



# Recuperación de Información Multimedia

## Detección de Líneas

**CC5213 – Recuperación de Información Multimedia**

Departamento de Ciencias de la Computación

Universidad de Chile

Juan Manuel Barrios – <https://juan.cl/mir/> – 2019



# Detección de Líneas

- Una vez detectados puntos de bordes en una imagen es común localizar puntos co-lineales o líneas.
  - Ej.: Detección de marcos, detección de figuras geométricas.
- Técnicas para encontrar una o más rectas en un conjunto de puntos candidatos:
  - Template Matching
  - Mínimos cuadrados
  - RANSAC
  - Transformada de Hough

# Template Matching

- Template: Modelo del patrón a buscar.
- Calcular la diferencia o correlación de la imagen (o una zona de ella) con cierta plantilla.
  - Ej.: Convolución con los siguientes kernels:

-1	-1	-1
2	2	2
-1	-1	-1

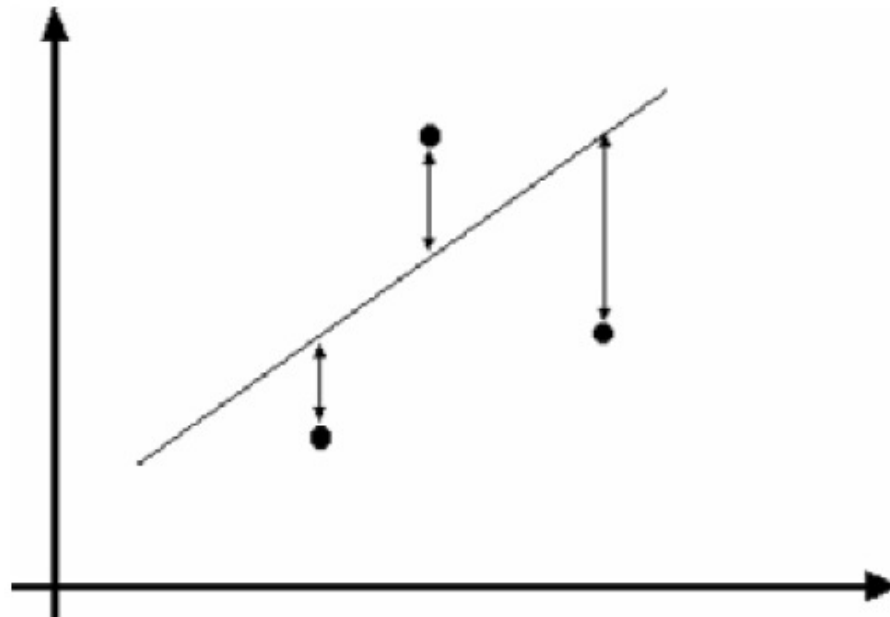
-1	-1	2
-1	2	-1
2	-1	-1

-1	2	-1
-1	2	-1
-1	2	-1

2	-1	-1
-1	2	-1
-1	-1	2

# Mínimos cuadrados

- Encontrar la línea que minimiza el error cuadrático global.
  - Afectado si es que existen outliers





# RANSAC

- Random Sample Consensus
- Para evitar el impacto de outliers, la idea es buscar solo inliers
- Escoger algunos puntos aleatoriamente.
  - Si se escogiera un outlier no se encontrará una recta que tenga mucho apoyo entre los demás puntos.



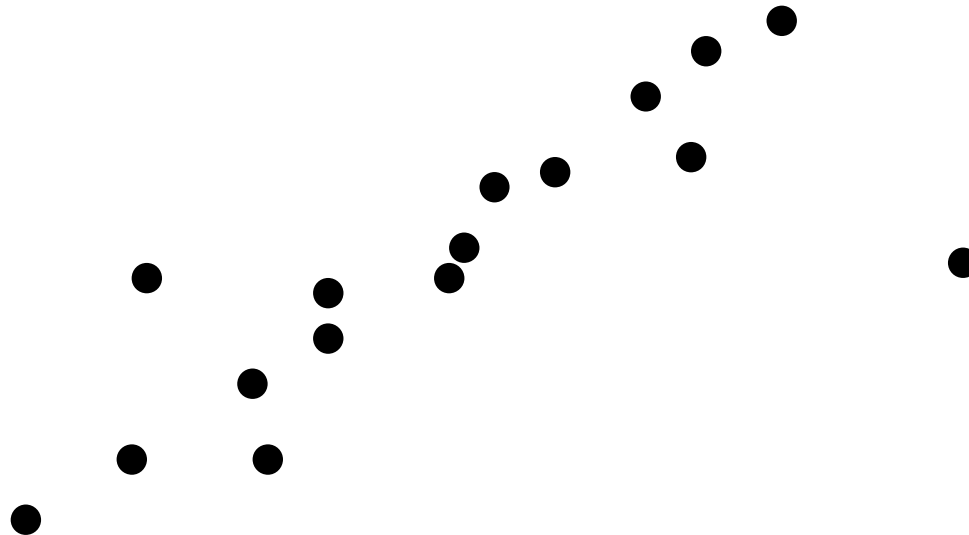
# RANSAC

- Realizar  $N$  veces:
  1. Seleccionar 2 puntos al azar (semillas) y definir la recta que pasa por ellos.
  2. Contar puntos que “apoyen” esa recta (inliers).
  3. Opcional: corregir recta iterativamente:
    - Usando mínimos cuadrados calcular la mejor recta para todos los inliers.
    - Contar los inliers para la recta corregida.
    - Fin de la corrección si no hay cambios en los inliers.
- Cada ciclo se inicia con distintos puntos semillas.
- Quedarse con la recta que tuvo más inliers.
- Con suficientes iteraciones el algoritmo encuentra (probablemente) la mejor recta.



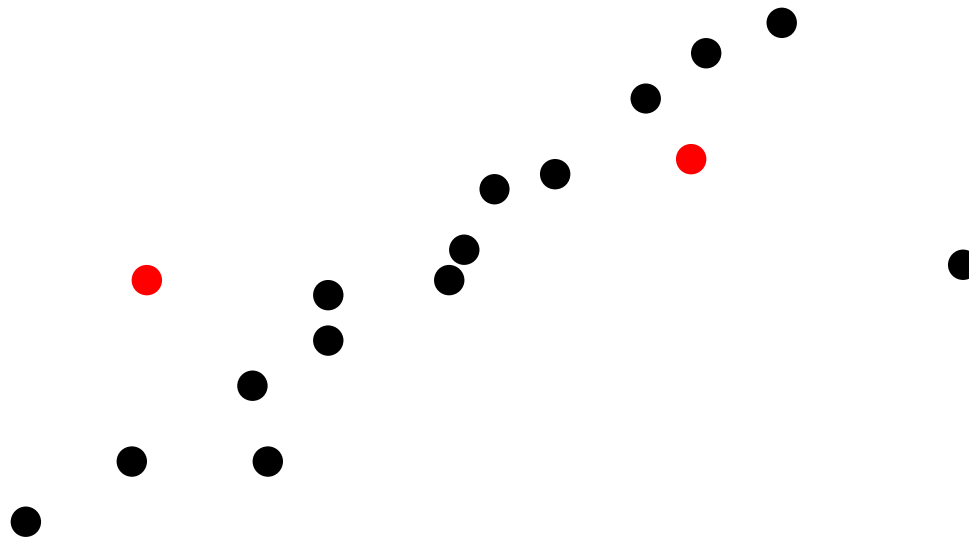
# RANSAC

■ Ejemplo:



