

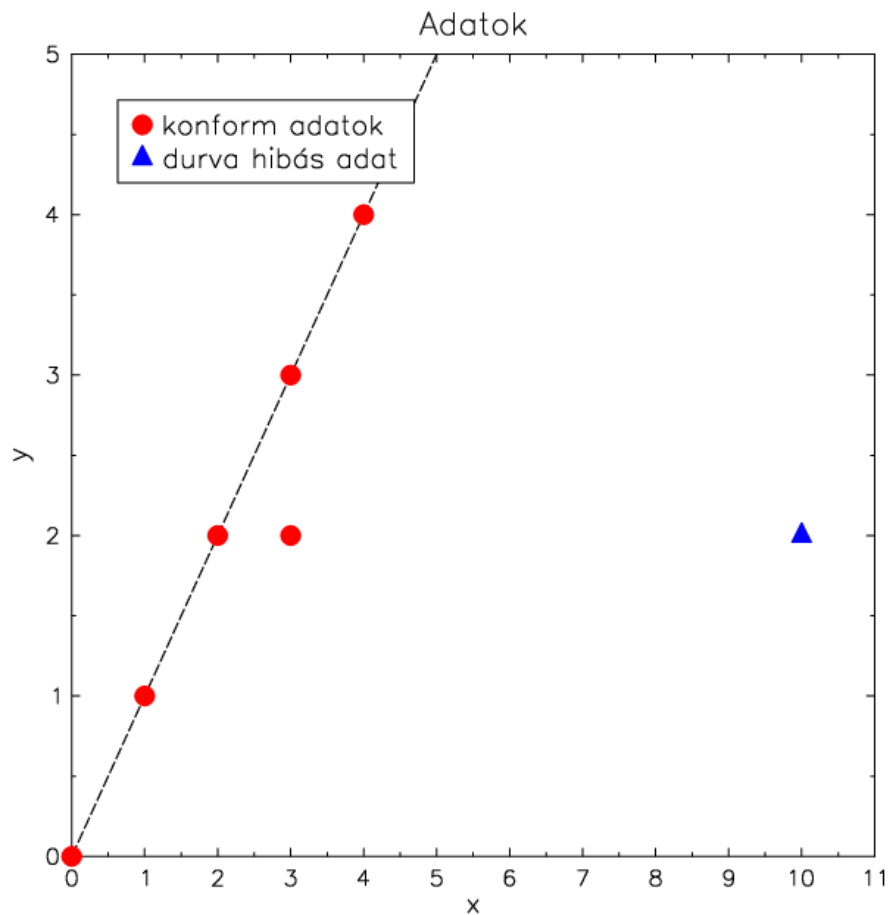


12. előadás

Miről lesz szó?

- RANSAC paraméter becslési eljárás
 - Bevezetés
 - Az eljárás alapötlete
 - Paraméterek és algoritmus
 - RANSAC változatok
 - Példák
 - egyenes illesztése
 - ellipszis illesztése
 - gömb illesztése
 - egyenes körhenger illesztése
 - több modell illesztése

Regressziós egyenes

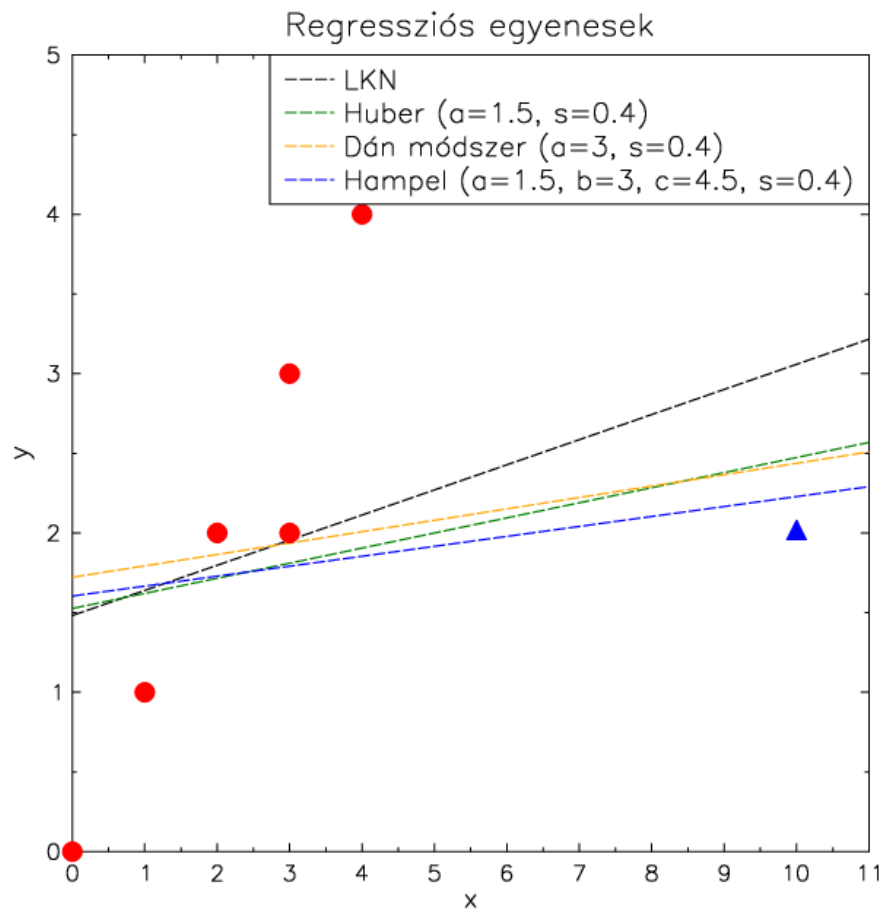


Fishler és Bolles
(1981) példája:

