Trabajo Práctico 3

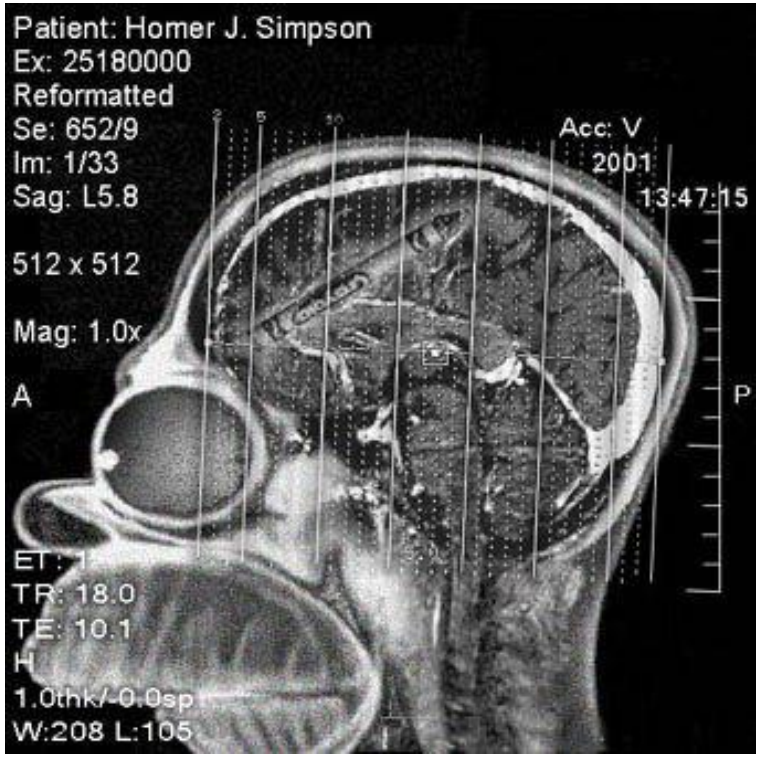
Tomografía computada

Luis Greco - Marcela Herrera - Cristian Kubrak - Alejo Salvador

5 de julio de 2018

INTRODUCCIÓN

- El objetivo de este trabajo práctico es evaluar un método para reconstruir imágenes tomográficas sujetas a ruido, utilizando el método de aproximación por cuadrados mínimos.



PREMISAS

- Nuestro "sujeto" es una imagen de nxn pixeles discretizada en celdas de kxk pixeles.
- La intensidad de un pixel se asocia al tiempo que demora un rayo en atravesar ese pixel.
- La distancia que recorre un rayo que atraviesa al sujeto es igual a la cantidad de pixeles por los que pasa.
- En cinemática $Velocidad = \frac{Espacio}{Tiempo}$. En este caso la "velocidad" promedio dentro de la celda k es:

$$v_k = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} I_{ij}}{n * n}$$

- Los rayos están sujetos a ruido.

Discretización

- Utilizamos valores divisores de la dimensión de la imagen (100 x 100 pixeles).
- En los tests de granularidad, variamos el tamaño de las celdas tomando los valores 4x4, 5x5, 10x10, 20x20, 25x25 y 50x50 pixeles.
- A medida que se achica el tamaño de las celdas crece el tiempo de procesamiento, lo cual condicionó la elección de los parámetros.

Recorrido de un rayo

- Los rayos son considerados como rectas en el plano, caracterizados por un punto de origen que llamamos (x_0, y_0) y un ángulo α.
- Se busca la intersección con la recta $y = 0$ la cual es el punto $(x - y/tangent(\alpha), 0)$.
- Usando la ecuación de la recta $y = tan(\alpha) * x + y_0$ y variando el valor de x, se calculan los puntos del plano por los que pasa el rayo.
- Si el punto hallado pertenece a la imagen se guardan las coordenadas del pixel correspondiente en un vector.
- Se usa el vector para el cálculo de distancias y de tiempos.

Recorrido de un rayo

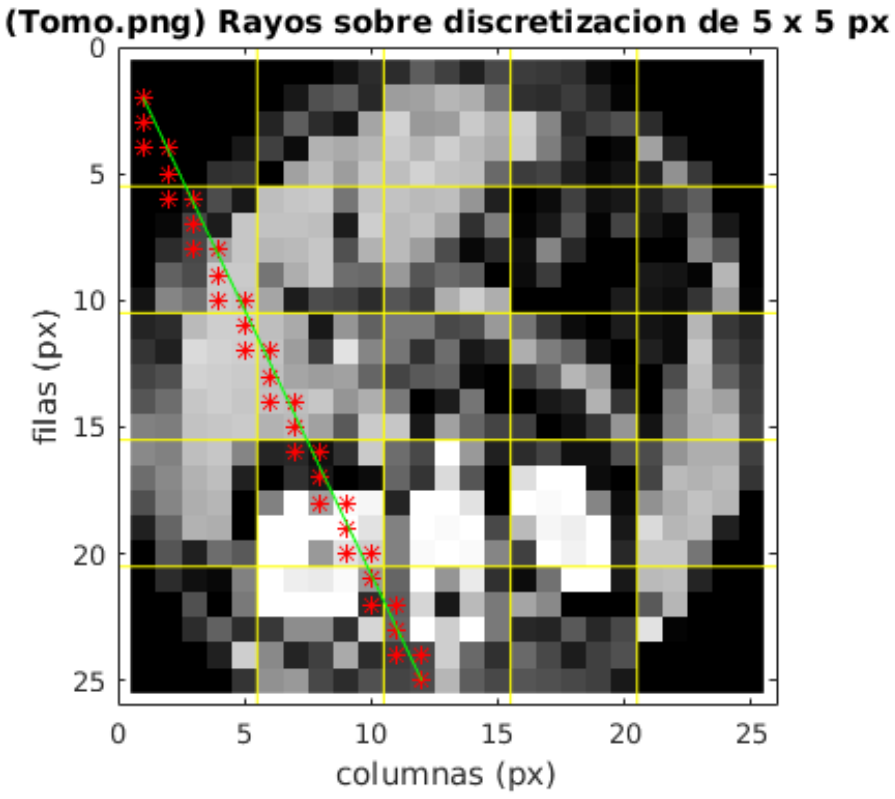


Figura: Pixeles/Celdas por la que pasa un rayo

Distancia y tiempo de recorrida de un rayo

- A partir del vector de pixeles por los que pasa un rayo, se averigua a qué celda pertenece. Se suma 1 unidad por pixel. El resultado se almacena en la matriz de distancias D (una fila por rayo, una columna por celda numeradas de arriba hacia abajo y de izquierda a derecha).
- Cada rayo toca a lo sumo $2n - 1$ celdas. Se usa la estructura de matriz esparza del TP1.
- A partir del vector de pixeles por los que pasa un rayo, se acumulan las intensidades de los mismos. Se almacenan los resultados en un vector de "tiempos" (un elemento por rayo).

