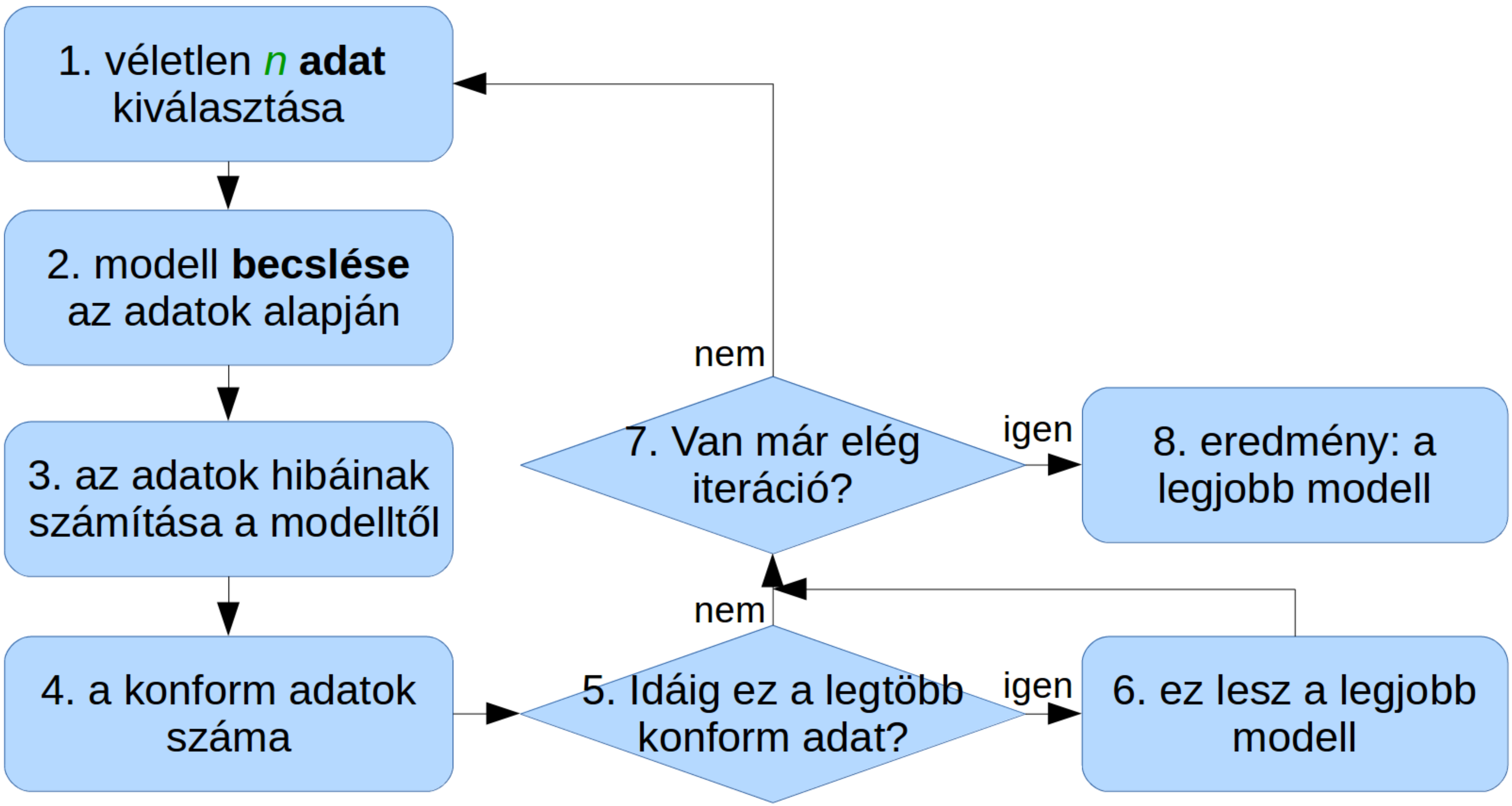
6. gyakorlat

Kiegyenlítő számítások MSc 2018/19

Áttekintés

- Számítógépes programok RANSAC becslésekre
  - egyenes illesztése
  - ellipszis illesztése
  - több modell illesztése
  - gömb és henger illesztése (Python)

A RANSAC folyamatábrája



Adatok és programok

- https://github.com/gyulat/kiegyenlito_szamitasok

Gyula Toth 4 lépcső egyenes illesztés		Latest commit 6d58c8c 18 hours ago
README.md	Create README.md	10 months ago
circle5.dat	sorozatos RANSAC kör illesztés	22 hours ago
ecdf.py	Tapasztalati eloszlásfüggvények (Python szkript)	3 months ago
ellipdata.txt	RANSAC illesztés demo	a day ago
ellipsedemo.m	Minor changes	a day ago
kstest2.m	Kétmintás Kolmogorov-próba	3 months ago
kstest2.py	Kétmintás Kolmogorov-próba (Python szkript)	3 months ago
linedata.txt	RANSAC illesztés demo	a day ago
mfv.m	Leggyakoribb érték számítása (Octave script)	3 months ago
ransacdemo_minimal.m	Minimális egyenes illesztés demo	a day ago
seq_ransac_circle.m	sorozatos RANSAC kör illesztés	22 hours ago
seq_ransac_line.m	sorozatos RANSAC egyenes illesztés demo	a day ago
seq_ransac_stairs.m	4 lépcső egyenes illesztés	18 hours ago
stairs4.dat	4 lépcső egyenes illesztés	18 hours ago
star5.dat	sorozatos RANSAC egyenes illesztés demo	a day ago

ZIP fájlban letölthető

Egyenes illesztés

adatok: `linedata.txt`

modell:

$$ax + by - 1 = 0$$

Egyenes illesztés – Octave (ransacdemo_minimal.m)

```
% RANSAC egyenes illesztés minimális demo
clear all
% minta adatok
d = load('linedata.txt');
x = d(:,1);  y = d(:,2); nd = length(x);
tol = 0.05;  % hiba küszöb
k = 16;  % iterációk száma
nmax = 0; % nincs még konszenzus halmaz
for i=1:k
  % két pont véletlenszerű kiválasztása
  is = randperm(nd,2);
  % ax+by-1=0 egyenes paramétereinek meghatározása
  A = d(is,:); b = [1; 1]; p = A\b;
  % az adatok egyenestől mért távolságai
  t = abs(p(1)*x+p(2)*y-1)/sqrt(p(1)^2+p(2)^2);
  xk = x(t<tol);  yk = y(t<tol); % illeszkedő adatok
  nin = length(xk);  % konszenzus halmaz elemszáma
  if nin > nmax  % jobb mint idáig
    xin = xk; yin = yk; nmax = nin;
    bp = p;  % legjobb egyenes
  end
end
printf("maximális konszenzus halmaz elemszáma: %d\n",nmax)
pls = polyfit(xin,yin,1); % LKN egyenes illesztés
printf("LKN egyenes egyenlete: y = %.3f*x %+.3f\n",pls(1),pls(2))
```