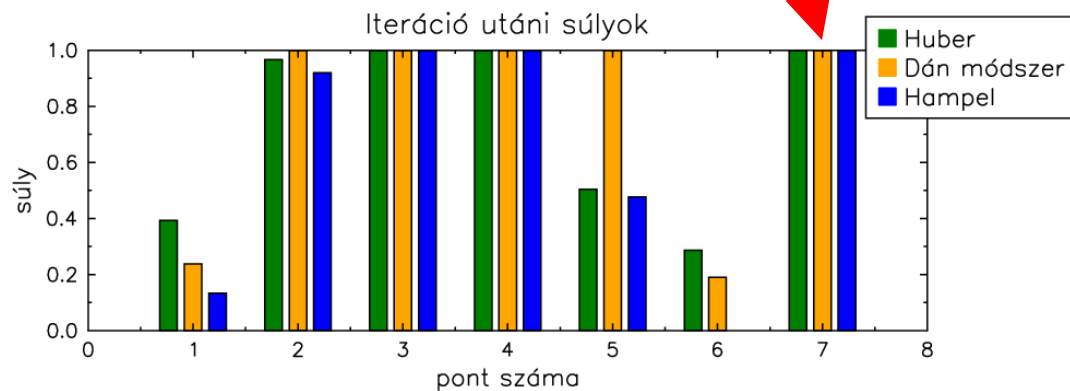
5 pont már egy
egyenesen van

a konform adatok
hibahatára: 0.8

„Rezisztens” M-becslések



a durva hibás 7.
pontot mindegyik
megtartotta!



A probléma oka

- a kezdeti modell paramétereit az összes adat alapján LKN becsléssel határoztuk meg
- ez a kezdeti becslés annyira rossz, hogy sohasem találjuk meg a jó modellt, hiába súlyozzuk újra rezisztens módon a méréseket
- *egyetlen* „kellően” kivágó adat (14%) jelenléte ennek az oka

A probléma megoldása

- *Fordított* megközelítésre van szükség:
 - a kezdeti modell paramétereit csak a *minimálisan szükséges adatmennyiség* alapján határozzuk meg
 - ezt a kezdeti becslést próbáljuk *bővíteni* a modellre illeszkedő, *konform* adatokkal
 - azt a modellt tartjuk meg, amelyre a *legtöbb* konform adat illeszkedik

Ez a RANSAC (iteratív robusztus becslés)

Az iteratív robusztus becslési eljárás (RANSAC)

- Fischler és Bolles 1981-ben alkották meg a módszert
- RANSAC = RANdOm SAMple Consensus
- Rendkívül *népszerű*, általánosan alkalmazható robusztus becslési eljárás
- *nem determinisztikus* algoritmus: csak egy adott valószínűséggel ad helyes eredményt – ez a valószínűség az iterációk számának növelésével növelhető

Fischler és Bolles cikke (1981)

Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography

MA Fischler, RC Bolles

Communications of the ACM, 1981 · dl.acm.org

A new paradigm, Random Sample Consensus (RANSAC), for fitting a model to experimental data is introduced. RANSAC is capable of interpreting/smoothing data containing a significant percentage of gross errors, and is thus ideally suited for applications in automated image analysis where i provided by error-prone feature detectors. A major application of RANSAC to the Location Determination Problem

Graphics and
Image Processing

J. D. Foley
Editor

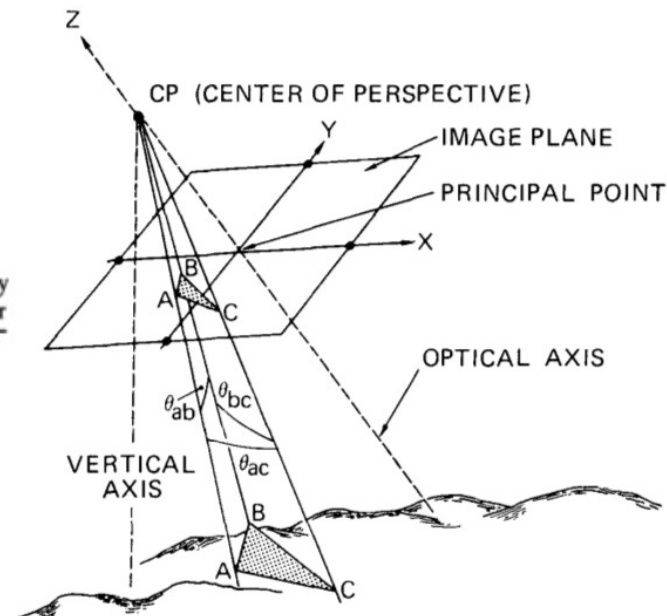
Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography

Martin A. Fischler and Robert C. Bolles
SRI International

TÖBB MEGJELENÍTÉSE ▾

☆ Mentés Hivatkozás Idézetek száma: 31934 Kap

Fig. 2. Geometry of the Location Determination Problem.



The LDP is formally defined as follows:

Given a set of m control points, whose 3-dimensional coordinates are known in some coordinate frame, and given an image in which some subset of the m control points is visible, determine the location (relative to the coordinate system of the control points) from which the image was obtained.

RANSAC – miért népszerű?

- *könnyen érthető és hatékony* (több mint 50% durva hibát is képes sikeresen kezelni – magas összeomlási pont)
- *gyakran ismétlődő problémát old meg* (automata mérőrendszerek durva hibás méréseinek kiszűrése)
- a durva hibák nagy valószínűséggel (99,99%) történő kiszűréséhez meglepően *kevés iteráció kell*
- a *számítási kapacitás növekedése* lehetővé tette nagy számú iteráció (több száz, ezer) elvégzését
- az eljárás *azonnal megállhat*, mihelyt egy jó egyezést találunk (ellentétben más eljárásokkal, melyek először nagy számú mintát állítanak elő és csak utána keresik az egyezéseket)