$$D = \begin{bmatrix} d_1^1 & d_2^1 & \dots & d_{r^2}^1 \\ \dots & \dots & \dots & \dots \\ d_1^k & d_2^k & \dots & d_{r^2}^k \\ \dots & \dots & \dots & \dots \\ d_1^m & d_2^m & \dots & d_{r^2}^m \end{bmatrix}, t = \begin{bmatrix} t^1 \\ t^2 \\ \dots \\ t^i \\ \dots \\ t^{nxn} \end{bmatrix}$$

Sistema de ecuaciones

Sea v es el vector de la velocidades por celda. Tenemos que:

$$v^t = \left[\; v^1 v^2 ... v^i ... v^m \; \right]$$

El tiempo que tarda el rayo k en atravesar al sujeto es:

$$t[k] = \sum_{j=1}^{nxn} t_j^k = \sum_{j=1}^{nxn} \frac{d_j^k}{v[j]} \Rightarrow Dv = t$$

Como la matriz D es rectangular con m>>nxn, se resuelve por cuadrados mínimos usando las ecuaciones normales:

$$Dv = t \Rightarrow D^t Dv = D^t t, D^t D \in R^{n^2 x n^2}$$

Generación de rayos

Cantidad y ubicación de los emisores

1. Primeras opciones:
   - Trazar rayos horizontales y verticales, formando una cuadrícula $\Rightarrow$ Filas de ceros.
   - Trazar rayos saliendo de los cuatro vértices de la imagen en distintos direcciones para barrer los 90° de cada ángulo $\Rightarrow$ Información similar, imágenes irreconocibles.
2. Elección final:
   - Trazar rayos desde los cuatro laterales de la imagen.
   - Los emisores se ubican en igual cantidad sobre los lados de la imagen.
   - La ubicación de los emisores se selecciona aleatoriamente con distribución uniforme entre 0 y k (k cantidad de pixeles por lado de la imagen.
   - Igual cantidad de rayos en ángulos que van entre 0° a 180°.
   - Ángulos seleccionados aleatoriamente con distribución

Generación de rayos

Cantidad y ubicación de los emisores

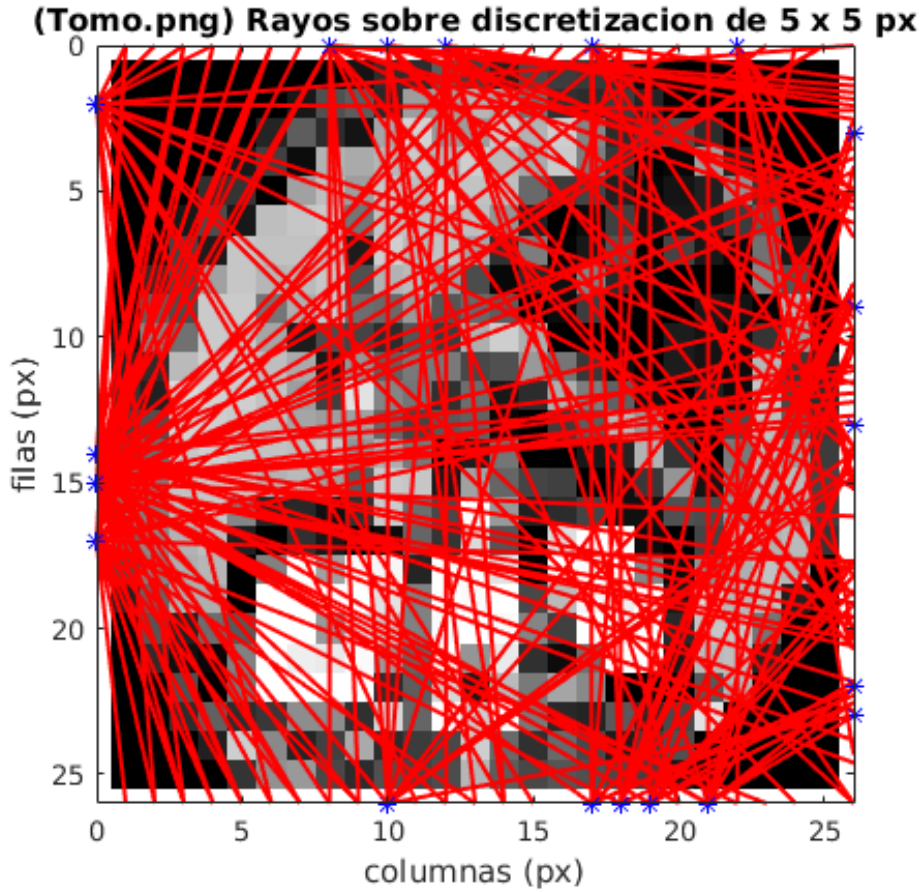
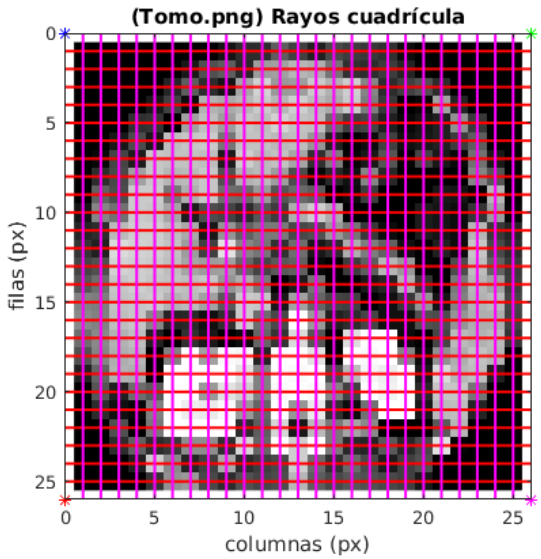


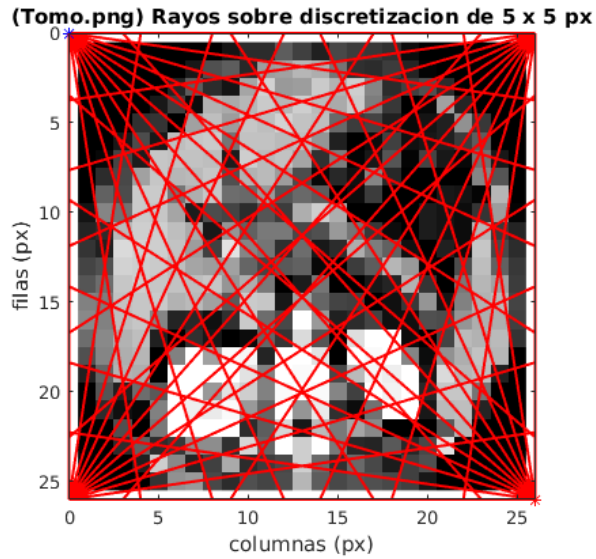
Figura: Trazado aleatorio de rayos

Generación de rayos

Casos fallidos



(a) Cuadrícula



(b) Desde los vértices

Figura: Casos fallidos de distribución de emisores, sin ruido.