# RANSAC

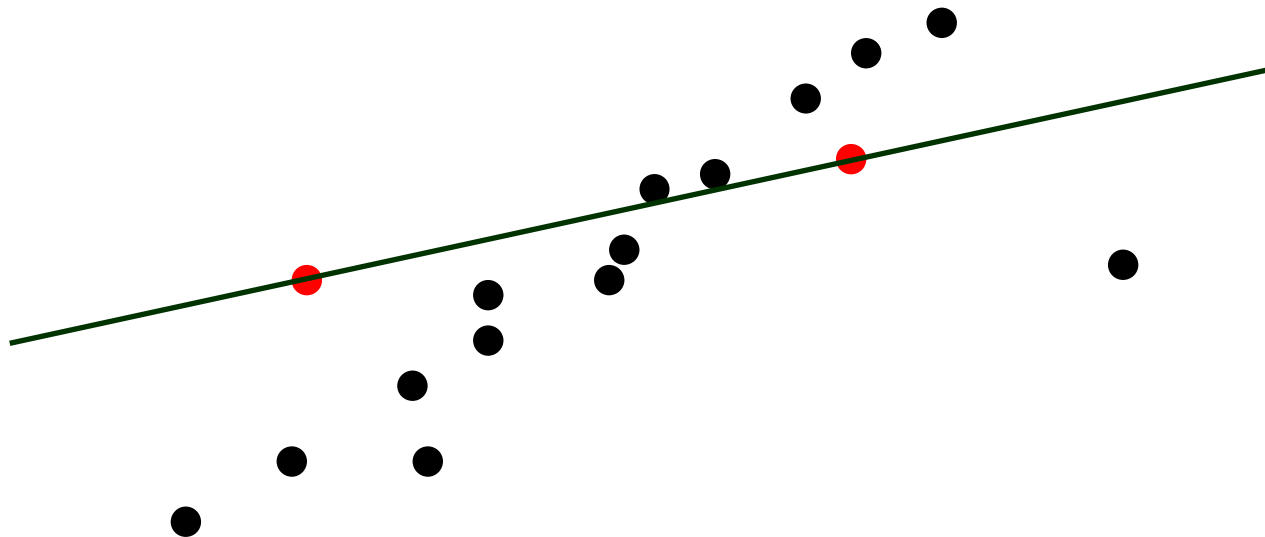
- 1) Seleccionar 2 semillas





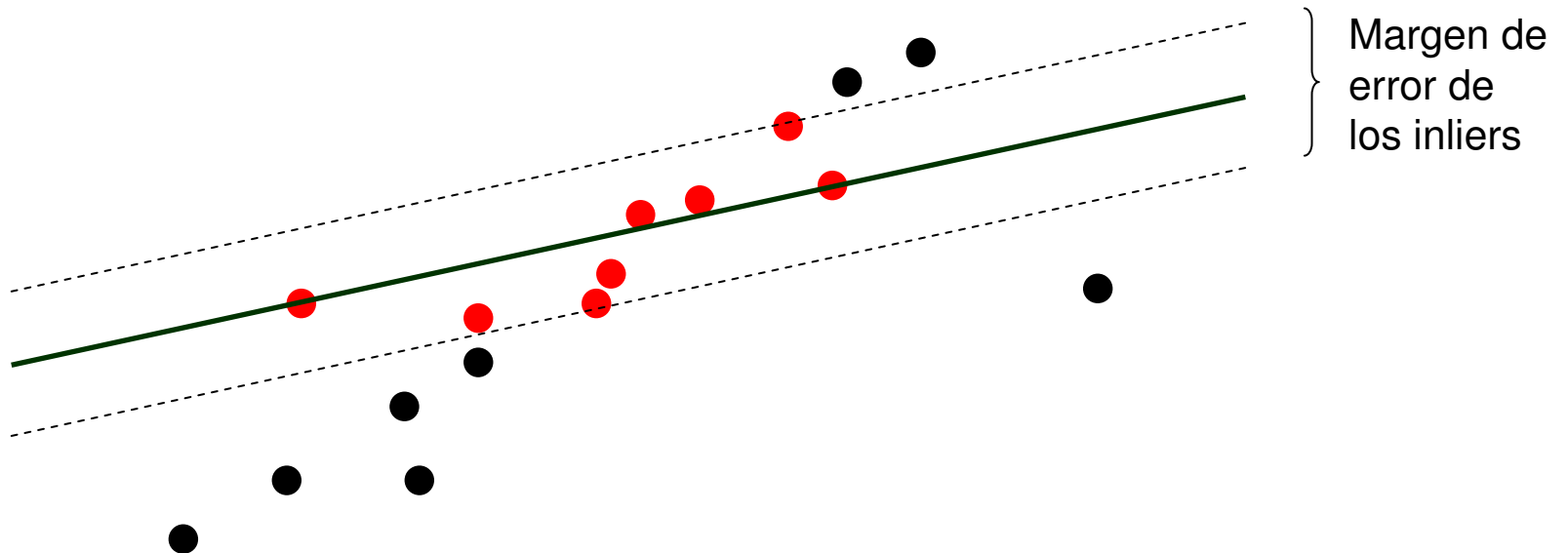
# RANSAC

- 2) Trazar una recta



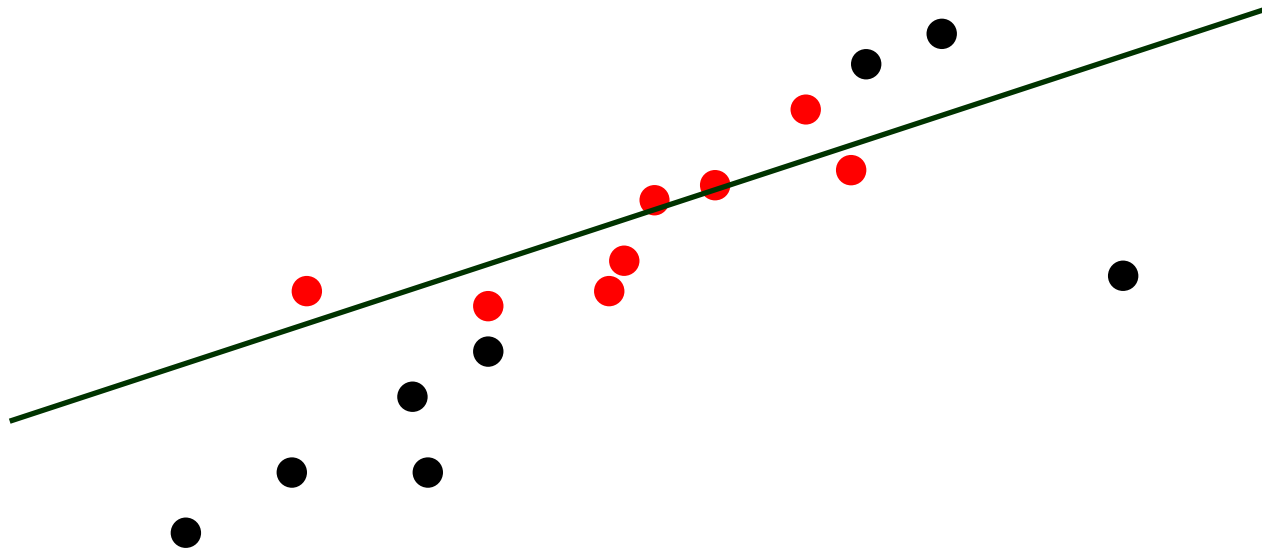
# RANSAC

## ■ 3) Buscar inliers



# RANSAC

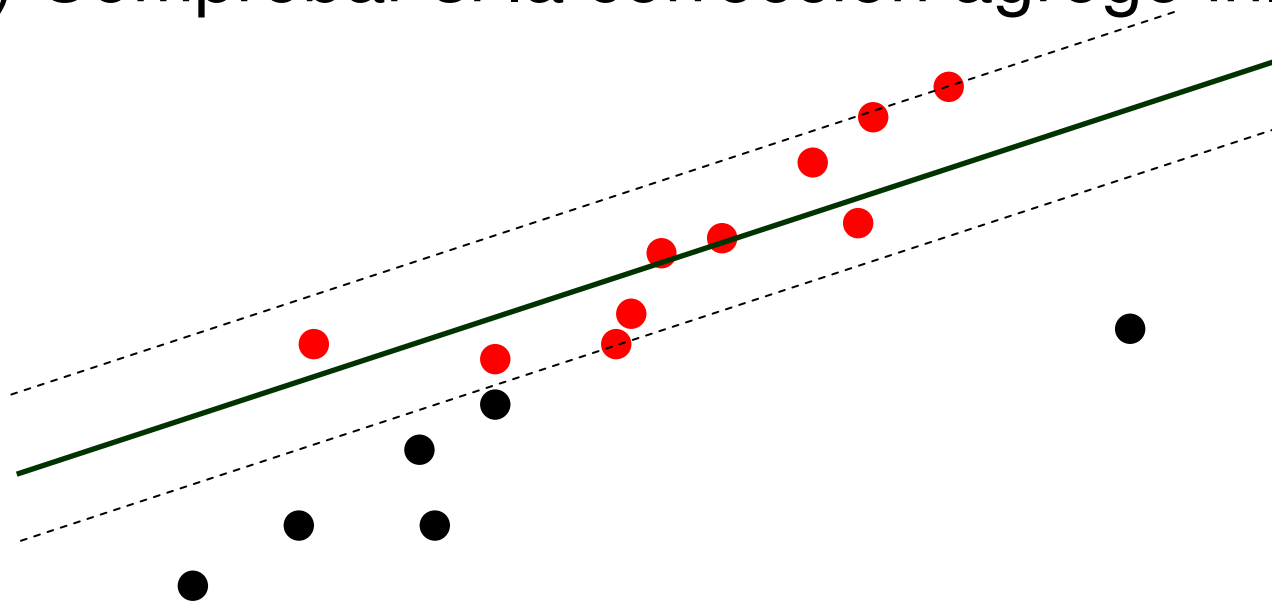
- 4) Corregir la recta según los inliers