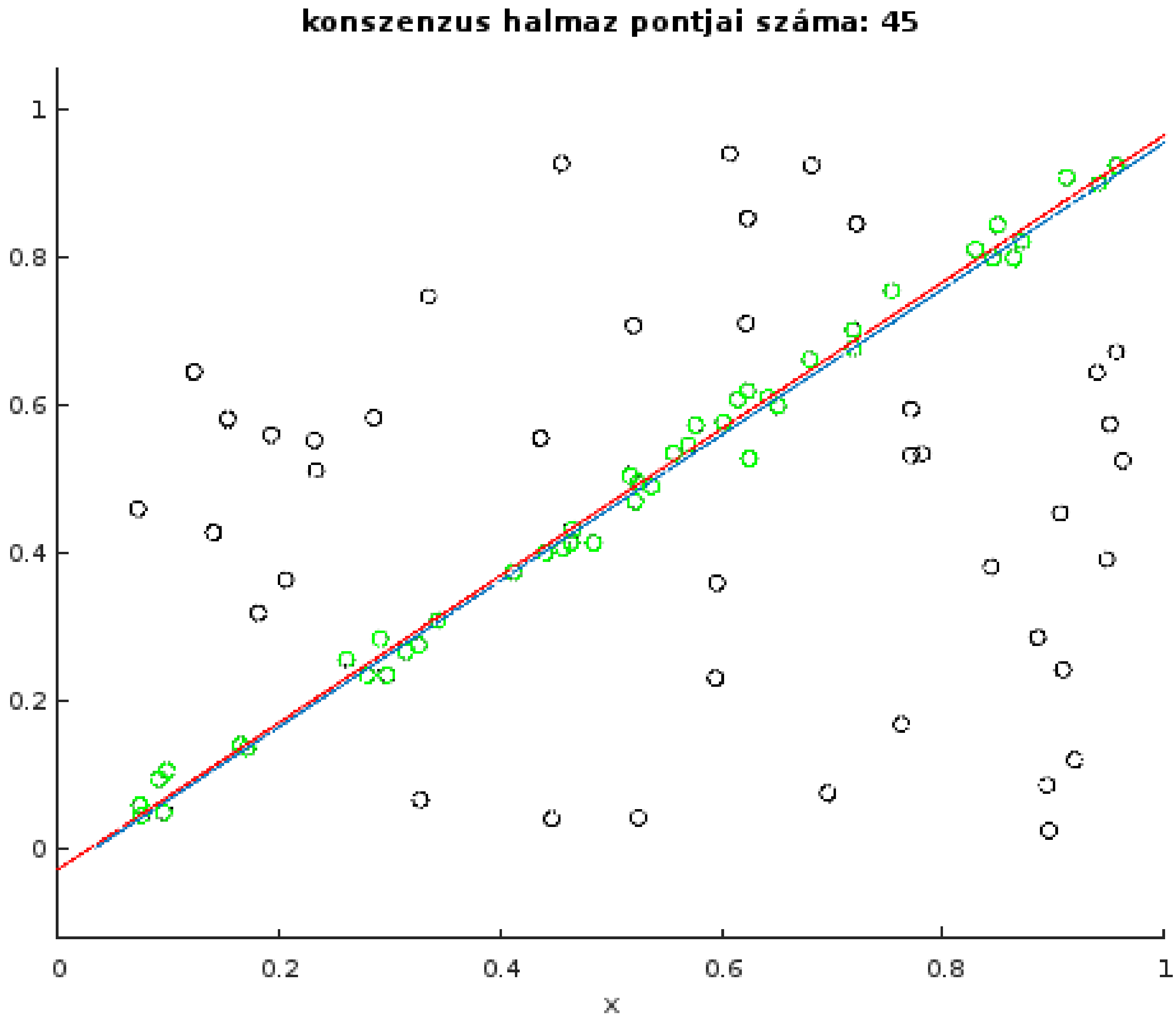
86 adat 50%-a durva hibás

Egyenes illesztés - Octave

```
>> run ransacdemo_minimal
maximális konszenzus halmaz elemszáma: 42
LKN egyenes egyenlete: y = 0.982*x -0.026
>> run ransacdemo_minimal
maximális konszenzus halmaz elemszáma: 45
LKN egyenes egyenlete: y = 0.994*x -0.030
>> run ransacdemo_minimal
maximális konszenzus halmaz elemszáma: 44
LKN egyenes egyenlete: y = 1.000*x -0.033
>> run ransacdemo_minimal
maximális konszenzus halmaz elemszáma: 38
LKN egyenes egyenlete: y = 1.027*x -0.052
>> run ransacdemo_minimal
maximális konszenzus halmaz elemszáma: 45
LKN egyenes egyenlete: y = 0.994*x -0.030
>> run ransacdemo_minimal
maximális konszenzus halmaz elemszáma: 41
LKN egyenes egyenlete: y = 1.014*x -0.039
>> run ransacdemo_minimal
maximális konszenzus halmaz elemszáma: 45
LKN egyenes egyenlete: y = 0.994*x -0.030
```

nem determinisztikus az algoritmus

Egyenes illesztés - Octave

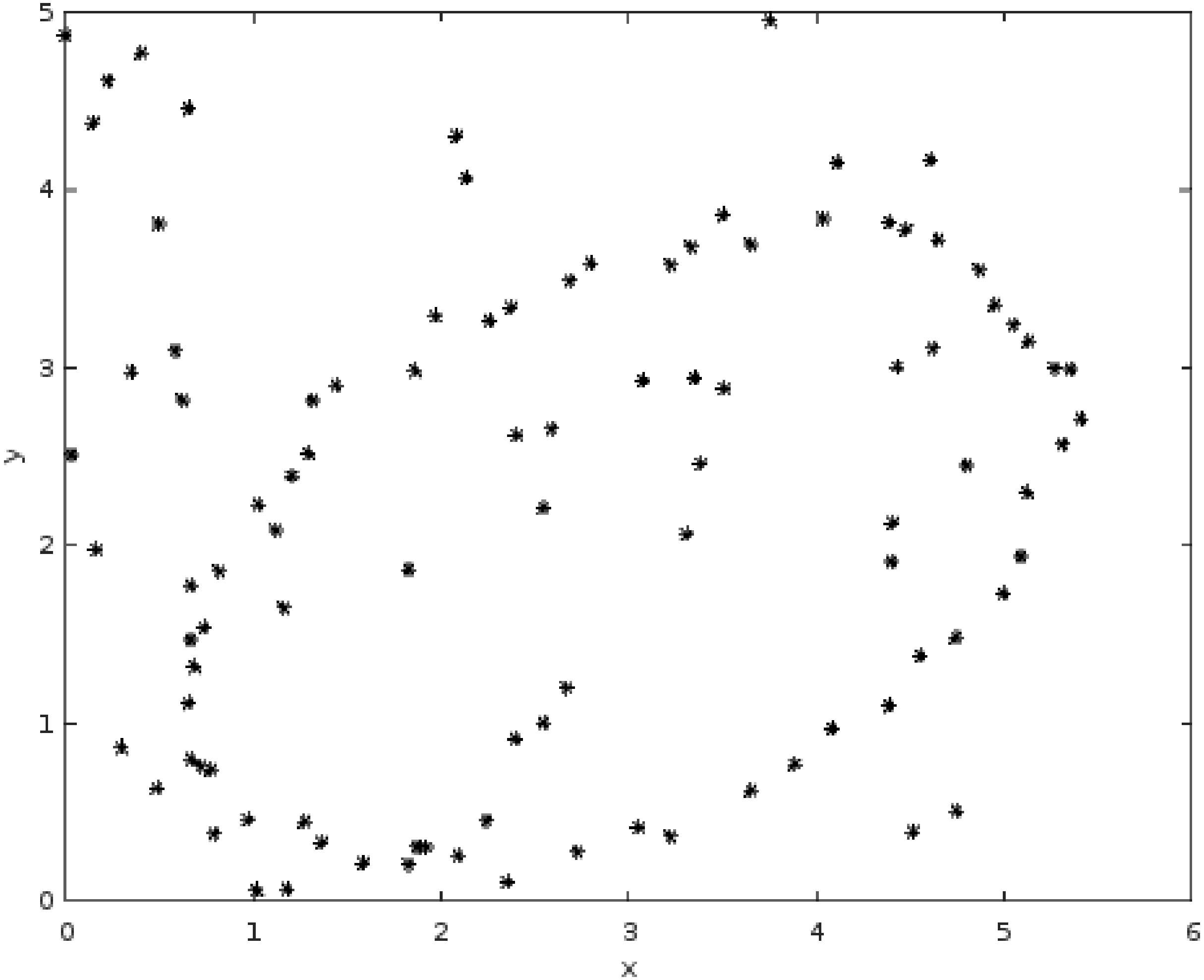


Ellipszis illesztés (Octave megoldás: ellipsedemo.m)

