Jellemzőkinyerés és morfológiai műveletek

Éldetektáló operátorok

Az éldetektáló operárokat az `edge()` függvény valósítja meg. Parameterként meg kell adni a szürkeárnyalatos input képet, az éldetektáló operátor nevét, valamint meg lehet adni egy küszöb értéket. Az eredmény egy bináris élkép.

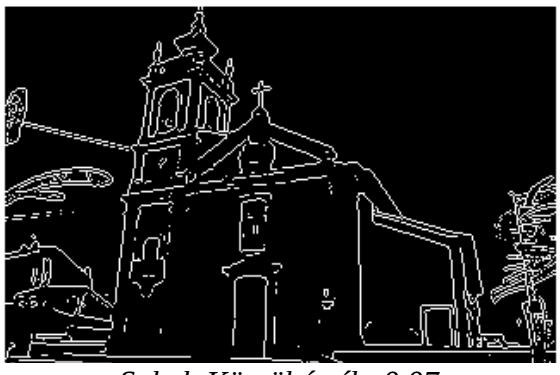
A gradiens operátorok gradiens magnitúdó képet eredményeznek.A Sobel, Roberts, Prewitt operátorkat az `edge()` függvény is alkalmazza. Ezeken a nevezetes gradiens operátorokon kívül a Matlabban meg van valósítva a Canny éldetektálás is.

```
>> templom = imread('templom.jpg');
>> templom_gray = rgb2gray(templom);
>> templom_S = edge(templom_gray, 'sobel', 0.02);
>> imshow(templom_S);
>> templom_S = edge(templom_gray, 'sobel', 0.03);
>> imshow(templom_S);
>> templom_S = edge(templom_gray, 'sobel', 0.07);
>> imshow(templom_S);
```

szürkeárnyalatos kép

Sobel, Küszöbérték: 0.02

Sobel, Küszöbérték: 0.03



Sobel, Küszöbérték: 0.07

szürkeárnyalatos kép

Canny, Küszöbérték: 0.02

Canny, Küszöbérték: 0.03

Canny, Küszöbérték: 0.07

```
>> templom_Canny = edge(templom_gray, 'canny', 0.02);
>> imshow(templom_Canny);
>> templom_Canny = edge(templom_gray, 'canny', 0.03);
>> imshow(templom_Canny);
>> templom_Canny = edge(templom_gray, 'canny', 0.07);
>> imshow(templom_Canny);
```

Sarokpont detektálás

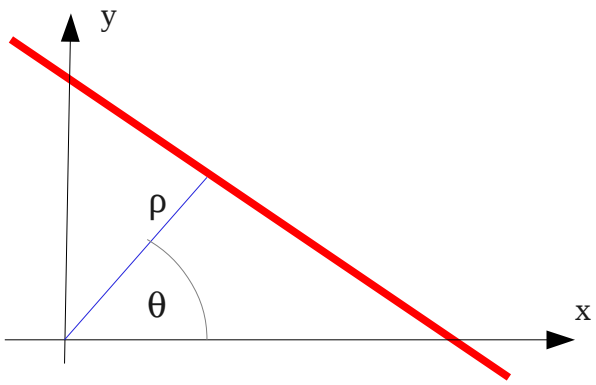
Matlab-ban két sarokpont detektáló algoritmus van implementálva, a Harris és Shi-Tomasi algoritmus. A `corner()` függvényt a `'MinimumEigenvalue'` paraméterrel meghívva tudjuk alkalmazni
Shi és Tomasi módszerét. Az eljárás nevén túl a függvény paramétereül meg kell adni az input képet, valamint a pontok maximális darabszámát. Ez utóbbira azért lehet szükség, mert a függvény egy *M*×2-es mátrixszal tér vissza, ahol *M* a visszaadott pontok száma.

```
>> templom = imread('templom.jpg');
>> templom_gray = rgb2gray(templom);
>> templom_HarrisPoints = corner(templom_gray, 'Harris', 50);
>> templom_Harris = templom;
>> templom_Harris(:,:,1)= templom_gray(:,:);
>> templom_Harris(:,:,2)= templom_gray(:,:);
>> templom_Harris(:,:,3)= templom_gray(:,:);
```

```
>> for i=1:50;
     x = templom_HarrisPoints(i,1);
     y = templom_HarrisPoints(i,2);
     templom_Harris(y-2:y+2,x-2:x+2 , 1) = 255;
     templom_Harris(y-2:y+2,x-2:x+2 , 2) = 0;
     templom_Harris(y-2:y+2,x-2:x+2 , 3) = 0;
end
>> imshow(templom_Harris);
```