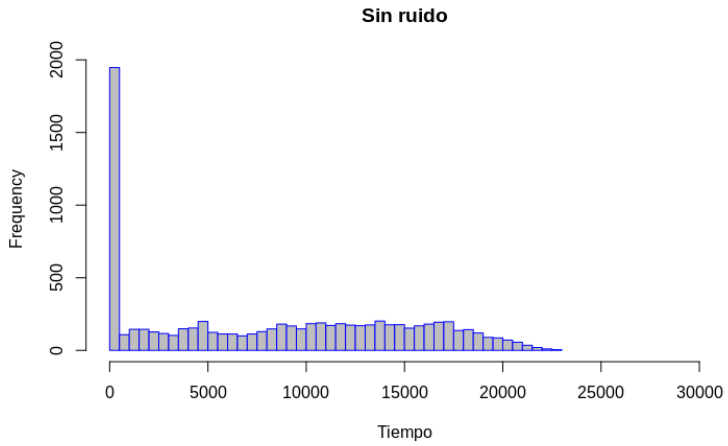
Ruido

- Lo que buscamos fue perturbar los datos con los que trabajamos a modo de probar qué tan robusto es nuestro sistema
- Para eso utilizamos ruido gaussiano con distribución $\mathcal{N}$ (0, 1), el más usual.
- Lo aplicamos al vector que contiene los tiempos del recorrido de cada rayo
- ¿Por qué lo aplicamos ahora y antes sobre la imagen original?

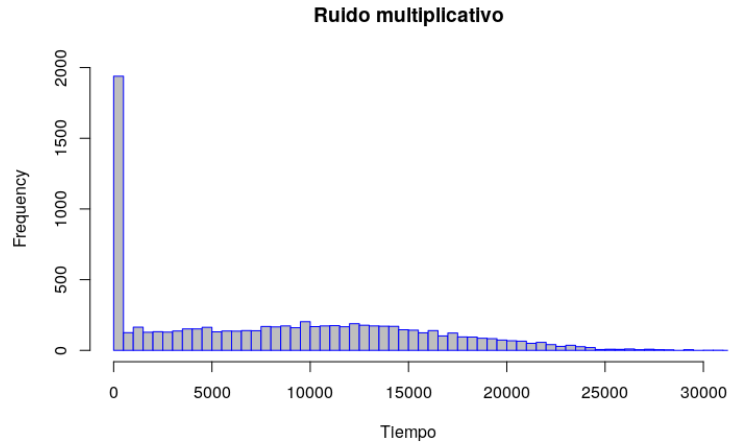
Ruido multiplicativo vs aditivo

- Una decisión a la hora de aplicar el ruido fue si tenía que ser aditivo (es decir, sumando una constante por un número aleatorio) o multiplicativo (proporcional al número al que le vamos a agregar ruido).
- Siendo a_i un elemento del vector de tiempos, r_i un elemento del vector de ruidos, k una constante y α el paramentro de entrada:
  $a_i + (a_i * r_i * \alpha)$ vs $a_i + (a_i * k * \alpha)$