- minta adatok betöltése, rajz készítése
- paraméterek megadása

```
% RANSAC ellipszis illesztés demo
clear all
% minta adatok
d = load('ellipdata.txt'); % teljes ellipszis
x = d(:,1);  y = d(:,2); nd = length(x);
tol = 0.25;  % hiba küszöb
k = 300;  % iterációk száma
nmax = 0; % nincs még konszenzus halmaz
figure(1); hold on
plot(x,y,'ko')
```

Ellipszis illesztés



adatok: `ellipdata.txt`

modell: $x^2 + Bxy + Cy^2 + Dx + Ey + F = 0$

ellipszis, ha: $B^2 - 4C < 0$

Ellipszis illesztés (Octave megoldás: ellipsedemo.m)

- 5 véletlen pont kiválasztása
- kúpszelet meghatározása: ellipszis lett-e?

```
for i=1:k
  while 1
    % öt pont véletlenszerű kiválasztása
    is = randperm(nd,5);
    % x^2+Bxy+Cy^2+Dx+Ey+F=0 ellipszis paramétereinek meghatározása
    A = [x(is).*y(is), y(is).^2, x(is), y(is), ones(5,1)];
    b = -x(is).^2; p = A\b;
    % ellipszis, ha B^2-4*C < 0
    if (p(1)^2-4*p(2))<0
      break
    end
  end
  ell = @(x,y) x.^2+p(1)*x.*y+p(2)*y.^2+p(3)*x+p(4)*y+p(5);
```

Ellipszis illesztés (Octave megoldás: ellipsedemo.m)

- adatok távolsága gradiens norma szerint
- konszenzus halmaz meghatározása; az új jobb-e?

```
  % gradiens norma
  gnorm = @(x,y) sqrt((2*x+p(1)*y+p(3)).^2+(p(1)*x+2*p(2)*y+p(4)).^2);
  % az adatok ellipszistől mért távolságai gradienssel súlyozva
  t = abs(ell(x,y))./gnorm(x,y);
  xk = x(t<tol);  yk = y(t<tol); % illeszkedő adatok
  nin = length(xk);  % konszenzus halmaz elemszáma
  if nin > nmax  % jobb mint idáig
    xin = xk; yin = yk; nmax = nin;
    bp = p;  % legjobb ellipszis
  end
end
```

$$\|\nabla t_a(x_i, y_i)\| = \sqrt{(2x_i + By_i + D)^2 + (Bx_i + 2Cy_i + E)^2}$$

Ellipszis illesztés

- Fitzgibbon et al. (1996) eljárása

Direct Least Squares Fitting of Ellipses

Andrew W. Fitzgibbon **Maurizio Pilu**
Robert B. Fisher

Department of Artificial Intelligence
The University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL
SCOTLAND
email: {andrewfg,maurizp,rbf}@aifh.ed.ac.uk

January 4, 1996

```
% x,y are lists of coordinates
function a = fit_ellipse(x,y)
% Build design matrix
D = [ x.*x x.*y y.*y x y ones(size(x)) ];
% Build scatter matrix
S = D'*D;
% Build 6x6 constraint matrix
C(6,6) = 0; C(1,3) = 2; C(2,2) = -1; C(3,1) = 2;
% Solve eigensystem
[gevec, geval] = eig(inv(S)*C);
% Find the positive eigenvalue
[PosR, PosC] = find(geval > 0 & ~isinf(geval));
% Extract eigenvector corresponding to positive eigenvalue
a = gevec(:,PosC);
```

Figure 7: Complete 6-line Matlab implementation of the proposed algorithm.

Ellipszis illesztés (Octave megoldás: ellipsedemo.m)

- ellipszis, konszenzus halmaz ábrázolása
- LKN ellipszis illesztés Fitzgibbon et al. (1996) eljárásával

```
printf("maximális konszenzus halmaz elemszáma: %d\n",nmax)
plot(xin,yin,'go')
ell = @(x,y) x.^2+bp(1)*x.*y+bp(2)*y.^2+bp(3)*x+bp(4)*y+bp(5);
ezplot(ell,[0,6,0,6])
% LKN ellipszis illesztés (Fitzgibbon et al. 1996)
D = [xin.*xin, xin.*yin, yin.*yin, xin, yin, ones(size(xin))];
S = D'*D;
C(6,6)=0; C(1,3)=2; C(2,2)=-1; C(3,1)=2;
[gevec, geval] = eig(inv(S)*C);
[posR, posC] = find(geval>0 & ~isinf(geval));
pls = gevec(:,posC);
a = (pls./pls(1))(2:end);
ells = @(x,y) x.^2+a(1)*x.*y+a(2)*y.^2+a(3)*x+a(4)*y+a(5);
h = ezplot(ells,[0,6,0,6]);
set(h,"Color","red");
title(["konszenzus halmaz elemszáma: ",num2str(nmax)])
legend("adatok","konform","ellipszis","LKN illesztés")
```

Geometriai paraméterek számítása

- Van Loan által közölt képletek

Using the Ellipse to Fit and Enclose Data Points

A First Look at Scientific Computing and Numerical Optimization

Charles F. Van Loan
Department of Computer Science
Cornell University

Paraméterek meghatározása

- Transzformáció paraméteres alakba

$$M_0 = \begin{bmatrix} F & D/2 & E/2 \\ D/2 & A & B/2 \\ E/2 & B/2 & C \end{bmatrix} \qquad M = \begin{bmatrix} A & B/2 \\ B/2 & C \end{bmatrix}$$

$$a = \sqrt{-\det(M_0)/(\det(M)\lambda_1)} \qquad b = \sqrt{-\det(M_0)/(\det(M)\lambda_2)}$$

$$h = (BE - 2CD)/(4AC - B^2) \qquad k = (BD - 2AE)/(4AC - B^2)$$

$$\tau = \operatorname{arccot}((A - C)/B)/2$$

λ_1, λ_2 az M mátrix rendezett sajátértékei: $|\lambda_1 - A| \le |\lambda_1 - C|$

Ellipszis illesztés (Octave megoldás: ellipsedemo.m)

