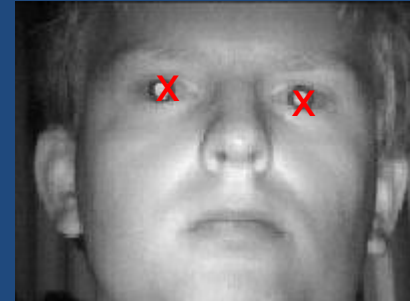


ÖZNİTELİK ALGILAMA

- Görüntü öznitelikleri: Önemli/ilginç yerel şablonlar (paternler)
- Öznitelikleri algılamak görüntüdeki nesneleri yerelleştirme ya da tanımlamada önemli bir adım olabilir.(“Öznitelik temelli yöntemler”)

-Örnek öznitelikler:

- *Kenarlar,
- *Çizgi ve eğriler,
- *Köşeler,
- *Uygulamaya özel şablonlar(paternler)



Kızıl ötesi aydınlatma ile uyuşukluk

algılama, N.EAGLE EE368 sınıf projesi

Kenar algılama

-Amaç(Sürekli-yer)(idea continuous-space):
Yerel eğimi algıla

$$|grad(f(x, y))| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

-Dijital görüntü: Sonlu farkları kullan

fark
merkezi fark
prewitt
sobel

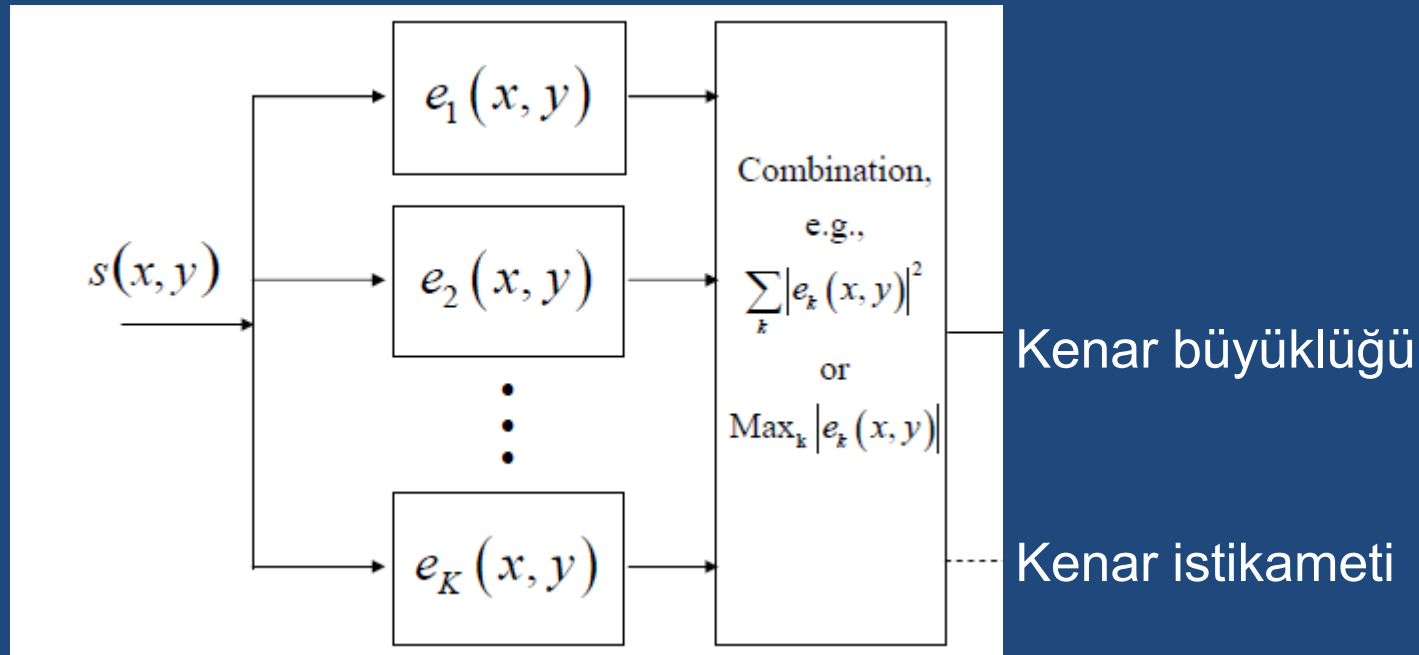
$$\begin{pmatrix} -1 & 1 \end{pmatrix}$$
$$\begin{pmatrix} -1 & [0] & 1 \end{pmatrix}$$
$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix}$$
$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & [0] & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Pratik kenar algılayıcılar

Kenarlar herhangi bir uyarlamaya sahip olabilirler

Tipik kenar algılama düeni $K=2$ kenar şablonunu (template) kullanır

Bazıları ise $K>2$ uygular



Kenar algılama filtreleri

Roberts

$$\begin{pmatrix} [0] & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} [1] & 0 \\ 0 & -1 \end{pmatrix}$$

Prewitt

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Sobel

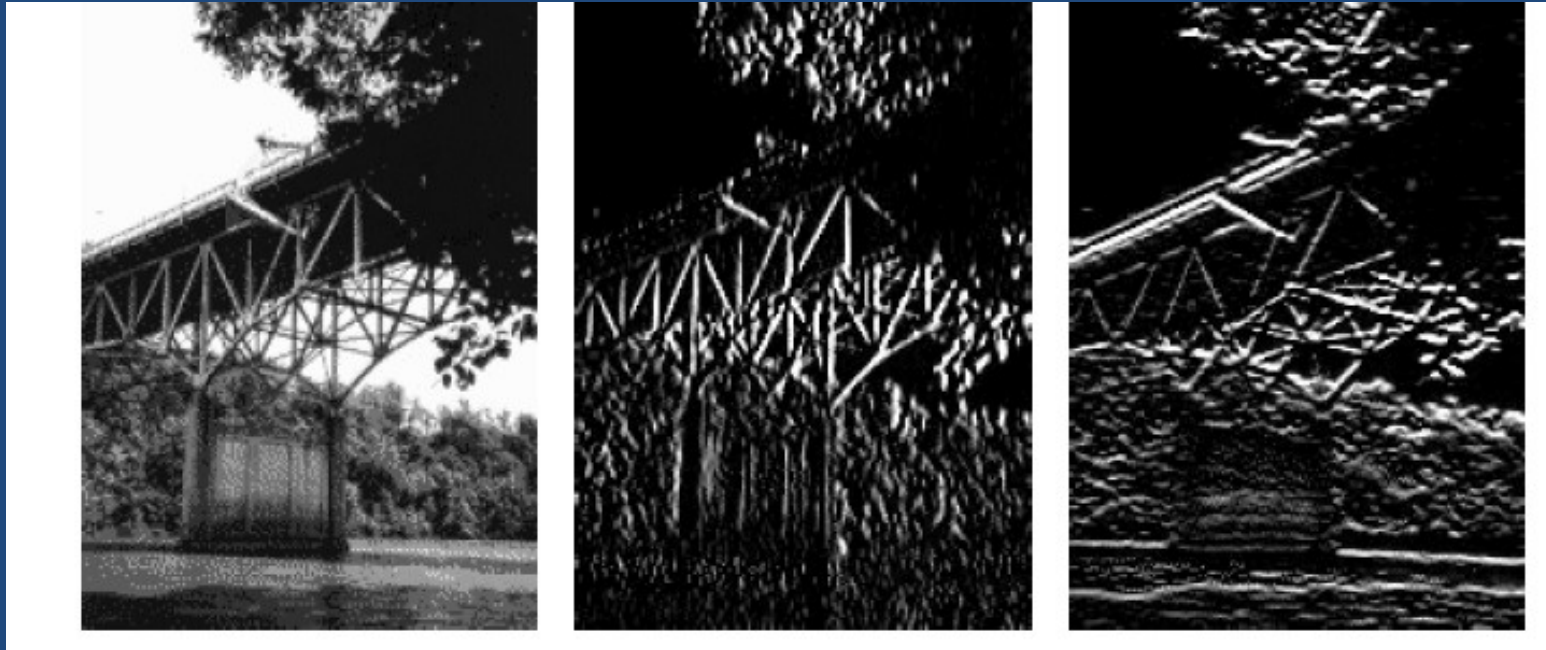
$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & [0] & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & -1 \\ 0 & [0] & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Kirsch

$$\begin{pmatrix} +5 & +5 & +5 \\ -3 & [0] & -3 \\ -3 & -3 & -3 \end{pmatrix} \begin{pmatrix} -3 & +5 & +5 \\ -3 & [0] & +5 \\ -3 & -3 & -3 \end{pmatrix} \begin{pmatrix} -3 & -3 & +5 \\ -3 & [0] & +5 \\ -3 & -3 & +5 \end{pmatrix} \begin{pmatrix} -3 & -3 & -3 \\ -3 & [0] & +5 \\ -3 & +5 & +5 \end{pmatrix}$$

$$\begin{pmatrix} -3 & -3 & -3 \\ -3 & [0] & -3 \\ +5 & +5 & +5 \end{pmatrix} \begin{pmatrix} -3 & -3 & -3 \\ +5 & [0] & -3 \\ +5 & +5 & -3 \end{pmatrix} \begin{pmatrix} +5 & -3 & -3 \\ +5 & [0] & -3 \\ +5 & -3 & -3 \end{pmatrix} \begin{pmatrix} +5 & +5 & -3 \\ +5 & [0] & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

Prewitt kullanıcı örneği



Gerçek köprü

220 x 160

alttaki matrislerle filtrelenmiş
görüntüler

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Prewitt kullanıcı örneği



Gerçek görüntü
büyüklüğü

310 x 241

Filtrelenmiş görüntülerin log (Laplace of Gaussian)

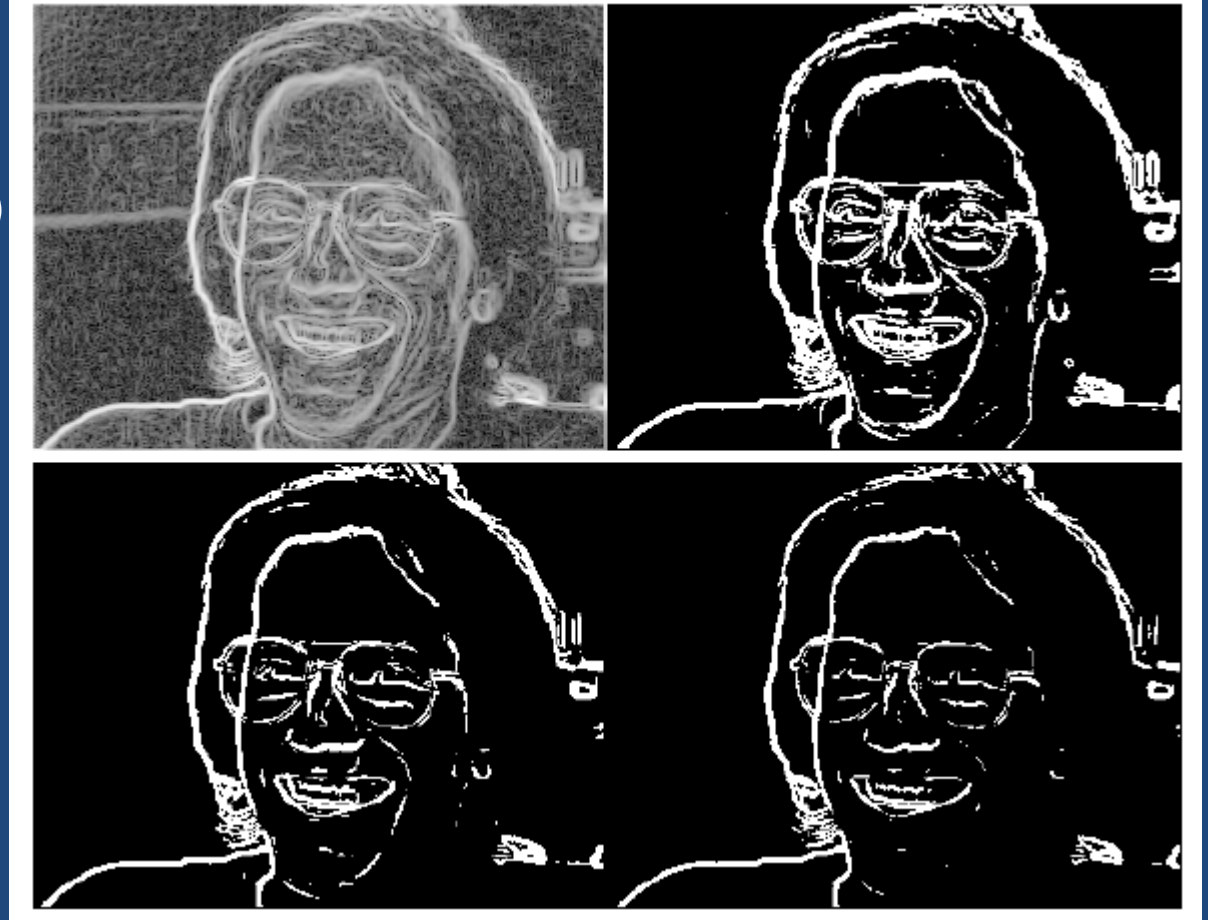
$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & [0] & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & -1 & -1 \\ 0 & [0] & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Prewitt kullanıcı örneği