A RANSAC eljárás lépései

- Állítsunk elő egy előre meghatározott M számú modellt (hipotézist), mindegyiket a modell egyértelmű megalkotásához szükséges minimális n darab adat alapján
(ez a modell paraméterek meghatározását jelenti)
- Értékeljük ki mindegyik modellt (hipotézist):
 - számítsuk ki az összes adat *illeszkedési eltéréseit* a modellhez képest egy adott hibaküszöbön belüli adatok alkotják a *konform* adathalmazt (ún. „*konszenzus*” *halmazt*)
- Azt a modellt (hipotézist) választjuk, amelynek konform (konszenzus) halmaza *a legtöbb elemből áll*
 - erre a modellre újra meghatározzuk a modell paramétereit a konform adatok alapján (általában LKN becsléssel)

A RANSAC eljárás paramétere

- A modellek megalkotásához szükséges paraméterek minimális n száma
- A maximálisan megengedhető iterációk k száma
- t küszöbérték annak eldöntéséhez, hogy egy adat illeszkedik-e a modellhez
- az a d adatszám, amely alapján eldöntjük, hogy egy modell jól illeszkedik-e az adatokhoz

A RANSAC eljárás paramétere

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

$$k = \frac{\ln(1 - p)}{\ln(1 - w^n)}$$

$$\sigma(k) = \frac{\sqrt{1 - w^n}}{w^n}$$

k : iterációk száma

n : egy iterációban kiválasztott adatok száma

p : annak a valószínűsége, hogy a RANSAC egy iterációban csakis konform adatokat választ ki (egy iterációban n darabot)

w : A konform (nem durva hibás) adat előfordulási valószínűsége

$\sigma(k)$: k szórása (közel egyezik k -val)

(k értékének helyes megválasztásához kell)

A RANSAC eljárás paramétere

Levezetés lépésenként:

$$w^n$$

annak valószínűsége, hogy *egy* durva hibás adat *sem* kerül be a modellbe

$$1 - w^n$$

annak valószínűsége, hogy *legalább egy* durva hibás adat bekerül a modellbe

$$(1 - w^n)^k$$

annak valószínűsége, hogy a k modell *mindegyikébe* bekerül durva hibás adat

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

annak valószínűsége, hogy a k modell *nem mindegyikébe* kerül durva hibás adat, vagyis a k modellből *lesz legalább egy* durva hiba mentes

Hány k iteráció lehet szükséges?

adatok száma	durva hibás adatok aránya ($1 - w$)									
	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
n										
2	2	3	4	5	6	7	8	10	13	16
3	2	4	5	6	8	11	14	19	25	34
4	3	4	6	9	12	17	23	33	48	71
5	3	5	8	12	17	25	37	57	89	145
6	3	6	10	15	23	37	59	96	164	292
7	4	7	12	20	32	54	92	162	300	587
8	4	8	14	25	44	78	142	272	548	1177

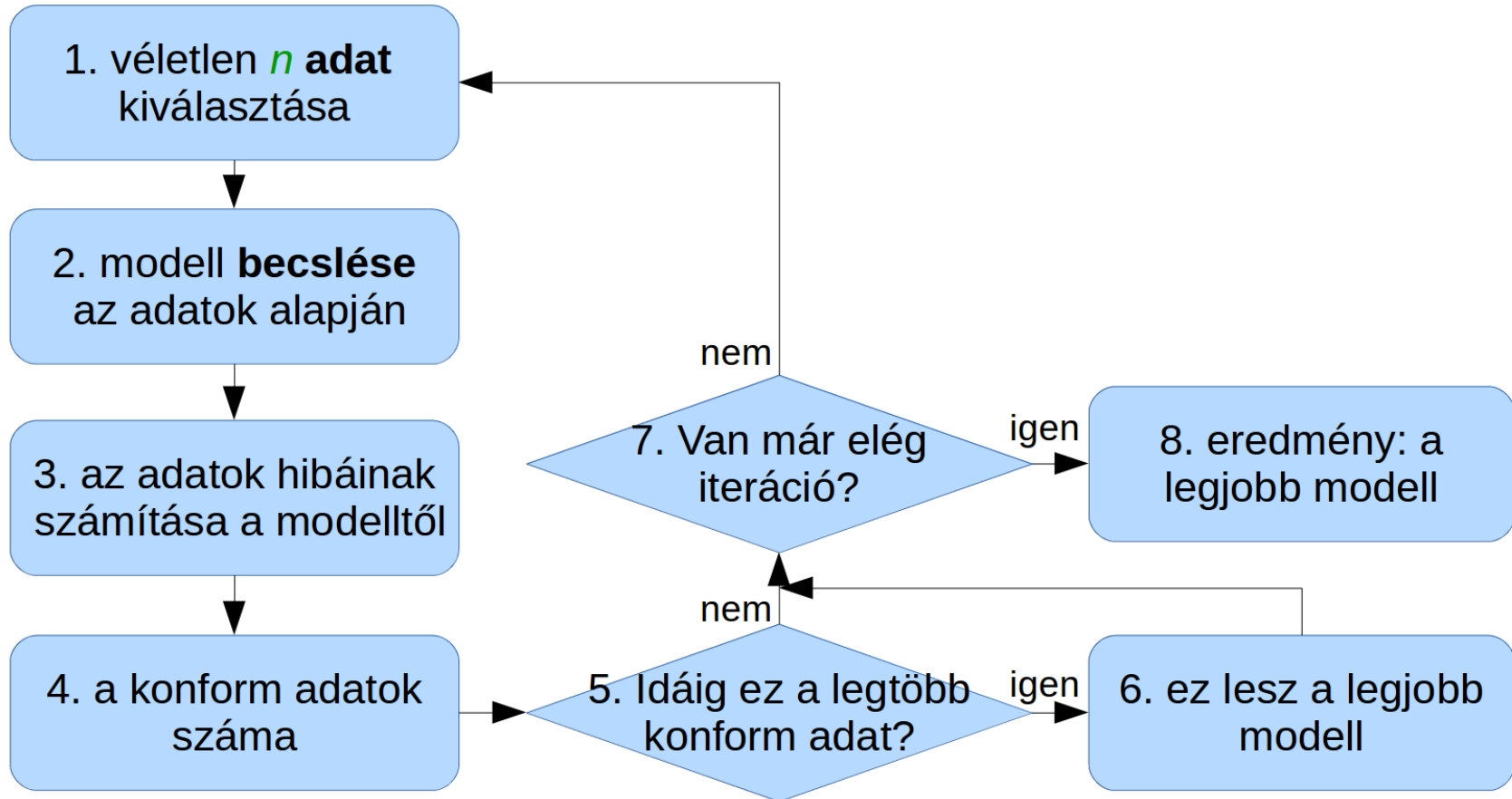
$p = 0.99$
(szokásos érték)