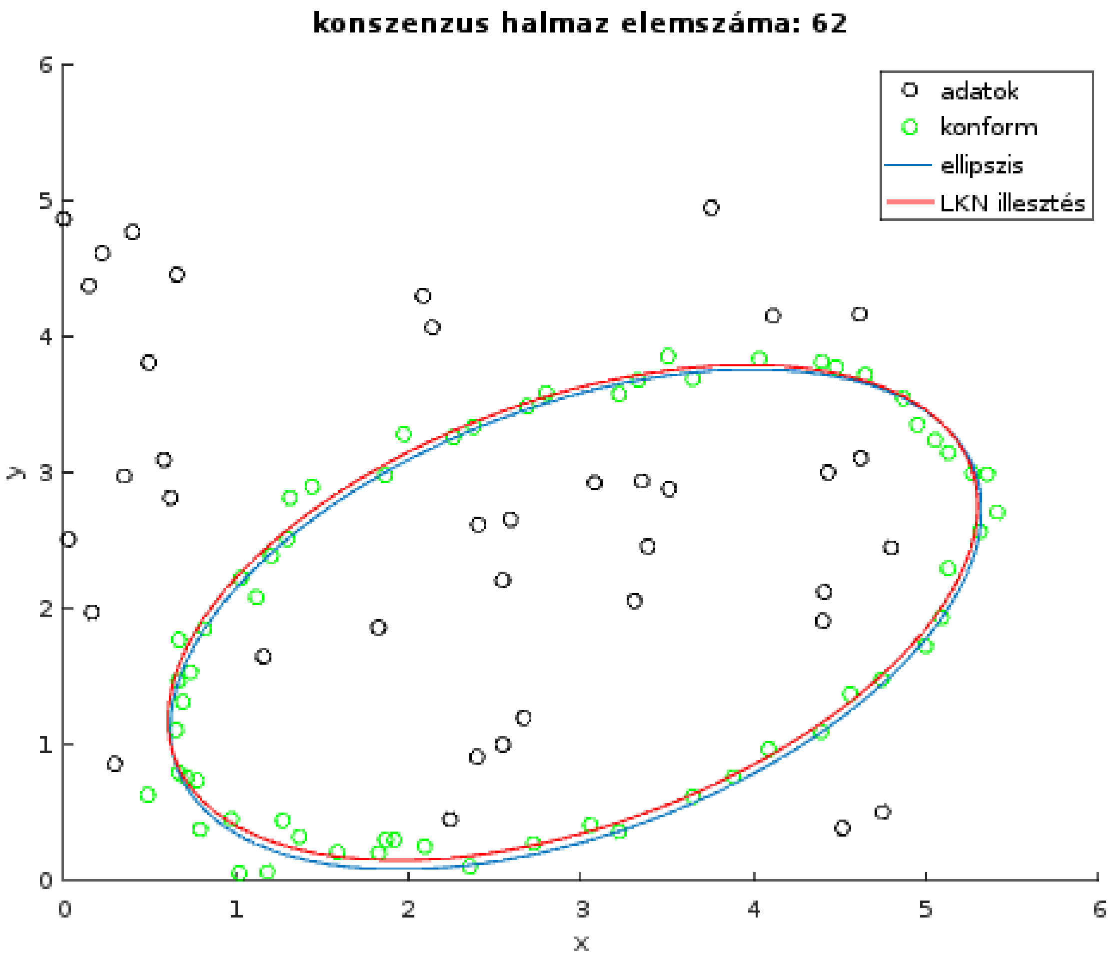
- ellipszis geometriai paramétereinek a számítása
- RANSAC nélküli LKN ellipszis illesztés

```
% ellipszis középpontja a=(A B C D E F), Rosin(1999)
a = pls;
xc = (a(2)*a(5)-2*a(3)*a(4))/(4*a(1)*a(3)-a(2)^2);
yc = (a(2)*a(4)-2*a(1)*a(5))/(4*a(1)*a(3)-a(2)^2);
% fél nagy és kistengely
a1 = sqrt(-2*(a(6)-(a(3)*a(4)^2-a(2)*a(4)*a(5)+a(1)*a(5)^2)/(4*a(1)*a(3)-a(2)^2)) ...
           /(a(1)+a(3) - sqrt(a(2)^2+(a(1)-a(3))^2)) );
a2 = sqrt(-2*(a(6)-(a(3)*a(4)^2-a(2)*a(4)*a(5)+a(1)*a(5)^2)/(4*a(1)*a(3)-a(2)^2)) ...
           /(a(1)+a(3) + sqrt(a(2)^2+(a(1)-a(3))^2)) );
% elfordulás szöge
theta = 0.5*atan(a(2)/(a(1)-a(3)));
printf("Az ellipszis adatai: \n");
printf("xc: %.3f  yc: %.3f\n",xc,yc);  % eredeti: (3,2)
printf("a1: %.3f  a2: %.3f\n",a1,a2);  % eredeti: 1.5, 2.5
printf("theta: %.1f\n fok", theta*180/pi); % eredeti: 30
```

```
% RANSAC nélküli LKN illesztés
% LKN ellipszis illesztés (Fitzgibbon et al. 1996)
D = [x.*x, x.*y, y.*y, x, y, ones(size(x))];
```

Illesztés eredményei

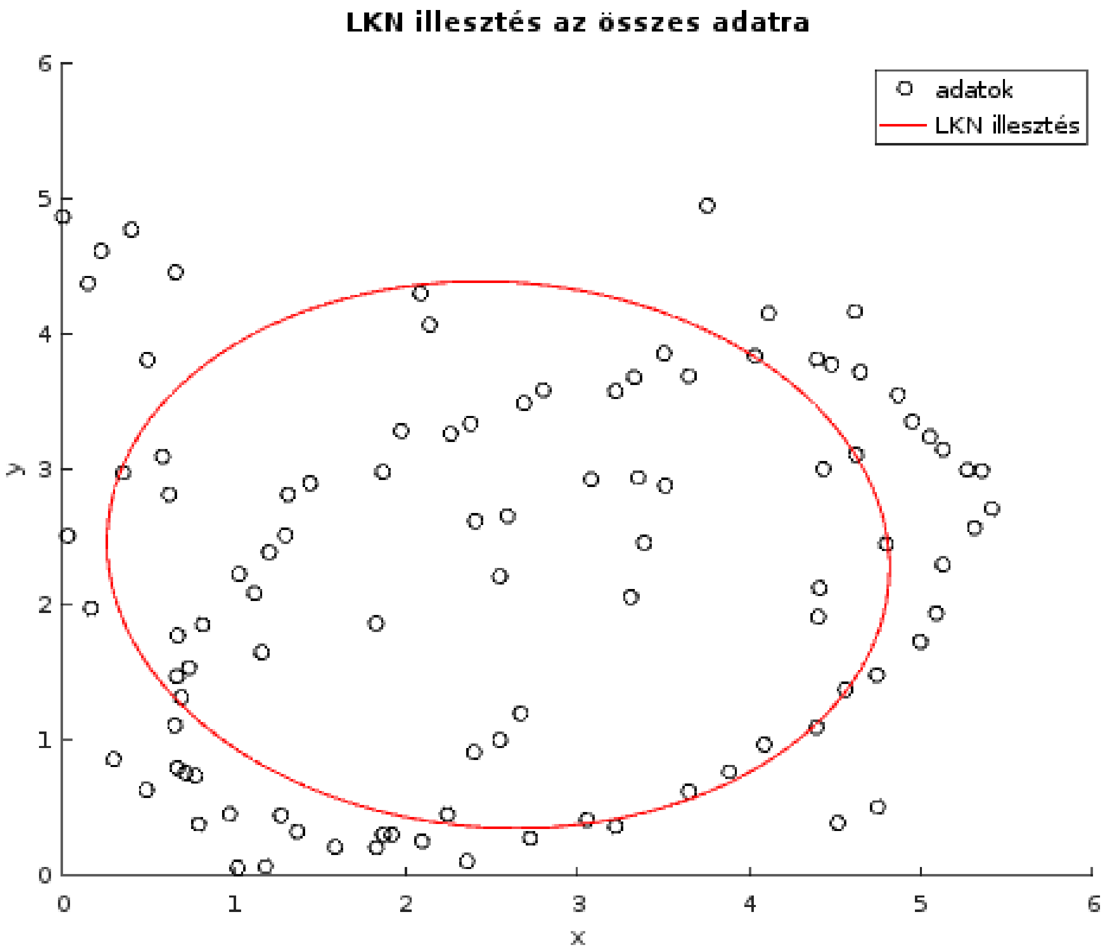
- A meghatározott ellipszis paraméterei (gradiens súlyozás):
  - középpont: (2.95, 1.97), fél tengelyhosszak: (2.56, 1.50)
  - elfordulás szöge: 29.7°