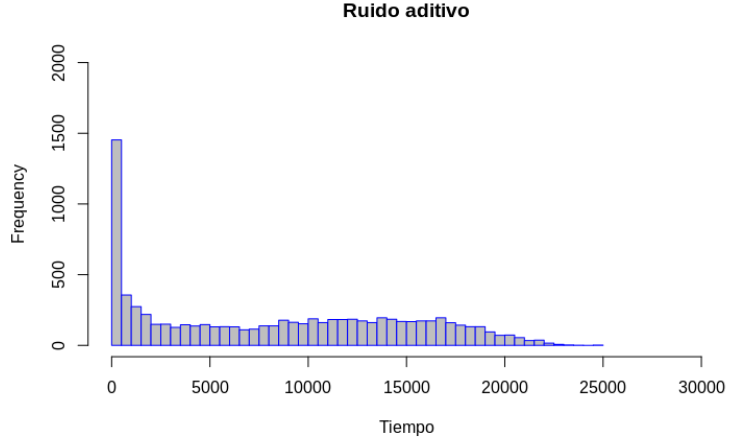
Ruido multiplicativo vs aditivo



(a) Tiempos originales

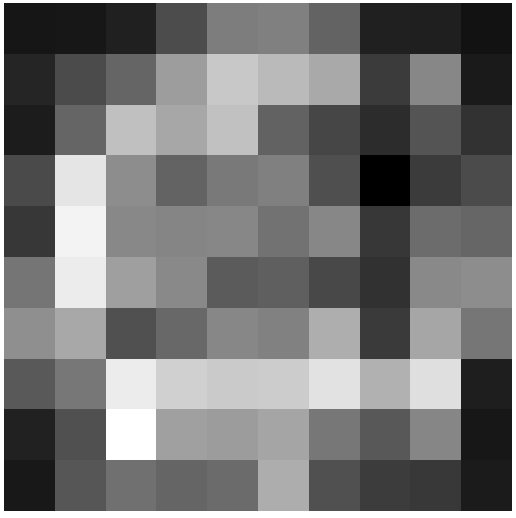


(b) Tiempos con ruido multiplicativo

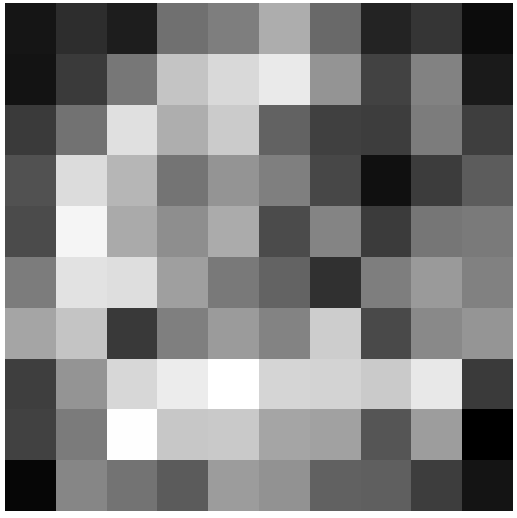


(c) Tiempos con ruido aditivo

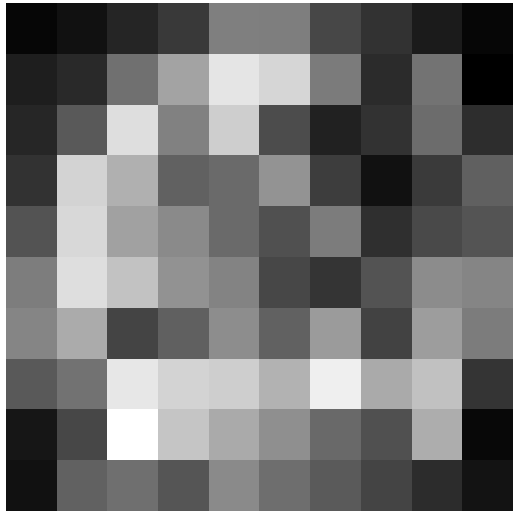
Comparación de resultados



(a) $\alpha = 0.001$



(b) $\alpha = 0.005$



(c) $\alpha = 0.01$

Imagen original

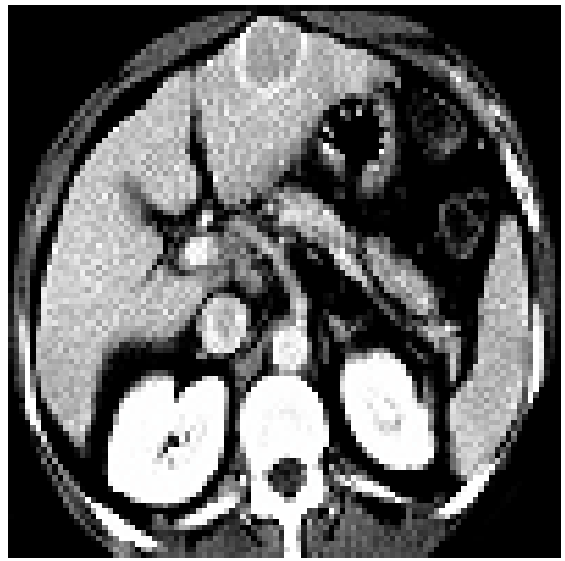


Figura: Imagen original

ECM vs. tamaño de la celda

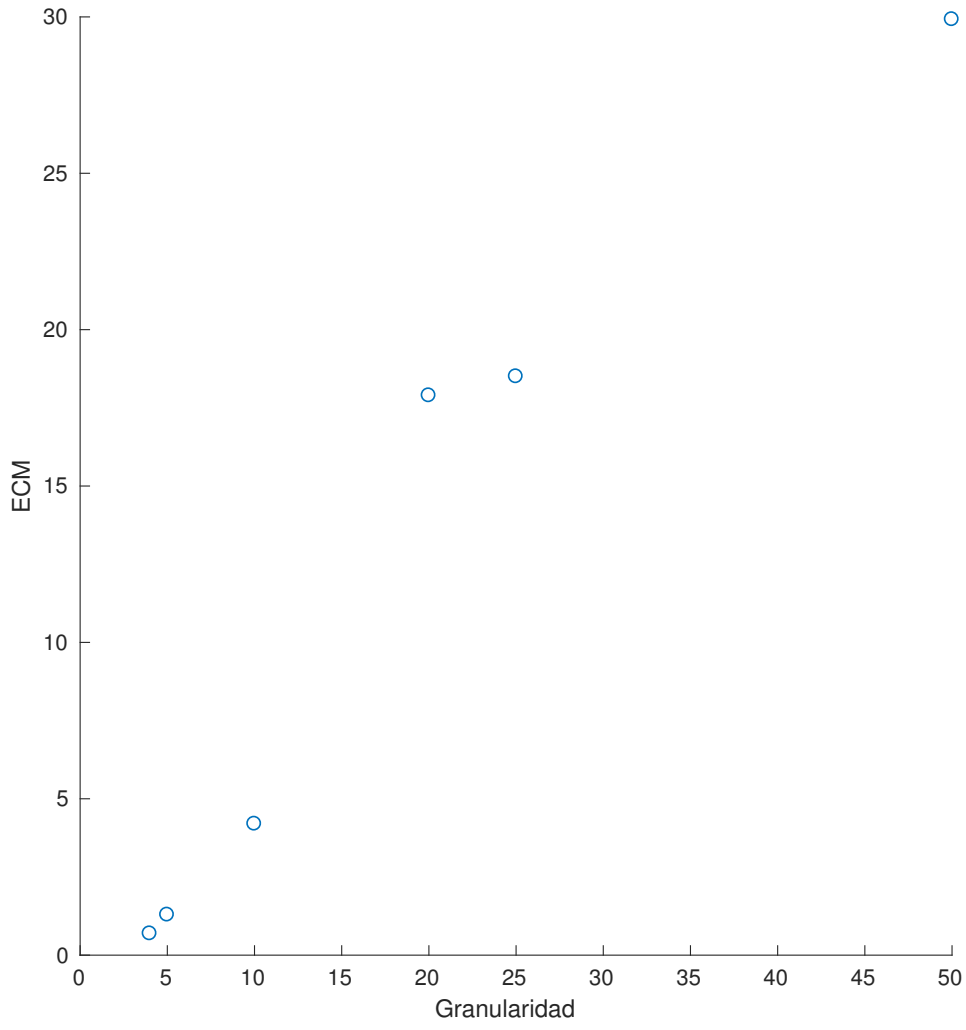
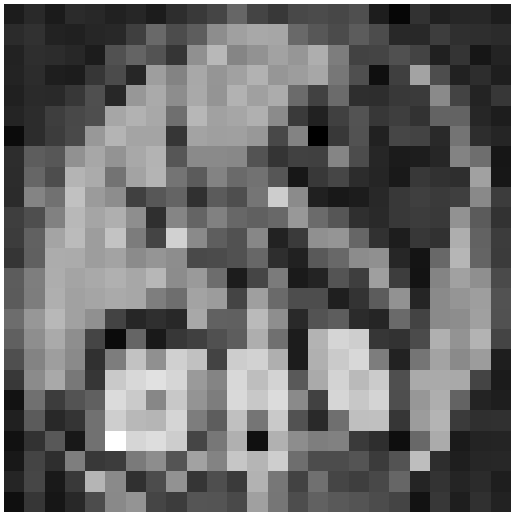
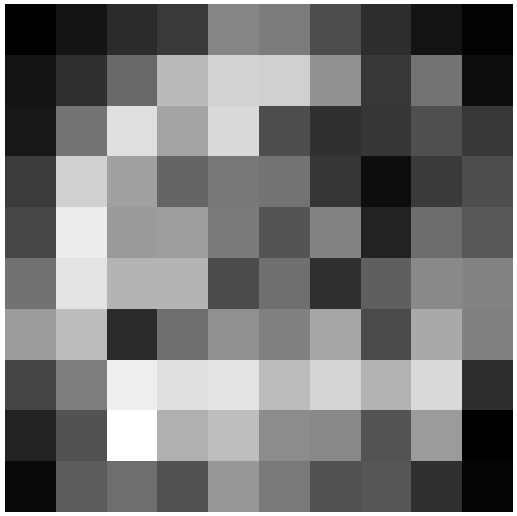