- La recta corregida se calcula usando mínimos cuadrados.

# RANSAC

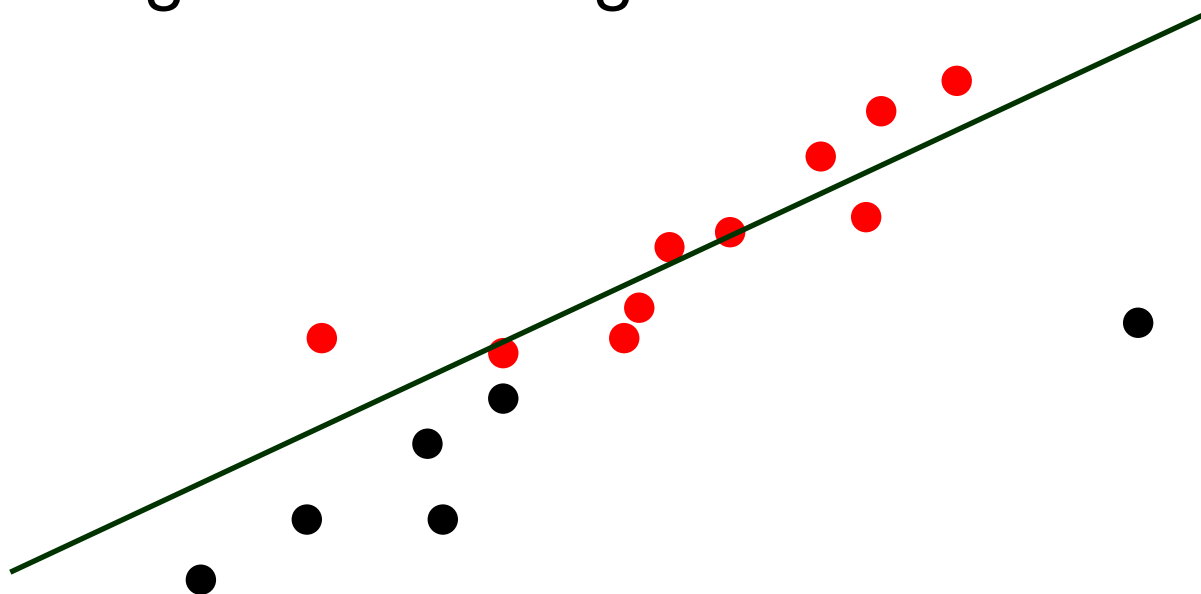
- 5) Comprobar si la corrección agregó inliers



- Hay 2 nuevos inliers → volver al paso 4...

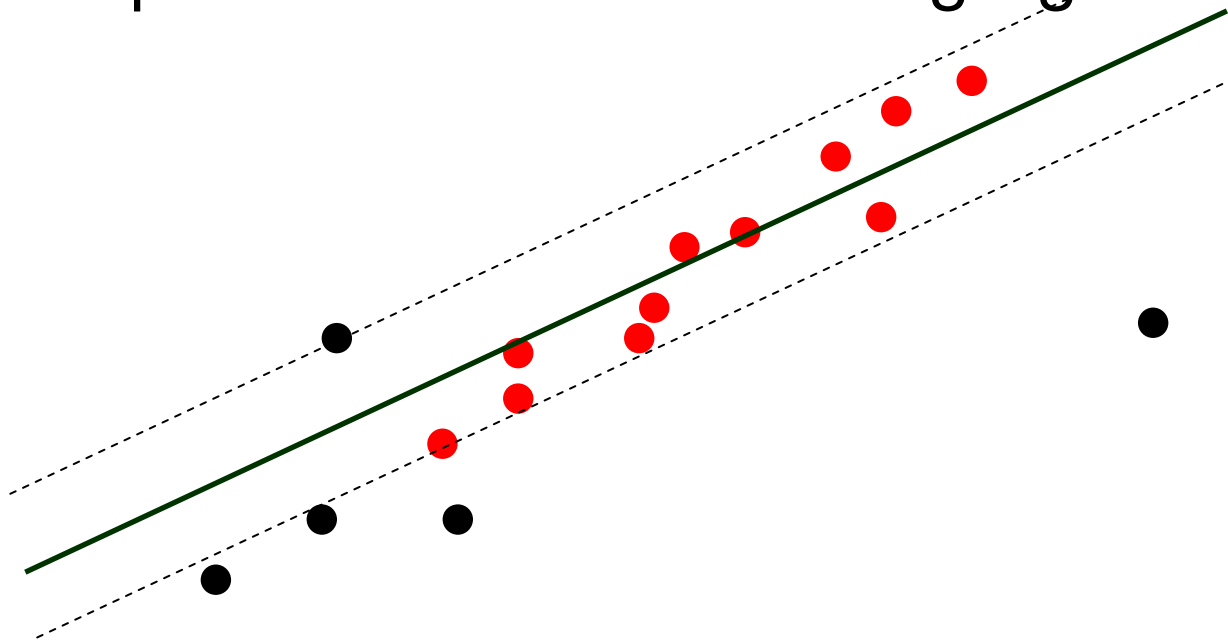
# RANSAC

- Corregir la recta según los inliers



# RANSAC

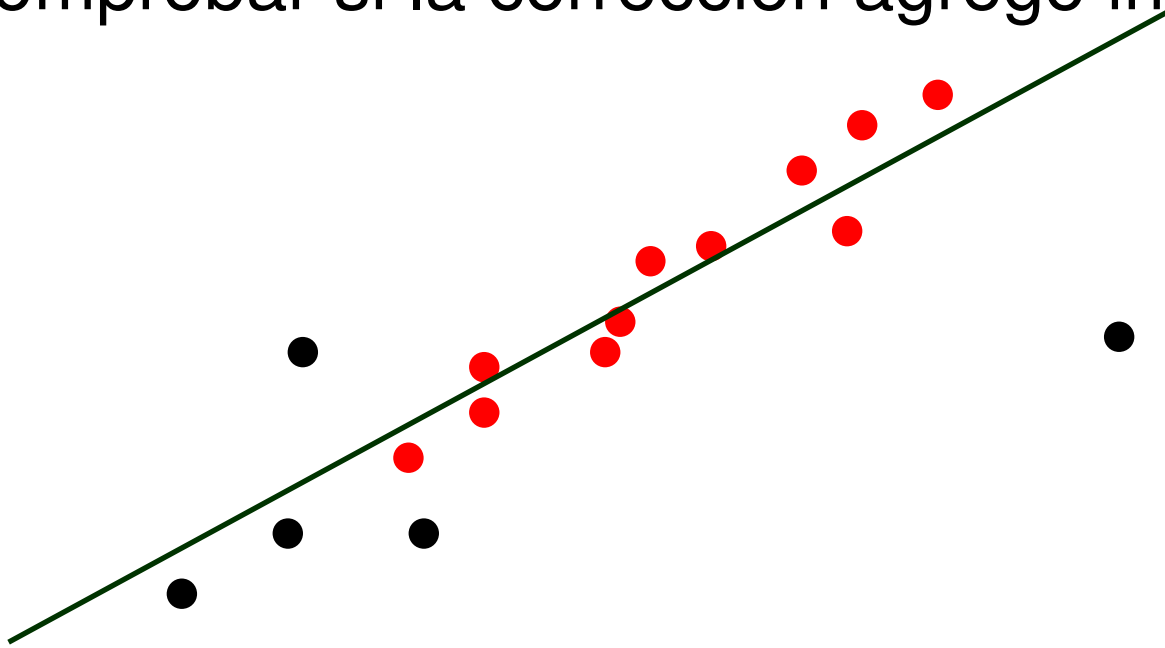
- Comprobar si la corrección agregó inliers



- Entren 2 inliers y sale 1 → volver al paso 4...