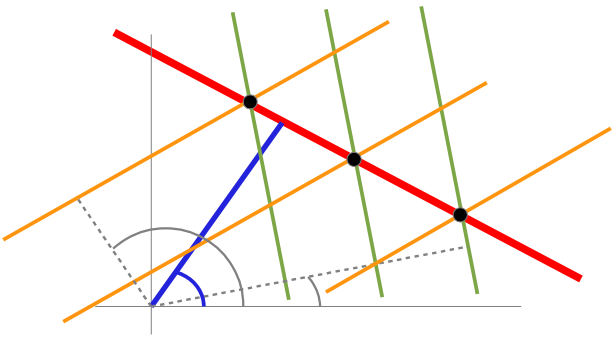
Egyenesek detektálása Hough transzformációval

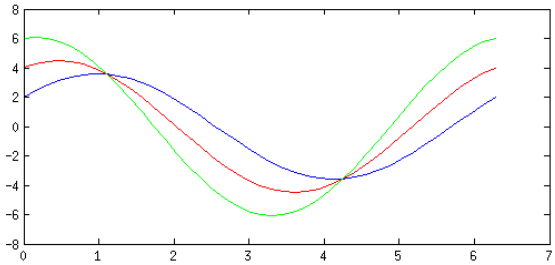
Egy egyenes megadható egy, az origóból az egyenesre bocsátott merőleges szakasz hosszával (ρ) és irányszögével (θ). A kép egy adott (u,v) pontján végtelen sok egyenes halad át és minden, egyenesre bocsátot θ szögű merőleges szakaszhoz más-más ρ hosszúság tartozik. A képlet alapján a paramétertérben így minden pont egy szinuszoiddal írható le.



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

Ha egy $\{(u_1, v_1), (u_2,v_2), ..., (u_n,v_n)\}$ ponthalmaz egy egyenesre esik, akkor ez az egyenes a ponthalmaz minden eleme esetén ugyanazzal a (θ,ρ) értékpárral rendelkezik. Ez azt jelenti, hogy ha ezen pontokat a Hough-térben egy-egy szinuszoiddal reprezentáljuk, akkor ezek a szinuszoidok pont abban a (θ,ρ) pontban metszik egymást, amelyekkel a pontokon áthaladó egyenes megadható.





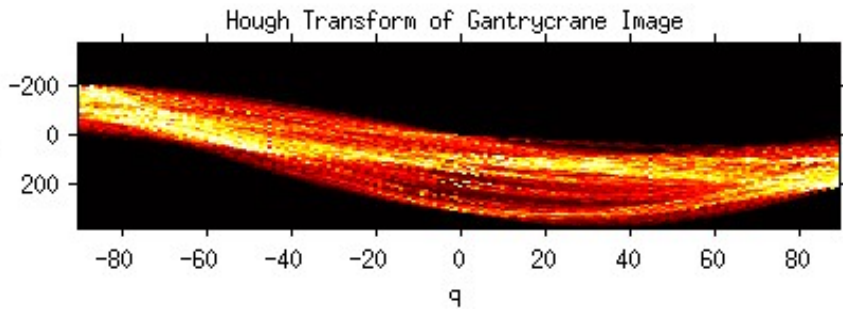
A három pontot három szinuszoid reprezentálja Hough térben

A Matlabban az egyenesek Hough transzformációval történő detektálása a `hough()` függvény használandó. A `hough()` függvény mellett vannak olyan segédfüggvények, amelyek a detektált egyenesek számát valamint azok kirajzolását hivatott segíteni. Nézzük meg a *`hough_pelda.m`* fájlban lévő példát!

