P_\theta(\tau)}{t-\tau}\, d\tau$$

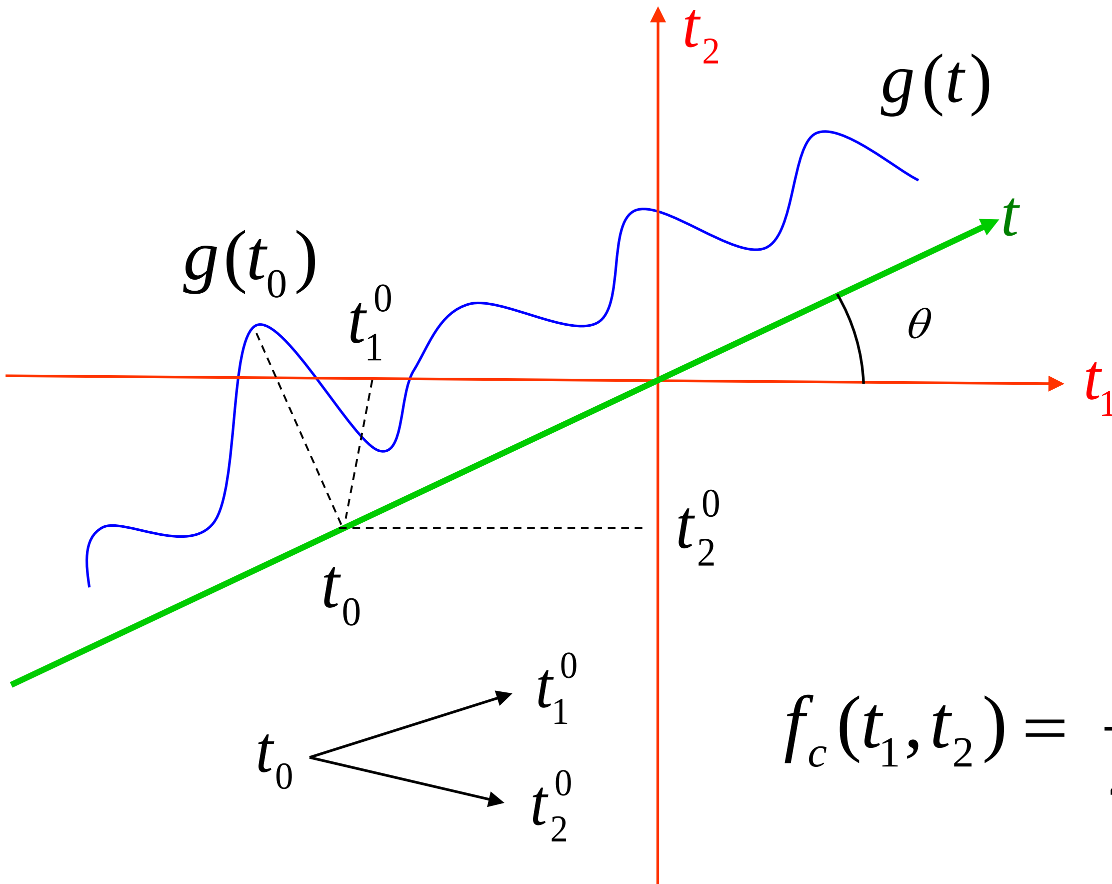
1- Encontramos $p(\theta,t)$

2- Hallamos $g_\theta(t) = p(\theta,t) \otimes IDFT\left\{|\omega|\right\} = \frac{d}{dt}\int_{-\infty}^{+\infty} \frac{P_\theta(\tau)}{t-\tau}\, d\tau$

3- Reemplazamos este resultado en $fc(t_1, t_2)$

$$f_c(t_1,t_2) = \frac{1}{4\pi^2}\int_0^{\pi} g_\theta(t_1\cos\theta + t_2\, sen\, \theta)\, d\theta$$