Yatay ve düşey
eğimlerin karelerinin
 \log (Laplace of Gaus.)
özeti

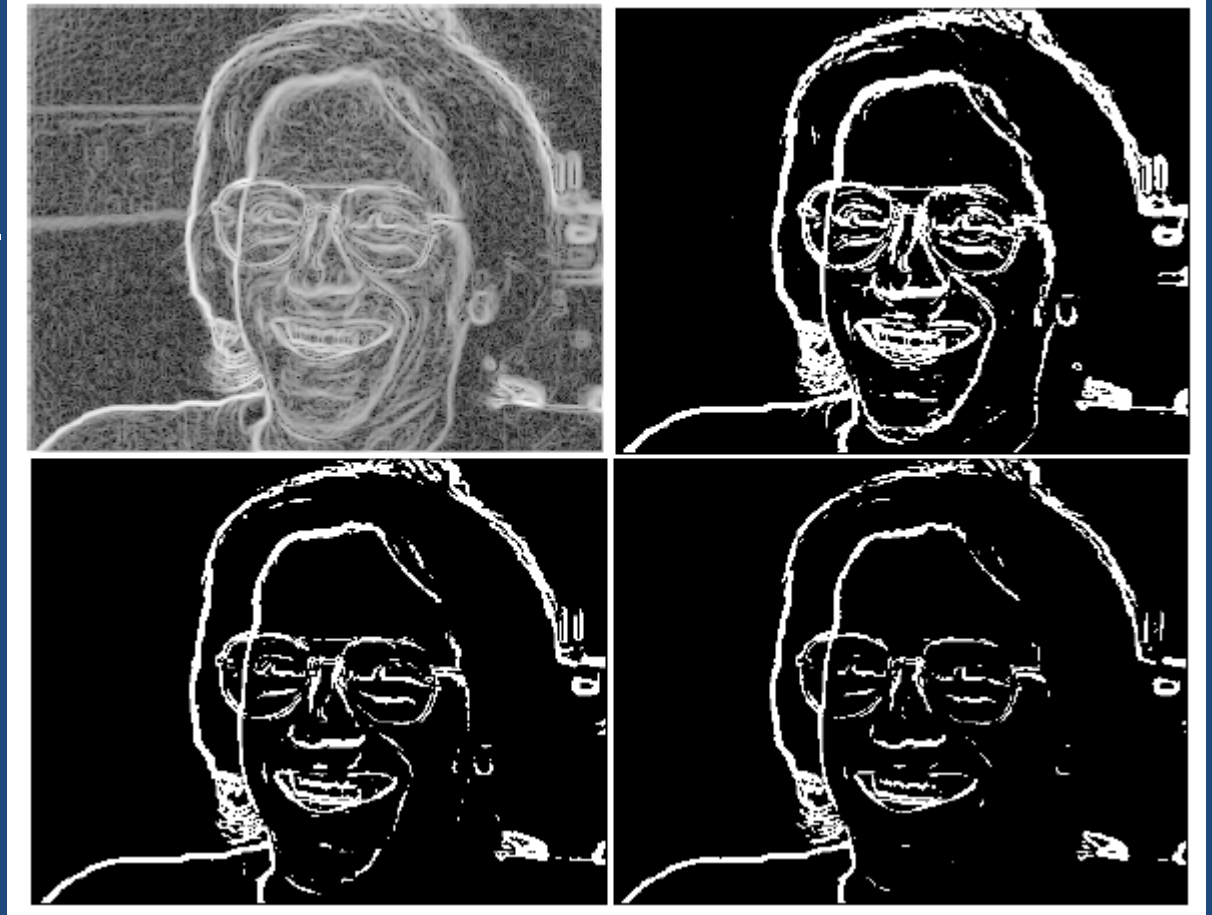
Farklı eşik değerleri



Sobel kullanıcı örneği

Yatay ve düşey
eğimlerin karelerinin
 \log (Laplace of Gaus.
özeti

Farklı eşik değerleri



Roberts kullanıcı örneği



Orijinal görüntü
büyüklüğü
309 x 240

Filtrelenmiş görüntülerin log (Laplace of Gaussian)

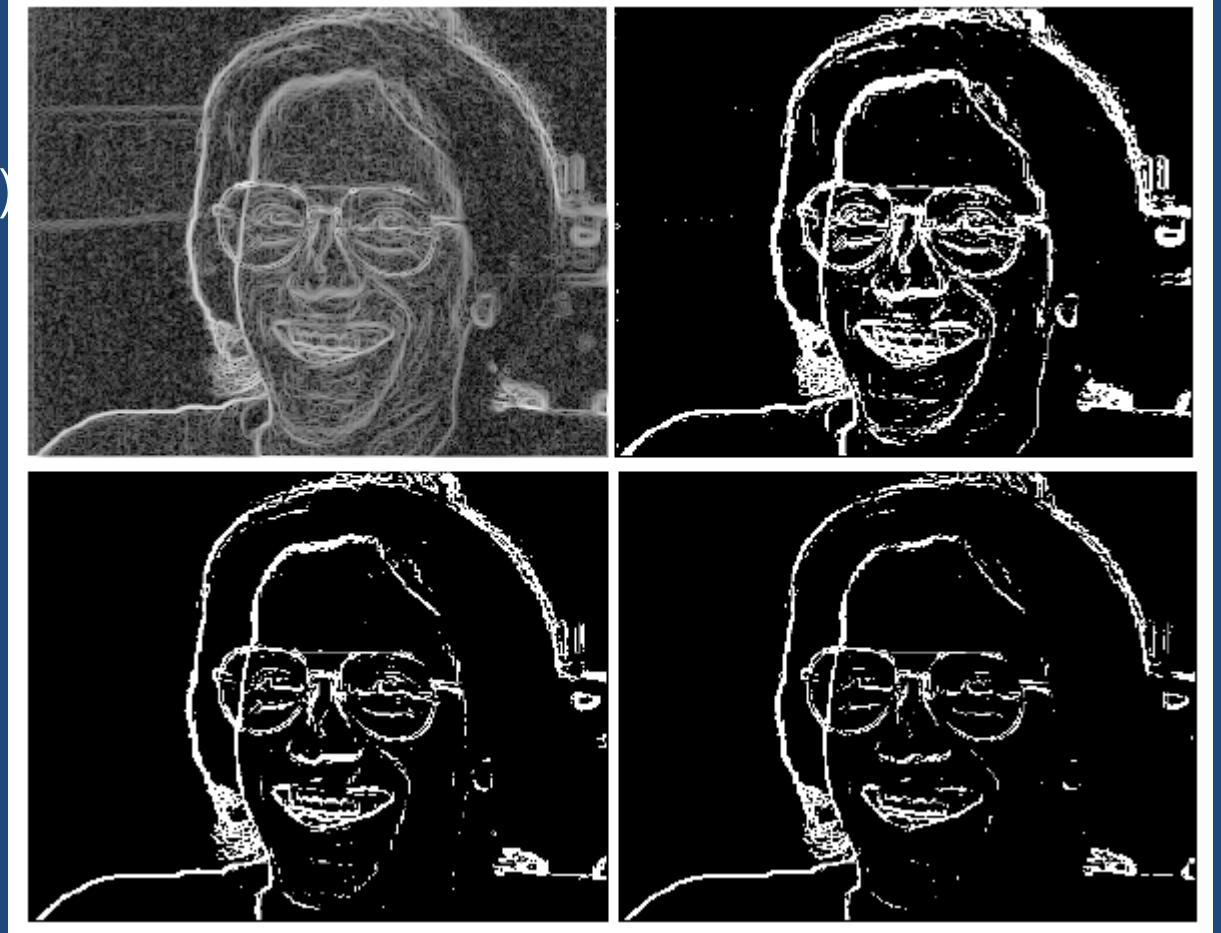
$$\begin{pmatrix} [1] & 0 \\ 0 & -1 \end{pmatrix}$$

$$\begin{pmatrix} [0] & 1 \\ -1 & 0 \end{pmatrix}$$

Roberts kullanıcı örneği

Köşegen
eğimlerinin karelerinin
 \log (Laplace of Gaus.)
özeti

Farklı eşik değerleri



Laplace kullanıcısı

- İkinci türevi hesaba katarak süreksizlikleri algıla

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

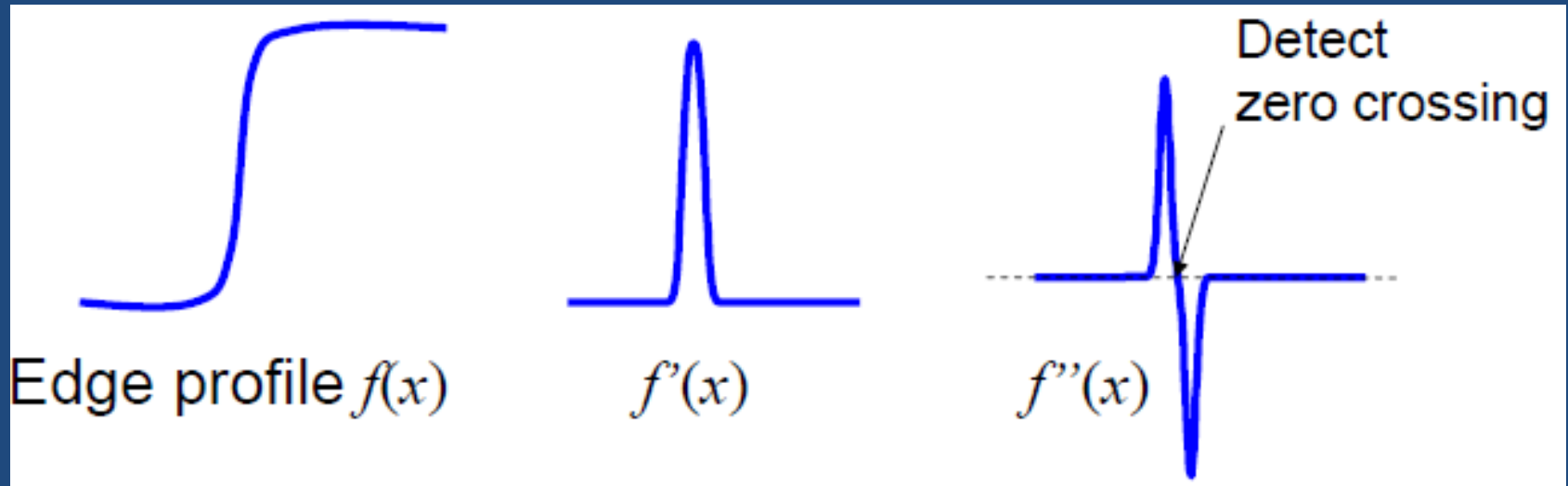
- Dönüsel değişmeyen kullanıcı
- Sıfır geçişleri noktası kenar konumu
- 3x3 itme tepkisiyle konvolüsyon sayesinde ayırık uzay tahmini

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & [-4] & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

or

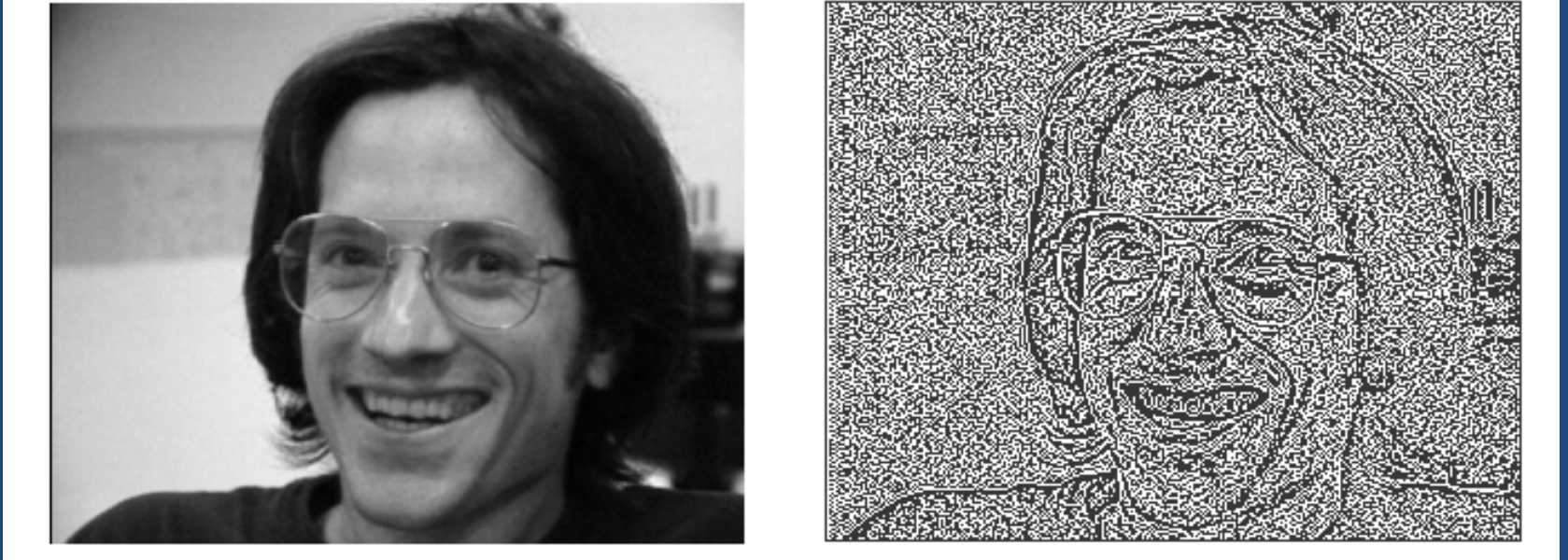
$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & [-8] & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