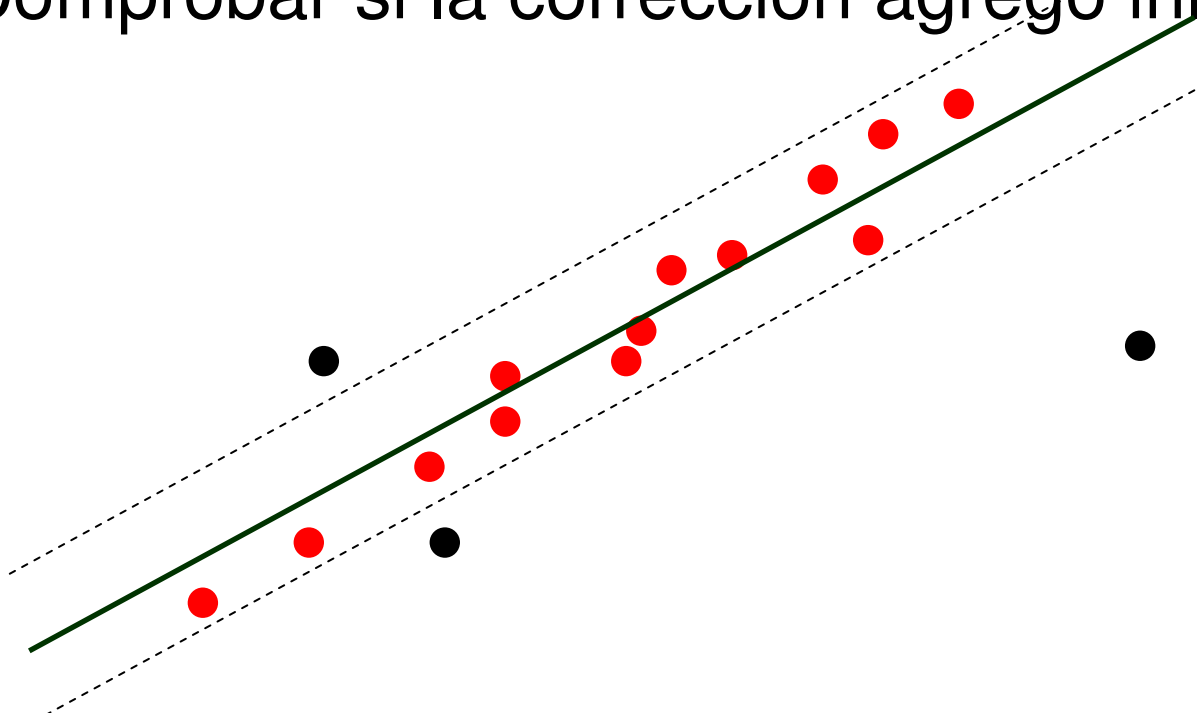
# RANSAC

- Comprobar si la corrección agregó inliers



# RANSAC

- Comprobar si la corrección agregó inliers

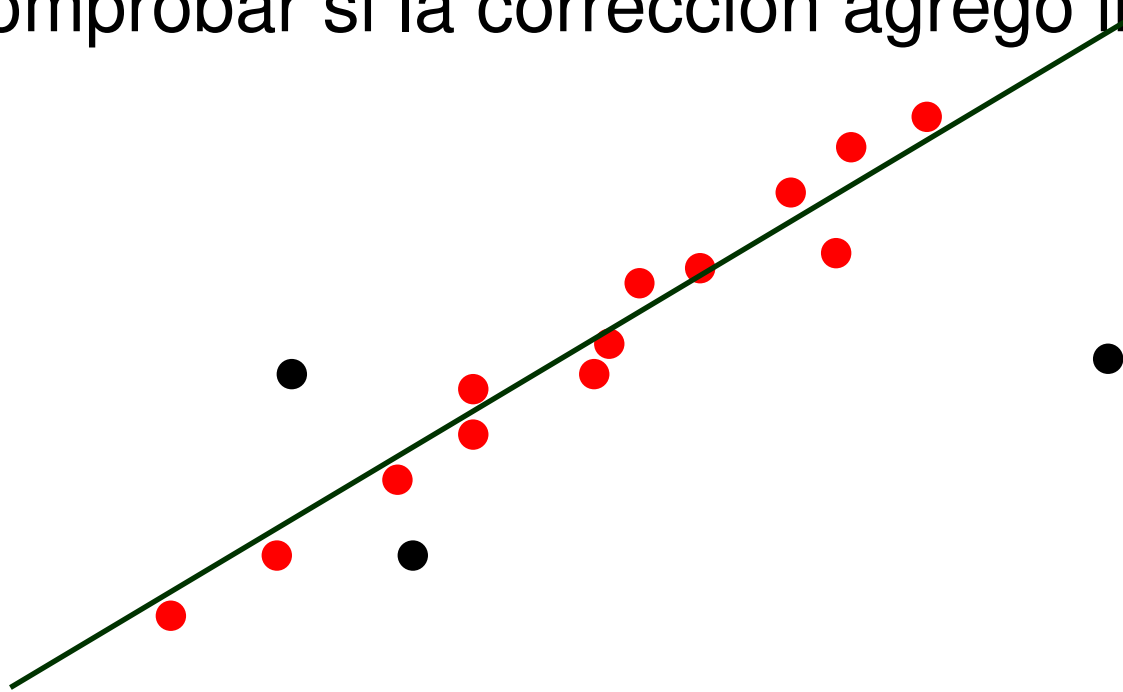


- Entren 2 inliers → volver al paso 4...