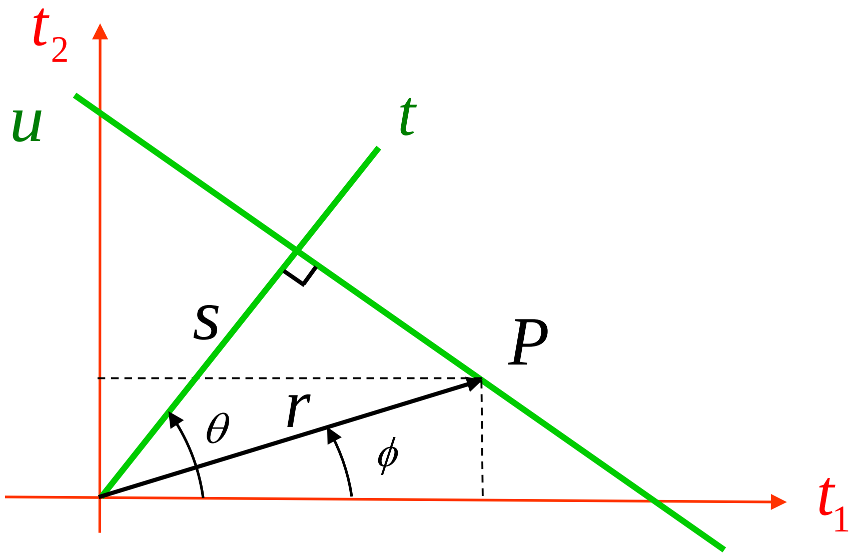
Convolution Backprojection Interpretación



$$f_c(t_1, t_2) = \frac{1}{4\pi^2} \int_0^{\pi} g_\theta(t)\, d\theta$$

$$f_c(t_1^0, t_2^0) = g(t_0)$$

Radon Transform



$$g(t,\theta) = \Re[\,f(t_1,t_2)] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(t_1,t_2)\,\delta(t_1\cos\theta + t_2\, sen\,\theta - t)\;\; dt_1\, dt_2$$

$$-\infty < t < \infty \quad 0 \le \theta < \pi$$

La función g(t,θ) es la transformada de Radon de f (t_1,t_2)

y es la proyeccion unidimensional de f (t_1,t_2) en la direccion θ

Radon Transform

Respecto del sistema rotado (t,u) g(t,θ) puede ser expresada como:

$$g(t,\theta)=\int_{-\infty}^{+\infty} f(t\cos\theta - u\, sen\,\theta, t\, sen\,\theta + u\cos\theta)\ \ du$$

$$-\infty < t < \infty \quad 0 \leq \theta < \pi$$

La funcion g(t,θ) es denominada suma de rayos dado que representa la suma de $f(t_1,t_2)$ en la dirreccion θ. La transformada de radon mapea el dominio espacial (t_1,t_2) al dominio (t,θ). Cada punto en el dominio (t,θ) corresponde a una linea en el dominio espacial (t_1,t_2)

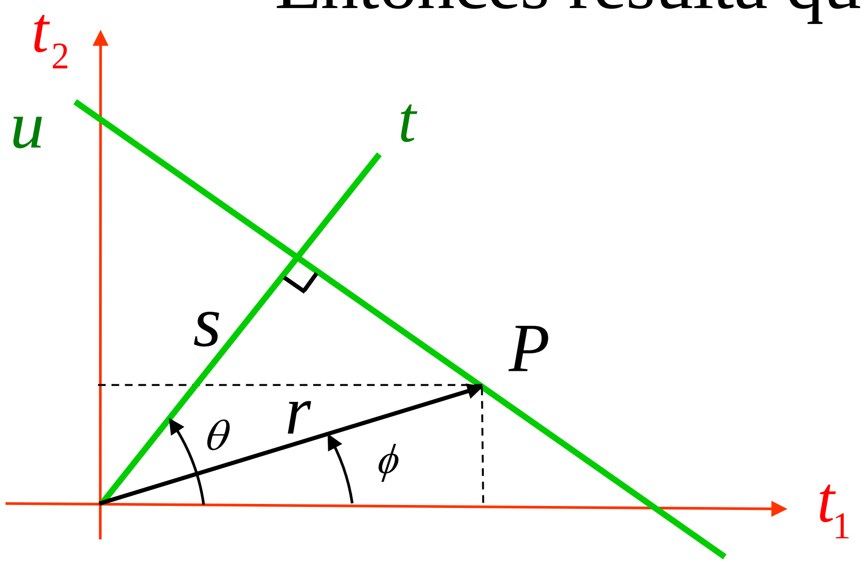
Radon Transform

Debe notarse que (t, θ) no son las coordenadas polares de (t_1, t_2).