2nci türev kenar algılayıcısının 1-d aydınlatması



Sıfır geçişini algıla

Laplace'in sıfır geçişleri



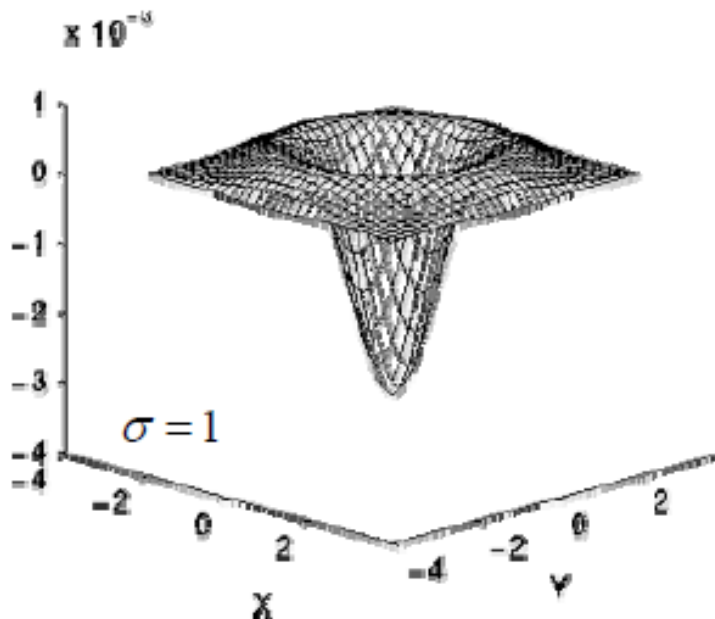
Çok iyi detaya ve gürültüye duyarlı -> Başlangıçta bulanık
Güçlü ve zayıf kenarlara eşit biçimde tepki verir -> Düşük
eğimle kenarları bastırır.

Gaus Laplace'ı

Görüntünün Laplace ve Gaus kullanıcılığıyla bulanıklaştırılması, Gaus Laplace(LoG) 'ıyla konvolüsyonla birleştirilebilir

$$LoG(x,y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Sürekli fonksiyon ve ayırık tahmin



$\sigma = 1.4$

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

LoG'un sıfır geçişleri

w/o
Gaus



$\sigma = 1.4$

$\sigma = 3$

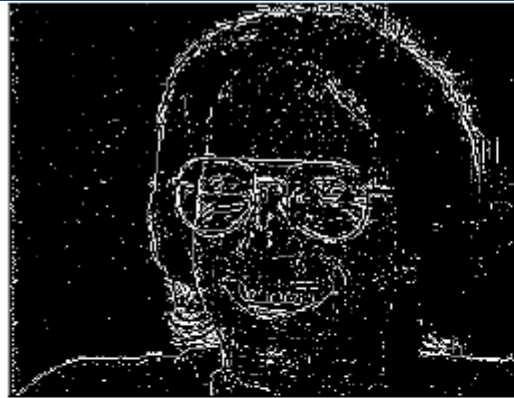


$\sigma = 6$

LoG'un sıfır geçişleri- Eğim(Gradyan) bazlı eşik

w/o
Gaus

$\sigma = 3$



$\sigma = 1.4$



$\sigma = 6$

Canny kenar algılayıcı

- Gaus ile filtrelenmiş düzgün (pürüzsüz) bir görüntü
- Eğimin büyüklüğünü ve açığı hesapla(rewitt,sobel)

$$M(x, y) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$
$$\alpha(x, y) = \tan^{-1} \left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right)$$

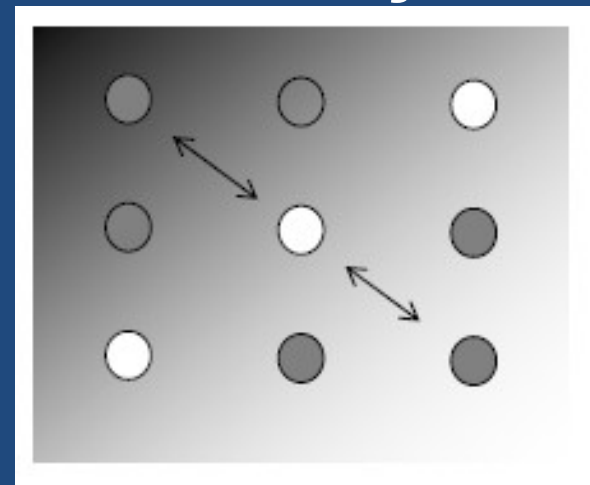
- Eğim büyüklüğü görüntüsüne maksimum olmayan bastırmayı uygula
- Güçlü ve zayıf kenar piksellerini algılamak için çift eşik değeri uygula
- Belirgin kenarla ilgisi olmayan zayıf kenar piksellerini çıkar (at)

Canny maksimum olmayan bastırma (nonmaxima suppression)

-Kenar normalini yönlerinden birine nicelendir:

yatay: -45, dikey: +45

- $M(x,y)$ kenar normali yönündeki komşularının herhangi birinden küçükse -> bastır, değilse devam et



Zayıf kenarların Canny ile eşiklemesi ve bastırılması

-Gradyan(eğim) büyüklüğünün çift eşiklemesi

Strong edge:	$M(x, y) \geq \theta_{high}$
Weak edge:	$\theta_{high} > M(x, y) \geq \theta_{low}$

$$\theta_{high} / \theta_{low} = 2...3$$

-Tipik ayar:

-Kenar piksellerin bölge etiketlemesi

-Belirgin(güçlü) kenar pikselleri dışındaki bölgelerin çıkarılması

Canny kenar algılayıcı



$\sigma = 1.4$



$\sigma = 3$



$\sigma = 6$

Canny kenar algılayıcı

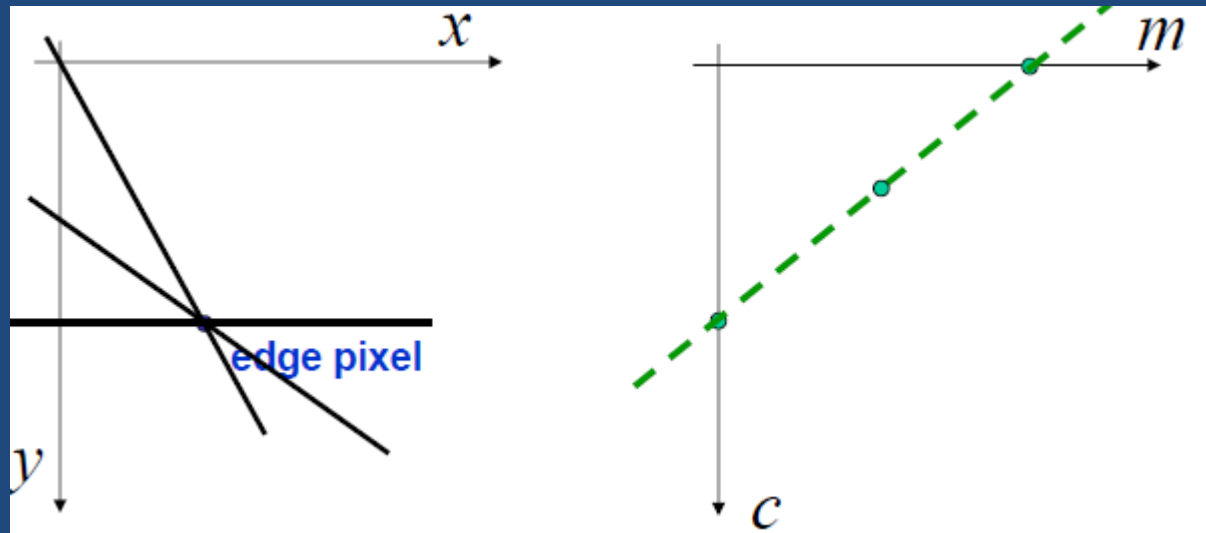


$\sigma = 1.4$

Hough(Temizleme) dönüşümü

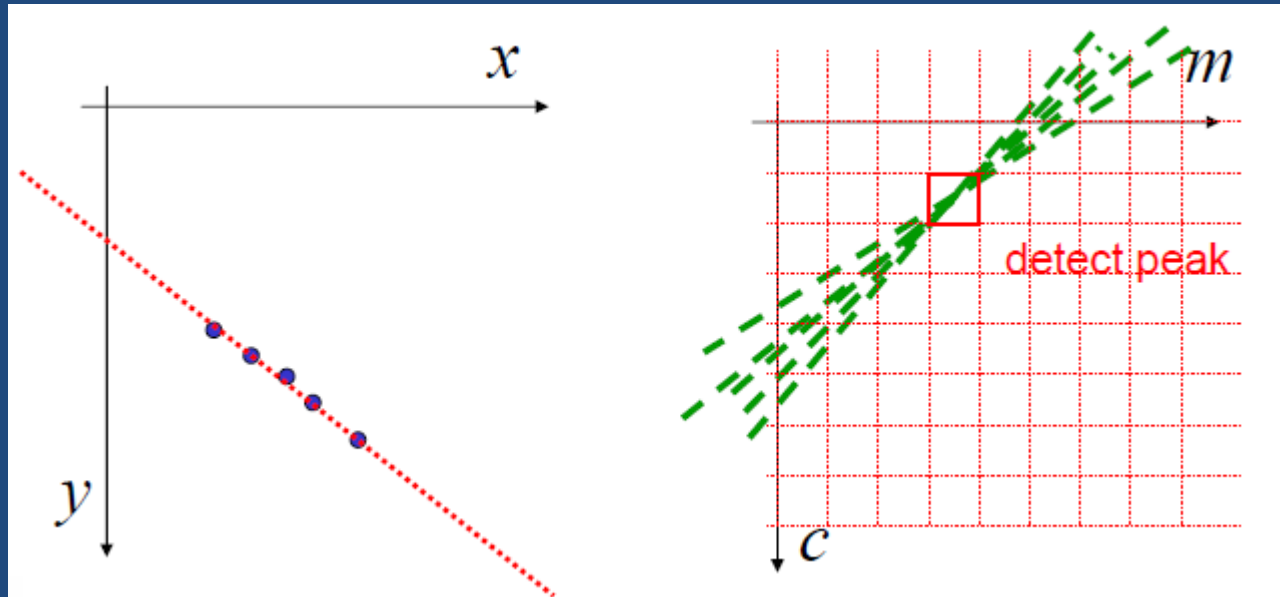
- Problem: Bir kenar piksel grubunun içine düz bir çizgi ya da eğri sığdır
- Hough dönüşümü(1962): Genel şablon eşleme tekniği
- Düz çizgilerin algılandığını göz önünde bulundur

$$y = mx + c$$



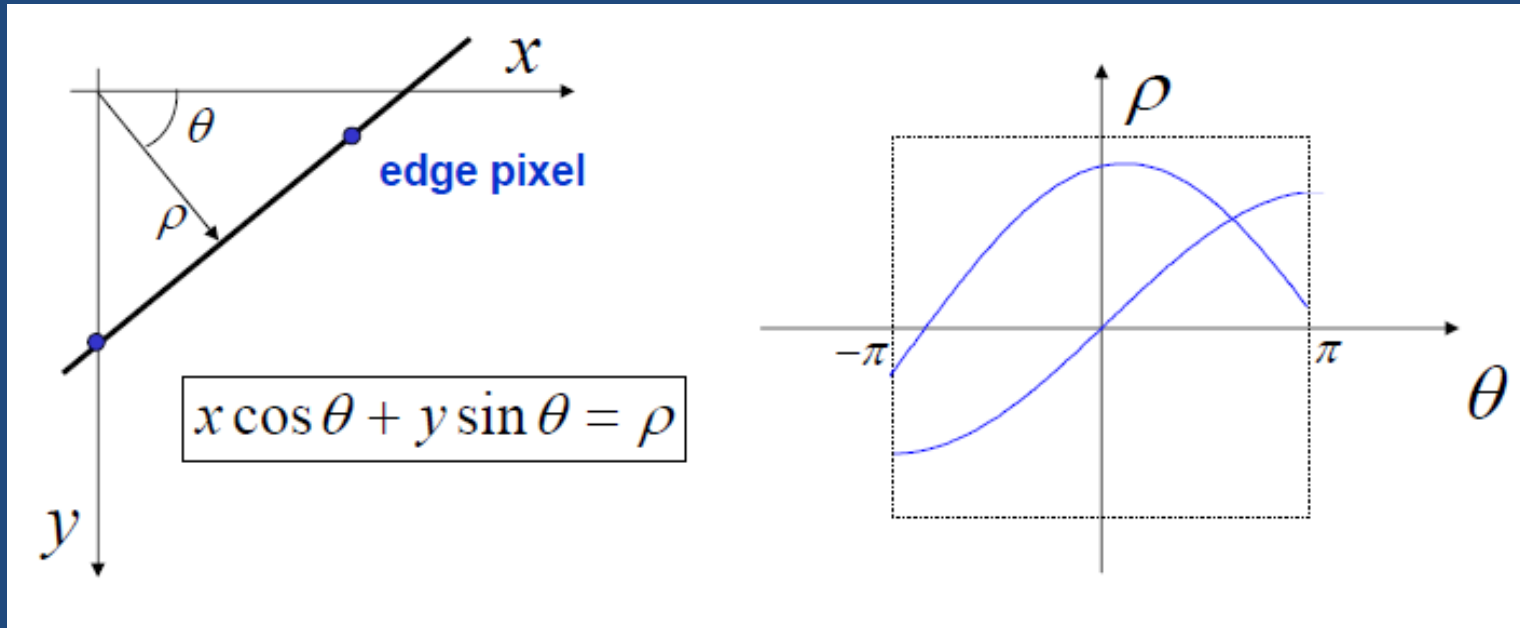
Hough dönüşümü

- m, c düzlemini ayırık karelere böl, tüm kareleri sıfırdan başlat
- m, c düzlemindeki her bir kenar pikseli x, y için bir doğru çiz ve doğru boyunca kare sayısını artır.
- m, c düzlemindeki tepe noktalarını tespit et (algıla)