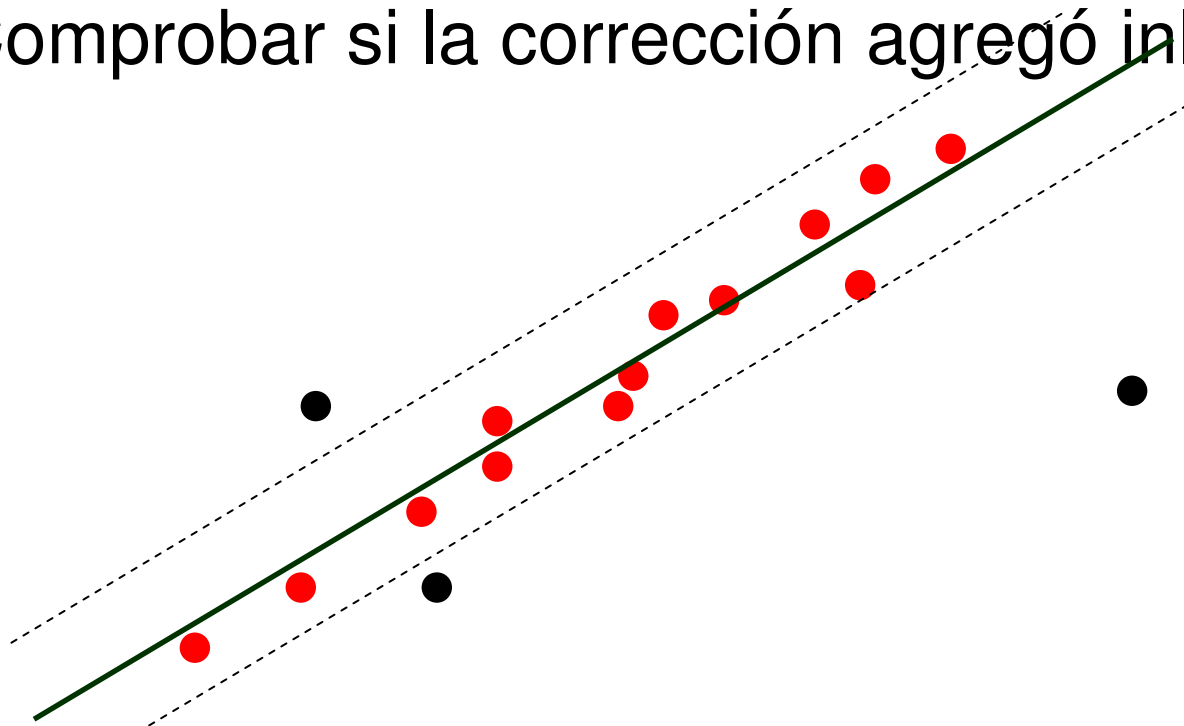
# RANSAC

- Comprobar si la corrección agregó inliers



# RANSAC

- Comprobar si la corrección agregó inliers

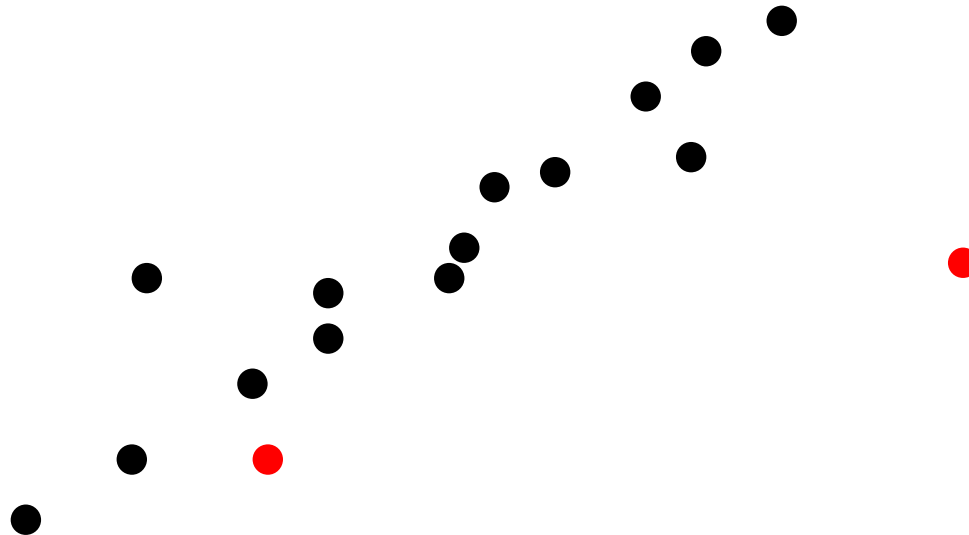


- No hay cambios en los inliers, fin de la corrección.
- Se encontraron 13 inliers con esas semillas.



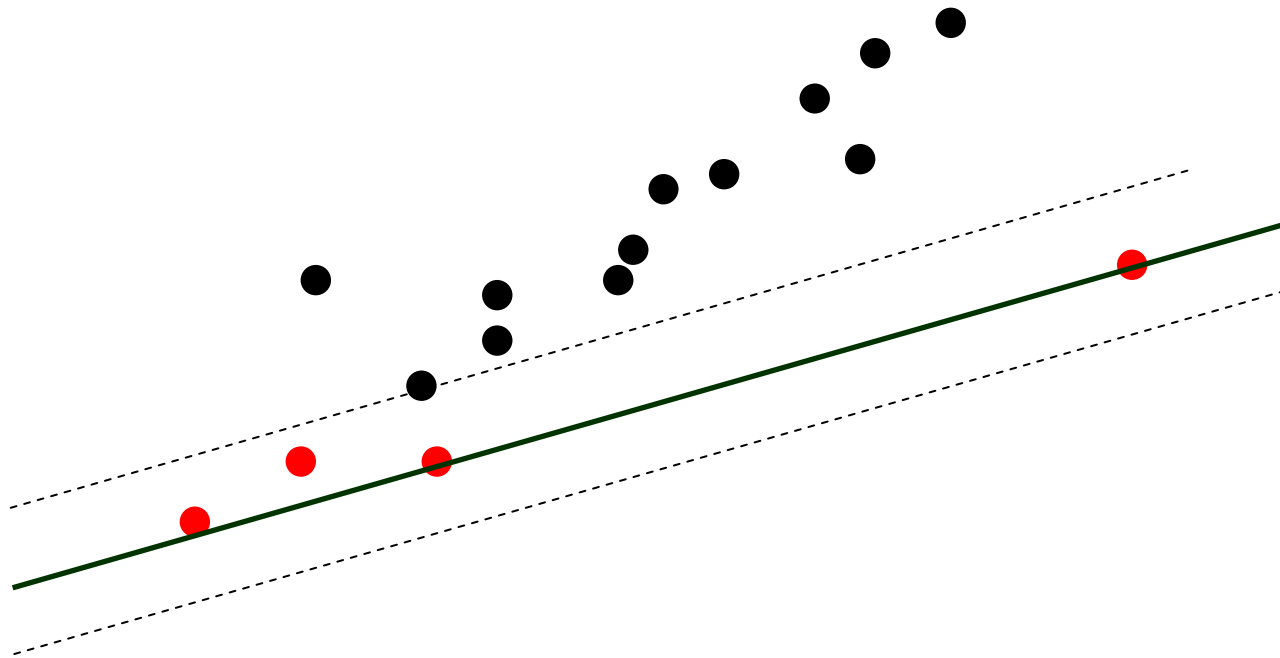
# RANSAC

- Intentar con nuevas semillas al azar:



# RANSAC

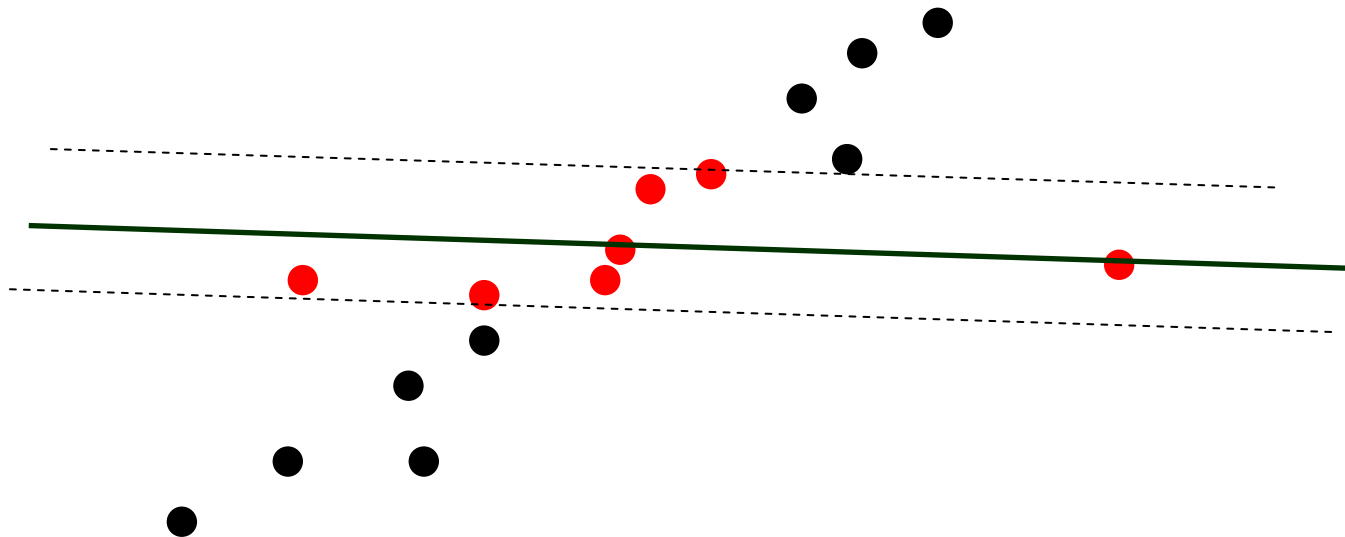
- Intentar con nuevas semillas al azar:



- Se encuentran dos inliers nuevos, corrigiendo usando mínimos cuadrados...