$$k = \frac{\ln(1 - p)}{\ln(1 - w^n)}$$

A RANSAC eljárás eredményei

- *A legjobb modell*: azok a paraméterek, amelyek a legjobban illeszkedő modellhez tartoznak (vagy semmi, ha nincs jól illeszkedő modell)
- *A legjobb konszenzus halmaz*: azok az adatok, amelyekből a legjobb modellt becsültük
- *A legkedvezőbb hiba*: a legjobb modell hibái a legjobb konszenzus halmaz adataihoz képest

A RANSAC folyamatábrája



RANSAC algoritmus pszeudokódja

Adott:

adat - a mért adatpontok
model - az adatokra illesztendő modell
n - a modellek megalkotásához szükséges paraméterek minimális száma
k - A maximálisan megengedhető iterációk száma
t - küszöbérték annak eldöntéséhez, hogy egy adat illeszkedik-e a modellhez
d - adatszám, amely alapján eldöntjük, hogy egy modell jól illeszkedik-e az adatokhoz

Eredmény:

legjobb - legjobban illeszkedő modellhez tartozó paraméterek (semmi, ha nincs jó modell)

```
iterációk = 0
legjobb = semmi
legjobb_hiba = valami nagyon nagy érték
amíg iterációk < k {
    lehet_konform = n véletlenszerűen választott adat
    lehet_model = a lehet_konform-hoz illesztett modell paraméterek
    szintén_konform = üres halmaz
    tedd minden adatpontra amely nincs a lehet_konform-ban {
        ha a pont illeszkedik a lehet_model-re t-nél kisebb hibával
            add a pontot a szintén_konform-hoz }
    ha az elemek száma a szintén_konform-ban > d {
        % ez jelzi, hogy egy jó modellt találtunk
        % most ellenőrizzük, hogy mennyire jó
        jobb_model = az összes pontra (a lehet_konform és szintén_konform pontokra)
            végzett illesztés paraméterei
        ez_a_hiba = annak mértéke, mennyire illeszkedik a modell a pontokra
        ha ez_a_hiba < legjobb_hiba {
            legjobb = jobb_model
            legjobb_hiba = ez_a_hiba }
    }
    következő iteráció
}
eredmény: legjobb
```