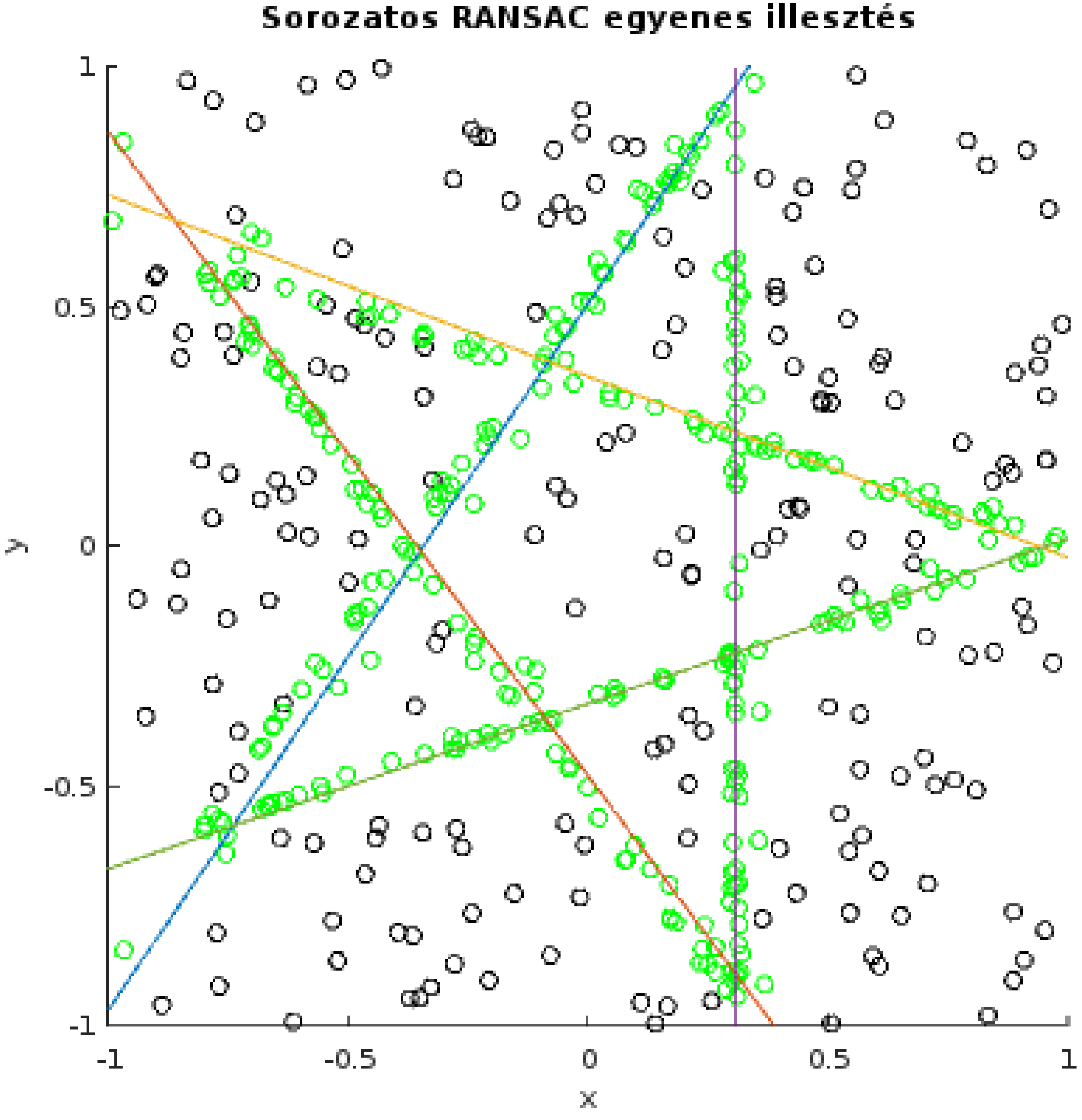
hiba küszöb: 0.25

Illesztés RANSAC nélkül

- A meghatározott ellipszis paraméterei (gradiens súlyozás):
  - középpont: (2.54, 2.37), fél tengelyhosszak: (2.29, 2.01)
  - elfordulás szöge: -10.4°



Sorozatos RANSAC



adatok: `star5.dat`

Egyenes illesztés: sorozatos RANSAC (Octave: seq_ransac_line.m)

- adatok betöltése, felrajzolása
- paraméterek megadása

```
% sorozatos RANSAC egyenes illesztés
clear all; close all
% minta adatok
d = load('star5.dat');
x = d(:,1);  y = d(:,2);
tol = 0.05;  % hiba küszöb
k = 100;  % iterációk száma
m = 5; % modellek száma
figure(1); hold on;
plot(x,y,"ko");
```