Figura: Error cuadrático medio versus granularidad

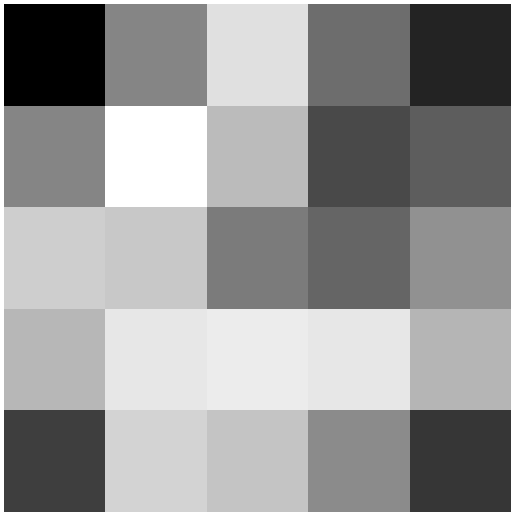
ECM vs. tamaño de la celda



(a) 4x4px por celda



(b) 10x10px por celda



(c) 20x20px por celda

Figura: Reconstrucción variando granularidad, sin ruido

ECM vs. cantidad de emisores

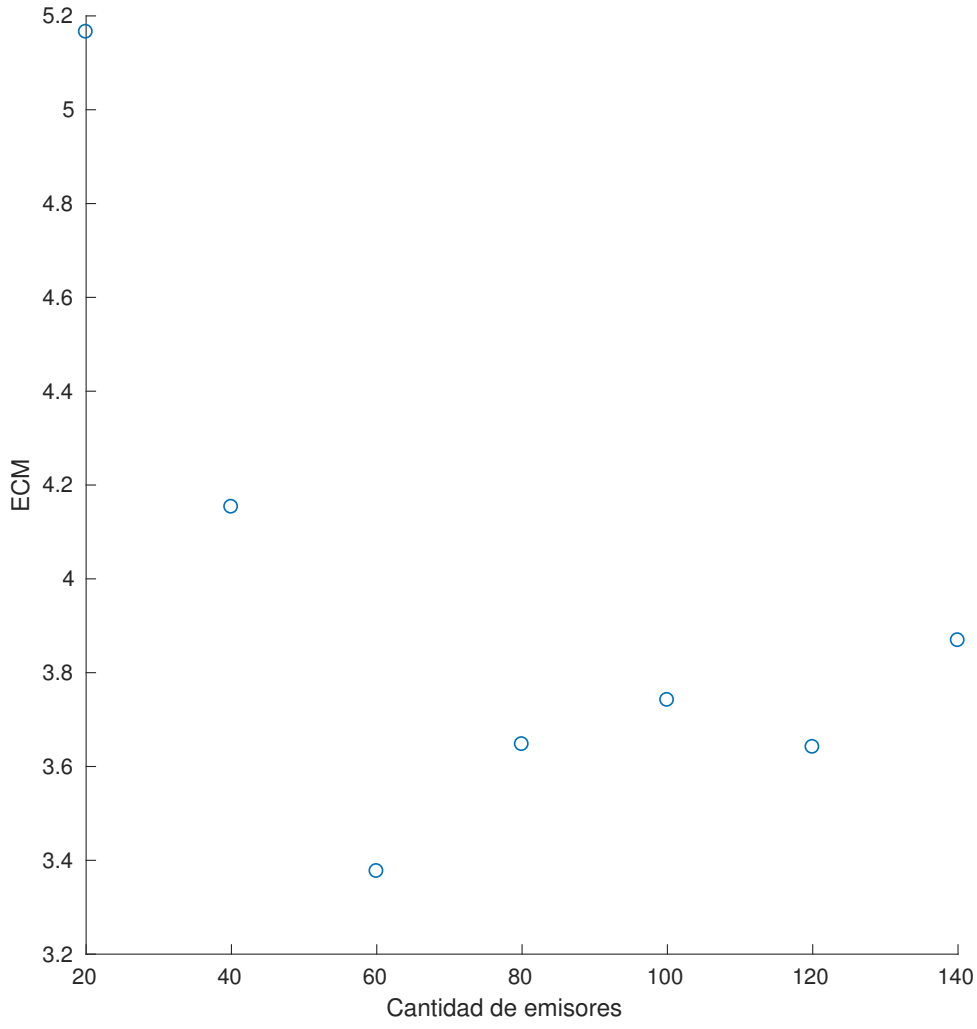
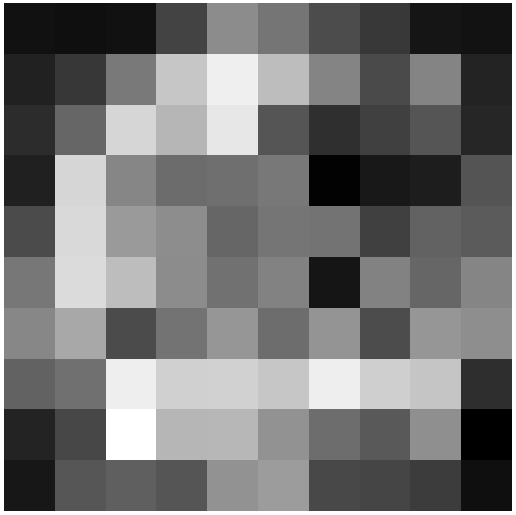
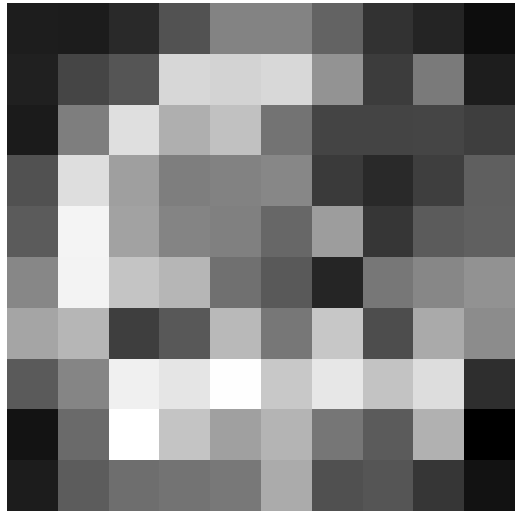