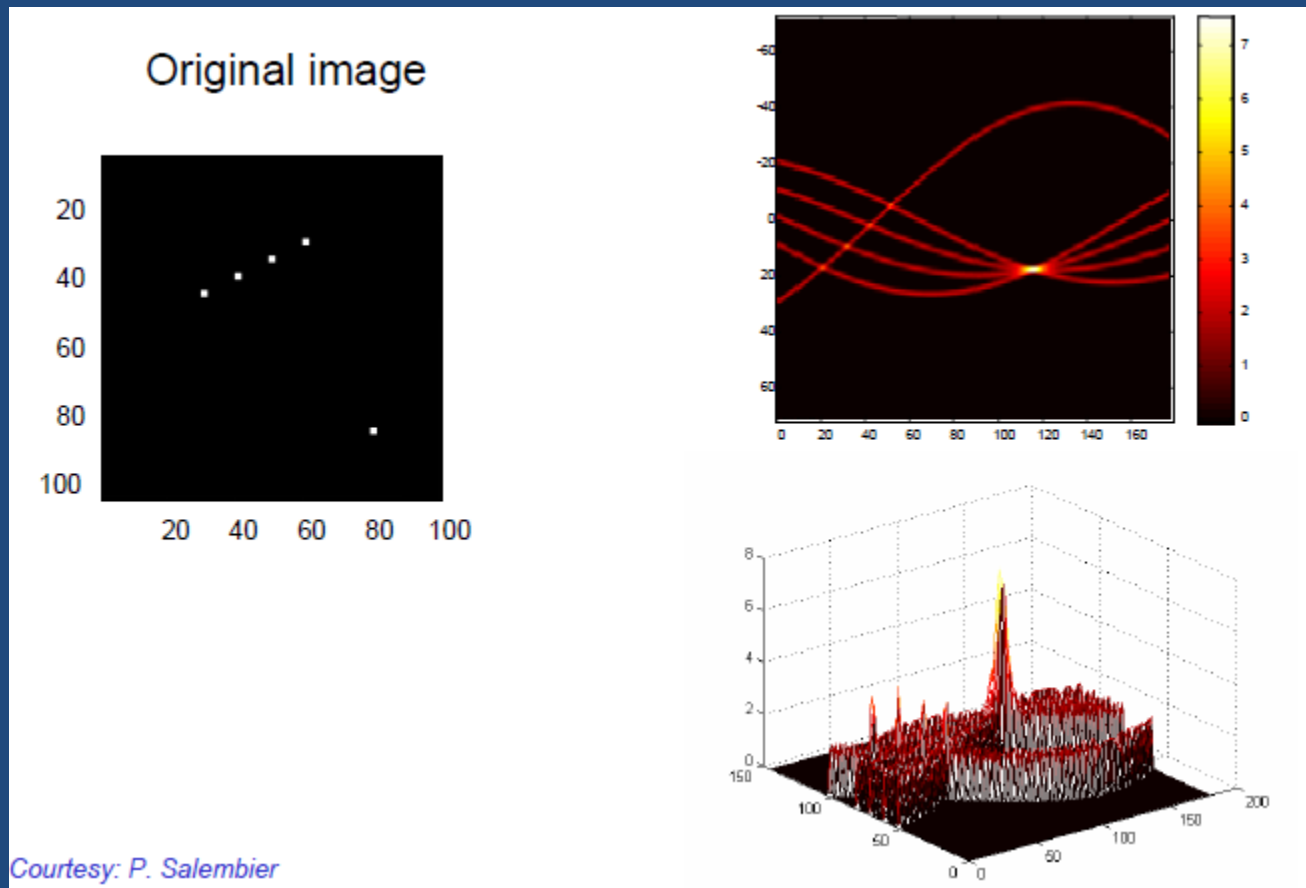
Hough dönüşümü

Alternatif ölçümler sonsuz eğim (infinite-slope) problemini ihmal eder.