Egyenes illesztés: sorozatos RANSAC

- modellenként RANSAC becslés

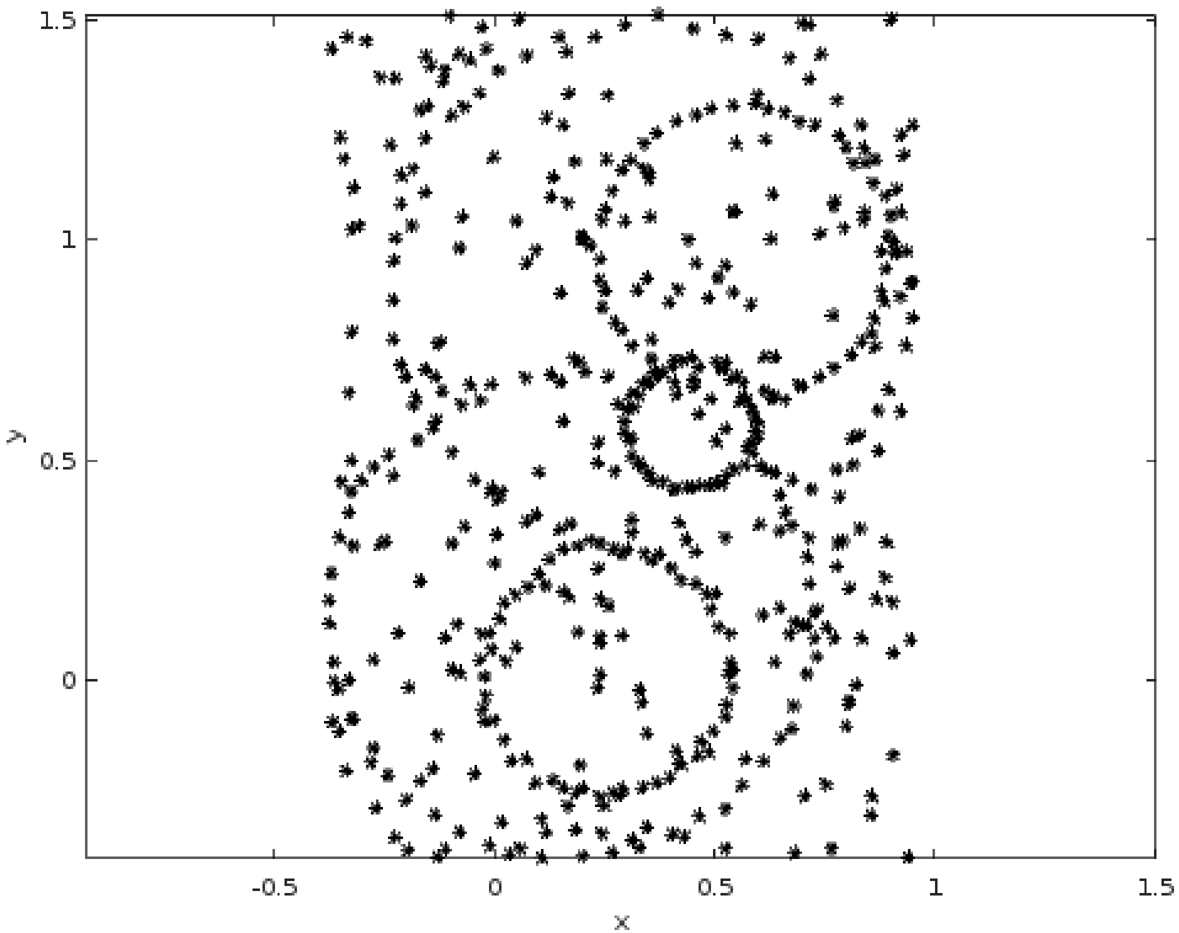
```
for n = 1:m
  nd = length(x);
  nmax = 0; % nincs még konszenzus halmaz
  for i=1:k
    % két pont véletlenszerű kiválasztása
    is = randperm(nd,2);
    % ax+by-1=0 egyenes paramétereinek meghatározása
    A = d(is,:); b = [1; 1]; p = A\b;
    % az adatok egyenestől mért távolságai
    t = abs(p(1)*x+p(2)*y-1)/sqrt(p(1)^2+p(2)^2);
    xk = x(t<tol);  yk = y(t<tol); % illeszkedő adatok
    nin = length(xk);  % konszenzus halmaz elemszáma
    if nin > nmax  % jobb mint idáig
      xin = xk; yin = yk; nmax = nin;
      xout = x(t>=tol); yout = y(t>=tol); % nem konszenzus halmaz
      bp = p;  % legjobb egyenes
    end
  end
```

Egyenes illesztés: sorozatos RANSAC

- LKN egyenes illesztés a maximális halmazra
- a konszenzus halmaz eltávolítása

```
  pls = polyfit(xin,yin,1); % LKN egyenes illesztés
  plot(xin,yin,"go")
  line = @(x,y) bp(1)*x + bp(2)*y - 1;
  ezplot(line,[-1,1,-1,1]) % egyenes felrajzolása
  % a konszenzus halmaz elemeit eltávolítjuk
  x = xout; y = yout; d = [x,y];
end
axis equal; ylim([-1,1]);
title("Sorozatos RANSAC egyenes illesztés");
```

Sorozatos RANSAC (körök illesztése)



adatok: `circles5.dat`

Kör illesztés 3 pontra

- Mindegyik pontra illeszkedik az *r* sugarú kör:

$$\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - r = 0, \quad i = 1, \ldots, 3$$

ezért mindegyik pontra

$$\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} + r = 2r = c = \text{áll}.$$

- A két egyenletet összeszorozva

$$(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 = 0$$

- Új α ismeretlent bevezetve *r* helyett lineáris egyenletrendszert kell megoldani:

$$x_i^2 + y_i^2 - 2x_0 x_i - 2y_0 y_i + \alpha = 0, \quad i = 1, \ldots, 3$$

$$\alpha = x_0^2 + y_0^2 - r^2$$

Sorozatos kör illesztés (Octave megoldás: seq_ransac_circle.m)

- adatok betöltése, felrajzolása
- paraméterek megadása

```
% sorozatos RANSAC kör illesztés
clear all; close all
% minta adatok
d = load('circle5.dat');
x = d(:,1);  y = d(:,2);
tol = 0.05;  % hiba küszöb
k = 100;  % iterációk száma
m = 5; % modellek száma
figure(1); hold on;
plot(x,y,"k*");
axis equal
```

Sorozatos kör illesztés (RANSAC)

- modellenként RANSAC becslés

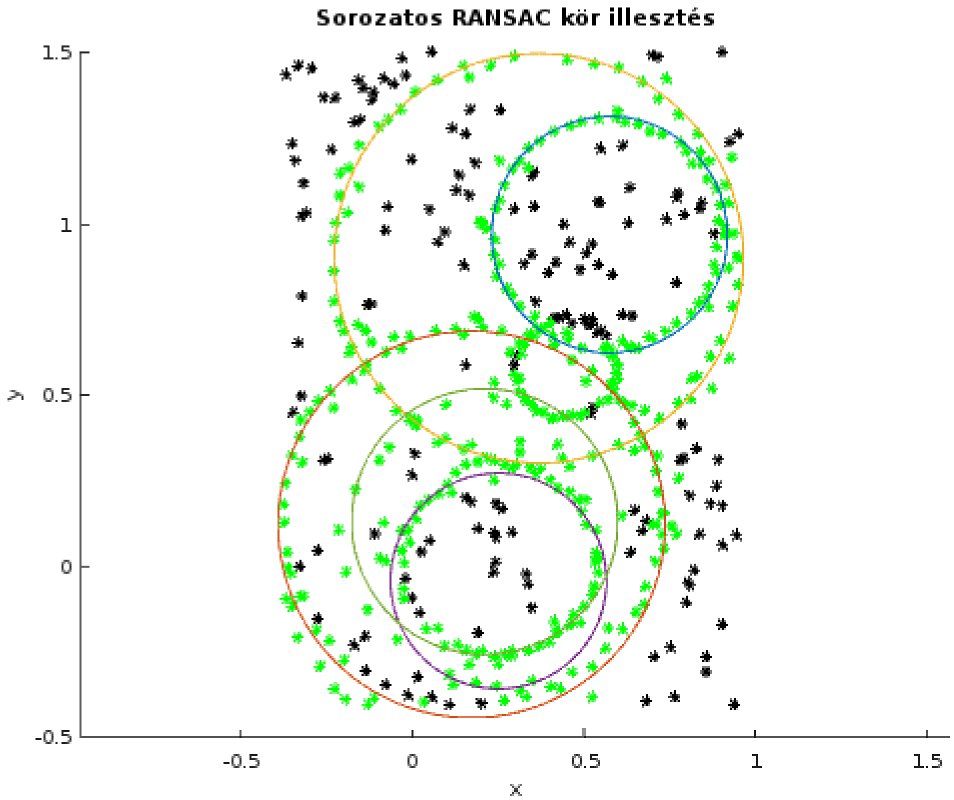
```
for n = 1:m
  nd = length(x);
  nmax = 0; % nincs még konszenzus halmaz
  for i=1:k
    % három pont véletlenszerű kiválasztása
    is = randperm(nd,3);
    % (x-x0)^2 + (y-y0)^2 + R^2 = 0 kör paramétereinek meghatározása
    A = [d(is,:),ones(3,1)]; b = -[d(is,1).^2+d(is,2).^2]; p = A\b;
    xc = -0.5*p(1); yc = -0.5*p(2);
    R = sqrt((p(1)^2+p(2)^2)/4-p(3));
    % az adatok körtől mért távolságai
    t = abs(sqrt((x-xc).^2+(y-yc).^2)-R);
    xk = x(t<tol);  yk = y(t<tol); % illeszkedő adatok
    nin = length(xk);  % konszenzus halmaz elemszáma
    if nin > nmax  % jobb mint idáig
      xin = xk; yin = yk; nmax = nin;
      xout = x(t>=tol); yout = y(t>=tol); % nem konszenzus halmaz
      bp = p;  % legjobb egyenes
    end
  end
```

