

رسالة محمد

# مبانی بینایی کامپیوتر

مدرس: محمدرضا محمدی

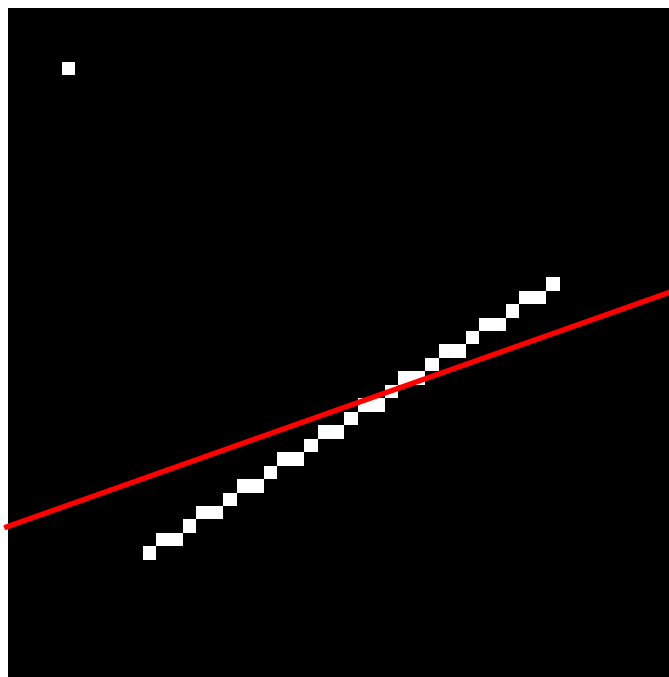
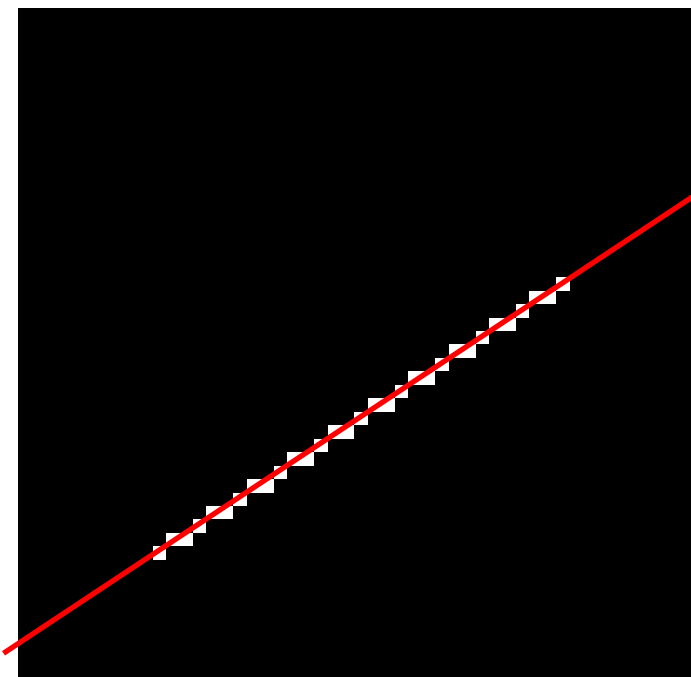
۱۴۰۱

استخراج شكل

Shape Extraction

# تشخیص خط

- لبه‌های دیگر چه اثری می‌گذارند؟



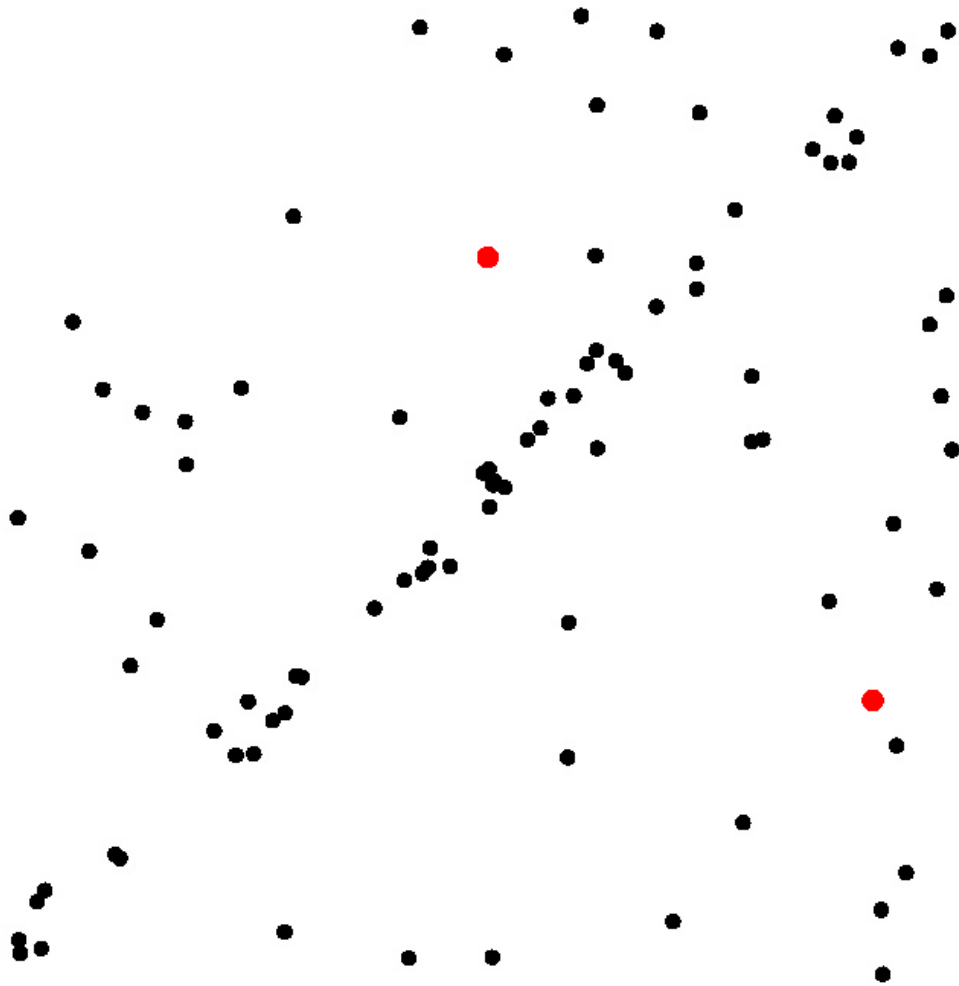
# RANSAC

- Random sample consensus
- با تعداد محدودی از نقاط یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم



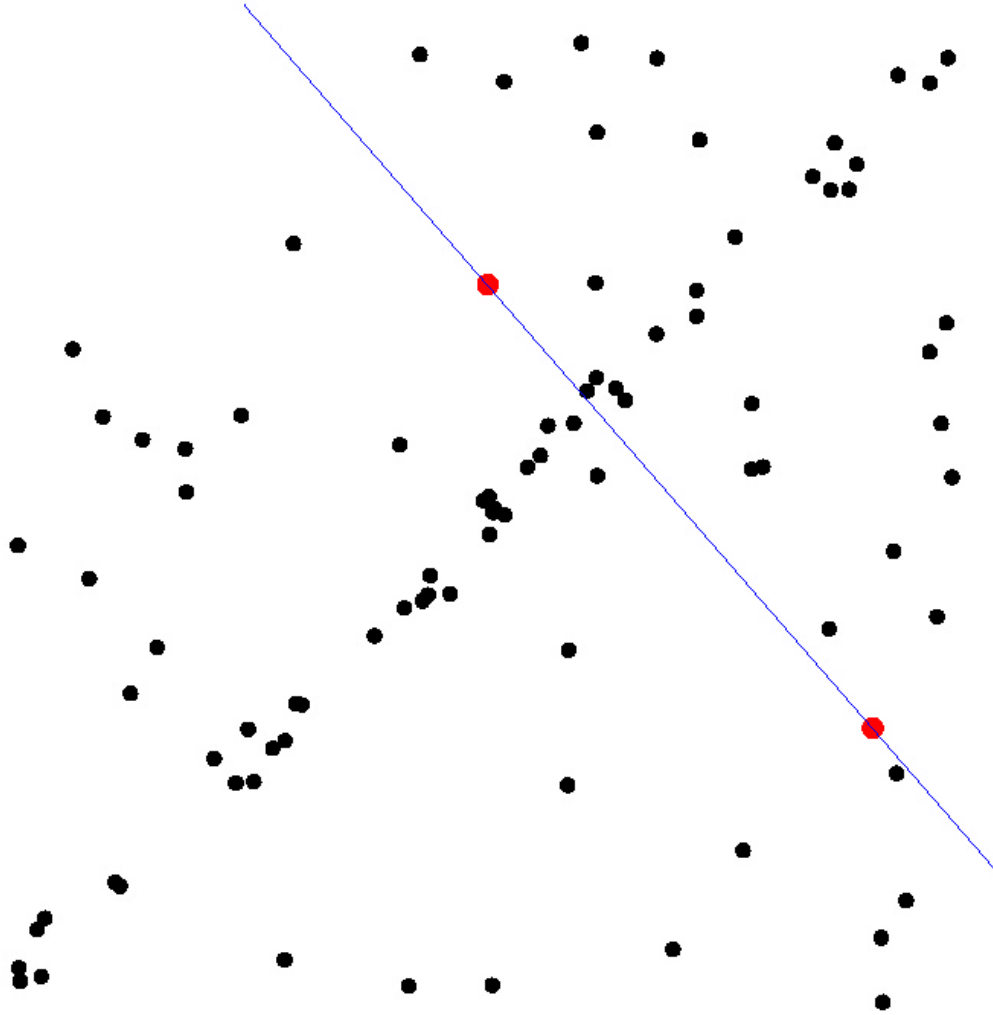
# RANSAC

- Random sample consensus
- با تعداد محدودی از نقاط یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم

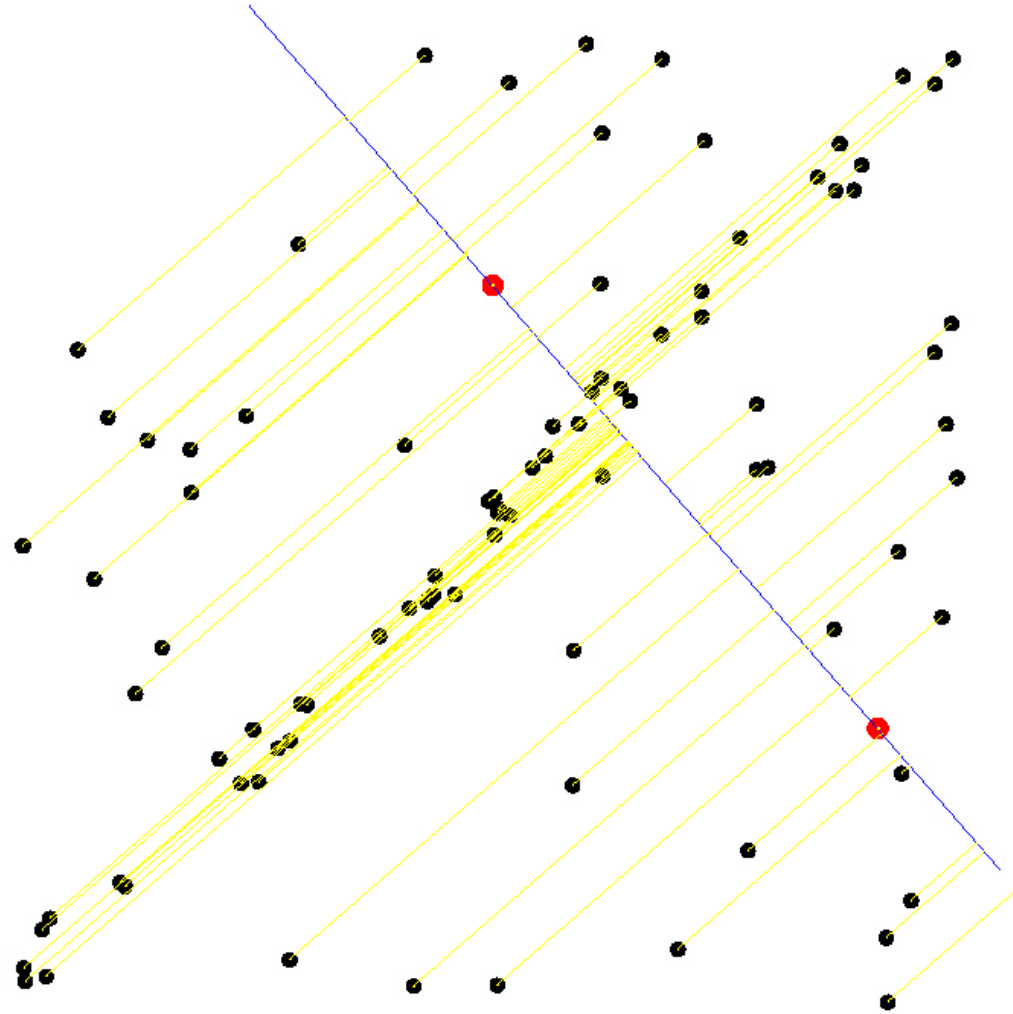


# RANSAC

- Random sample consensus
- با تعداد محدودی از نقاط یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم



# RANSAC

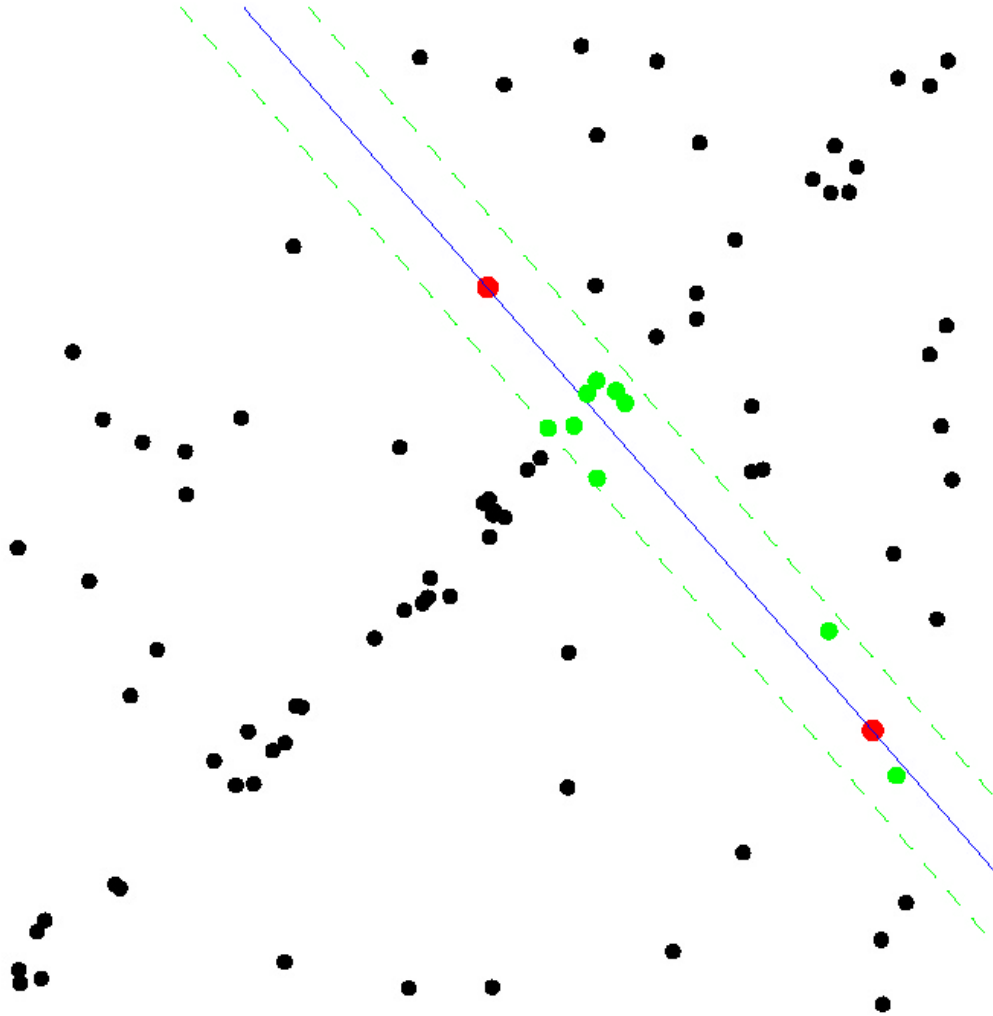


- Random sample consensus
- با تعداد محدودی از نقاط یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم



# RANSAC

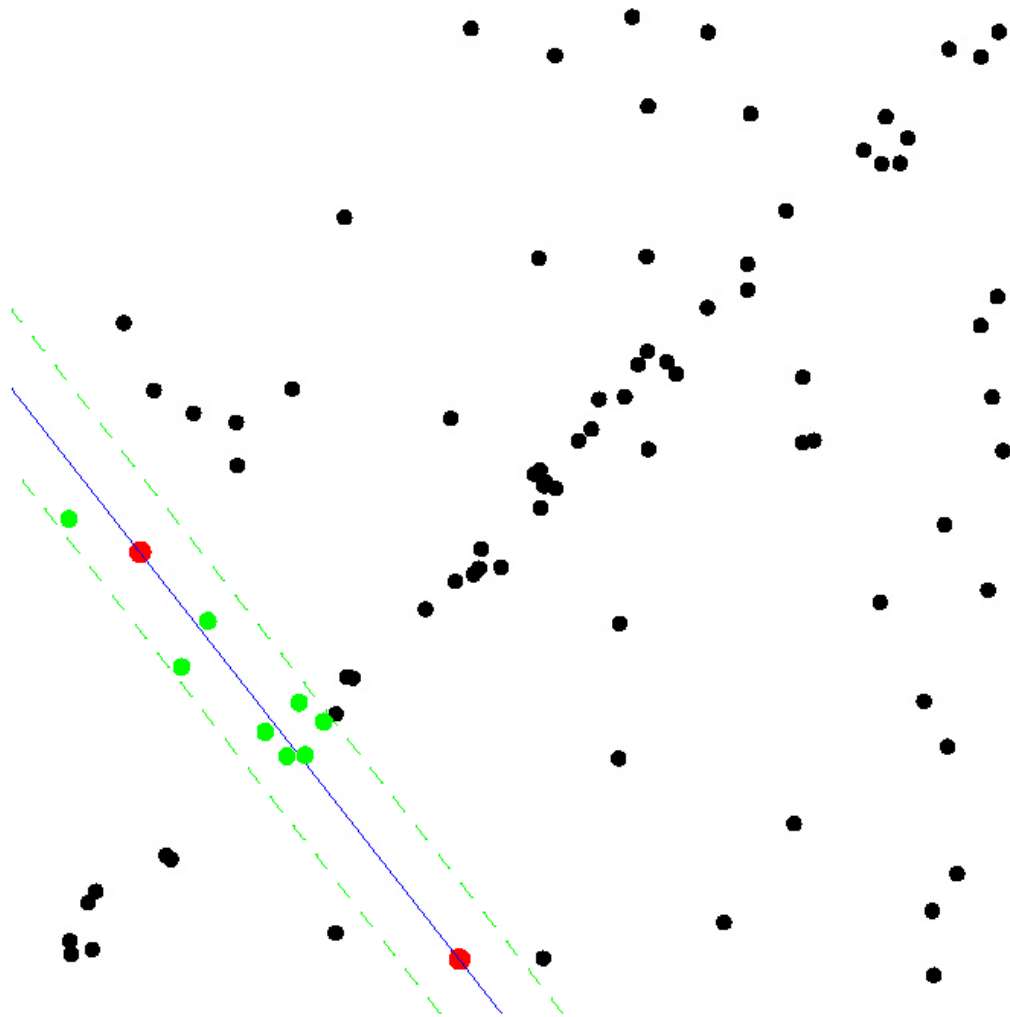
- Random sample consensus
- با تعداد محدودی از نقاط یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم



# RANSAC

- **Random sample consensus**

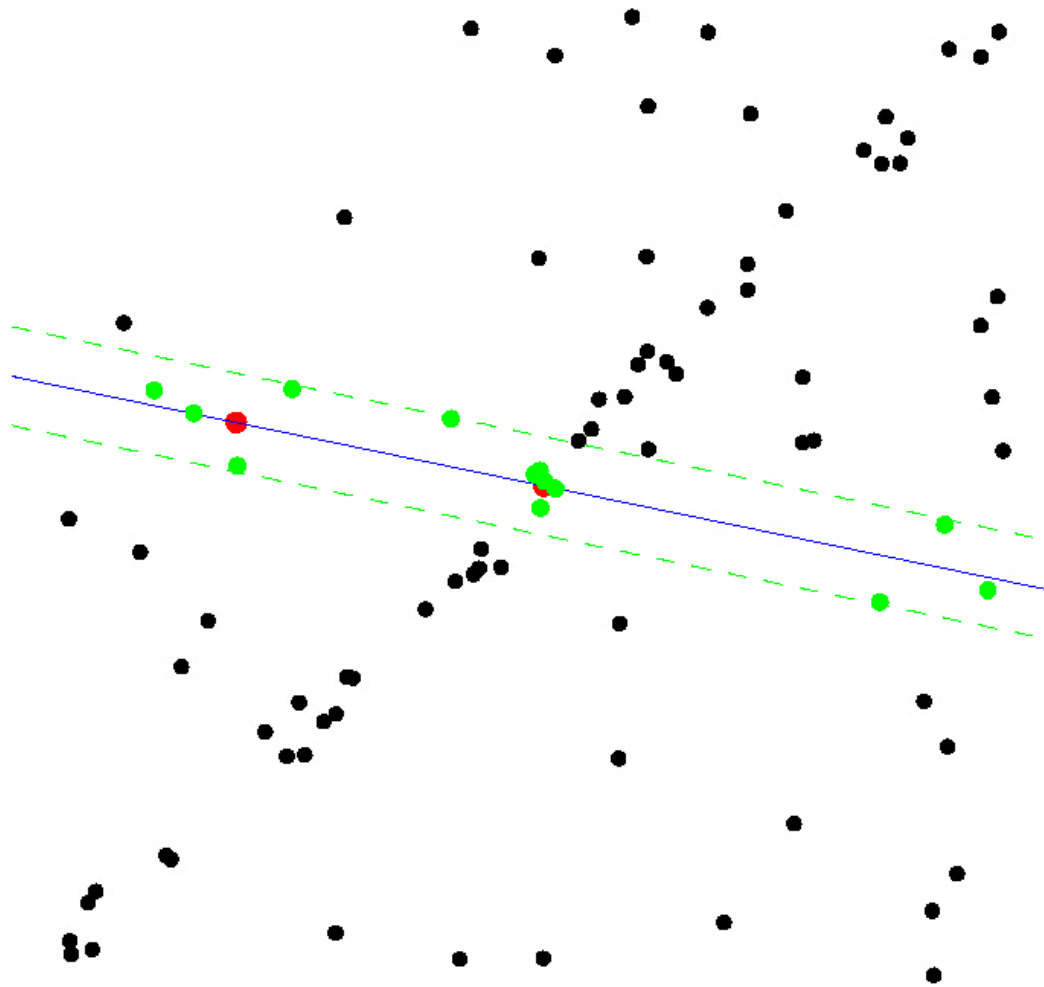
- با تعداد محدودی از نقاط یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم



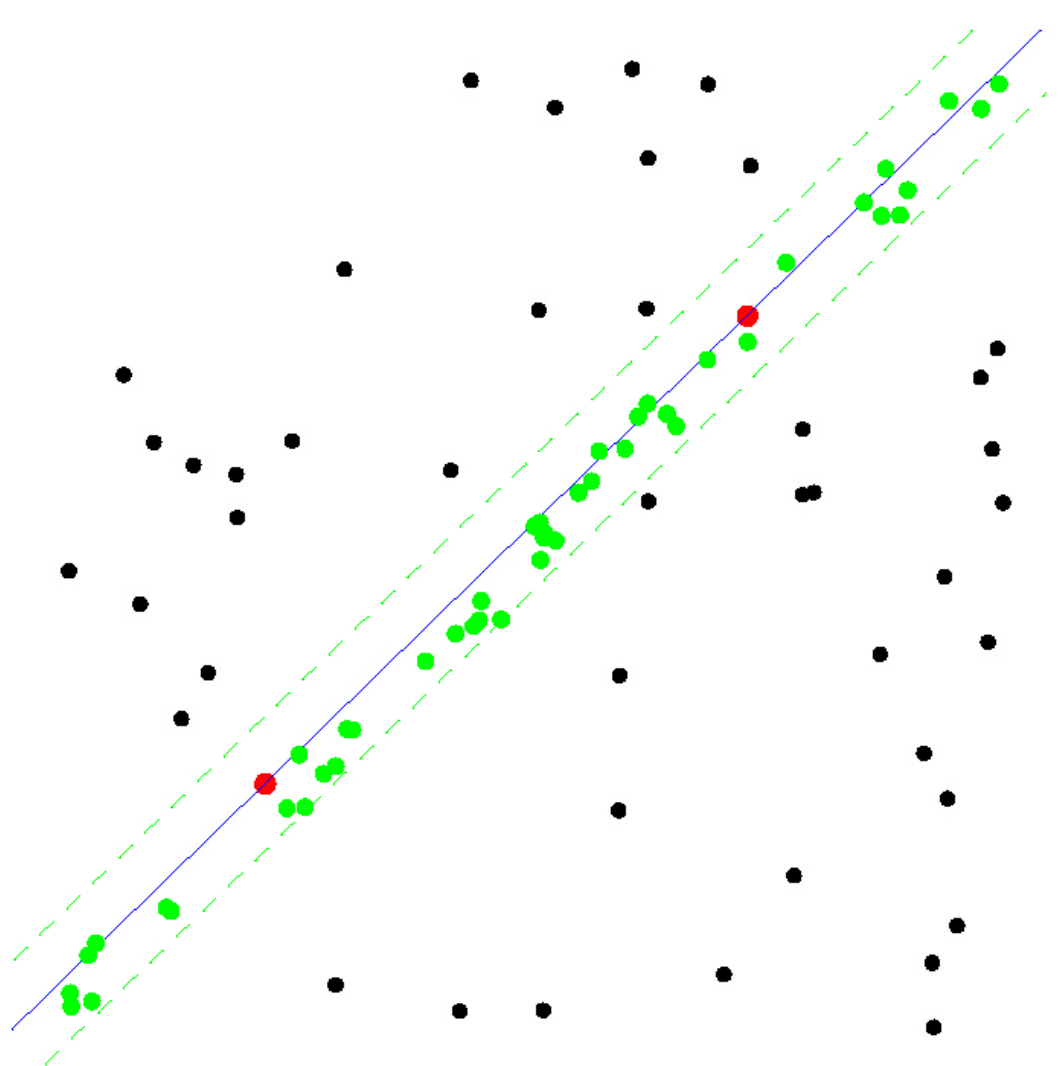
# RANSAC

Random sample consensus •

• با تعداد محدودی از نقاط یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم



# RANSAC



- Random sample consensus
- با تعداد محدودی از نقاط یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم

# RANSAC

- در این الگوریتم، ابتدا مدل با برخی از داده‌ها که به صورت تصادفی نمونه‌برداری شده‌اند آموزش می‌بیند و سپس با استفاده از داده‌های دیگر صحت‌سنجی می‌شود
- در نهایت بهترین مدل انتخاب می‌شود
- سوال: RANSAC به چه تعداد تکرار نیازمند است؟
- در حالت ایده‌آل، برای یافتن خط با این روش باید تمام ترکیب‌های دو نقطه‌ای را بررسی کنیم

$$\frac{N(N-1)}{2}$$

# RANSAC

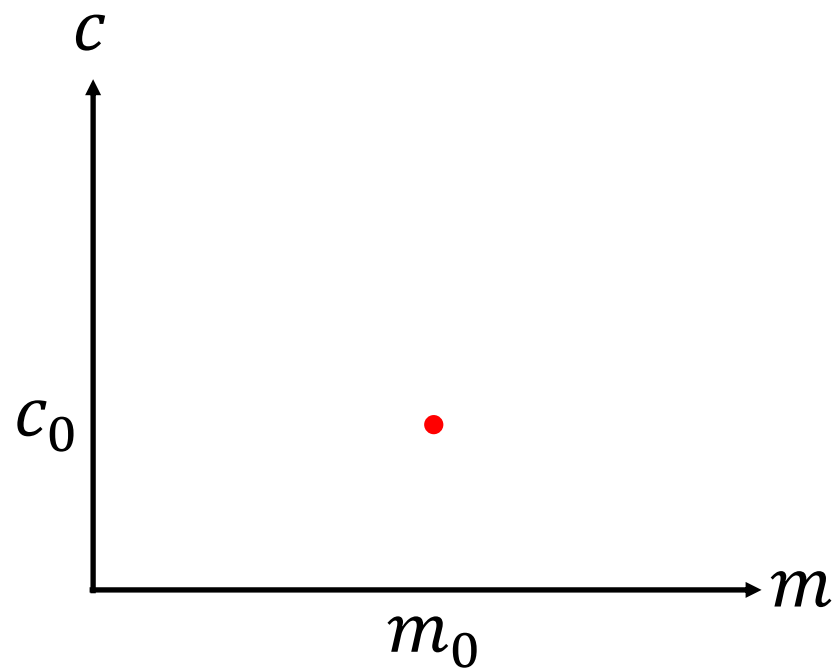
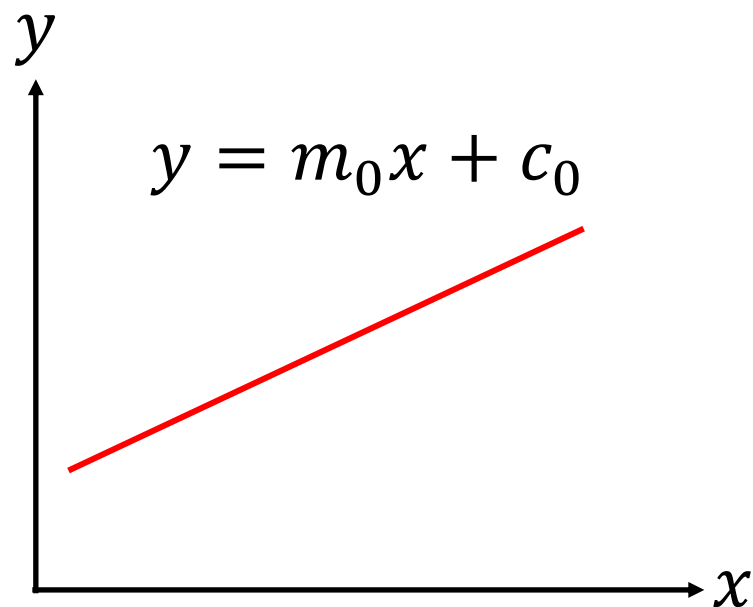
- اگر  $w$  نسبت تعداد نقاط inlier به تمام نقاط باشد
- اگر  $p$  احتمال یافتن یک مجموعه از نقاط بدون outlier باشد
- اگر فرض کنیم برای تخمین یک خط تنها به دو نقطه نیاز داریم
- احتمال آنکه یک مجموعه کاملاً از نقاط inlier تشکیل شود:  $w^2$
- اگر  $k$  تعداد تکرار باشد، احتمال آنکه هیچ مجموعه درستی انتخاب نشده باشد برابر است با:

$$1 - p = (1 - w^2)^k$$

$$k = \frac{\log(1 - p)}{\log(1 - w^2)}$$

# تبدیل Hough

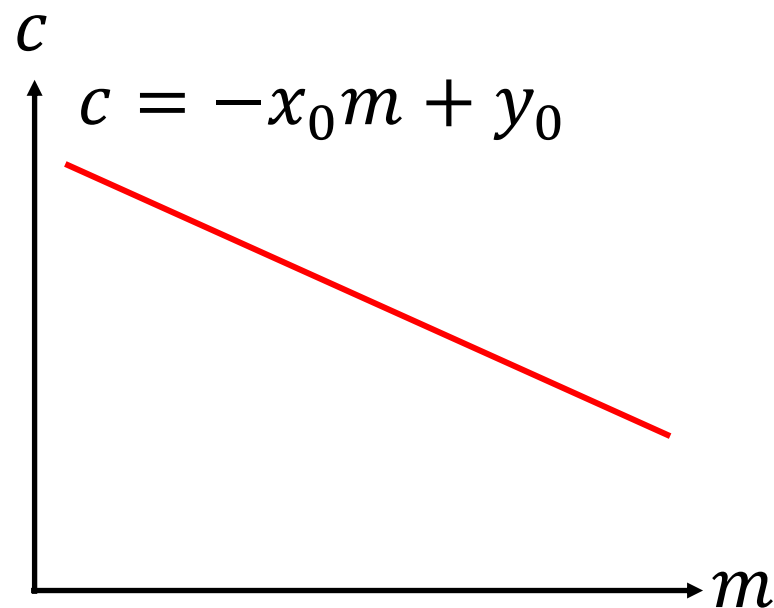
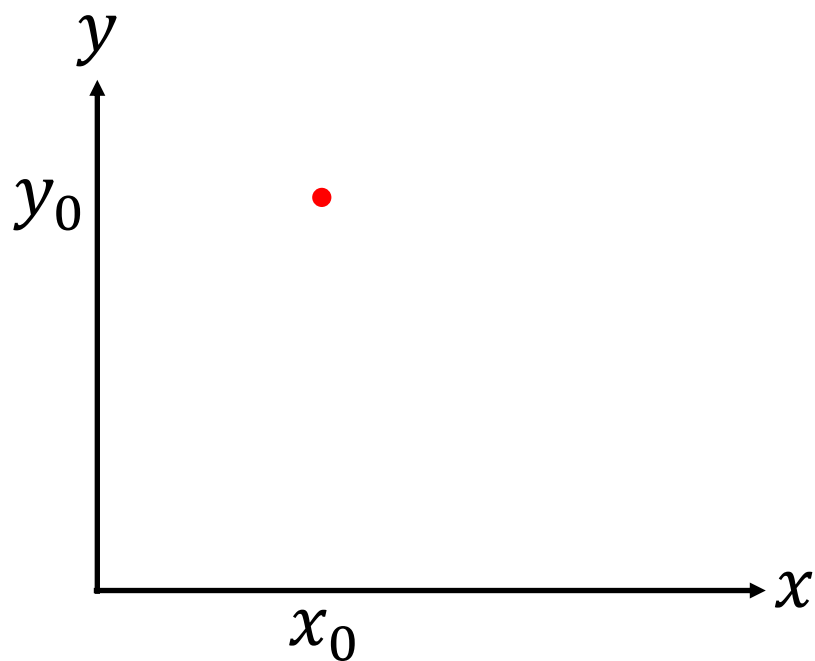
- ایده اصلی تبدیل Hough بر تغییر فضا و رای گیری است
- هر خط در فضای  $(x,y)$  معادل با یک نقطه در فضای  $(m,c)$  است



# تبدیل Hough

- هر نقطه در فضای تصویر معادل با چه تابعی در فضای هاف است؟

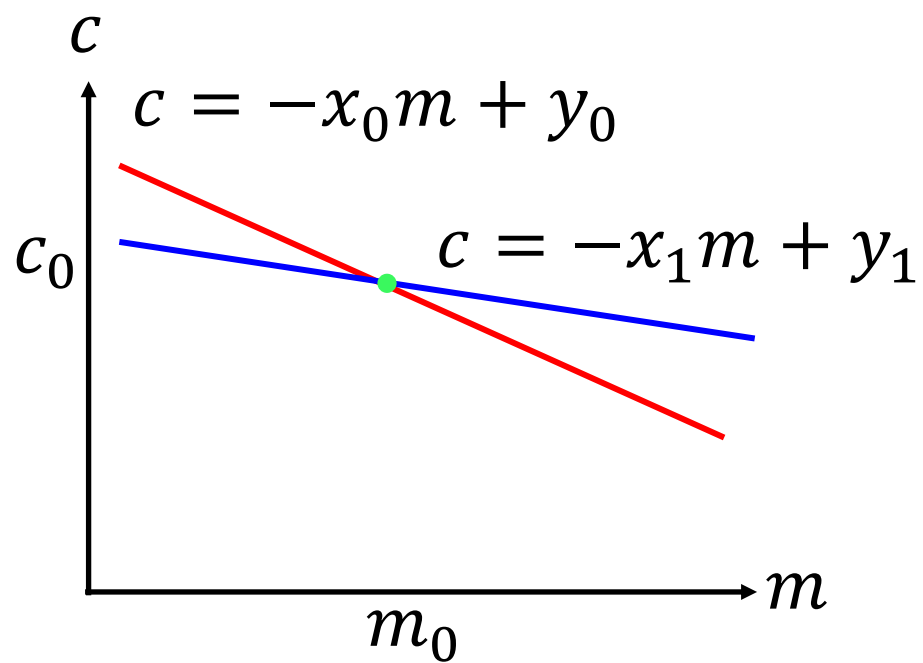
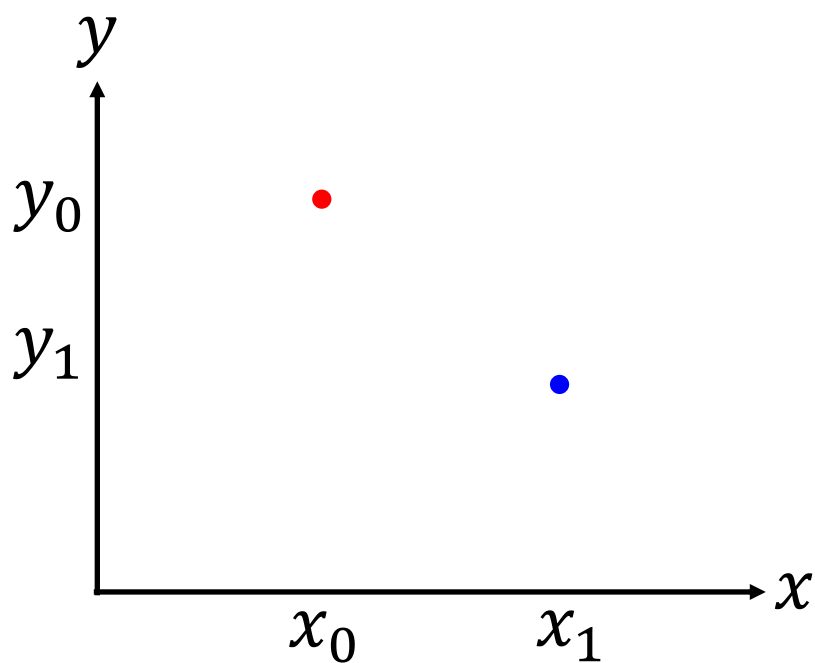
$$y = mx + c$$





# تبدیل Hough

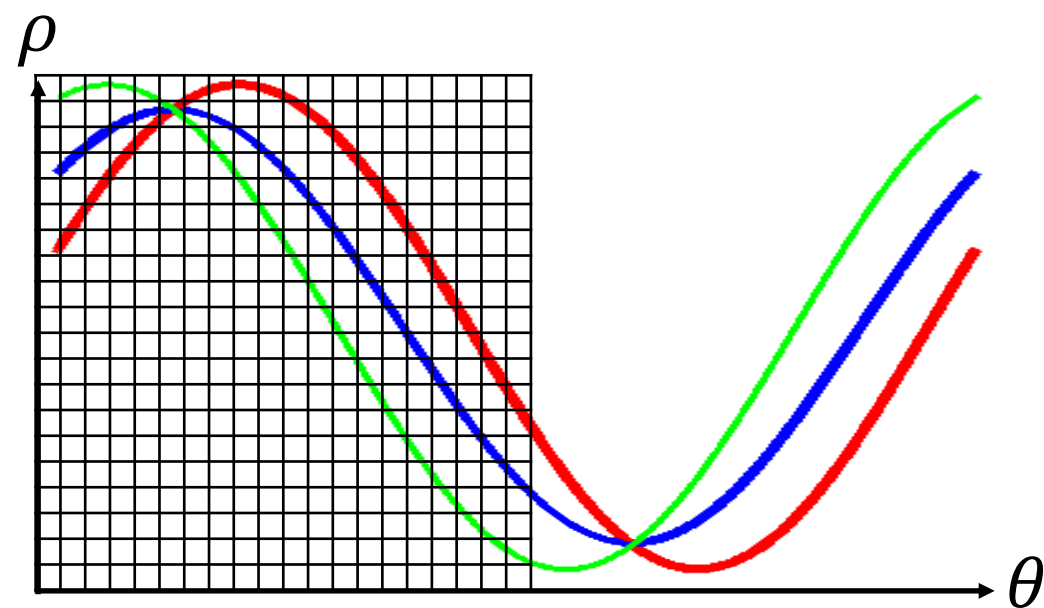
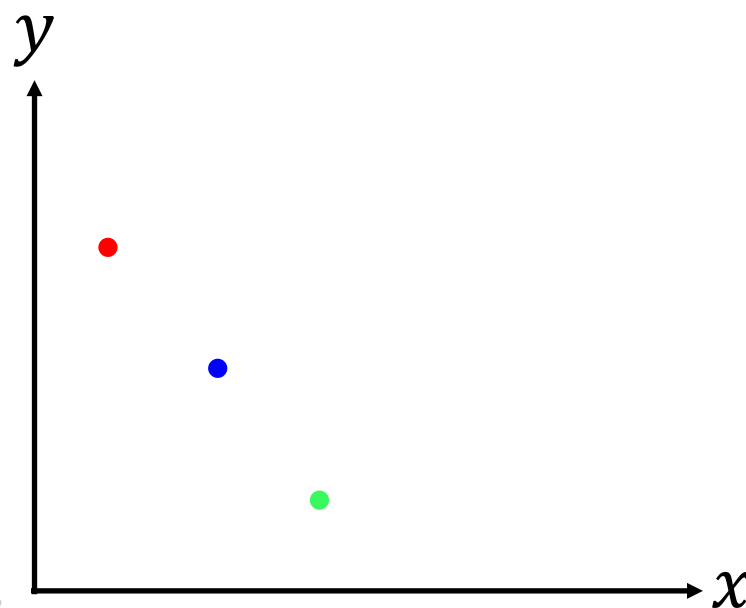
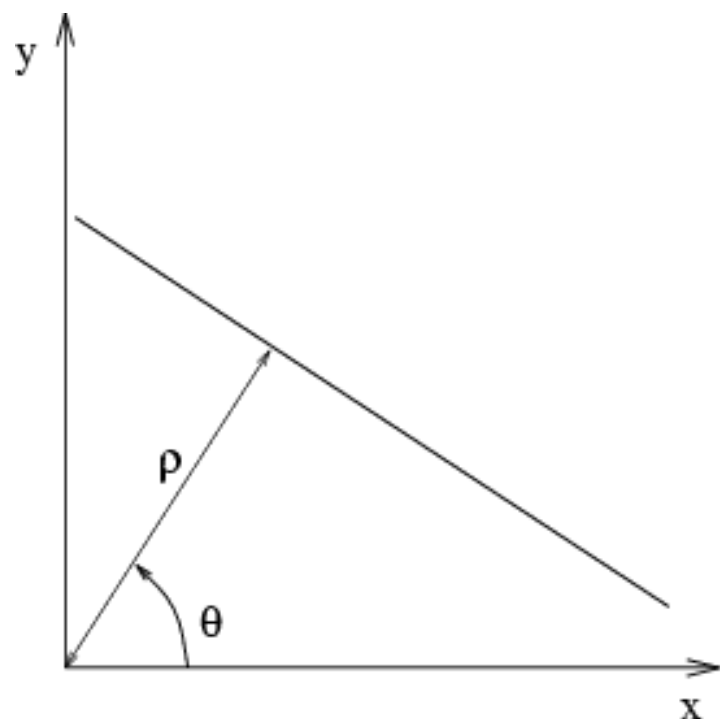
- وضعیت دو نقطه از تصویر اصلی در تبدیل هاف چگونه خواهد بود؟
- خطی که از هر دو نقطه عبور می کند تقاطع دو خط در فضای هاف است