En efecto si llamamos a (r, θ) a las coordenadas de (t_1, t_2)

$$t_1 = r\cos(\phi) \qquad t_2 = r\,sen(\phi)$$

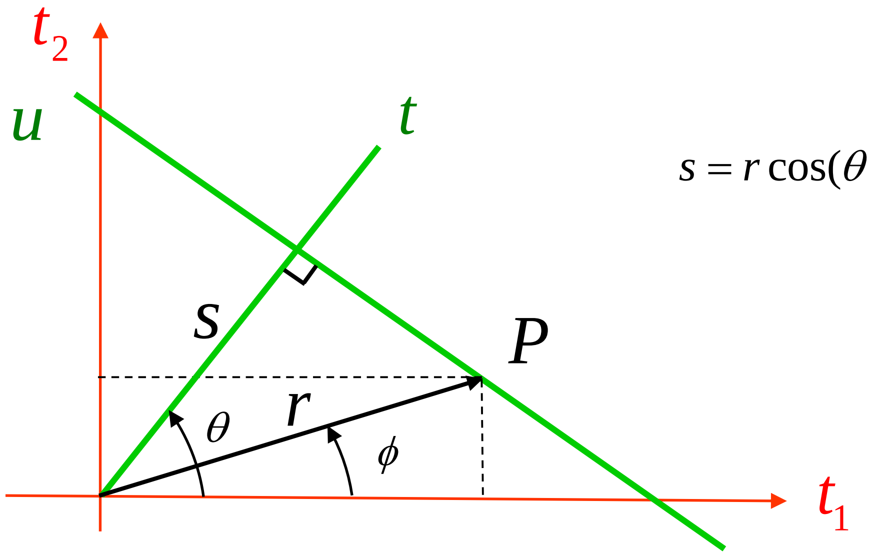
Entonces resulta que $s = r\cos(\theta - \phi)$

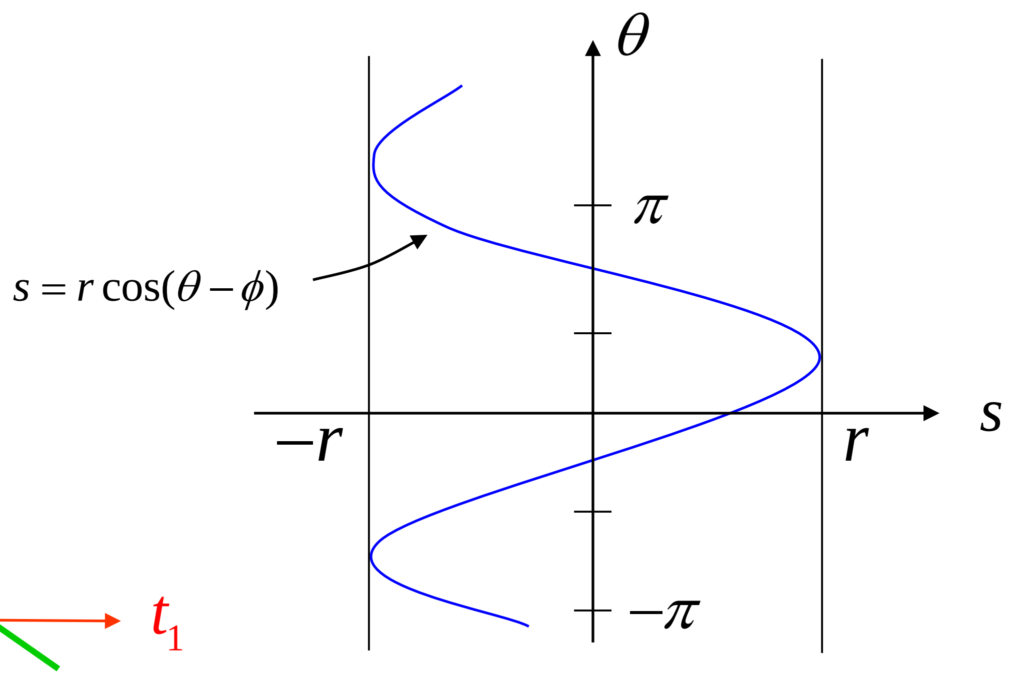


Radon Transform

Esto ultimo nos dice que el punto P se transforma en una senoidal en el plano (t, θ)