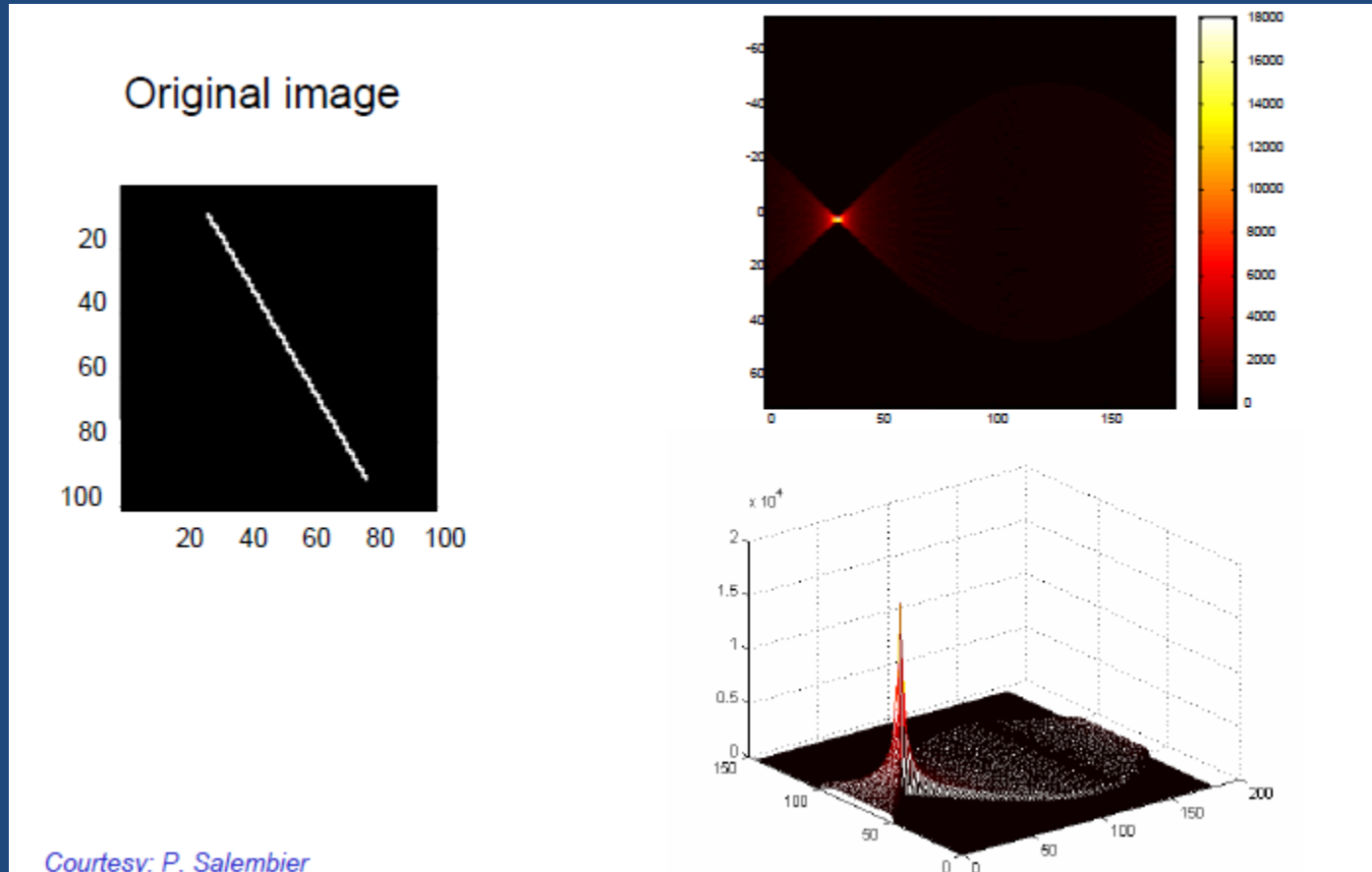


Radon dönüşümüne benzer

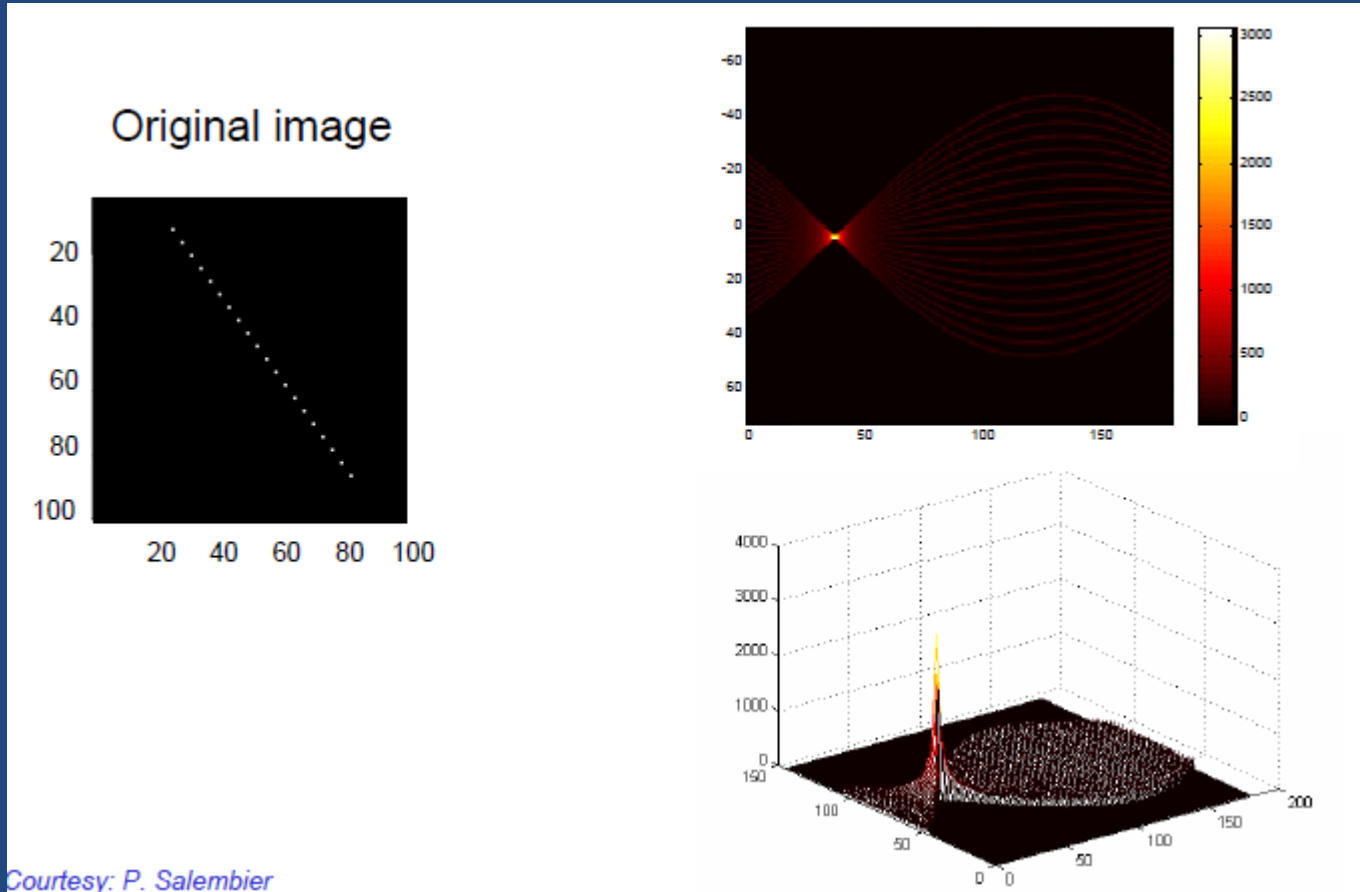
Hough dönüşümü Örnek A



Hough dönüşümü Örnek B

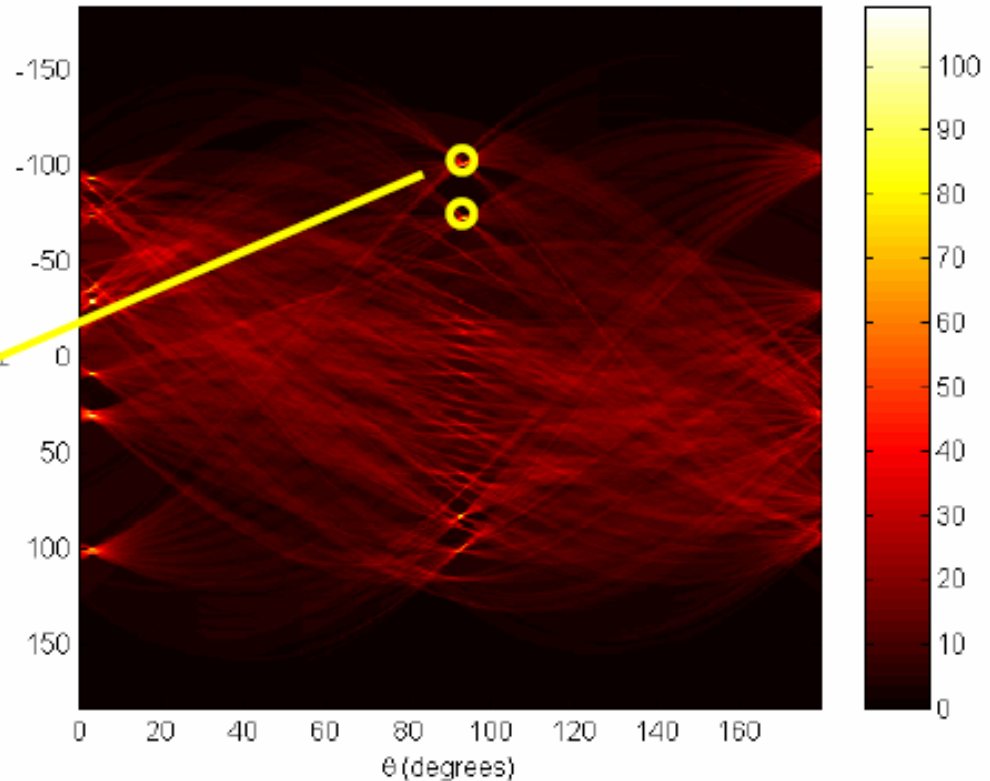
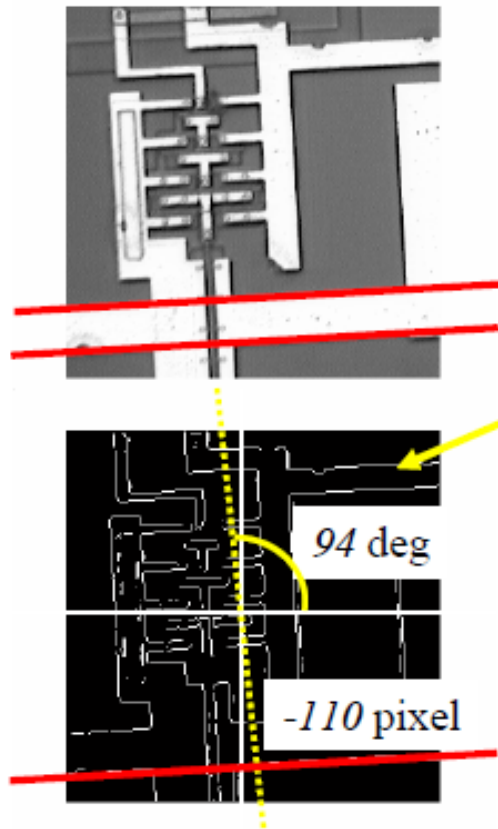


Hough dönüşümü Örnek C



Hough dönüşüm örneği

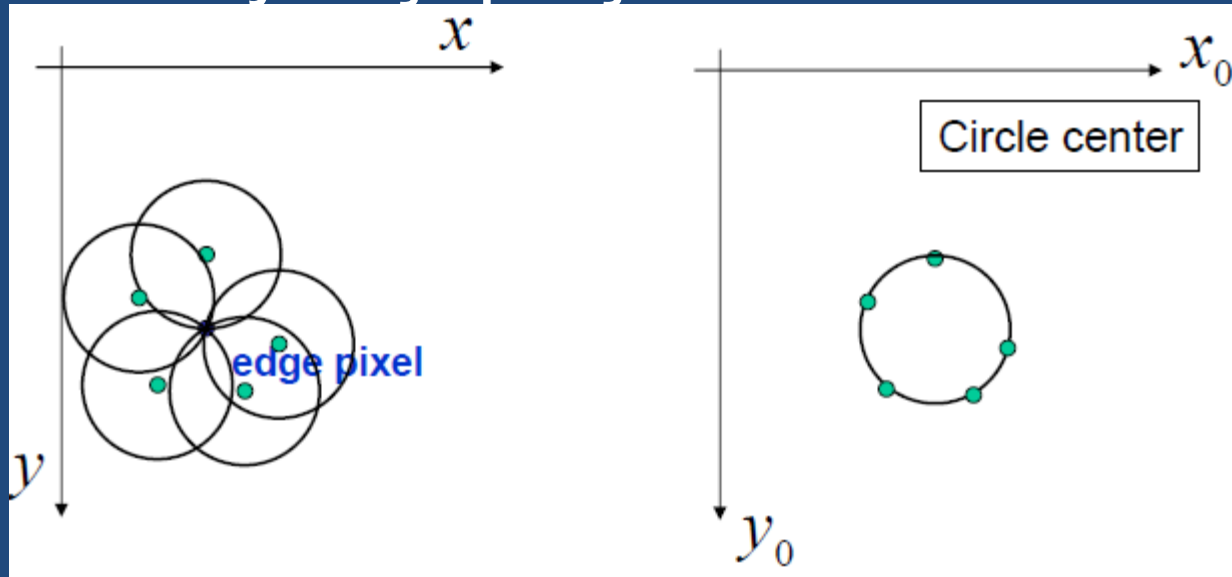
Original IC image (256x256)



**Kenar algılama
(Prewitt)**

Hough dönüşümü ile yuvarlak (çember) algılama

-İşaretli r yarıçaplı çemberleri bul

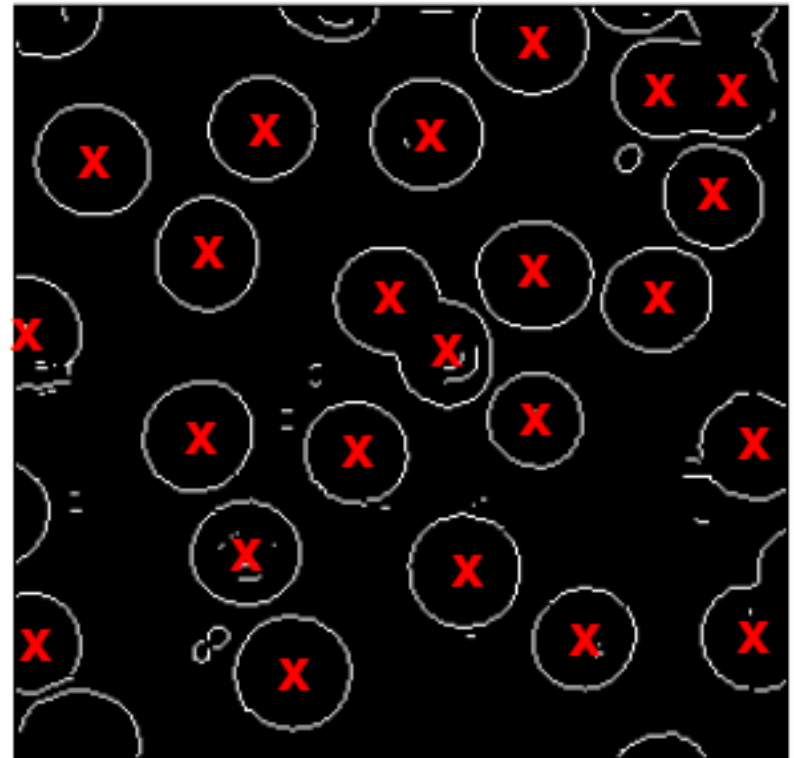
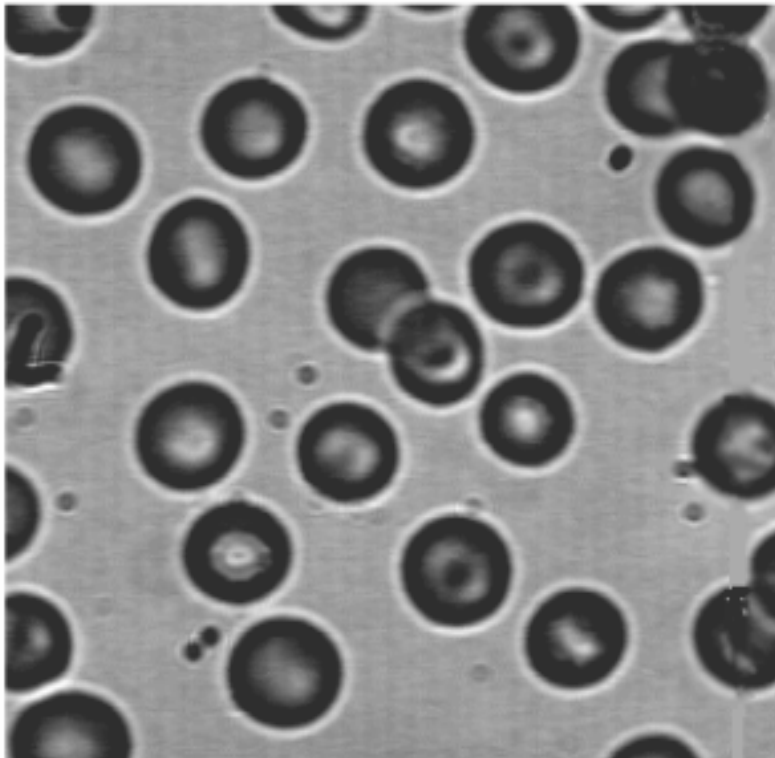


-Belirsiz yarıçaplı çemberler için, (x_0, y_0, r) parametrelili 3-d hough dönüşümünü kullan

Örnek: Hough dönüşümü ile çember algılama

Gerçek kan görüntüsü

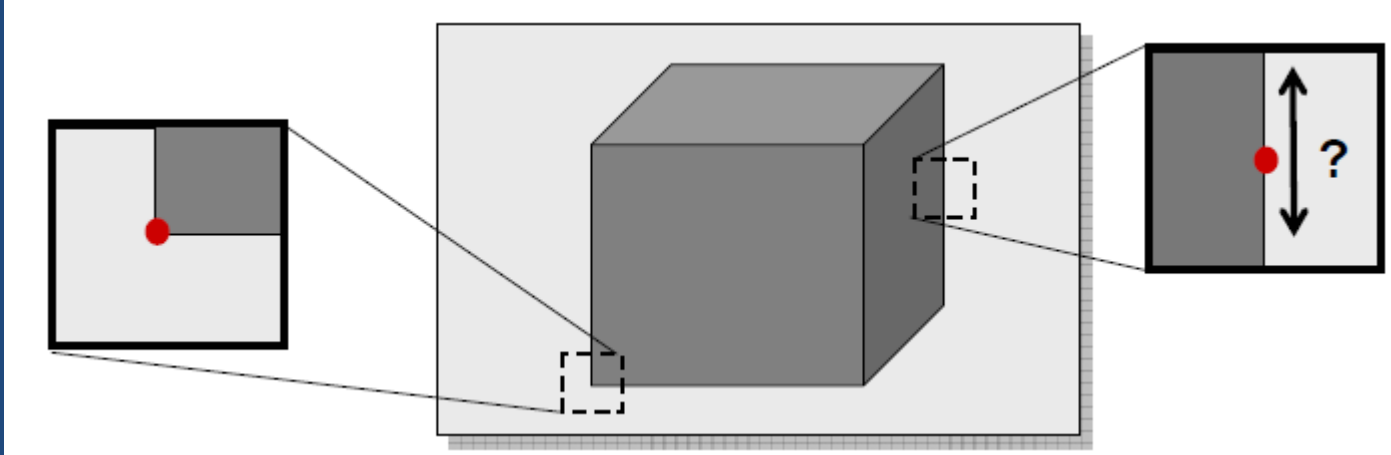
Prewitt kenar algılama



Köşe noktaları algılama

Birçok uygulamalar (x,y) konumlarındaki özniteliklerden faydalanır.

Kenarlar sadece bir yönde daha iyi (çok) konumlanır. -> Köşeleri algıla



Köşe algılayıcı için istenen özellikler:

- Doğru konum(yer)
- Ölçek, parlaklık, dönü ve yöne göre değişmez olma
- Gürültü (istenmeyen görüntü) ve tekrarlamalara karşı dayanıklı olma

En doğru konumlanan paternler hangileridir?

-Yerel uzaklık duyarlılığı (local displacement sensitivity)

$$S(\Delta x, \Delta y) = \sum_{(x,y) \in \text{window}} [f(x, y) - f(x + \Delta x, y + \Delta y)]^2$$

-Küçük $\Delta x, \Delta y$ için doğrusal tahmin

$$f(x + \Delta x, y + \Delta y) \approx f(x, y) + f_x(x, y) \Delta x + f_y(x, y) \Delta y$$

$$\begin{aligned} S(\Delta x, \Delta y) &\approx \sum_{(x,y) \in \text{window}} \left[\begin{pmatrix} f_x(x, y) & f_y(x, y) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right]^2 \\ &= (\Delta x \quad \Delta y) \left(\sum_{(x,y) \in \text{window}} \begin{bmatrix} f_x^2(x, y) & f_x(x, y) f_y(x, y) \\ f_x(x, y) f_y(x, y) & f_y^2(x, y) \end{bmatrix} \right) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \\ &= (\Delta x \quad \Delta y) \mathbf{M} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \end{aligned}$$

-Eşit duyarlılık (iso-sensitivity) eğrileri elipstir

Anahtar nokta algılama

Sıklıkla yapı (normal) matrisinin özdeğerleri esasına dayanır.