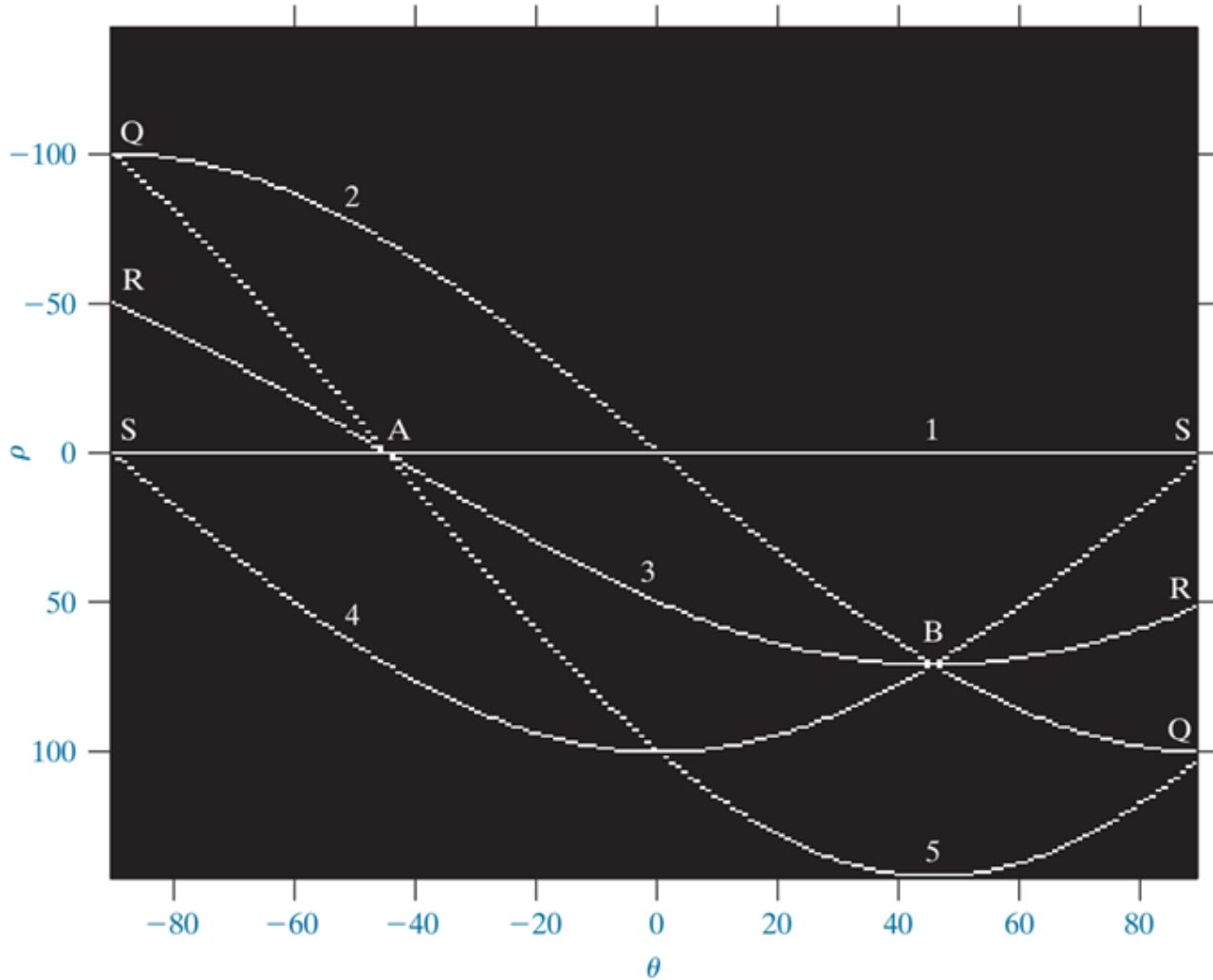
# تبدیل Hough

- البته فضای  $(m, c)$  به دلیل آنکه  $m$  می تواند نامحدود باشد مناسب نیست
- هر نقطه در فضای تصویر معادل با چه منحنی در فضای  $(\rho, \theta)$  خواهد بود؟

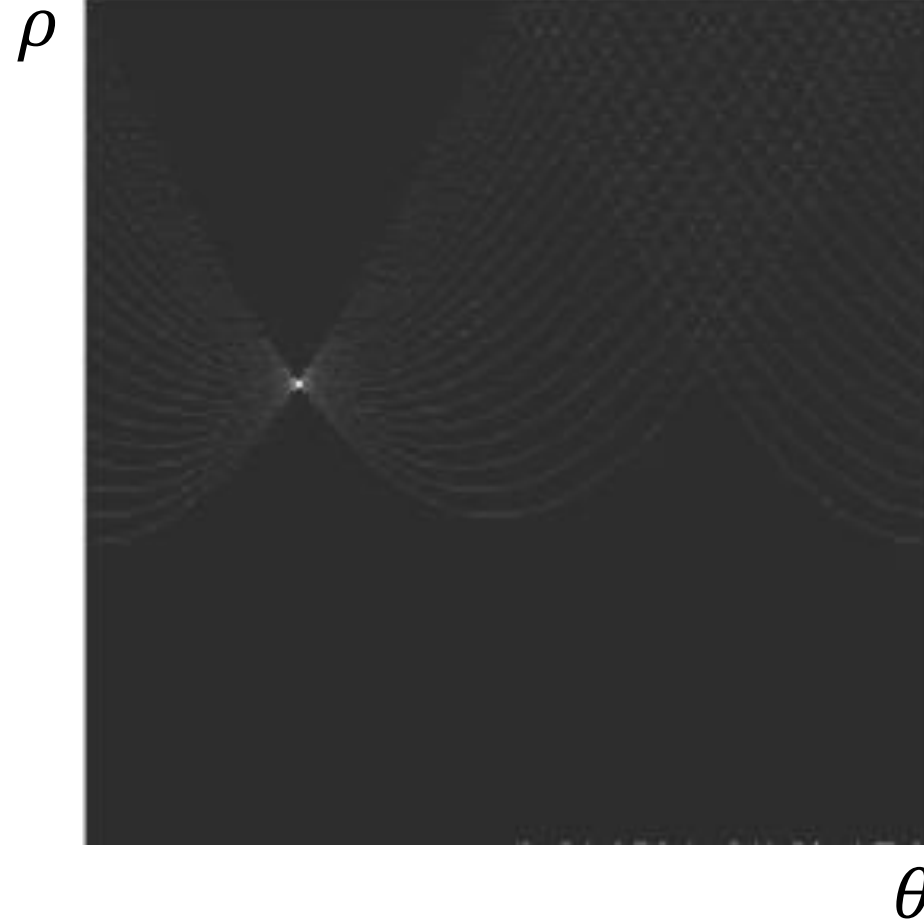
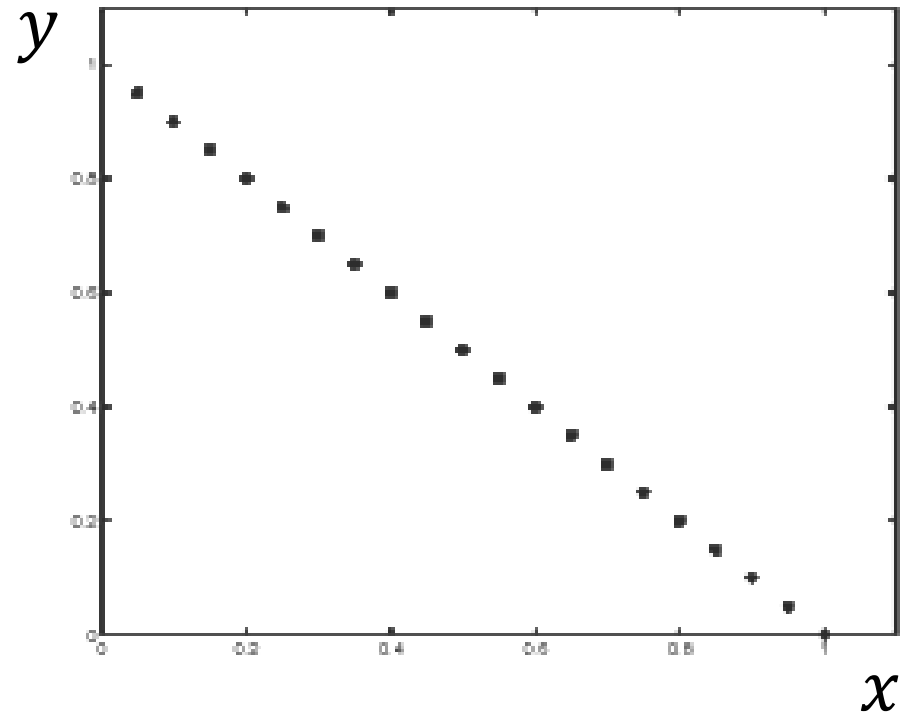
$$x \cos(\theta) + y \sin(\theta) = \rho$$