A `hough()` függvény egy bináris élképet vár paraméterül, majd visszaadja a Hough-mátrixot valamint az egyenesekhez tartozó a *theta* szögeket és az egyenesek origótól mért *rho* távolságértékeit.
A `houghpeaks()` függvény paramétereként kiválaszhatjuk, hogy hány egyenest szeretnénk meghatározni, ez alapján a függvény meghatározza az N legerősebb metszéspontot a Hough mátrixban. A `houghlines()` függvény pedig visszaad egy olyan mátrixot, amely tartalmazza az N legvalószínűbb egyenest a képen. A példában ezeket az egyeneseket rávetítjük a képre.

```
templom = imread('templom.jpg');
templom_gray = rgb2gray(templom);
templom_Canny = edge(templom_gray, 'canny');

[HM, theta, rho] = hough(templom_Canny);
peaks = houghpeaks(HM, 10);
lines = houghlines(templom_Canny, theta, rho, peaks);

%Display the original image.
subplot(2,1,1);
imshow(templom);
title('Templom');



% Display the Hough matrix.
subplot(2,1,2);
imshow(imadjust(mat2gray(HM)),'XData',theta,'YData',rho,'InitialMag
nification',
'fit');

title('Parameter Space');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;
colormap(hot);
```

```
figure, imshow(templom_gray); hold on;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',3,'Color','green');

   % Plot beginnings and ends of lines
   plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

   max_len = 0;
   % Determine the endpoints of the longest line segment
   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
      max_len = len;
      xy_long = xy;
   end
end

% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','red');
```



Bináris morfológiai műveletek

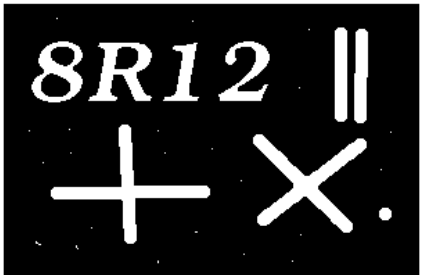
Az elemi morfológiai műveletek közé tartozik az erózió, a dilatáció, nyitás, zárás. A morfológiai operátorkat nagyon gyakran alkalmazzák jellemzőkinyerésre, mint például a határ kinyerése, vázszerű jellemzők kinyerése.

A Matlab-ban külön függvény van bináris morfolóiai műveletekre és szürkeárnyalatos morfológiai műveletekre. A bináris morfológiai műveleteket a `bwmorph()` függvénnyel végezhetjük el. Ezeknek a műveleteknek a szerkesztőeleme általában egy 3×3-as, 1-esekből álló mátrix. A függvény parametereként megadhatjuk, hogy hányszor fusson le a művelet.

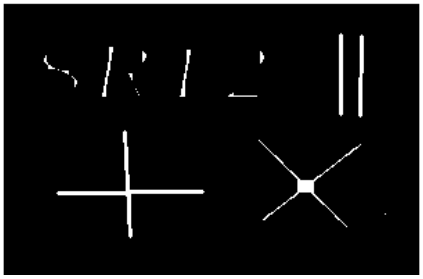
Példa erózióra:

```
>> img = imread('binimage.png');
>> img = img(:,:,1);
>> img = img ./255;   % osszunk le minden értéket 255-tel
>> img_erode = bwmorph(img, 'erode');
>> imshow(img_erode);
>> img_erode = bwmorph(img, 'erode', 4);
>> imshow(img_erode);
```

eredeti kép



erodált kép (1 iteráció után)



erodált kép (4 iteráció után)

Példa dilatációra:

```
>> img_dilate = bwmorph(img, 'dilate');
>> imshow(img_dilate);
>> img_dilate = bwmorph(img, 'dilate', 4);
>> imshow(img_dilate);
```

eredeti kép

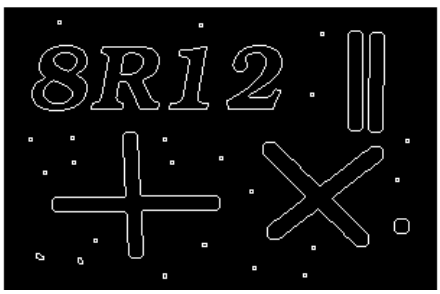
dilatált kép (1 iteráció után)

dilatált kép (4 iteráció után)

Példa határvonal detektálásra:

```
>> img_border = bwmorph(img, 'remove');
>> imshow(img_border);
```

eredeti kép

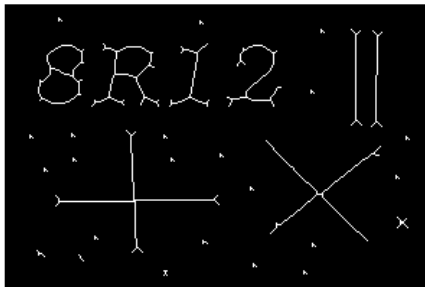


Az input kép 4-összefüggő határa

Példa vázszerű jellemző kinyerésére:

```
>> img_skel = bwmorph(img, 'skel', Inf);
>> imshow(img_skel);
>> img_thin = bwmorph(img, 'thin', Inf);
>> imshow(img_thin);
```

eredeti kép



img_skel

img_thin