Figura: ECM versus cantidad de emisores

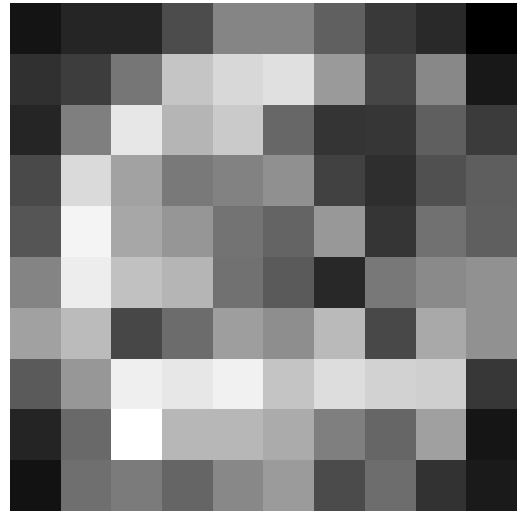
ECM vs. cantidad de emisores



(a) reconstrucción 20 emisores



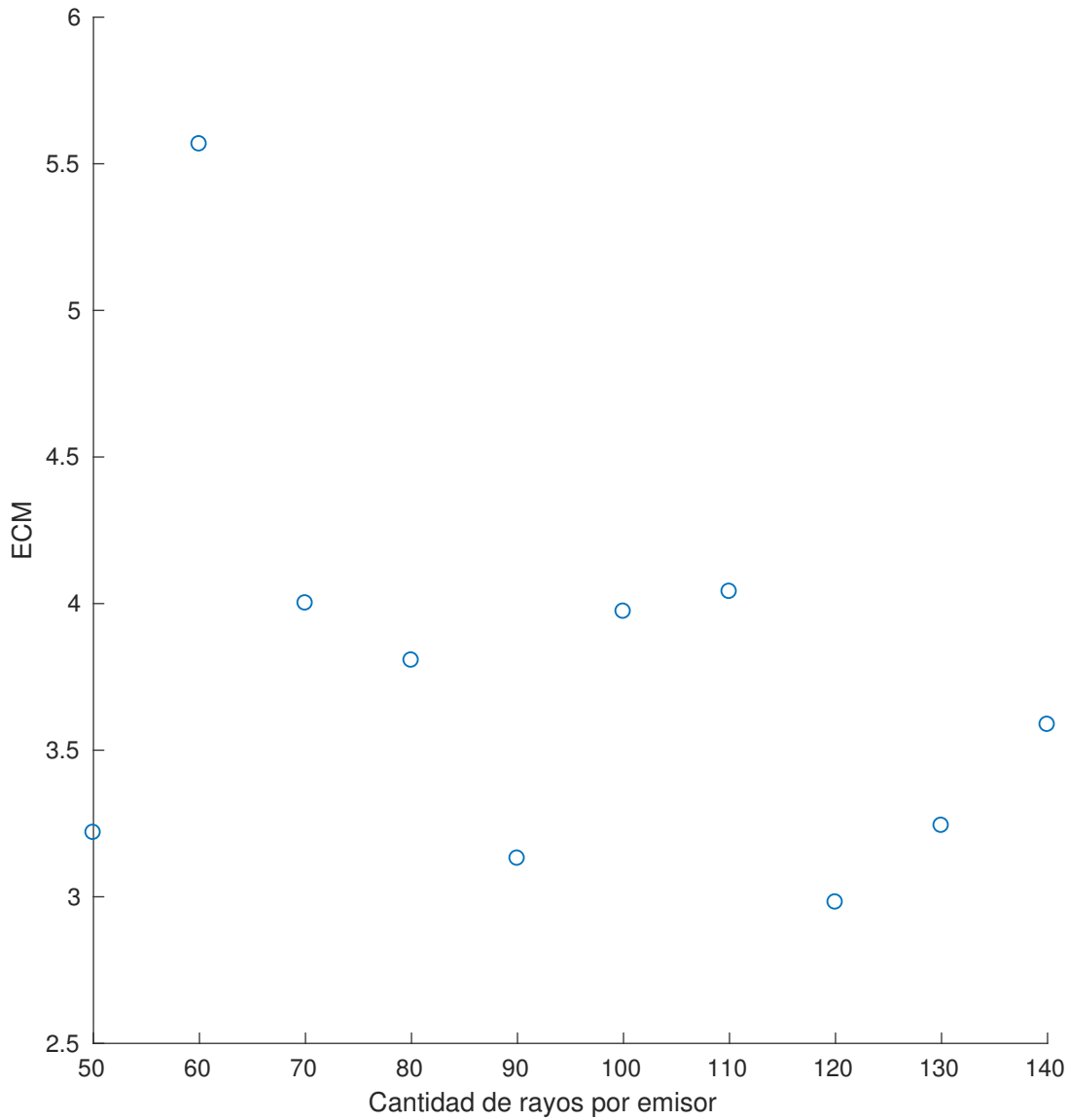
(b) reconstrucción 60 emisores



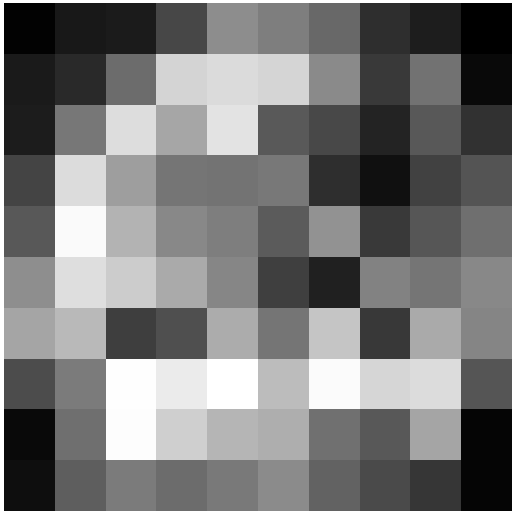
(c) reconstrucción 140 emisores

Figura: Reconstrucción variando cantidad de emisores, sin ruido

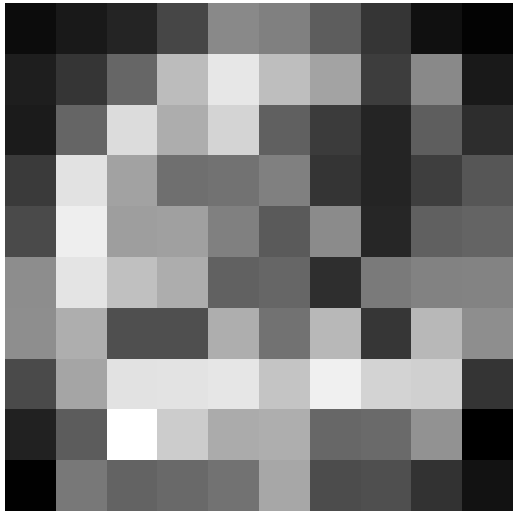
ECM vs. cantidad de rayos por emisor



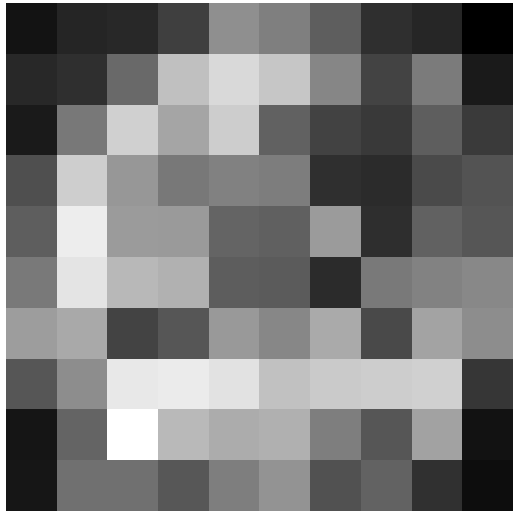