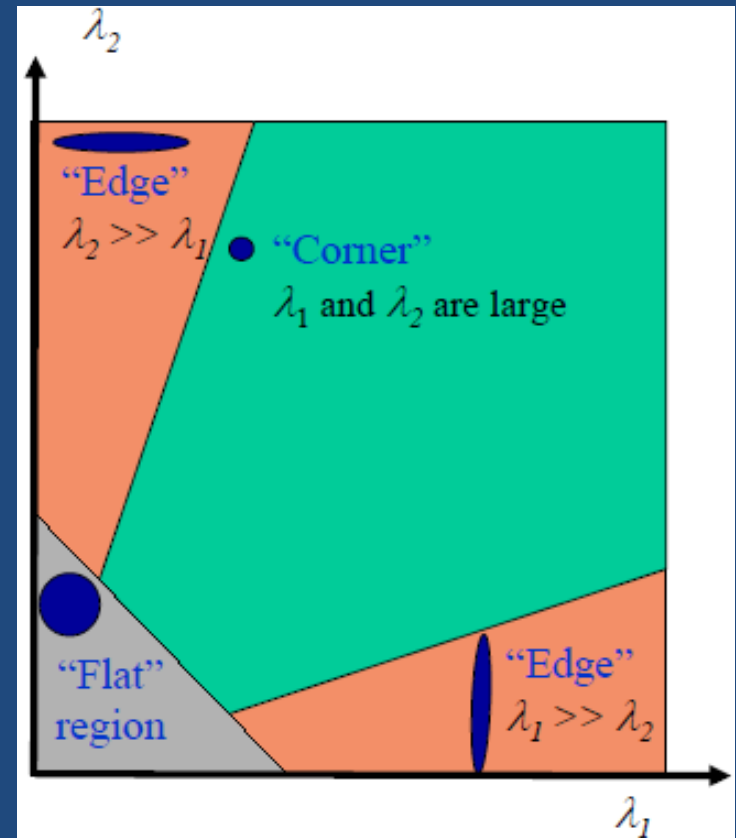
λ_1, λ_2

$$\mathbf{M} = \begin{bmatrix} \sum_{(x,y) \in \text{window}} f_x^2(x,y) & \sum_{(x,y) \in \text{window}} f_x(x,y) f_y(x,y) \\ \sum_{(x,y) \in \text{window}} f_x(x,y) f_y(x,y) & \sum_{(x,y) \in \text{window}} f_y^2(x,y) \end{bmatrix}$$

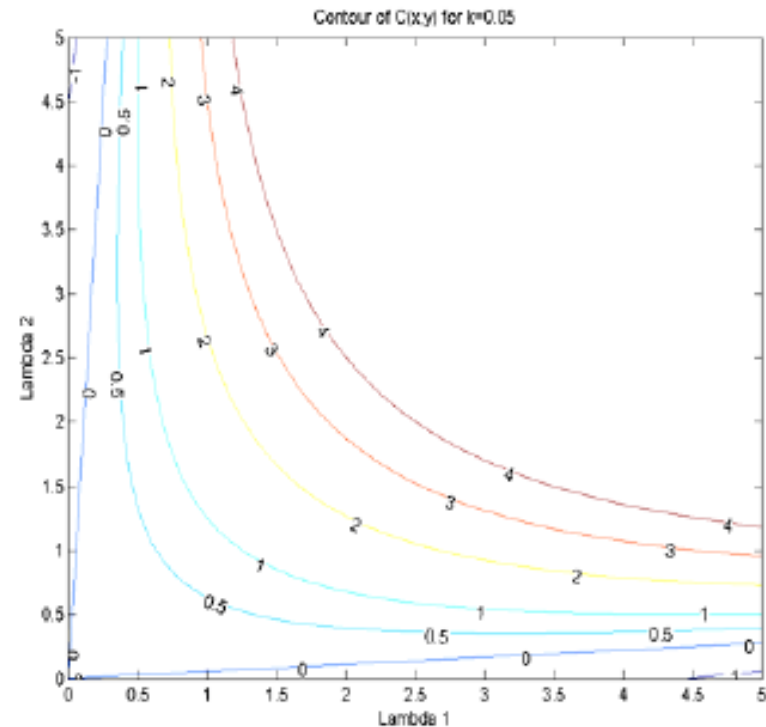
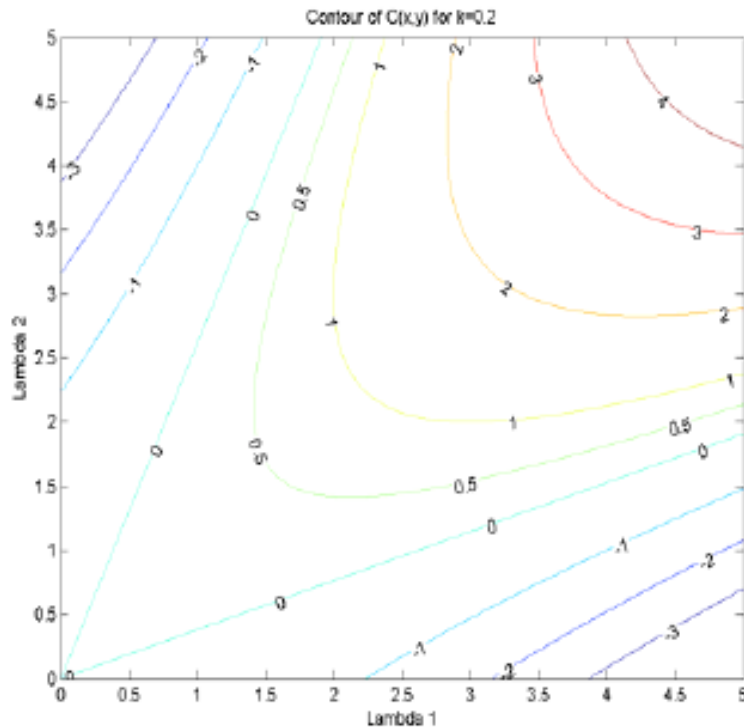
$f_x(x,y)$ – horizontal image gradient
 $f_y(x,y)$ – vertical image gradient

Köşe'lik ölçümü

$$\begin{aligned} C(x,y) &= \det(\mathbf{M}) - k \cdot (\text{trace}(\mathbf{M}))^2 \\ &= \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2) \end{aligned}$$



Harris köşeliğinin eş yükselti grafiği

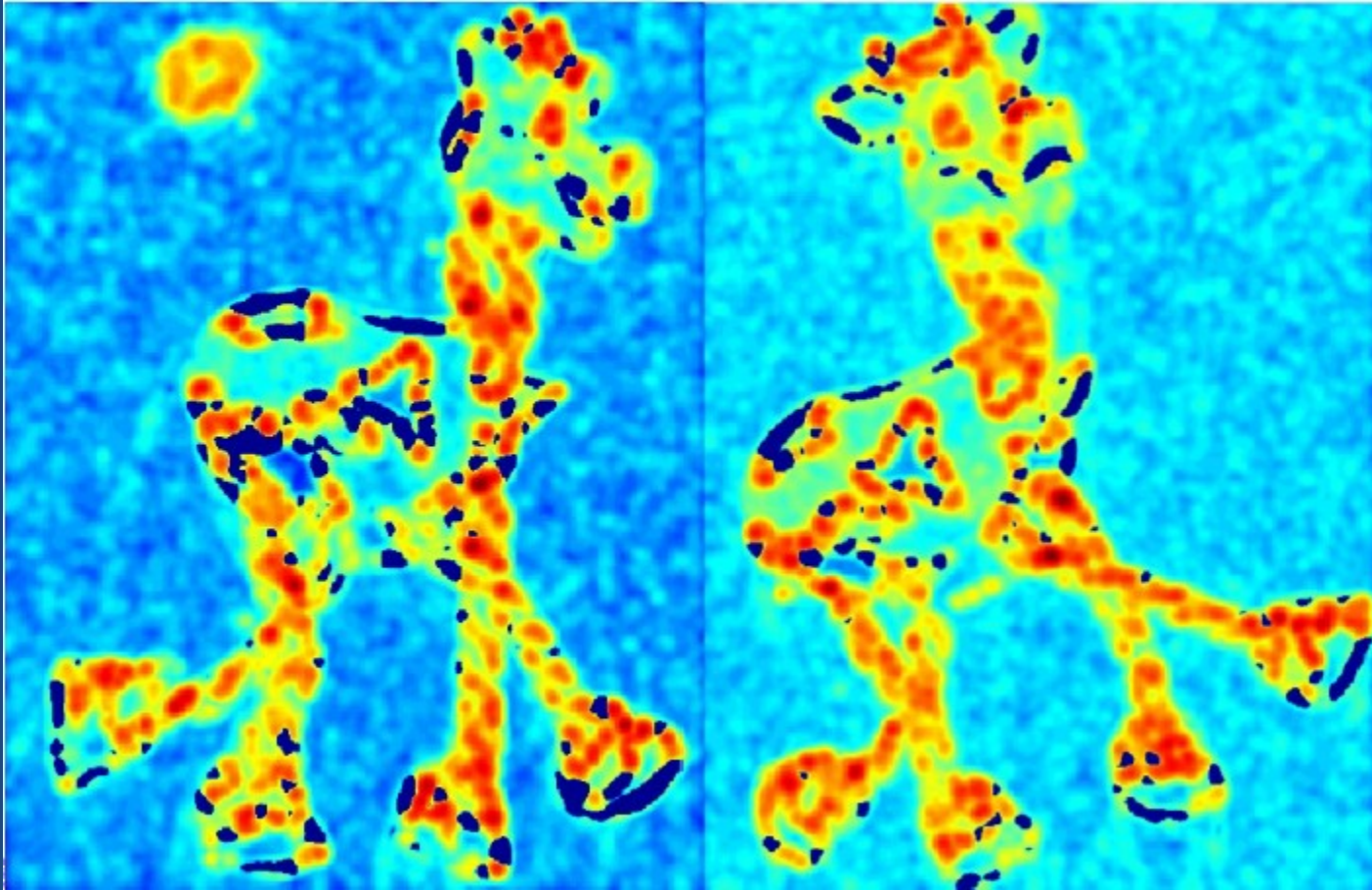


$$\begin{aligned} C(x,y) &= \det(\mathbf{M}) - k \cdot (\text{trace}(\mathbf{M}))^2 \\ &= \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 \end{aligned}$$