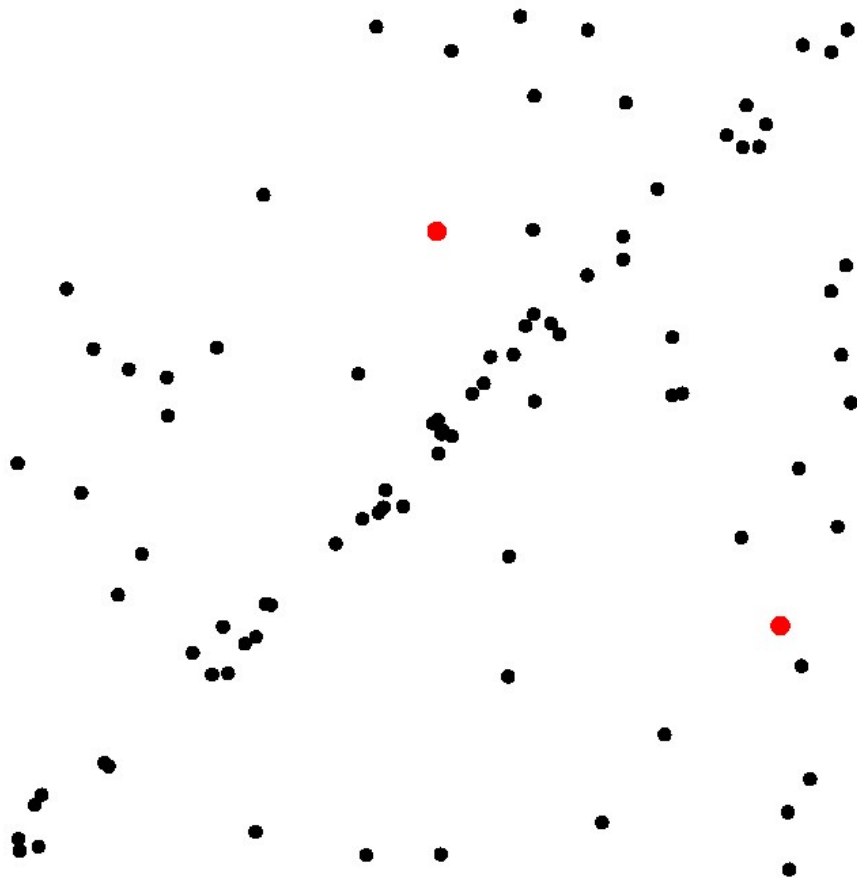
Egyenes illesztés



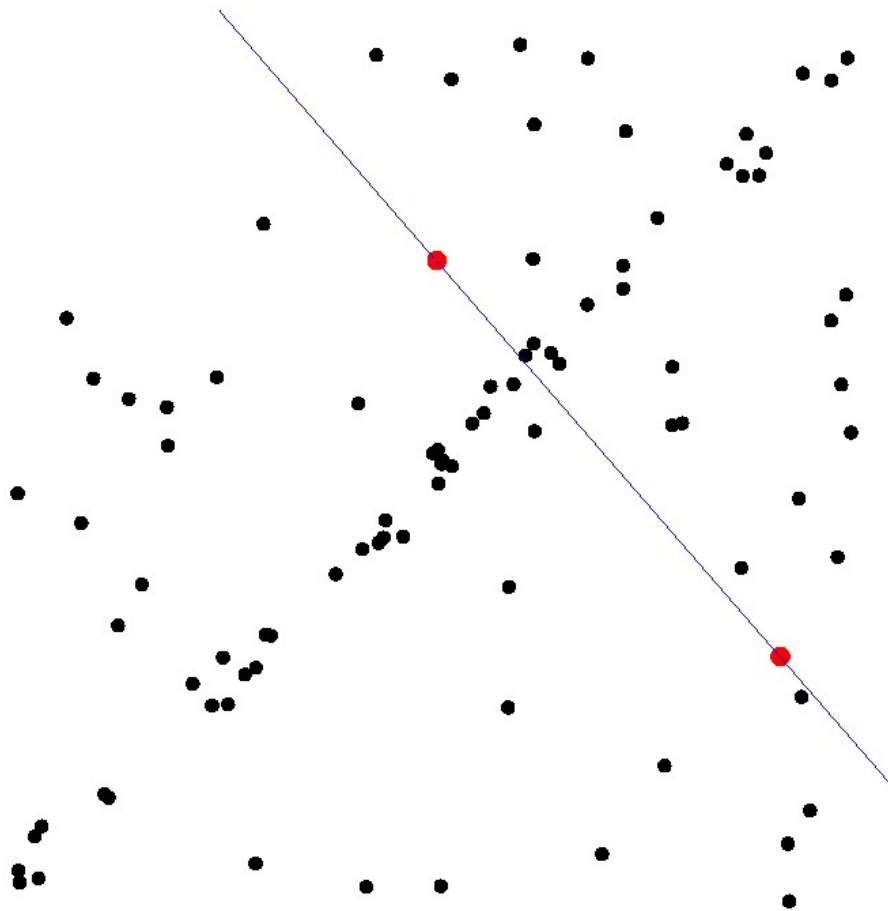
R. Raguram

Egyenes illesztés

$n = 2$ pontot véletlenszerűen kiválasztok



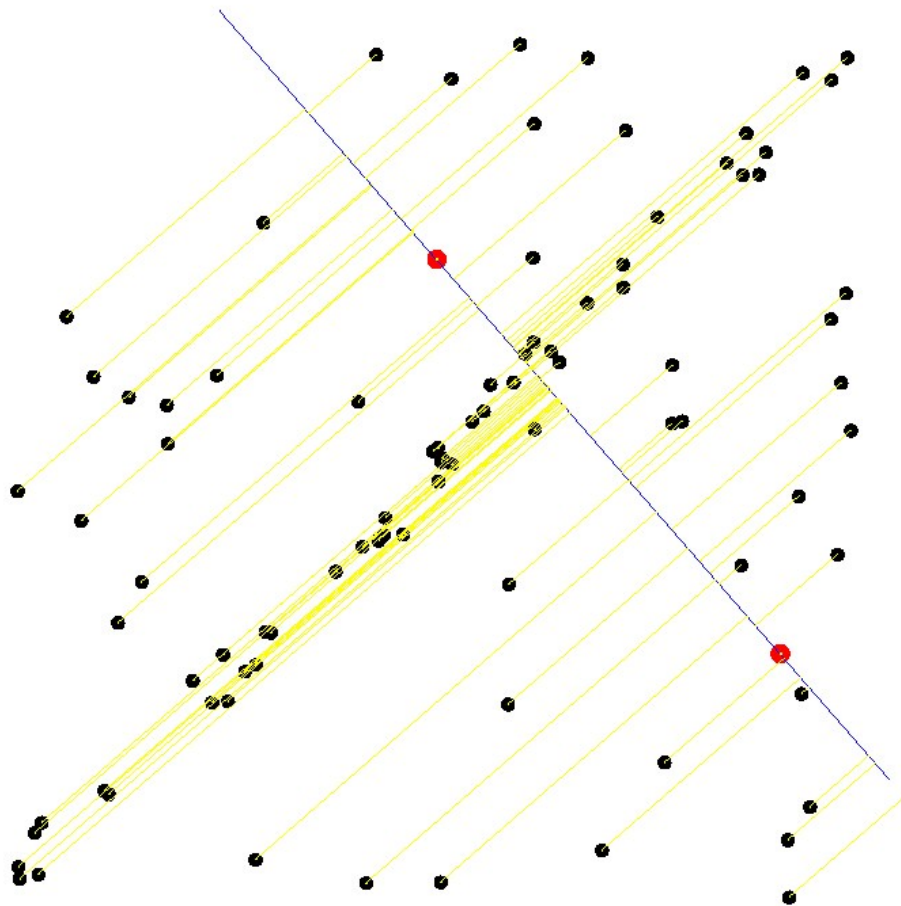
Egyenes illesztés



$n = 2$ pontot véletlenszerűen kiválasztok

kiszámítom az illeszkedő modell paramétereit