# RANSAC

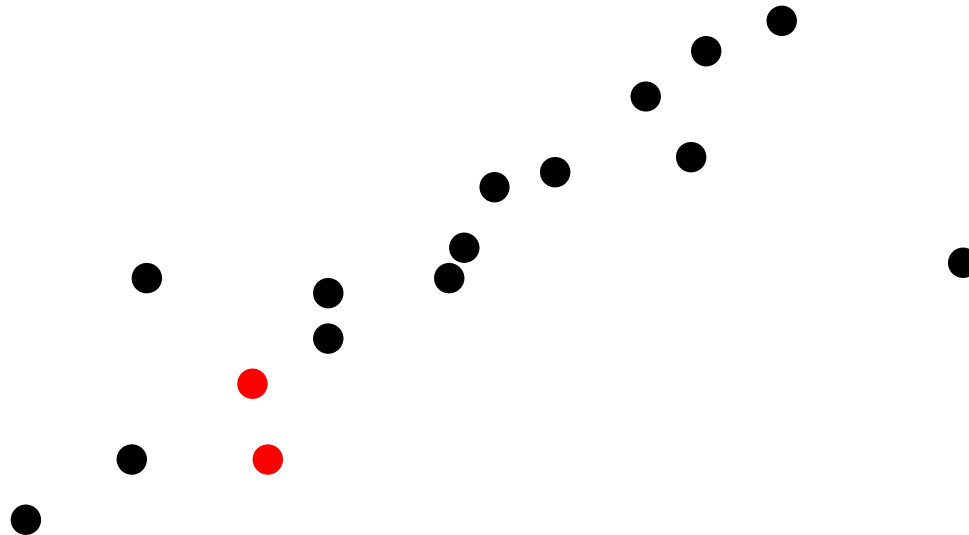
- La corrección puede converger a otra solución:



- Se encuentran 7 inliers con esas semillas

# RANSAC

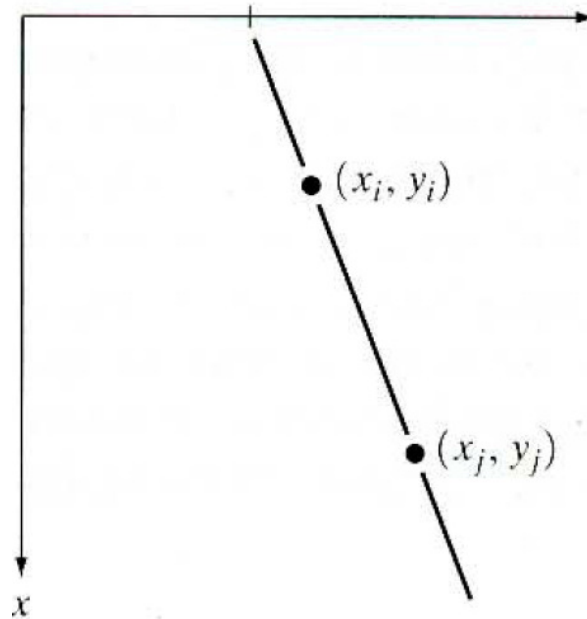
- Intentar con nuevas semillas al azar...



- Realizar  $N$  iteraciones (semillas+corrección).
- Seleccionar la recta con más inliers.

# Transformada de Hough

- Idea: Probar todas las posibles rectas y quedarse con la que pasa por más puntos.

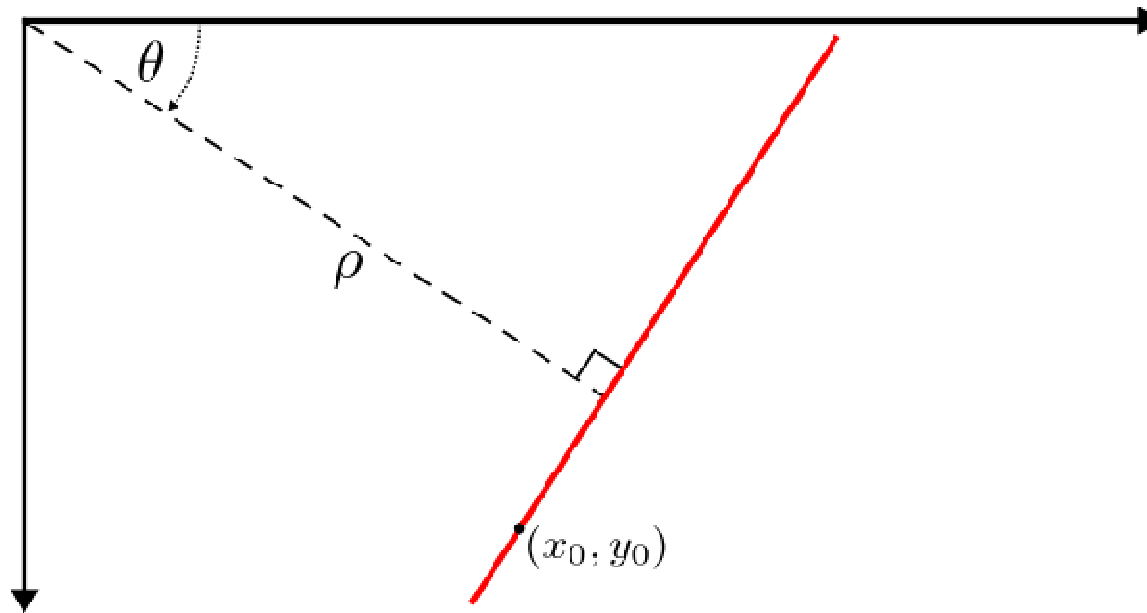


Ver González et al. cap 10

# Recta en coordenadas polares

- Todas las rectas que pasan por  $(x_0, y_0)$ :

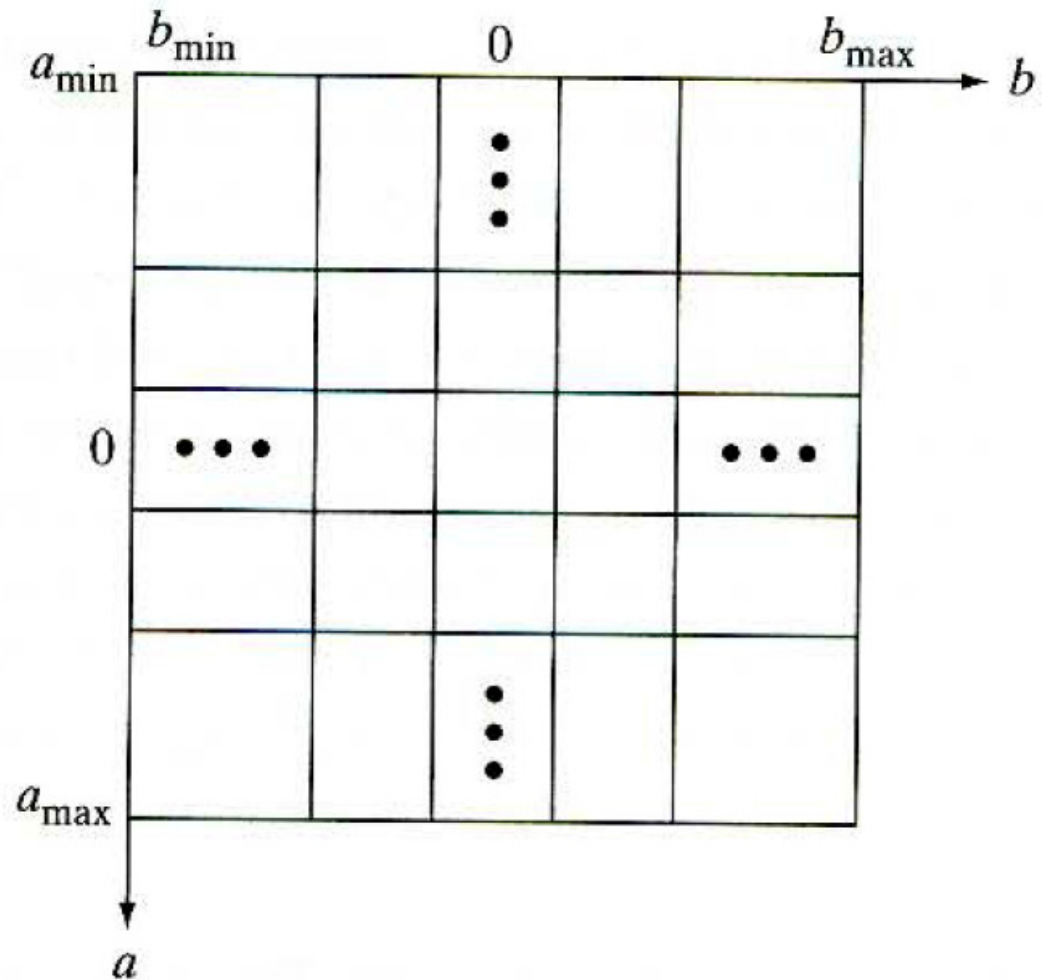
$$\rho(\theta) = x_0 \cos \theta + y_0 \sin \theta$$





# Transformada de Hough

- Dividir el espacio de parámetros en una tabla de contadores.