Entonces resulta que $s = r\cos(\theta - \phi)$





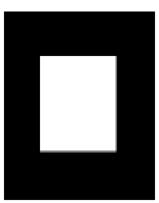
Matlab Support

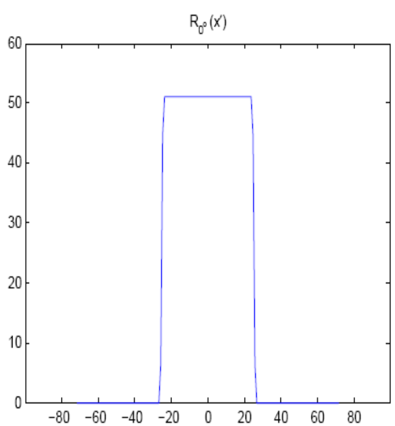
RADON TRANSFORM

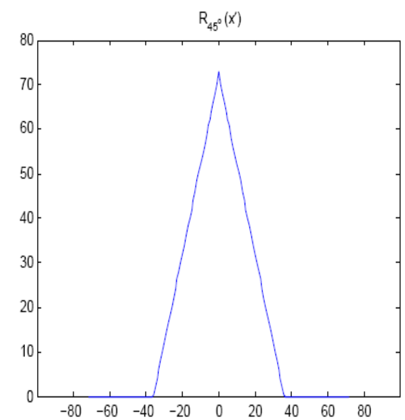
Obtiene las proyecciones a partir de la imagen

```
[R,xp] = radon(I,theta);
```

```
I = zeros(100,100);
I(25:75, 25:75) = 1;
imshow(I)
[R,xp] = radon(I,[0 45]);
figure; plot(xp,R(:,1)); title('R_{0^o} (x\prime)')
```







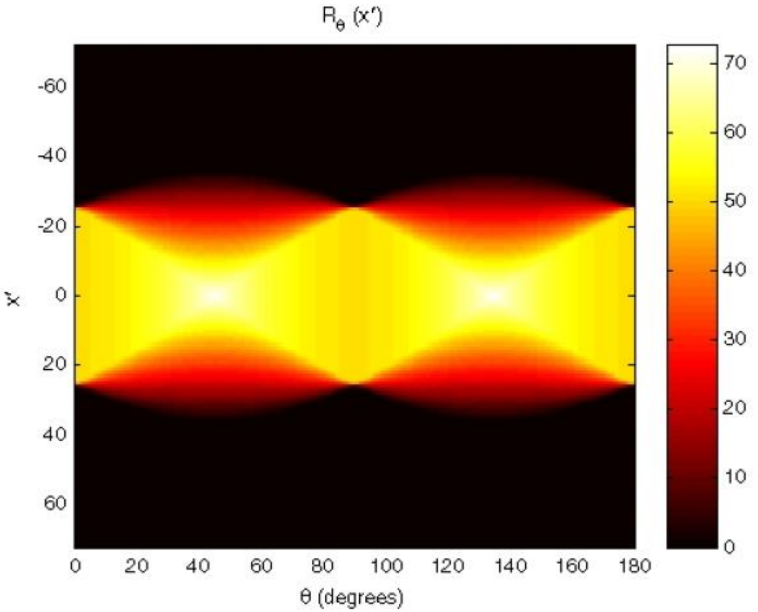

```
figure; plot(xp,R(:,2)); title('R_{45^o} (x\prime)')
```

Matlab Support

RADON TRANSFORM

Para ver la transformación para varios ángulos se suele verlas como una imagen.

```
theta = 0:180;
[R,xp] = radon(I,theta);
imagesc(theta,xp,R);
title('R_{\theta} (X\prime)');
xlabel('\theta (degrees)');
ylabel('X\prime');
set(gca,'XTick',0:20:180);
colormap(hot);
colorbar
```



Matlab Support

INVERSE RADON TRANSFORM

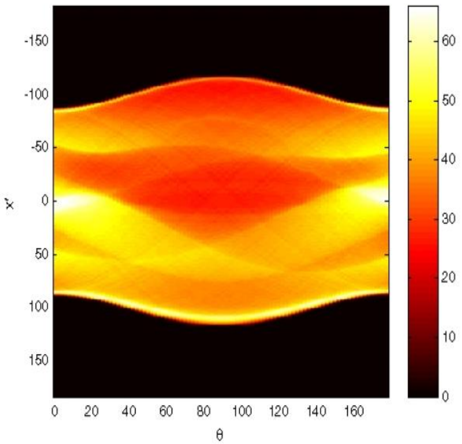
Podemos reconstruir la imagen a partir de las proyecciones.

```
IR = iradon(R,theta);
```

```
P = phantom(256);
imshow(P)
```

```
theta1 = 0:10:170; [R1,xp] = radon(P,theta1);
theta2 = 0:5:175;  [R2,xp] = radon(P,theta2);
theta3 = 0:2:178;  [R3,xp] = radon(P,theta3);
```

```
figure, imagesc(theta3,xp,R3); colormap(hot); colorbar
xlabel('\theta'); ylabel('x\prime');
```



Matlab Support

INVERSE RADON TRANSFORM

Podemos reconstruir la imagen a partir de las proyecciones.

```
IR = iradon(R,theta);
```

```
I1 = iradon(R1,10);
I2 = iradon(R2,5);
I3 = iradon(R3,2);
imshow(I1)
figure, imshow(I2)
figure, imshow(I3)
```