Egyenes illesztés

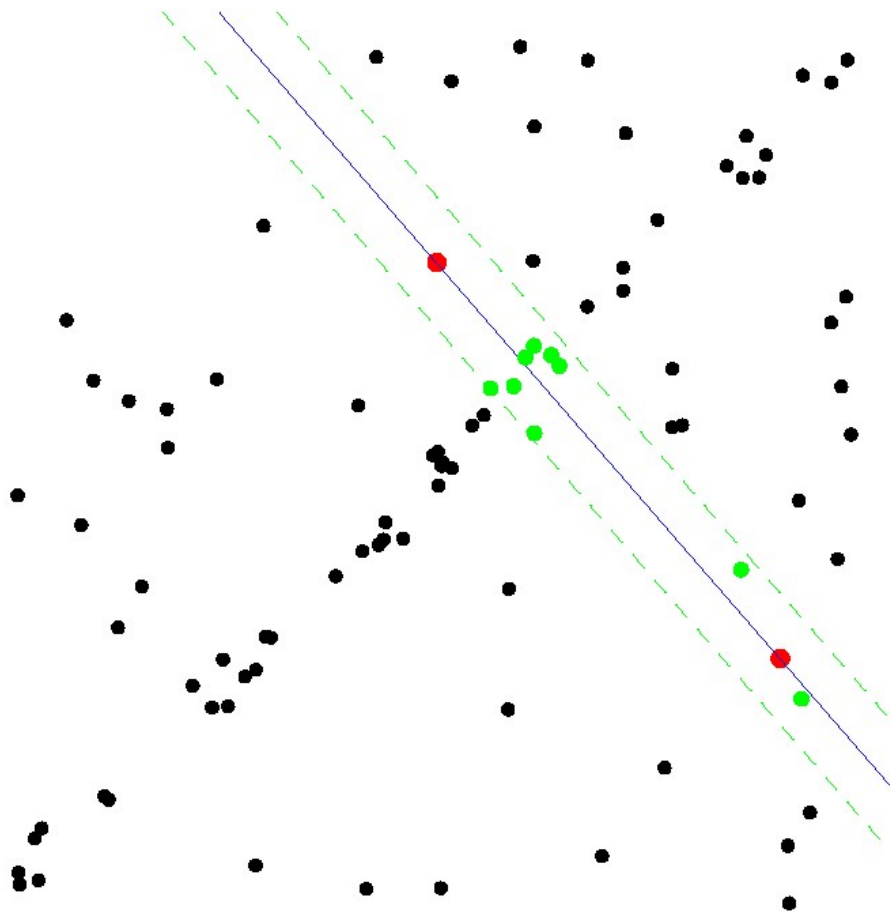


$n = 2$ pontot véletlenszerűen kiválasztok

kiszámítom az illeszkedő modell paramétereit

minden adatpont illeszkedési eltérését kiszámítom

Egyenes illesztés



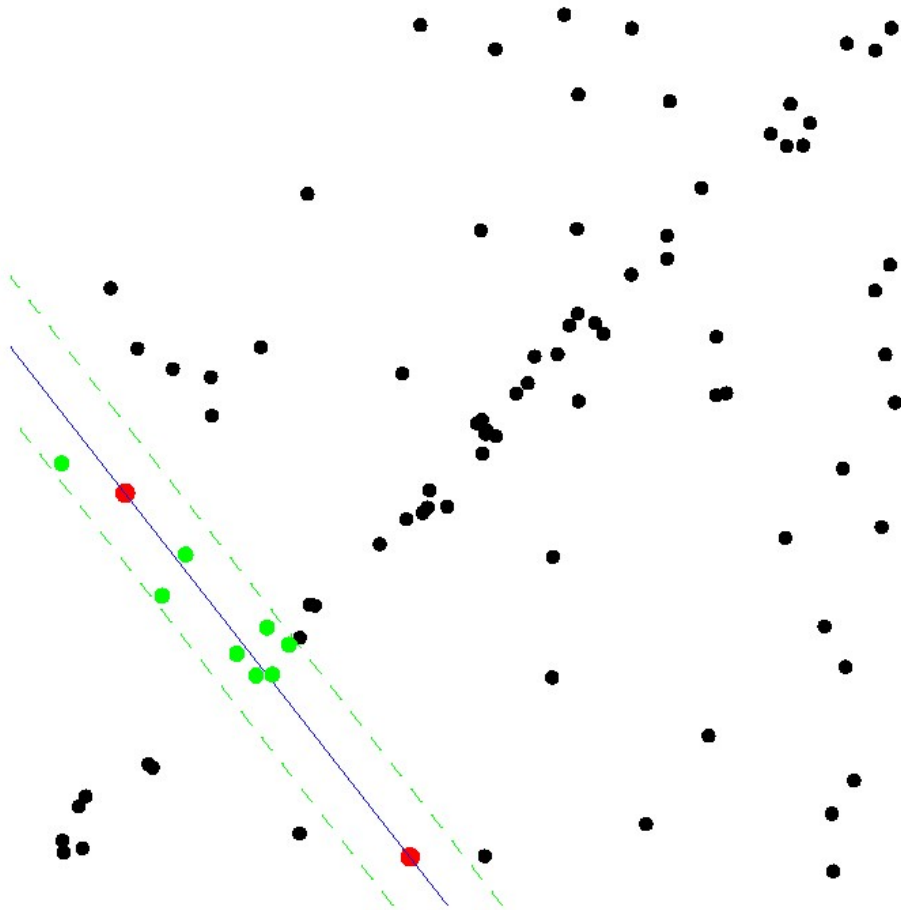
$n = 2$ pontot véletlenszerűen kiválasztok