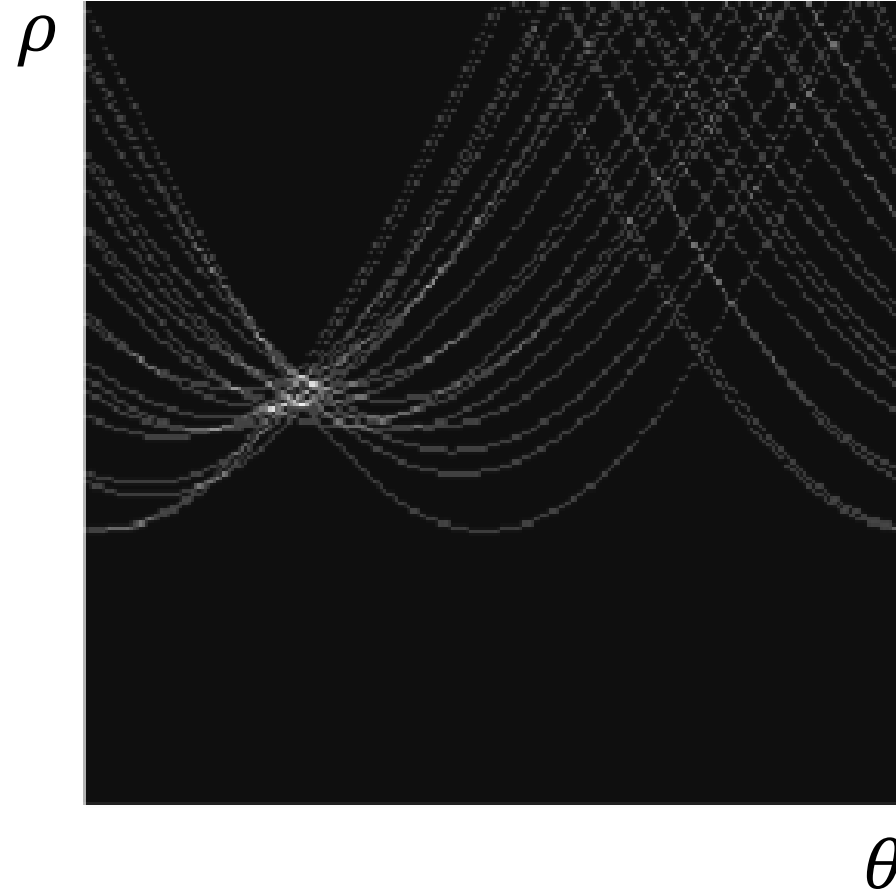
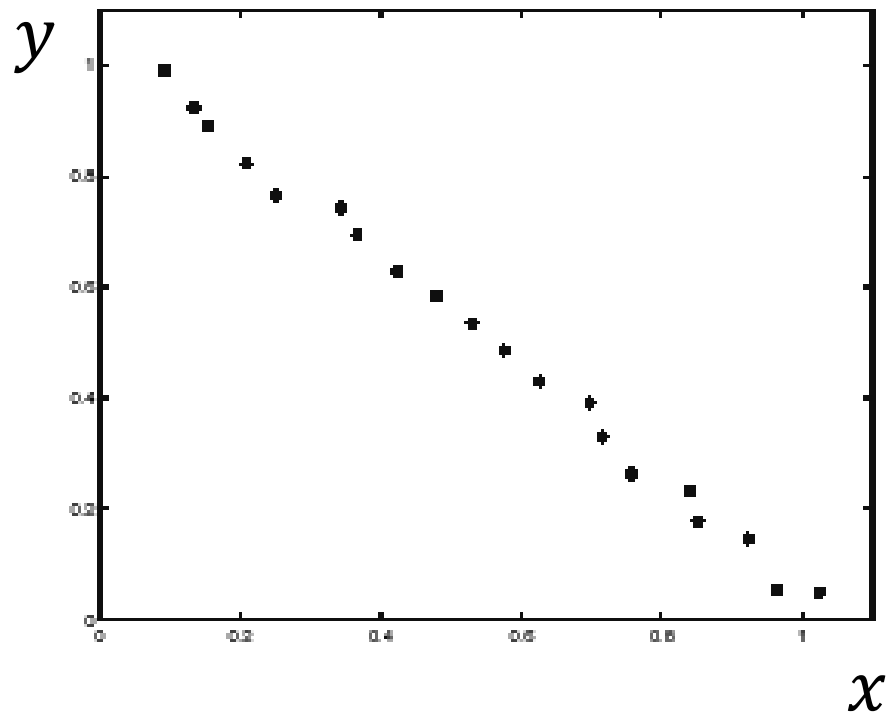
# تبدیل Hough



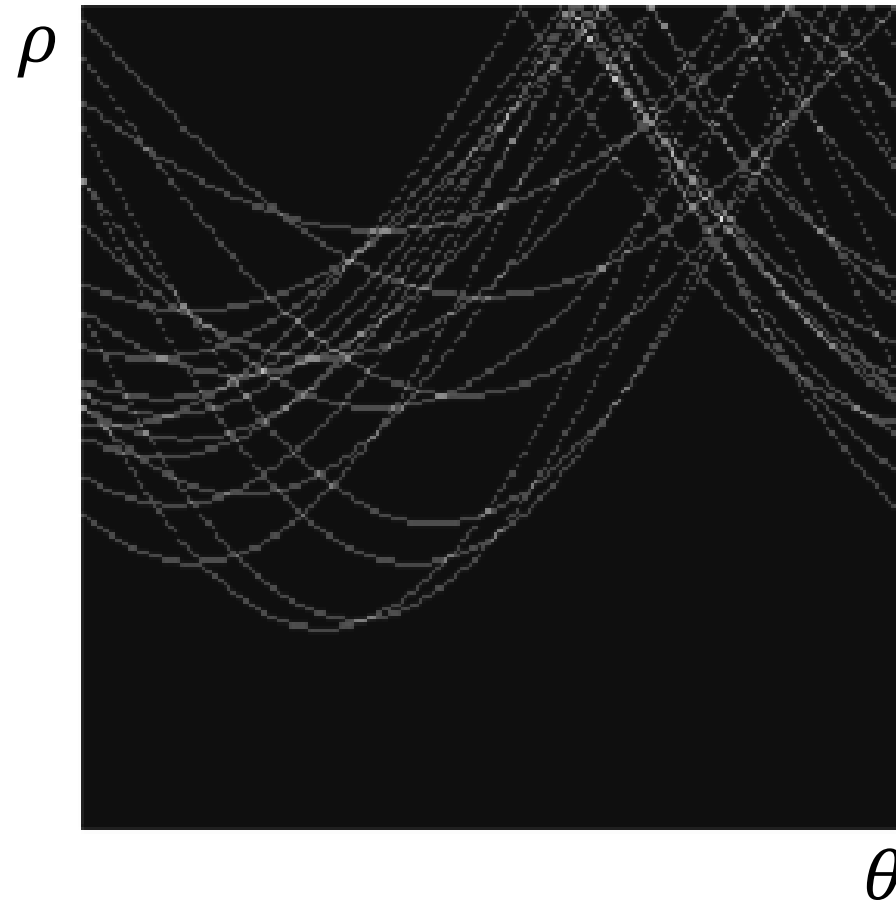
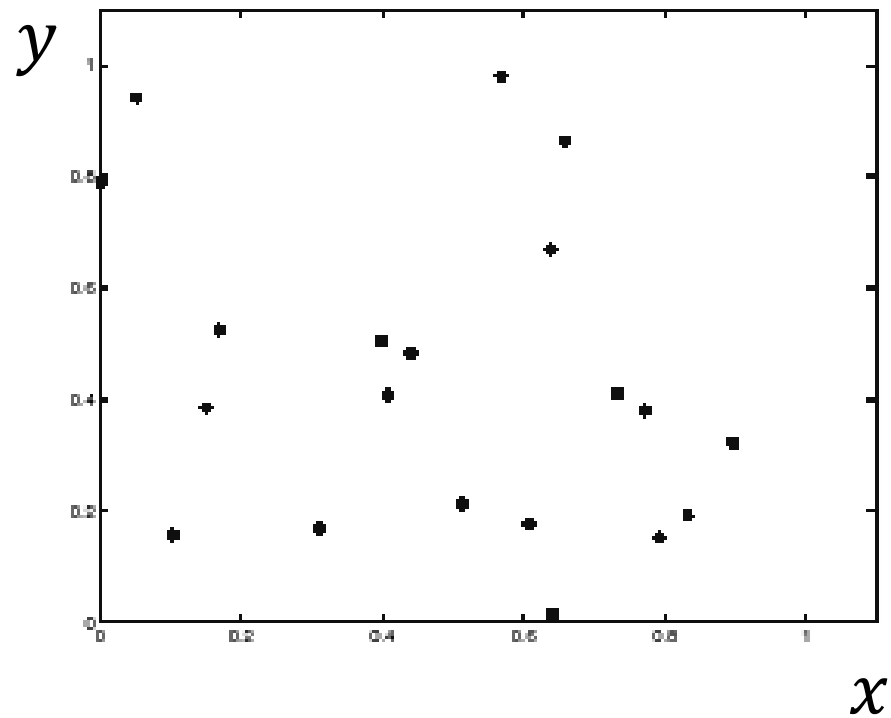
# تبدیل Hough