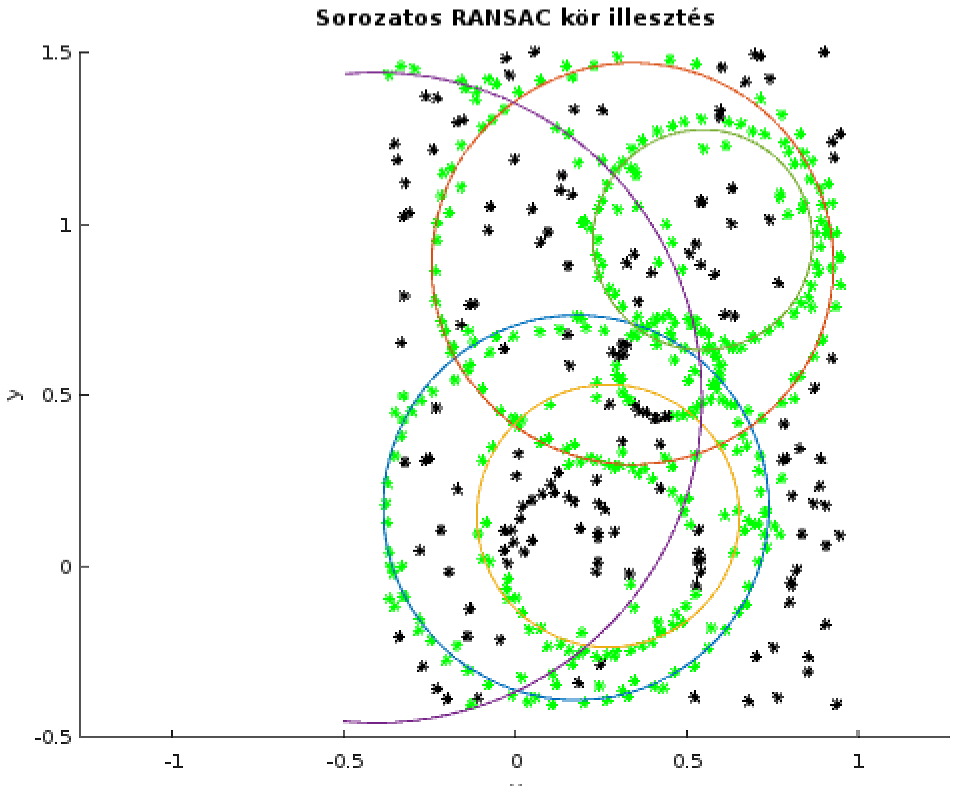
Sorozatos kör illesztés (RANSAC)

- LKN kör illesztés a maximális halmazra
- a konszenzus halmaz eltávolítása

```
  % LKN kör illesztés
  A = [xin,yin,ones(length(xin),1)]; b = -[xin.^2+yin.^2]; p = A\b;
  xc = -0.5*p(1); yc = -0.5*p(2);
  R = sqrt((p(1)^2+p(2)^2)/4-p(3));
  plot(xin,yin,"g*")
  circle = @(x,y) sqrt((x-xc).^2+(y-yc).^2)-R;
  ezplot(circle,[-0.5,1,-0.5,1.5]) % kör felrajzolása
  % a konszenzus halmaz elemeit eltávolítjuk
  x = xout; y = yout; d = [x,y];
end
axis equal;
title("Sorozatos RANSAC kör illesztés");
```