kiszámítom az illeszkedő modell paramétereit

minden adatpont illeszkedési eltérését kiszámítom

megkeresem a konszenzus halmaz pontjait

Egyenes illesztés



$n = 2$ pontot véletlenszerűen kiválasztok

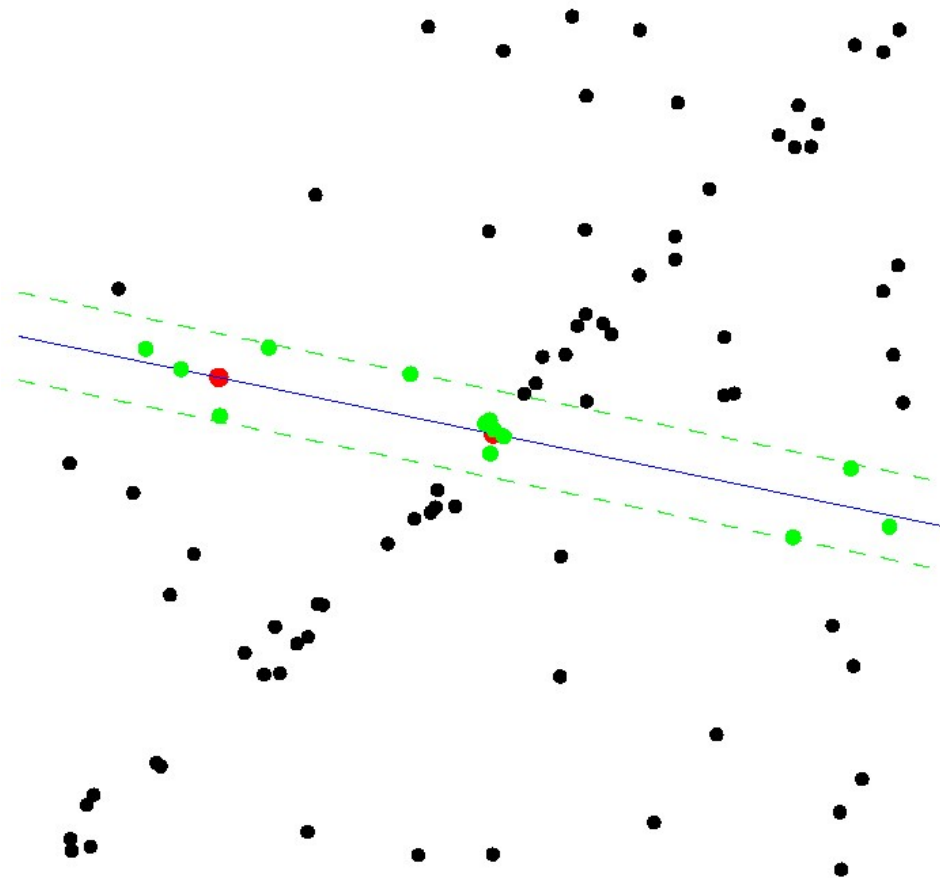
kiszámítom az illeszkedő modell paramétereit

minden adatpont illeszkedési eltérését kiszámítom

megkeresem a konszenzus halmaz pontjait

megismétlem az iterációt újabb 2 pont kiválasztásával

Egyenes illesztés



$n = 2$ pontot véletlenszerűen kiválasztok

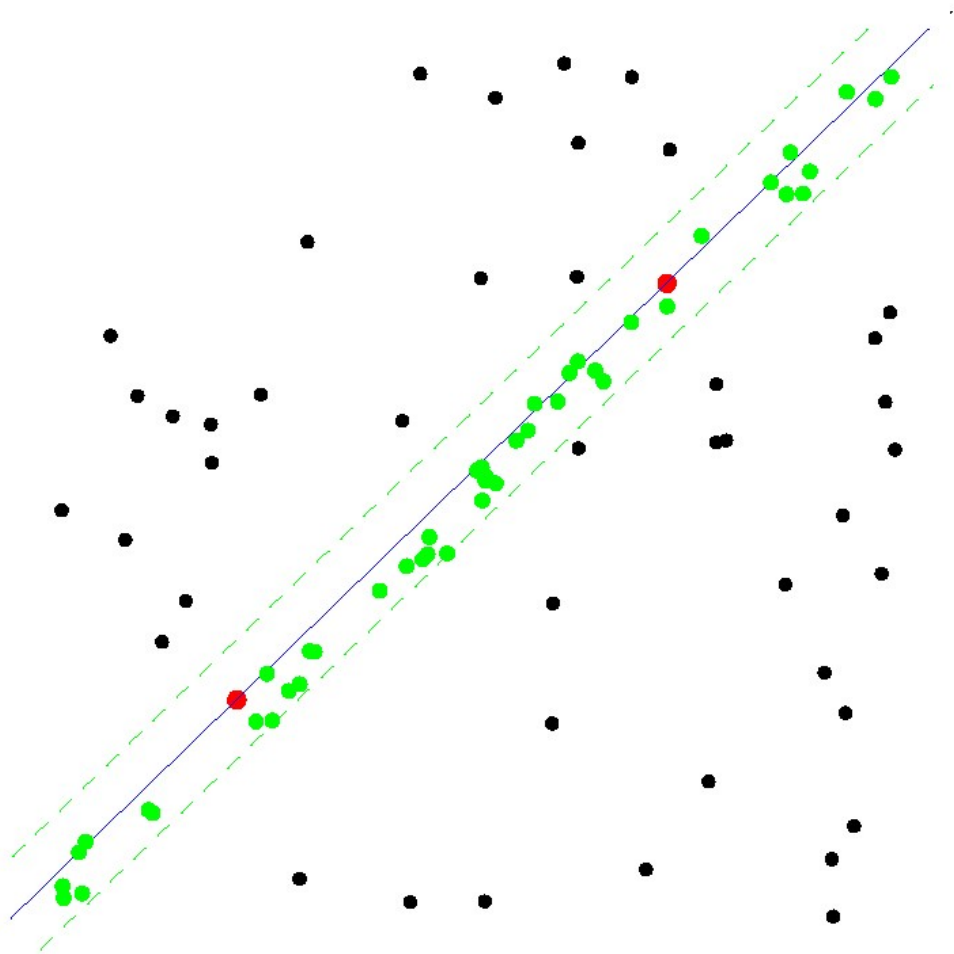
kiszámítom az illeszkedő modell paramétereit