# تبدیل Hough

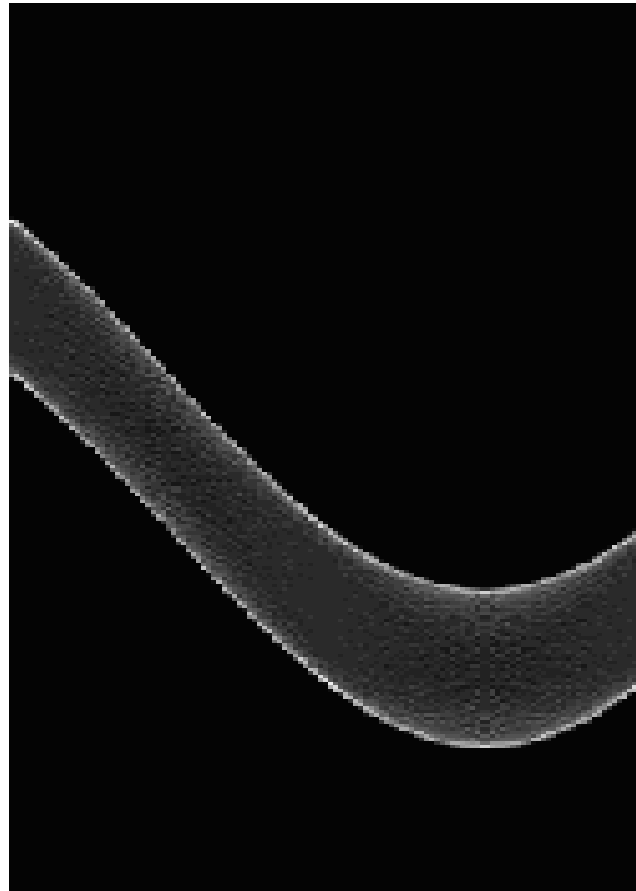


# تبدیل Hough

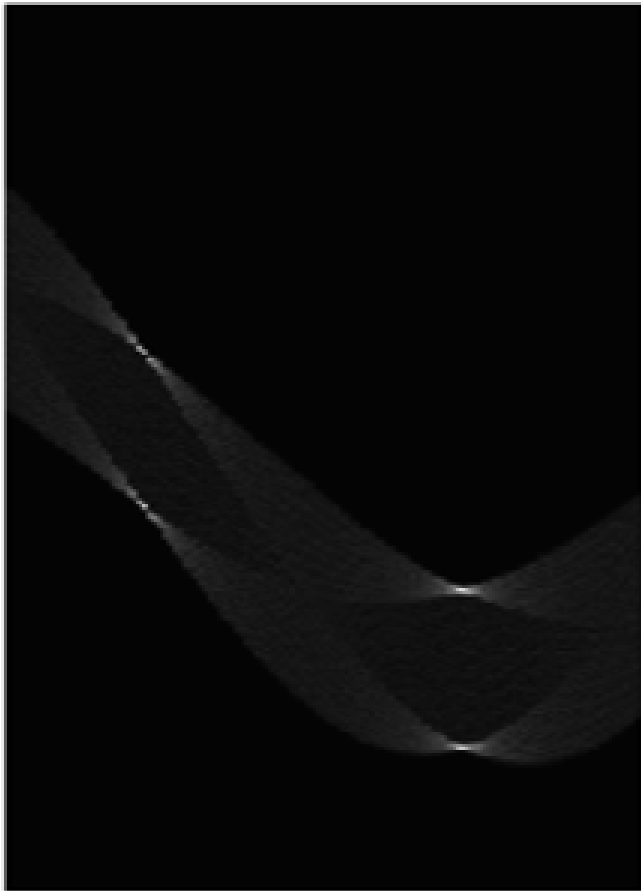


# تبدیل Hough

دایره



مستطیل



# شبه کد تبدیل Hough

- Initialize accumulator H to all zeros
- For each edge point  $(x, y)$  in the image

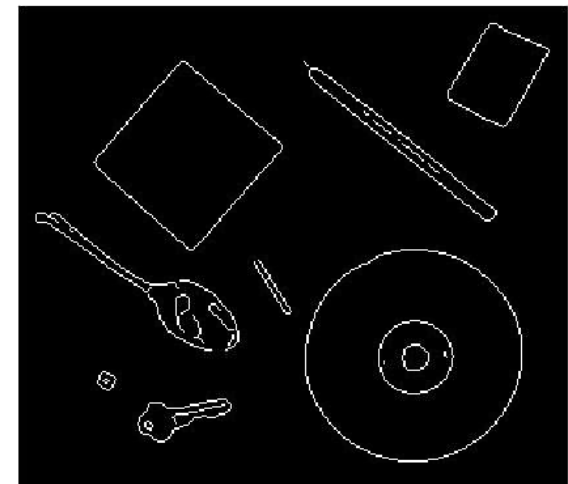
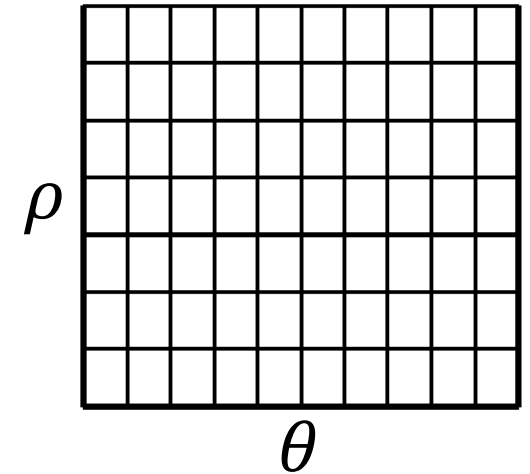
For  $\theta = 0$  to  $180$

$$\rho = x \cos \theta + y \sin \theta$$

$$H(\rho, \theta) = H(\rho, \theta) + 1$$

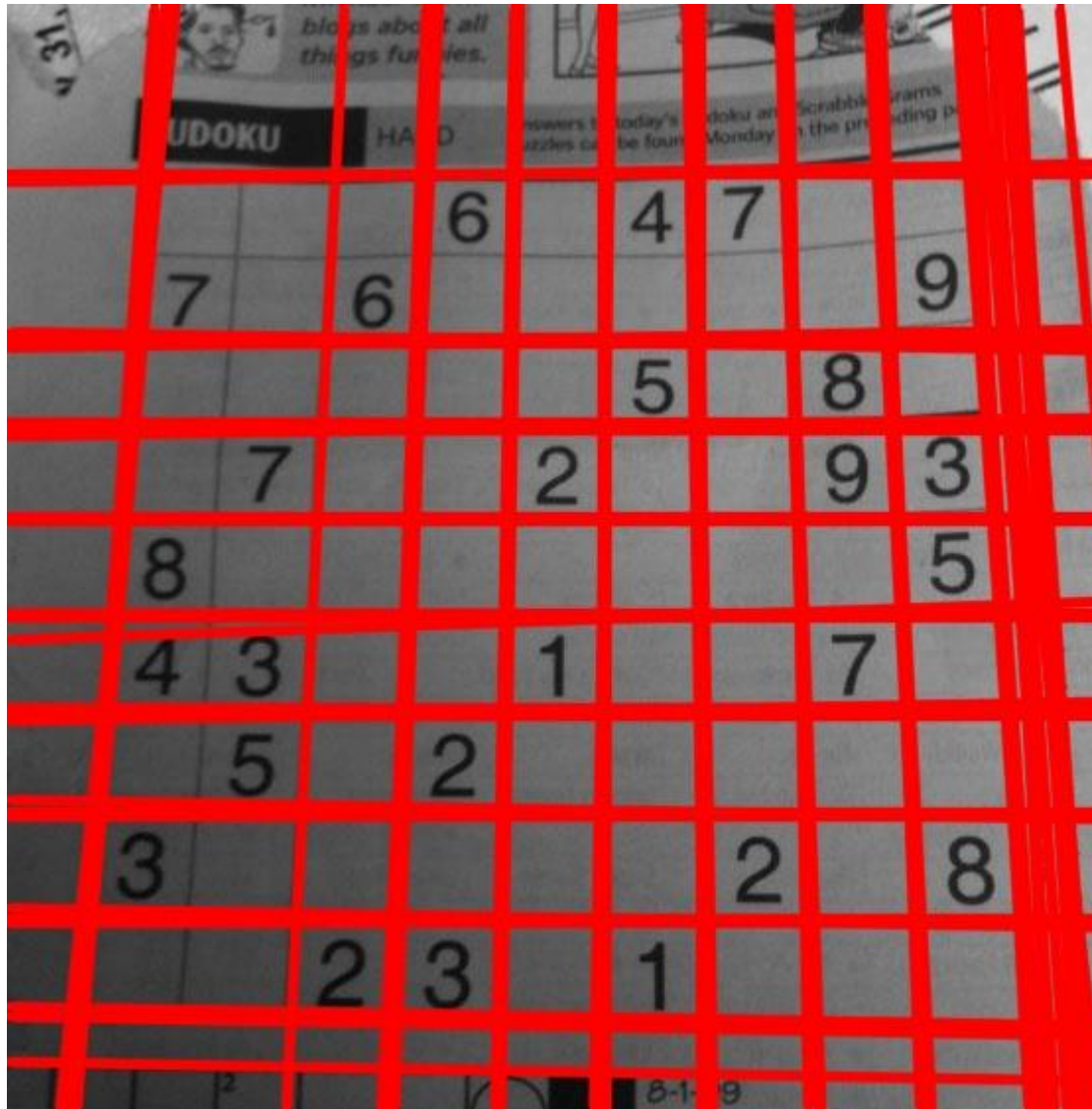
- Find the value(s) of  $(\rho, \theta)$  where  $H(\rho, \theta)$  is a large local maximum

H: accumulator array (votes)