# Transformada de Hough

- Para cada punto:
  - Recorrer cada celda en  $\theta$  y calcular  $\rho$  según la fórmula.
  - Sumar 1 en todas las celdas  $(\theta, \rho)$  correspondiente.
- Seleccionar las celdas con más votos.
  - Una celda con  $N$  votos implica una recta  $(\theta, \rho)$  con  $N$  puntos colineales.



# Transformada de Hough

- Decidir:

- ☐ Rango de  $\theta$  y  $\rho$ .
- ☐ Cantidad de contadores para  $\theta$  y  $\rho$ .

- Mejoras:

- ☐ Voto ponderado, cada voto se reparte entre celdas cercanas.
- ☐ Matriz dispersa para la votación, guardando sólo los contadores  $> 0$ .

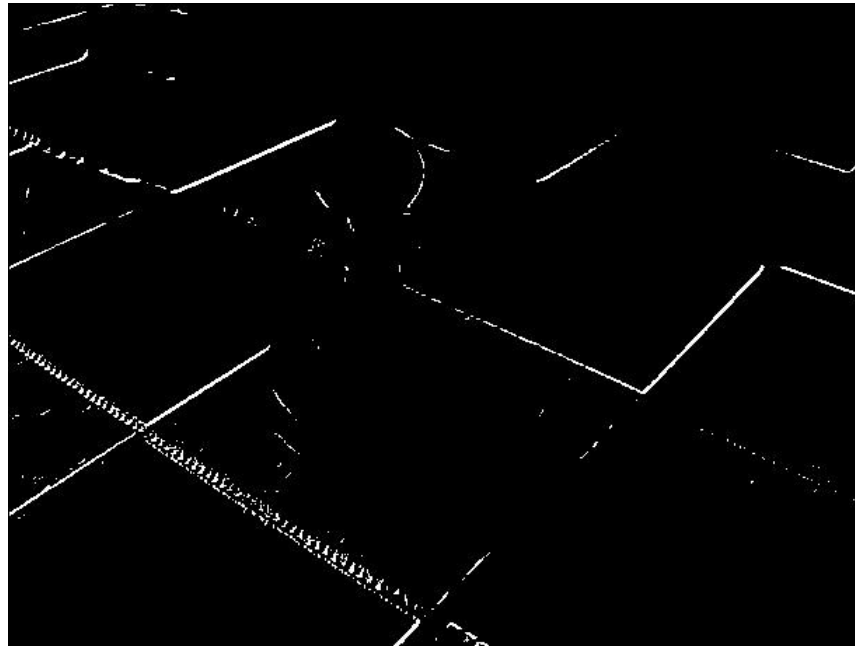


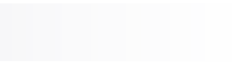
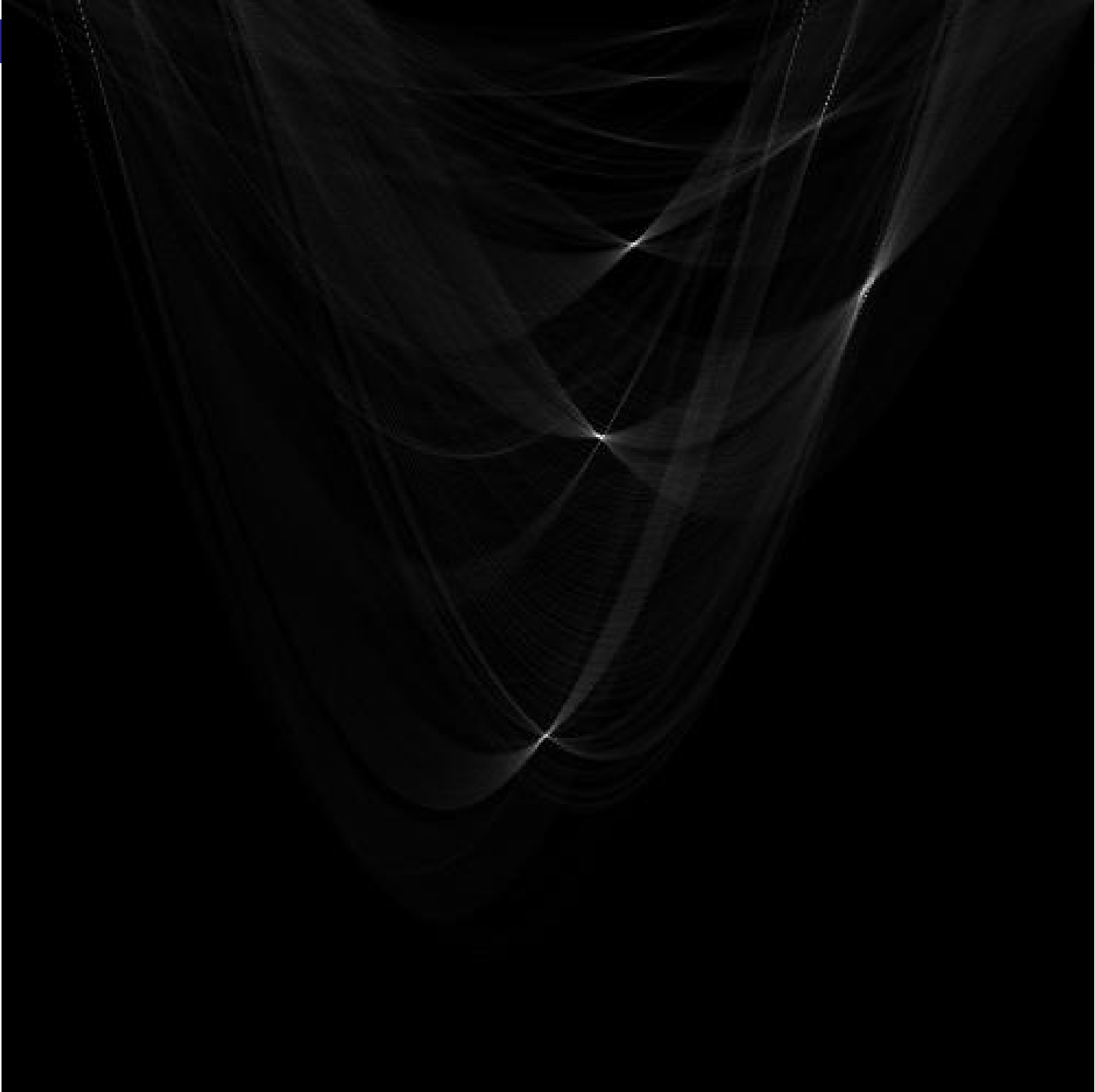
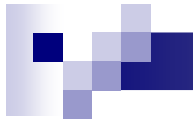
# Ejemplo