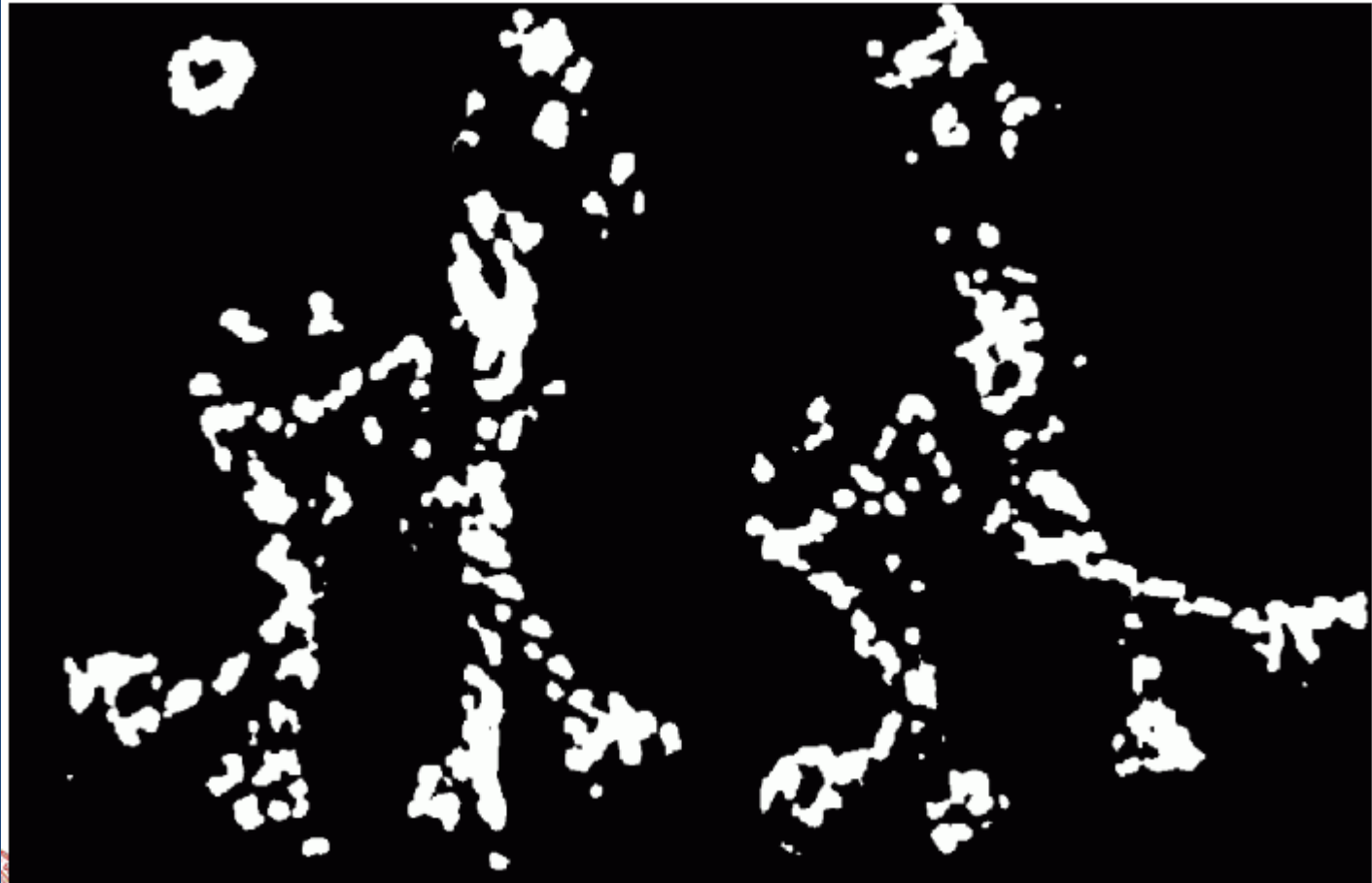
Anahtar nokta algılama:Girdi



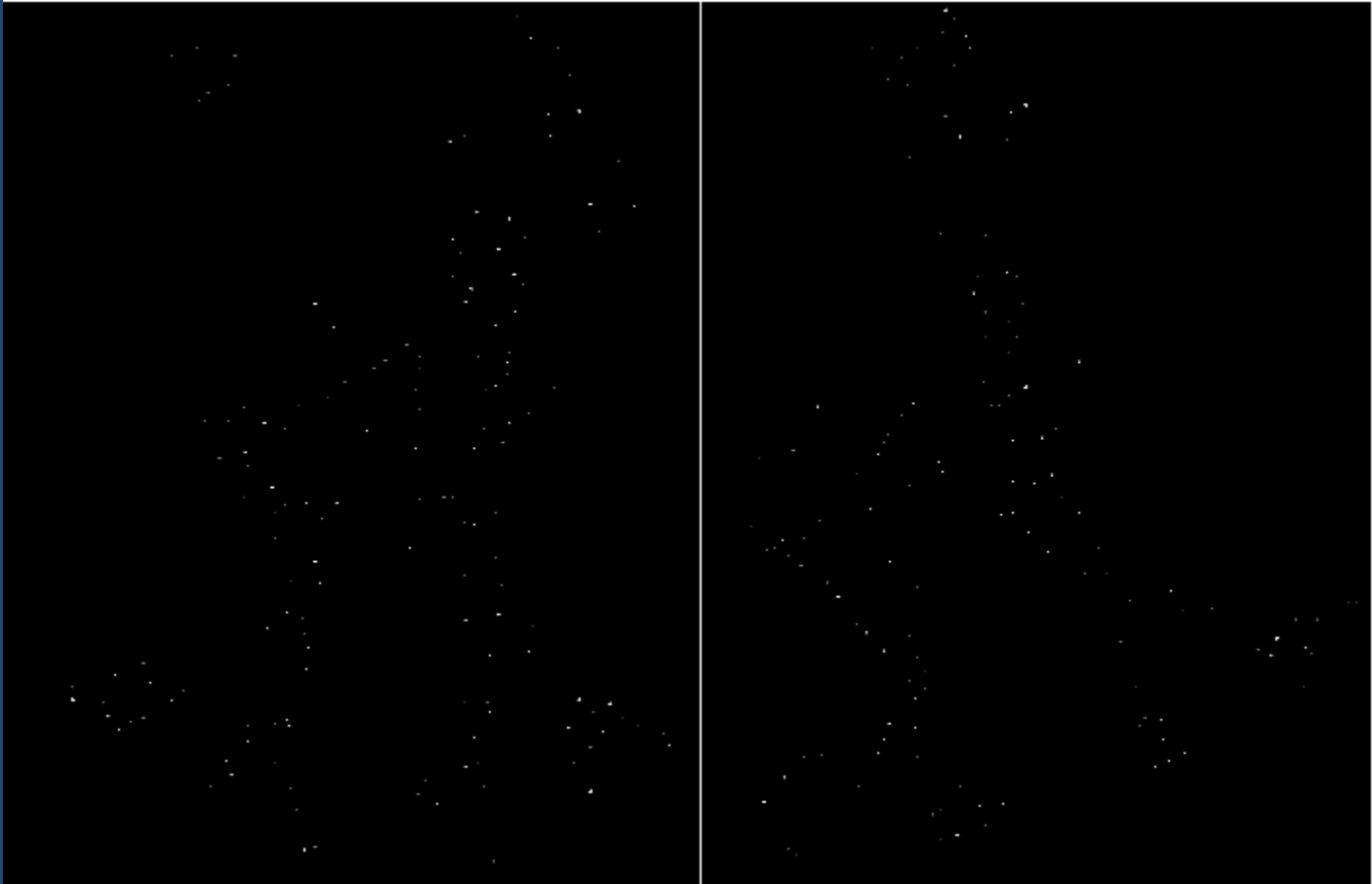
Harris köşeliği



Eşiklenen köşelik



Köşeliğin yerel maksimumu

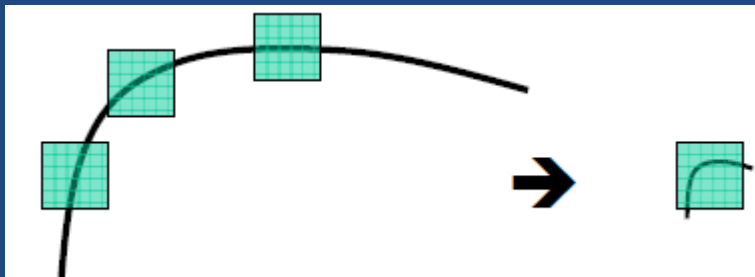
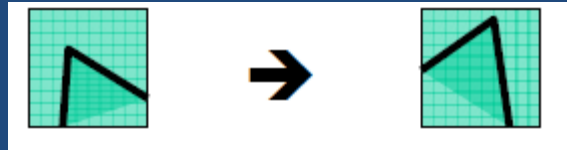


Birleřtirilmiř anahtar noktalar



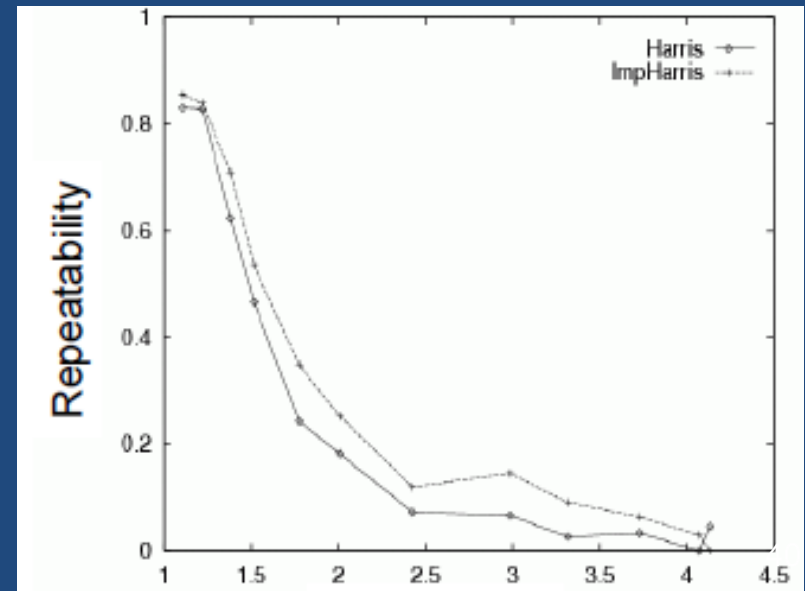
Harris köşe algılayıcısının dayanıklılığı

- Parlaklığa göre farklılık göstermez $f(x,y) \rightarrow f(x,y) + c$
- Yöne ve rotasyon 'a (dönü) göre farklılık göstermez
- Ölçeğe göre değişir:



Kenar

Köşe



Haralick köşe algılayıcı

Adım 1: İlgili algılama penceresi

- Sobel kullanıcısı kullanarak yatay gradyan $f_x(x,y)$ ve dikey gradyan $f_y(x,y)$ 'yi hesapla
- Her ölçüm penceresi için, aşağıdakilerin

determinantını al

$$\mathbf{M} = \begin{bmatrix} \sum_{(x,y) \in \text{window}} f_x^2(x,y) & \sum_{(x,y) \in \text{window}} f_x(x,y)f_y(x,y) \\ \sum_{(x,y) \in \text{window}} f_x(x,y)f_y(x,y) & \sum_{(x,y) \in \text{window}} f_y^2(x,y) \end{bmatrix}$$

Measure of "cornerness": $\det(\mathbf{M}) = \lambda_1 \lambda_2$

Circularity of ellipse: $q = 1 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2$

λ_1, λ_2 - eigenvalues of \mathbf{M}

-Eşikleme:

$$w = \begin{cases} \det(\mathbf{M}) & \text{if } \det(\mathbf{M}) > w_{\min} \text{ and } q > q_{\min} \\ 0 & \text{else} \end{cases}$$

-w üzerinde maksimum olmayan (non-maximum) baskı

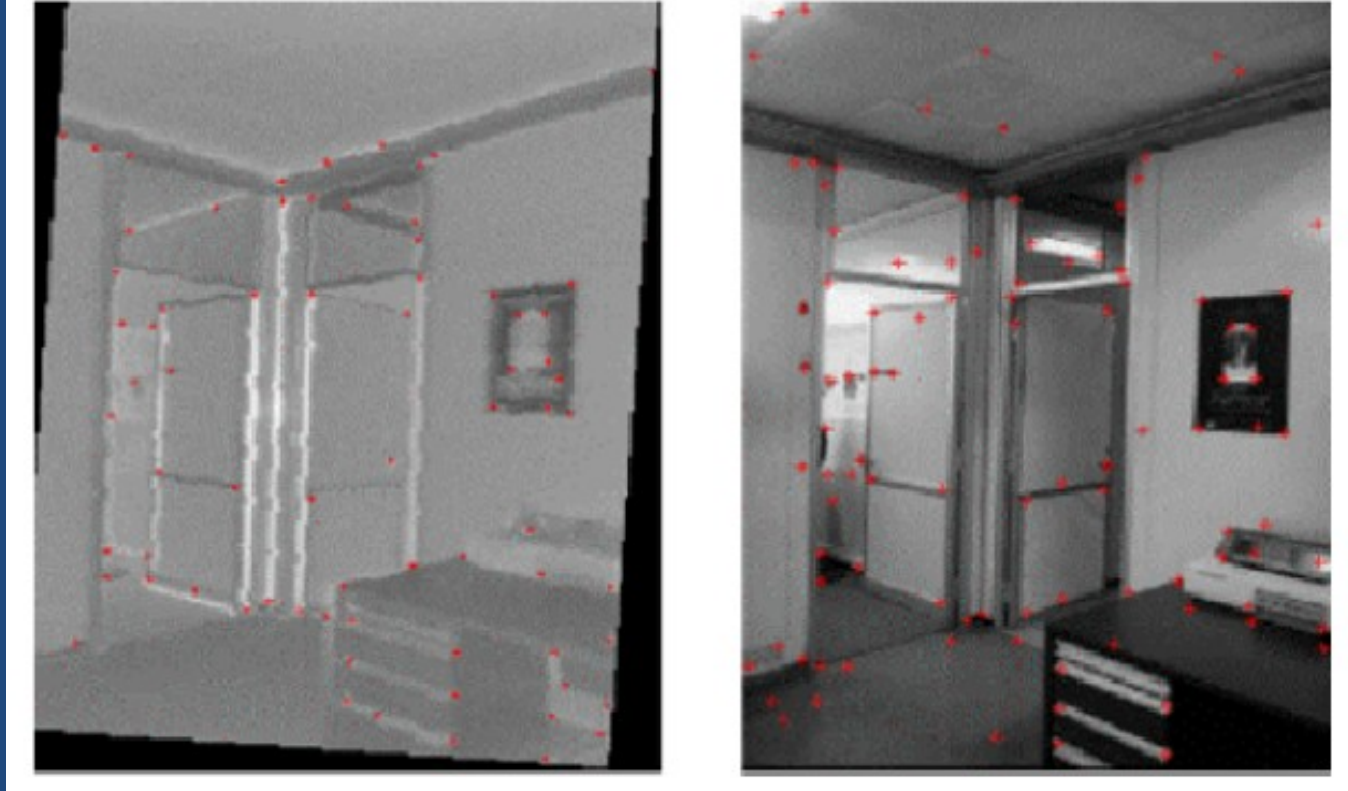
Haralick köşe algılayıcı

Adım 2:

İlgi penceresindeki ilginç noktanın çıkarımı: yatay ve dikey gradyanın ağırlık merkezinde olduğu gibi

$$x = \frac{\sum_{x,y \in \text{window}} (f_x(x,y) \times f_y(x,y) \times x)}{\sum_{x,y \in \text{window}} (f_x(x,y) \times f_y(x,y))}$$
$$y = \frac{\sum_{x,y \in \text{window}} (f_x(x,y) \times f_y(x,y) \times y)}{\sum_{x,y \in \text{window}} (f_x(x,y) \times f_y(x,y))}$$