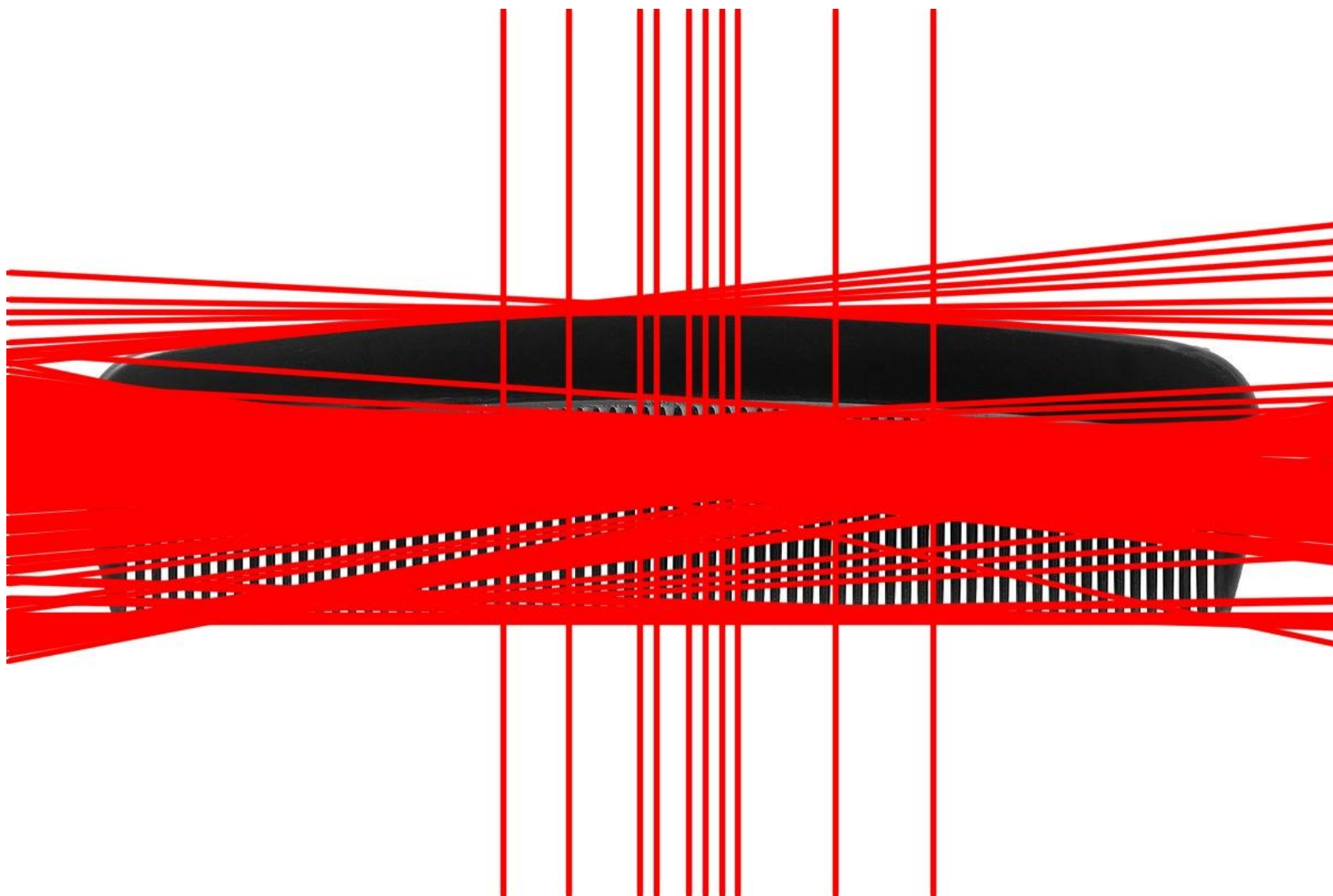




# تبدیل Hough



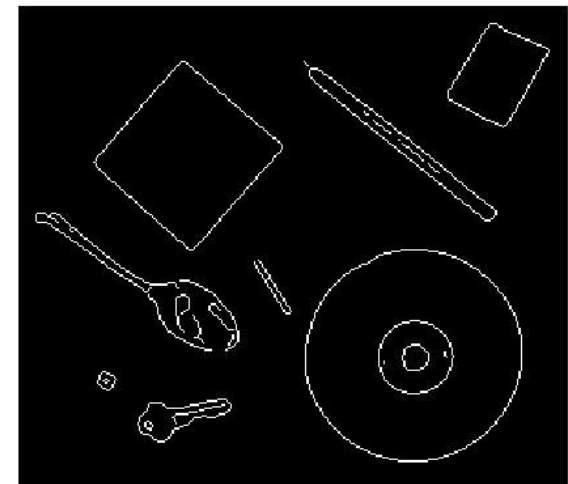
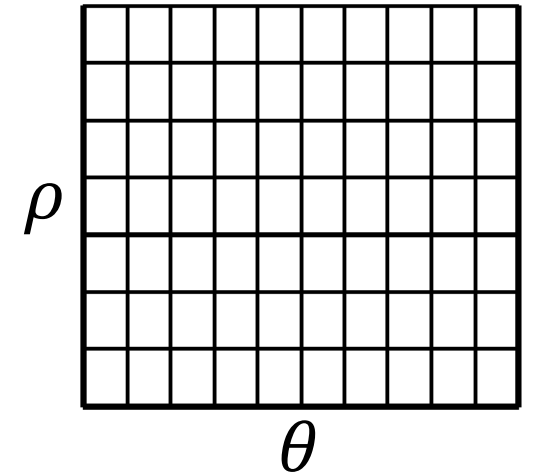
# تبدیل Hough



# شبه کد تبدیل Hough

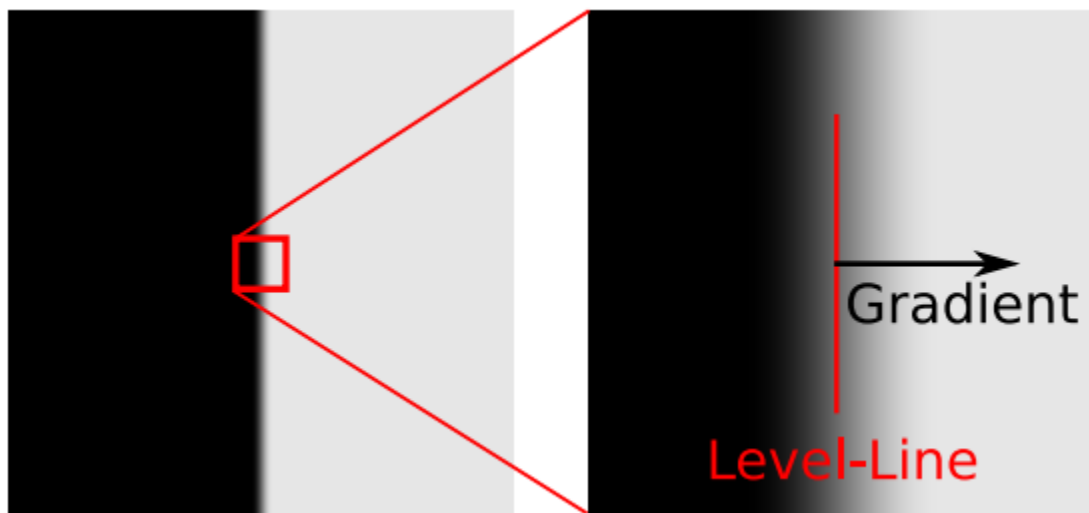
- Initialize accumulator  $H$  to all zeros
- For each edge point  $(x, y)$  in the image  
For  $\theta = 0$  to  $180$   
if  $|\cos(\theta - \text{dir}(x, y))| > \text{threshold}$   
 $\rho = x \cos \theta + y \sin \theta$   
 $H(\rho, \theta) = H(\rho, \theta) + 1$
- Find the value(s) of  $(\rho, \theta)$  where  $H(\rho, \theta)$  is a large local maximum

H: accumulator array (votes)

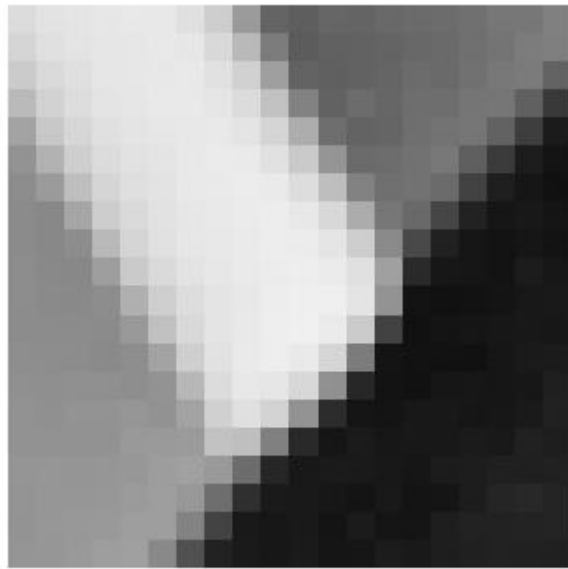
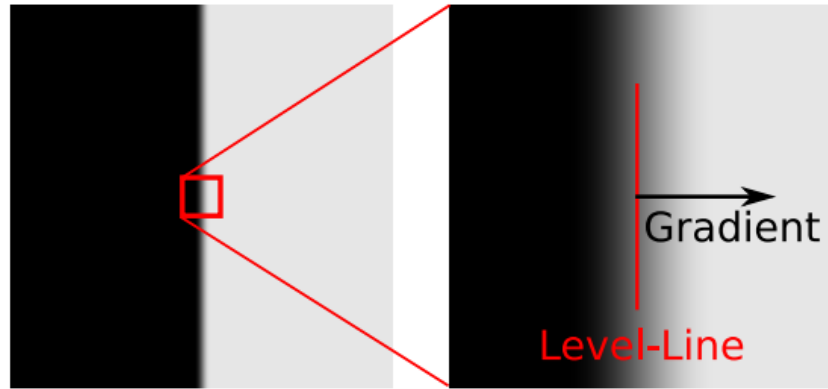


# تشخیص پاره خط (LSD)

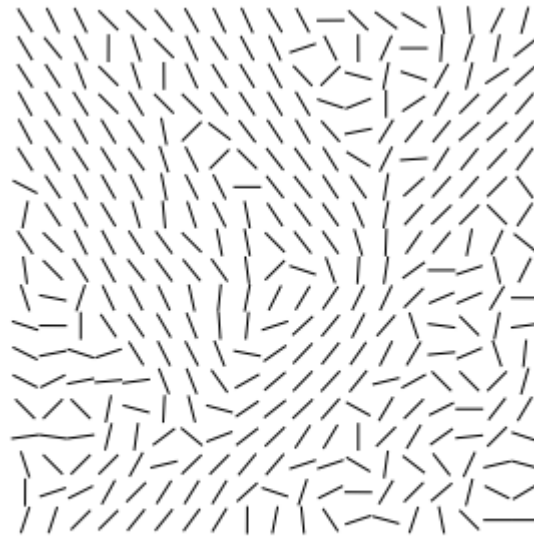
- هدف از این الگوریتم یافتن نقاط ابتدا و انتهای پاره خط‌های موجود در تصویر است
- هر پاره خط بجای ۲ پارامتر توسط ۴ پارامتر مشخص می‌شود
- مزیت اصلی الگوریتم LSD آن است که به خوبی از جهت گرادیان استفاده می‌کند



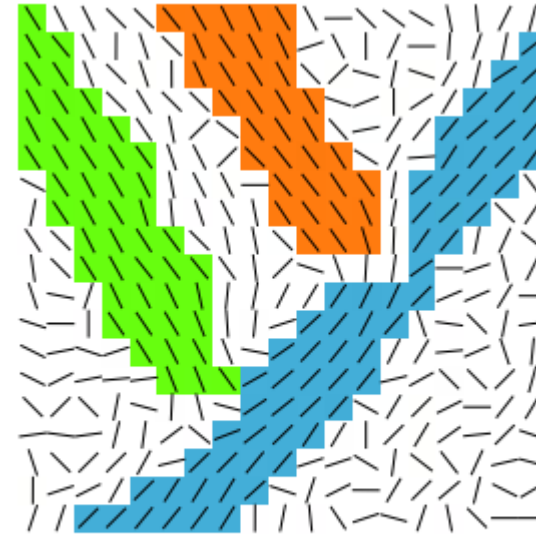
# تشخیص پاره خط (LSD)



Image



Level-line Field



Line Support Regions

# تشخيص پاره خط (LSD)





# تشخیص پاره خط (LSD)