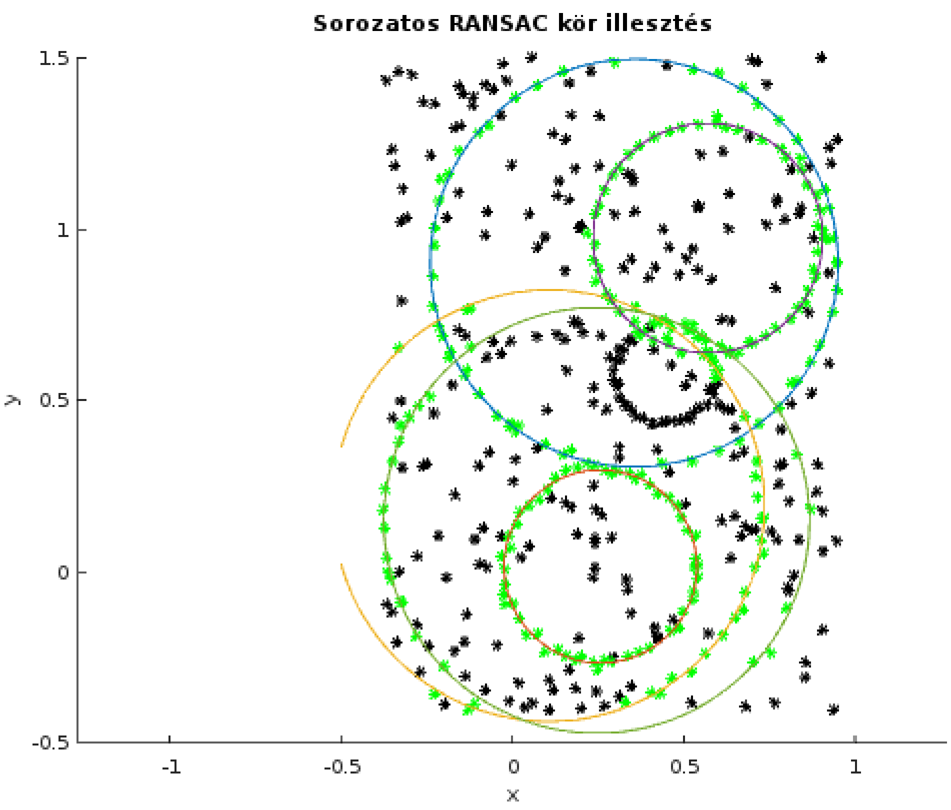
Eredmények

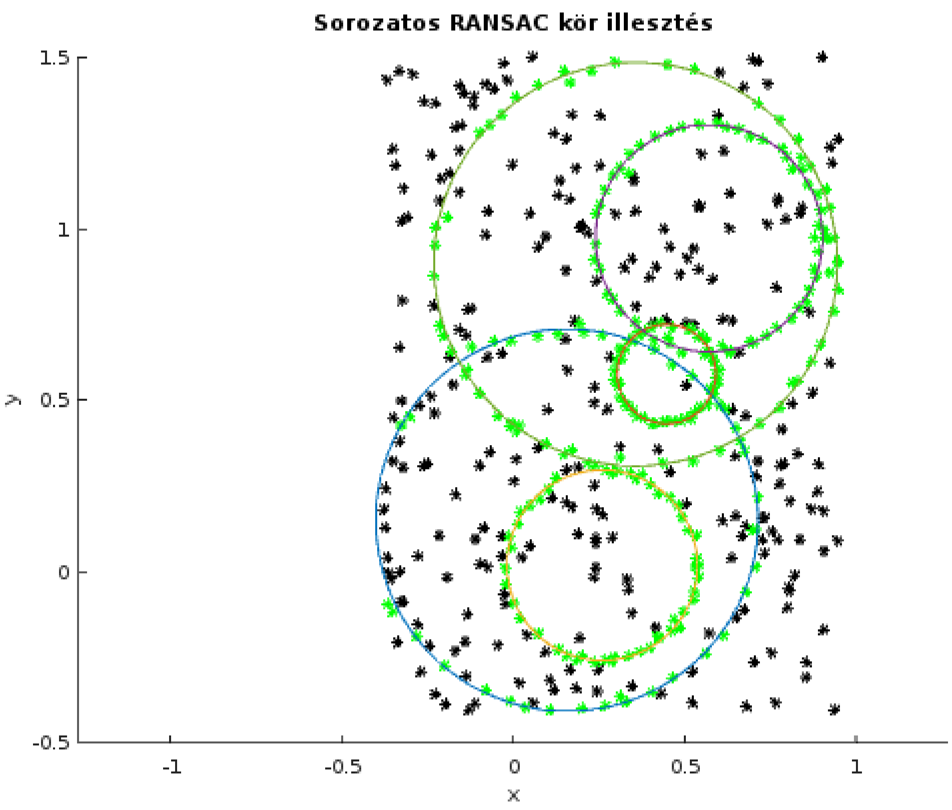




hiba küszöb: `0.05`

Eredmények





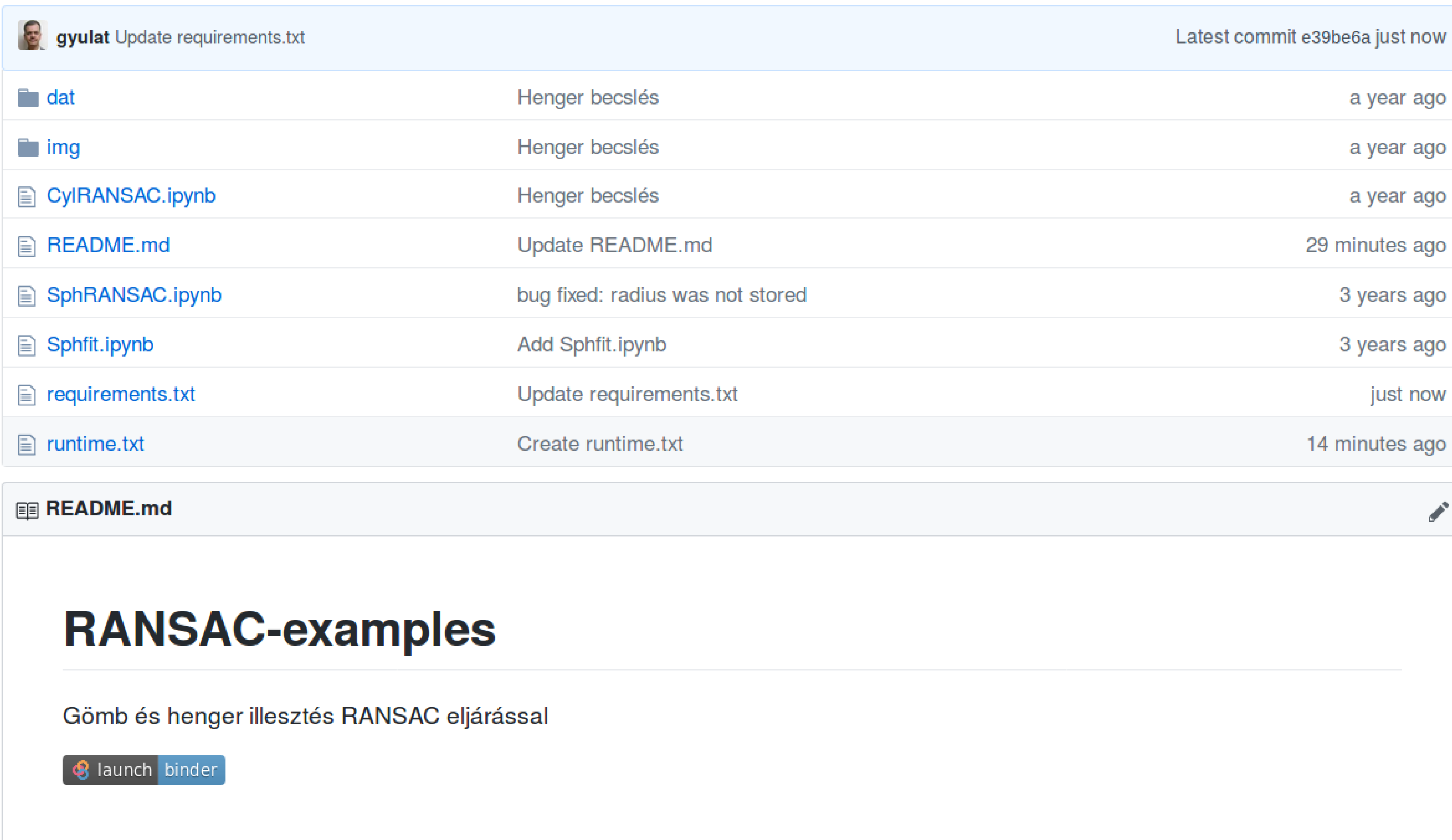
hiba küszöb: `0.02`

További lehetőségek

- A pontok egyenlő valószínűségű kiválasztása helyett a közelebbi pontok nagyobb valószínűséget kaphatnak
- Más eljárásokat is kipróbálhatunk:
  - MultiRANSAC
  - Residual Histogram Analysis, RHA
  - J-linkage, Merging J-Linkage, Kernel Fitting
  - PEARL

Gömb és henger illesztés

- https://github.com/gyulat/RANSAC-examples



gyulat Update requirements.txt — Latest commit e39be6a just now

dat	Henger becslés	a year ago
img	Henger becslés	a year ago
CylRANSAC.ipynb	Henger becslés	a year ago
README.md	Update README.md	29 minutes ago
SphRANSAC.ipynb	bug fixed: radius was not stored	3 years ago
Sphfit.ipynb	Add Sphfit.ipynb	3 years ago
requirements.txt	Update requirements.txt	just now
runtime.txt	Create runtime.txt	14 minutes ago

README.md

RANSAC-examples

Gömb és henger illesztés RANSAC eljárással

launch binder

Jupyter munkafüzet

- Python kód, megjegyzések, ábrák, képletek
- Futtató környezet:
  - Helyi számítógépen telepíthető
  - Felhőben elérhető ideiglenes környezet
    - mybinder.org
      - scikit-image még nem működik :(