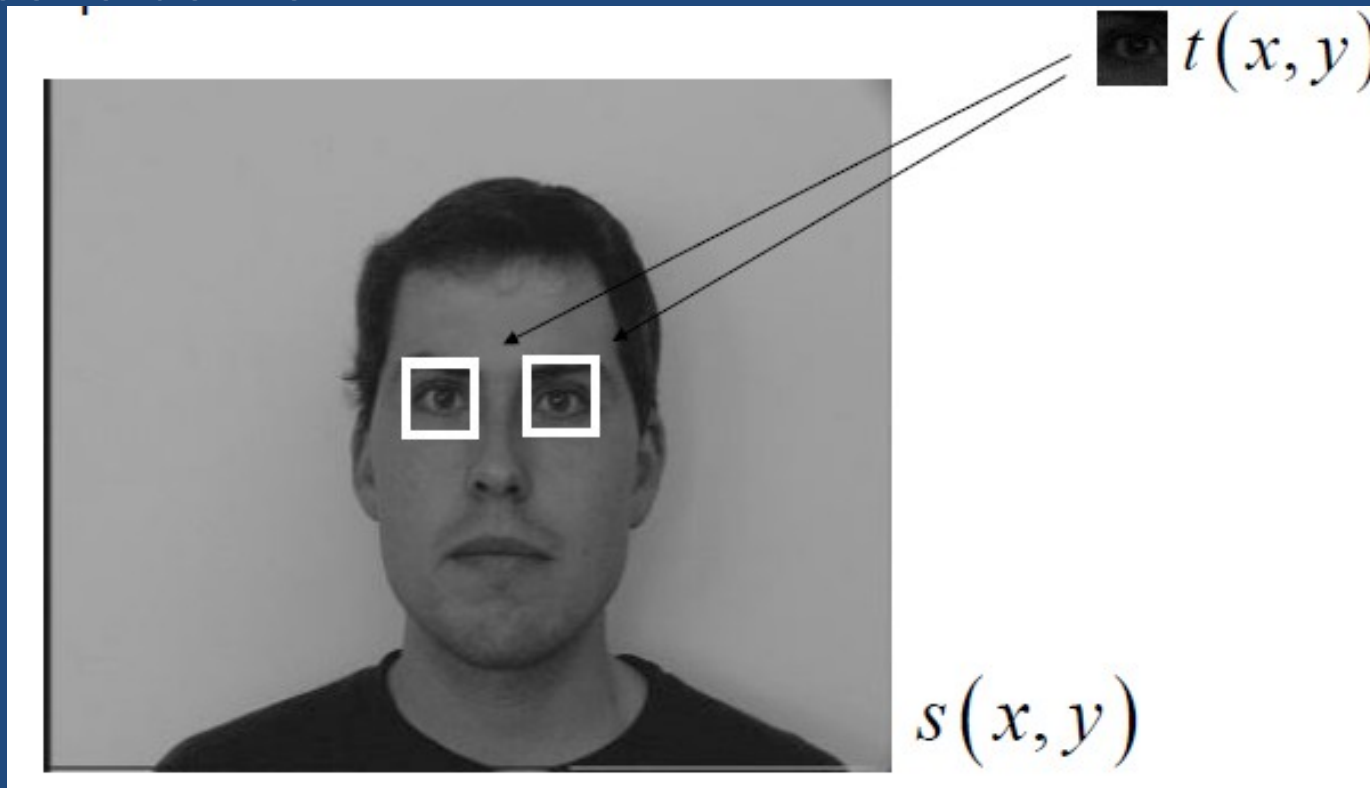
Haralick köşe algılayıcı



Köşe algılayıcı ,görüntü çakıştırma uygulayabilmek için lazer tarayıcı(solda) ve kamera(sağda) 'dan elde edilen bir yansıma görüntüsüne başvurur.

Şablon eşleştirme (Template matching)

Problem: $s(x,y)$ görüntü alanında $t(x,y)$ koordinatındaki bir şablonla ifade edilen bir nesne belirle



Şablon eşleştirme

-En iyi eşleştirme için kare ortalaması hatasını (mean-squared error) asgariye indirerek araştır

$$\begin{aligned} E(p, q) &= \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} [s(x, y) - t(x - p, y - q)]^2 \\ &= \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |s(x, y)|^2 + \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |t(x, y)|^2 - 2 \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} s(x, y) \cdot t(x - p, y - q) \end{aligned}$$

-Eşit biçimde, alan korelasyonunu maksimum yap

$$r(p, q) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} s(x, y) \cdot t(x - p, y - q) = s(p, q) * t(-p, -q)$$

-Alan korelasyonu, $s(x, y)$ görüntüsünün etki tepki ile konvolüsyonuna (evrişim) eşittir $t(-x, -y)$

Şablon eşleştirme

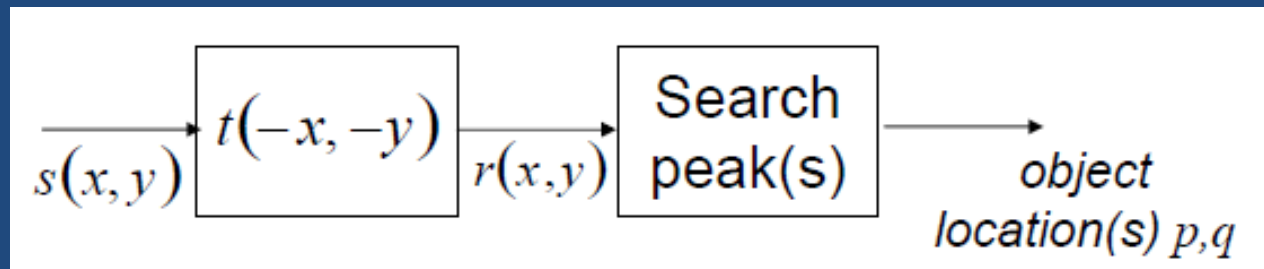
-Cauchy-schwarz eşitsizliğinden:

$$r(p, q) = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} s(x, y) \cdot t(x-p, y-q) \leq \sqrt{\left[\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |s(x, y)|^2 \right]} \cdot \sqrt{\left[\sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} |t(x, y)|^2 \right]}$$

-Eşitlik:

$$s(x, y) = \alpha \cdot t(x-p, y-q) \quad \text{with } \alpha \geq 0$$

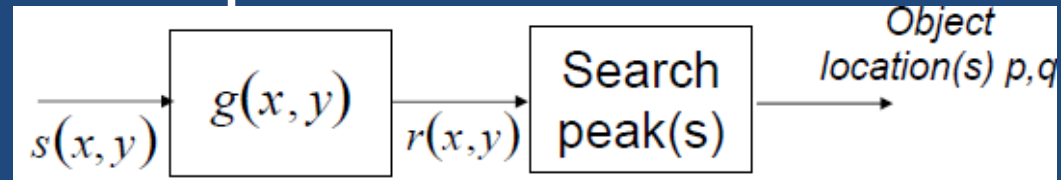
-Şablon eşleştirici blok diyagramı:



-Şablon eşleştirmeden önce parlak bölgelere sapmayı önlemek için ortalamayı at(sil)

Eşleştirilmiş filtreleme

-Sinyal algılama problemini hesaba kat



-Sinyal modeli

değiştirilen(shifted) şablon

gürültü

$$s(x, y) = t(x - p, y - q) + n(x, y)$$

- $g(x, y)$ filtresini maksimuma ayarla

$$SNR = \frac{|r(p, q)|^2}{E \left\{ |n(x, y) * g(x, y)|^2 \right\}}$$

correct peak

false readings

Doğru üst değer
(tepe noktası)

Yanlış
değerler

Eşleştirilmiş filtreleme

-Optimum filtre frekans tepkisine sahiptir

$$G(e^{j\omega_x}, e^{j\omega_y}) = \frac{T^*(e^{j\omega_x}, e^{j\omega_y})}{\Phi_{nn}(e^{j\omega_x}, e^{j\omega_y})}$$

-İspat:

$$SNR = \frac{|r(p, q)|^2}{E\{|n(x, y) * g(x, y)|^2\}} = \frac{\left| \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} G(e^{j\omega_x}, e^{j\omega_y}) T(e^{j\omega_x}, e^{j\omega_y}) d\omega_x d\omega_y \right|^2}{\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |G(e^{j\omega_x}, e^{j\omega_y})|^2 \Phi_{nn}(e^{j\omega_x}, e^{j\omega_y}) d\omega_x d\omega_y}$$

$$= \frac{\left| \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} [G \Phi_{nn}^{1/2}] [\Phi_{nn}^{-1/2} T] d\omega_x d\omega_y \right|^2}{\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |G|^2 \Phi_{nn} d\omega_x d\omega_y} \leq \frac{\left[\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |G|^2 \Phi_{nn} d\omega_x d\omega_y \right] \left[\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |T|^2 \Phi_{nn}^{-1} d\omega_x d\omega_y \right]}{\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |G|^2 \Phi_{nn} d\omega_x d\omega_y}$$

$$= \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |T|^2 \Phi_{nn}^{-1} d\omega_x d\omega_y$$

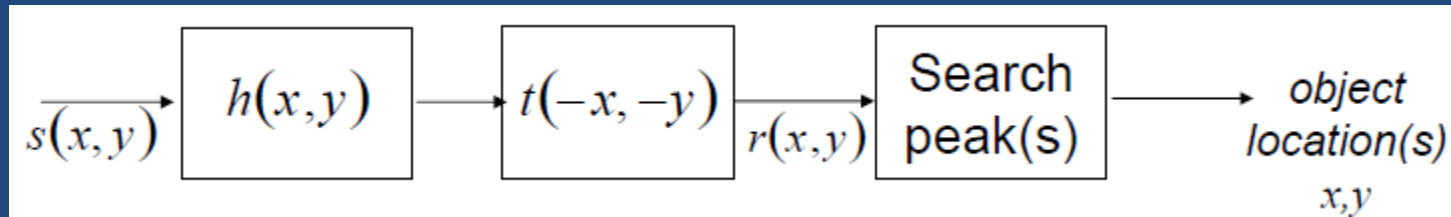
max. SNR

$$G \Phi_{nn}^{1/2} = \alpha \cdot [\Phi_{nn}^{-1/2} T]^*$$

eşitliğiyle **schwarz** eşitsizliği

Eşleştirilmiş filtreleme

-Optimum algılama: ön filtreleme & şablon eşleştirme



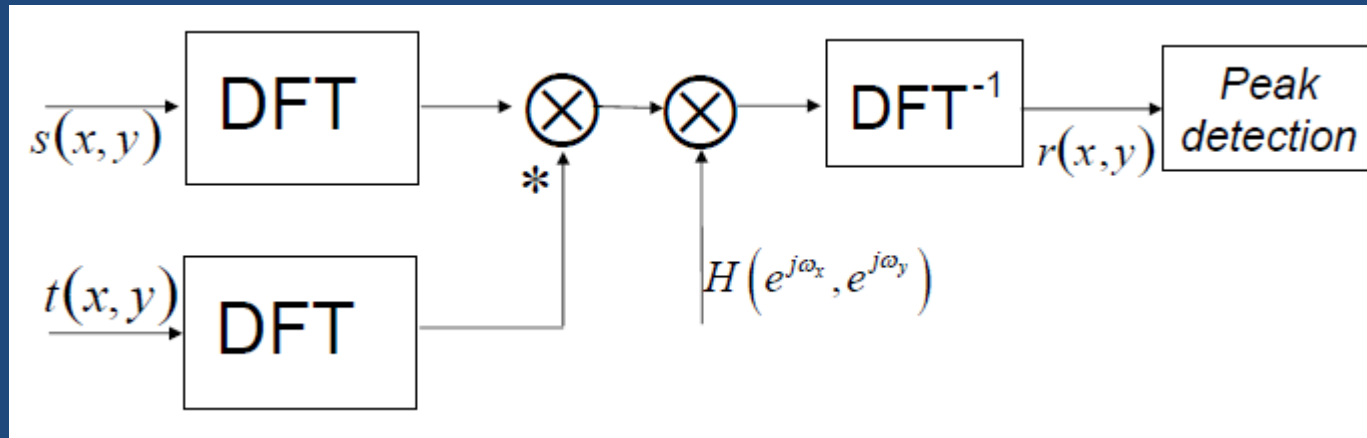
$$h(x, y) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \frac{1}{\Phi_m(e^{j\omega_x}, e^{j\omega_y})} e^{j\omega_x x + j\omega_y y} d\omega_x d\omega_y$$

-Beyaz renk istenmeyen görüntü(noise) $n(x, y)$ için $h(x, y)$ ön filtrelemesi istenmez

-Düşük frekans parazit: yüksek geçiren ön filtre(süzgeç)

Frekans etki alanı ilişkisi (korelasyonu)

-Ayırık Fourier dönüşümünü kullanan verimli uygulama



Tepe
noktası
algılama

-Faz korelasyonu

$$H(e^{j\omega_x}, e^{j\omega_y}) = \frac{1}{\left| S(e^{j\omega_x}, e^{j\omega_y}) \right| \left| T(e^{j\omega_x}, e^{j\omega_y}) \right|}$$