ECM vs. cantidad de rayos por emisor



(a) reconstrucción 50 rayos



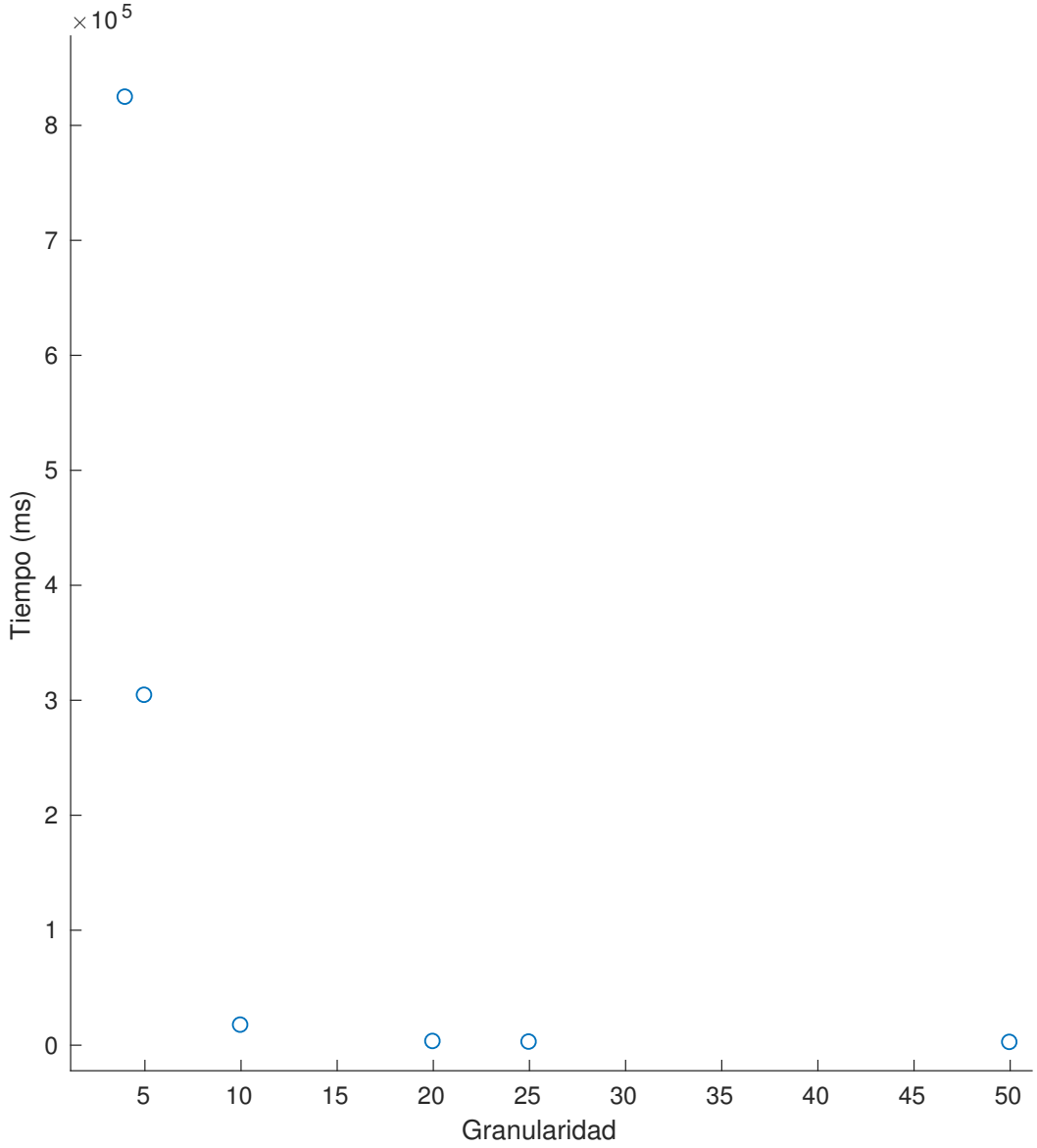
(b) reconstrucción 100 rayos



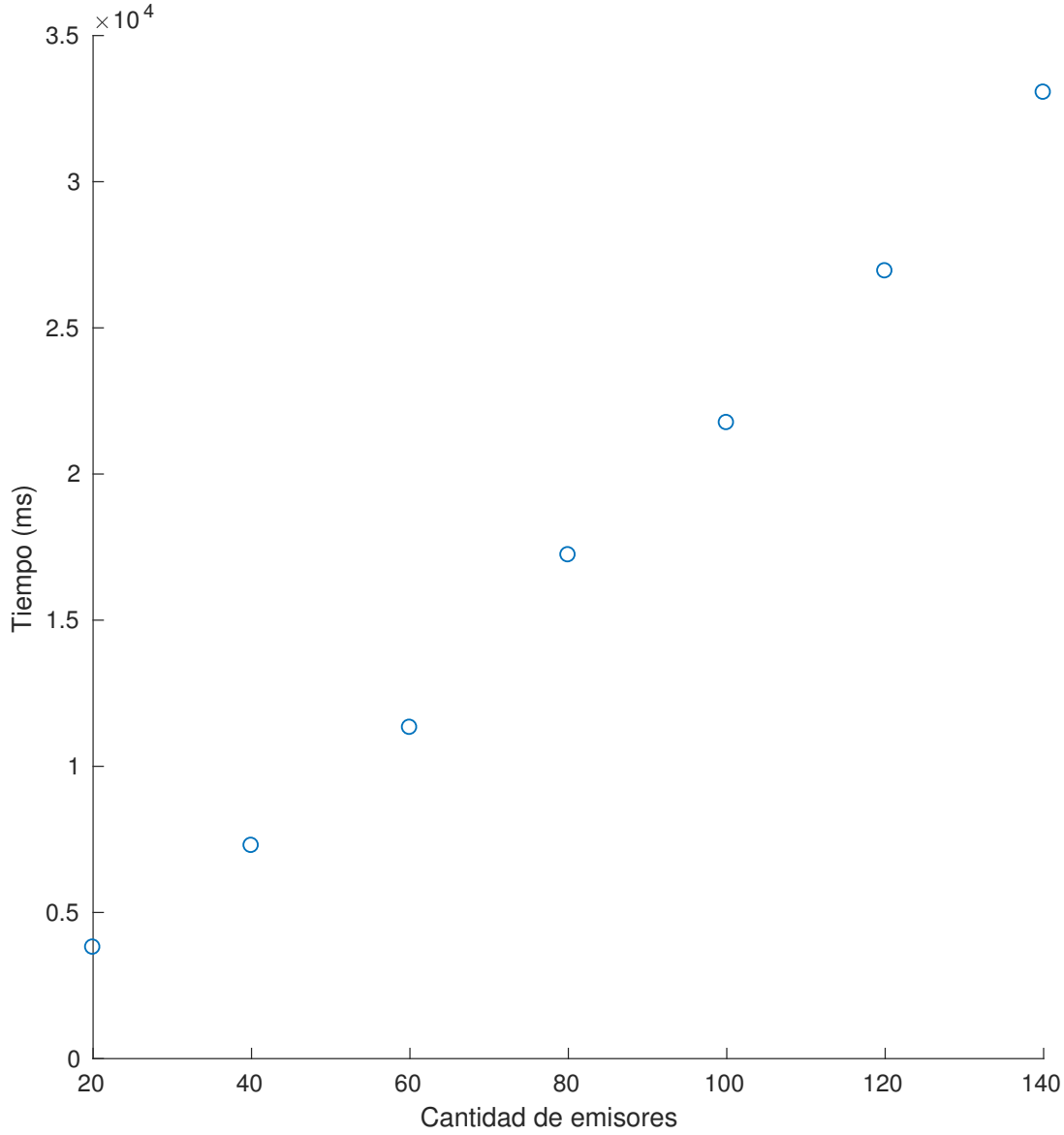
(c) reconstrucción 140 rayos

Figura: Reconstrucción variando cantidad de rayos, sin ruido

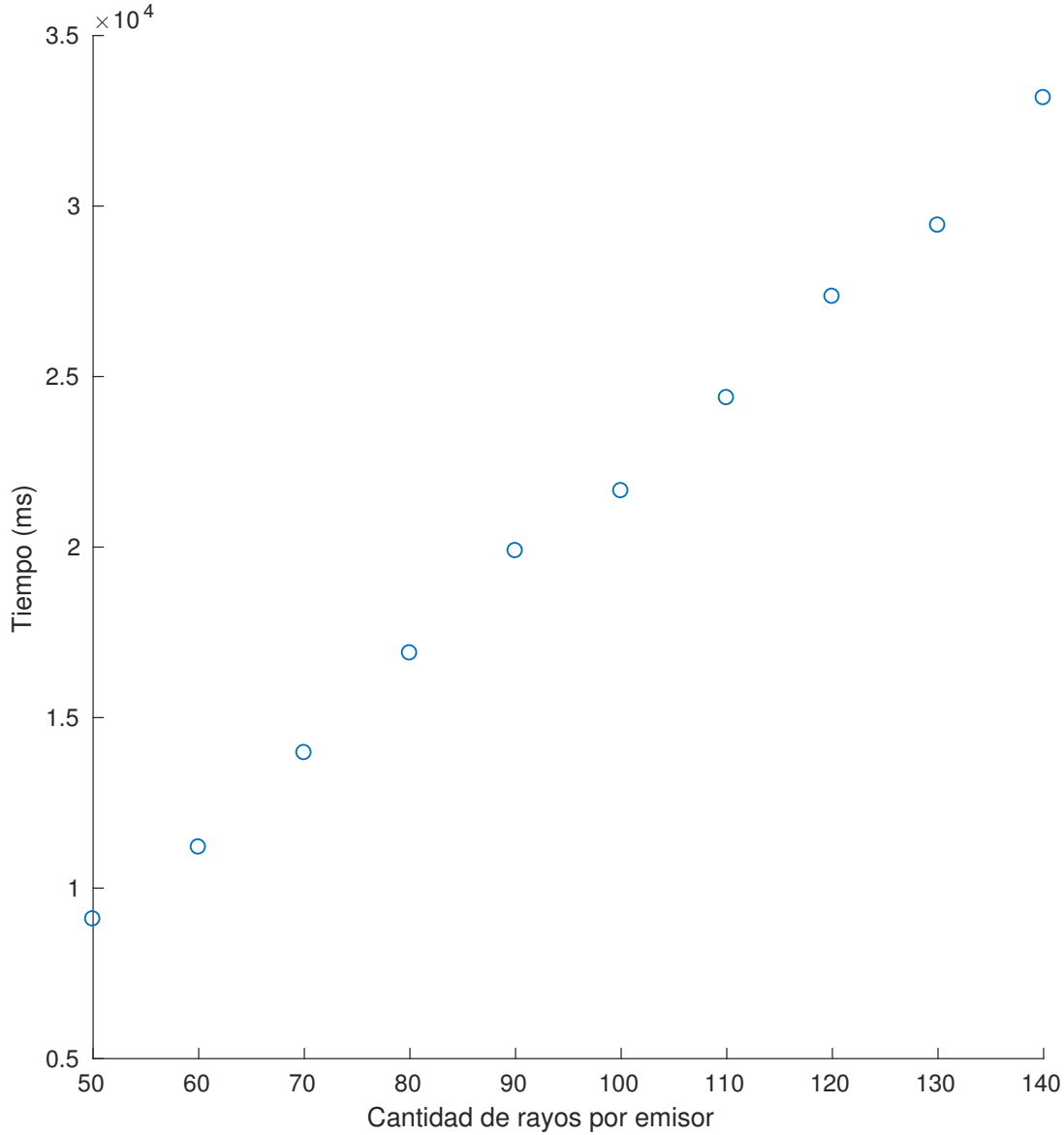
Tiempo vs. granularidad



Tiempo vs. cantidad de emisores



Tiempo vs. cantidad de rayos por emisor



Conclusiones e ideas a futuro

- ¿Qué nos hubiera gustado mejorar?
  - Minimizar la cantidad de rayos necesarios para lograr una reconstrucción aceptable
  - Otra mejora sería una implementación que no utilice elementos aleatorios, esto haría que sea más plausible una implementación real de este sistema.
  - Sería interesante analizar qué otros tipos de ruido pueden presentarse en este tipo de trabajos y analizar la robustez del sistema en esos casos (por ejemplo sal y pimienta, ruido periorico, etc).

REFERENCIAS

FIN