# Ejemplo

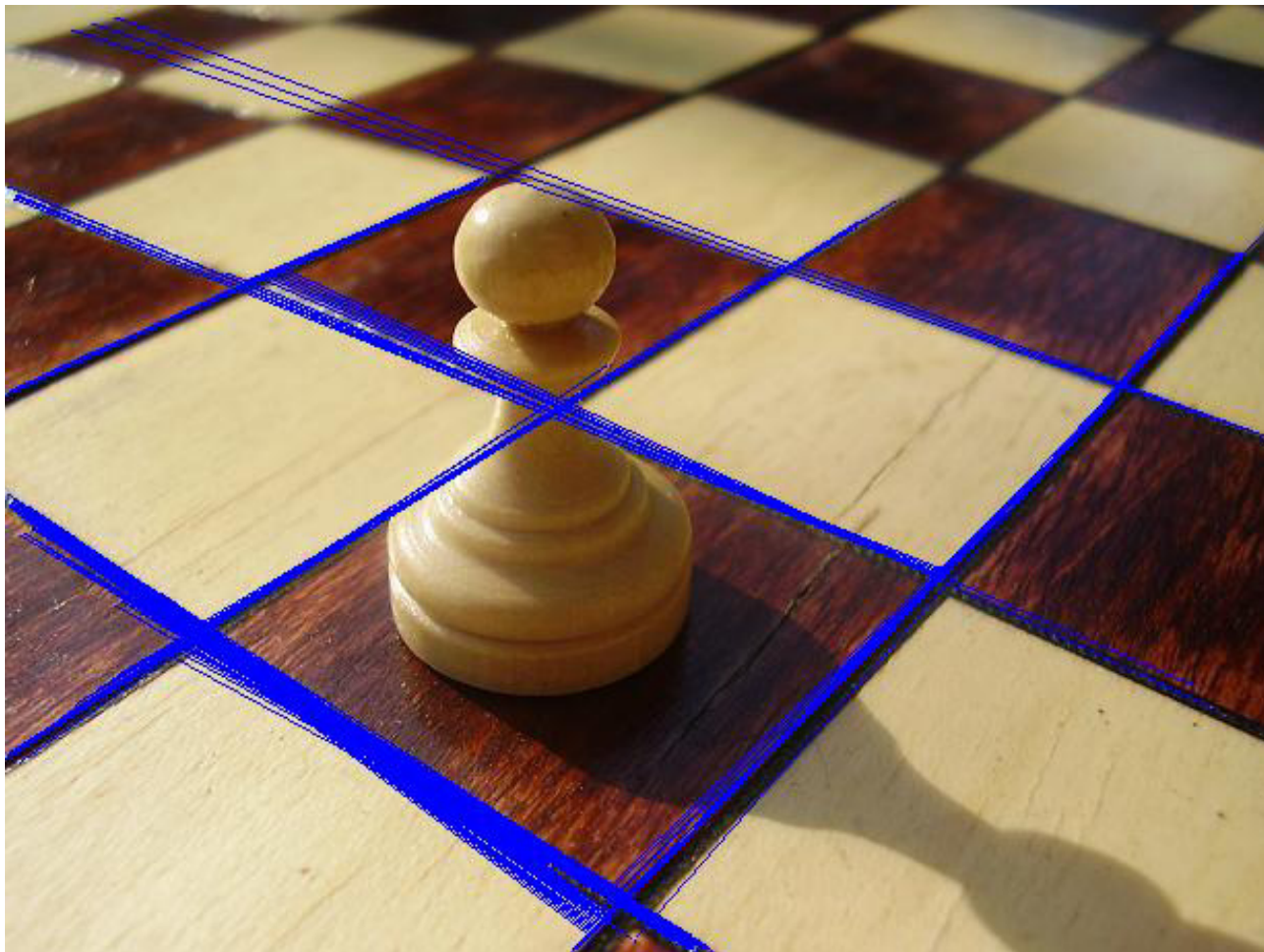
- Puntos de bordes detectados según un umbral en la magnitud del gradiente.
- Estos puntos participarán en la T. de Hough.



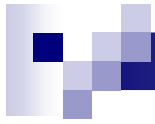




# Transformada de Hough

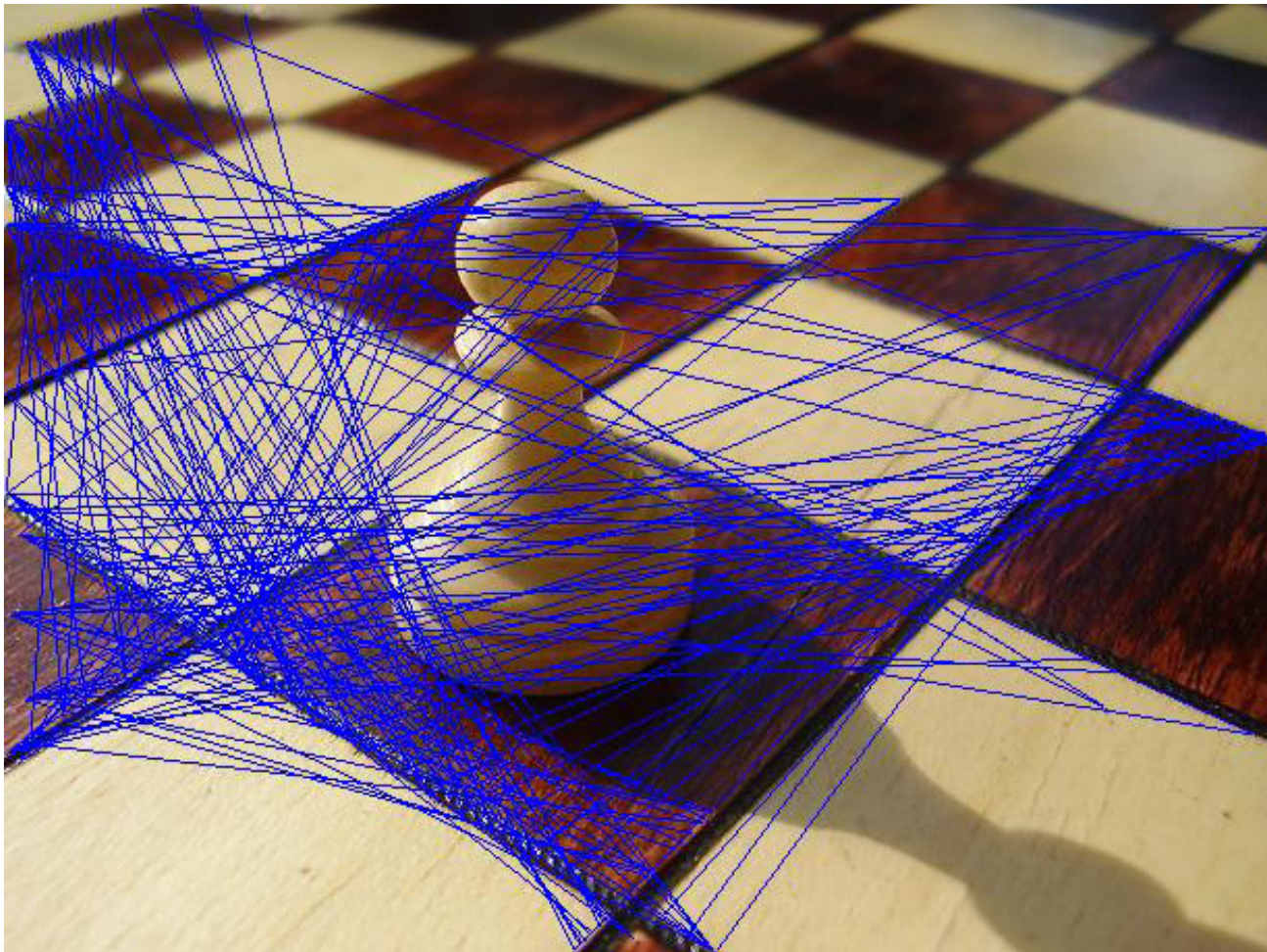






# Transformada de Hough

Usando malos parámetros...







# Resumen Detección De Líneas

- Mínimos cuadrados
  - Funciona cuando todos son inliers.
- RANSAC
  - Rápido, funciona cuando es probable encontrar al azar puntos inliers.
- Transformada de Hough
  - Lento, funciona cuando hay muchos outliers.
  - Puede encontrar más de una recta.



# Otros Usos

- Con modificaciones mínimas se puede usar RANSAC y T. Hough para buscar círculos y otras figuras.
  - El número de parámetros puede aumentar.
- RANSAC y T. Hough se usan para determinar transformaciones espaciales en descriptores locales.

# Bibliografía

- **Digital Image Processing.**  
González et al. 2008
  - Cap. 10.