minden adatpont illeszkedési eltérését kiszámítom

megkeresem a konszenzus halmaz pontjait

megismétlem az iterációt újabb 2 pont kiválasztásával

Egyenes illesztés



$n = 2$ pontot véletlenszerűen kiválasztok

kiszámítom az illeszkedő modell paramétereit

minden adatpont illeszkedési eltérését kiszámítom

megkeresem a konszenzus halmaz pontjait

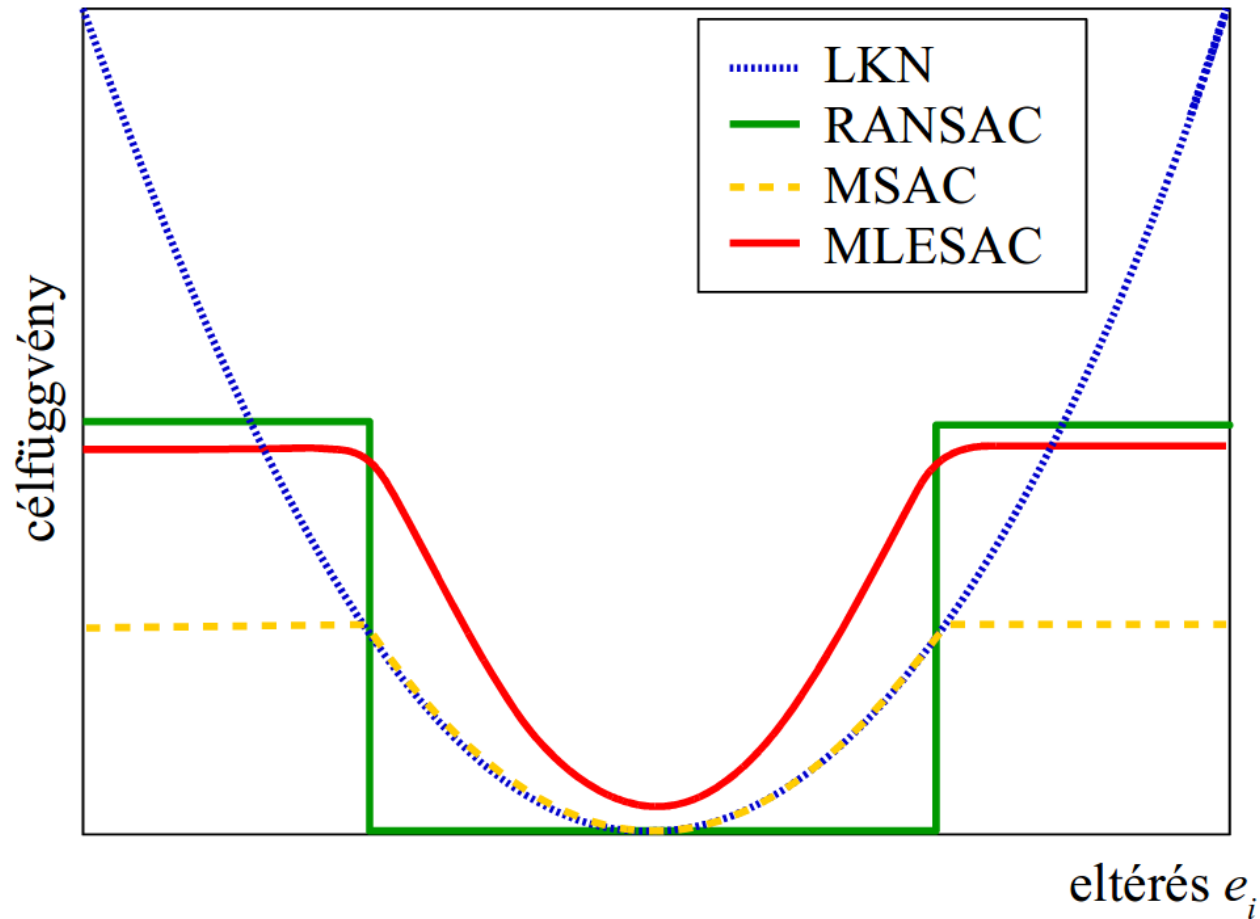
megismétlem az iterációt újabb 2 pont kiválasztásával

maximális konszenzus halmaz

A RANSAC változatai

- MSAC – M-becslő RANSAC
- MLESAC – Maximum Likelihood RANSAC
- PROSAC – progresszív RANSAC (a lehetséges modell kiválasztását előzetes információ segíti)
- R-RANSAC – nem a teljes adathalmazra értékeljük ki a modellt (gyorsabb)
- KALMANSAC – Kálmán-szűrő és RANSAC

Célfüggvények



Ellipszis illesztése

- Gyakran előforduló feladat a fotogrammetriában
- Hány pont szükséges a feladat megoldásához?
- Hogyan lehet meghatározni az ellipszis helyzetét, irányát?

Az ellipszis különböző egyenletei

- paraméteres: h, k, a, b, τ

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} h \\ k \end{bmatrix} + \begin{bmatrix} \cos \tau & -\sin \tau \\ \sin \tau & \cos \tau \end{bmatrix} \begin{bmatrix} a \cos t \\ b \sin t \end{bmatrix}, \quad 0 \leq t \leq 2\pi$$

- kúpszeletes: B, C, D, E, F

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

- fókusz-húr: $\alpha_1, \beta_1, \alpha_2, \beta_2, s$

$$\sqrt{(x - \alpha_1)^2 + (y - \beta_1)^2} + \sqrt{(x - \alpha_2)^2 + (y - \beta_2)^2} = s$$

Paraméterek meghatározása

- kúpszeletes egyenlet *lineáris* a B, C, D, E, F paraméterekre
5 pontra 5 egyenlet felírható és az 5 ismeretlen kiszámítható
- Transzformáció paraméteres alakba

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

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

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

$$h = (BE - 2CD)/(4AC - B^2)$$

$$k = (BD - 2AE)/(4AC - B^2)$$

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

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

Ellipszistől mért távolságok

- algebrai távolság

$$t_a(x_i, y_i) = x_i^2 + Bx_i y_i + Cy_i^2 + Dx_i + Ey_i + F$$

- gradienssel súlyozott algebrai távolság

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

- geometriai távolság

$$t_g(x_i, y_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$

- ahol az x, y az ellipszisen legközelebb levő pont koordinátái

Ellipszisen legközelebb levő pont

- az alábbi két ismeretlenes nemlineáris egyenletrendszer megoldása adja a legközelebbi pont x, y helyzetét

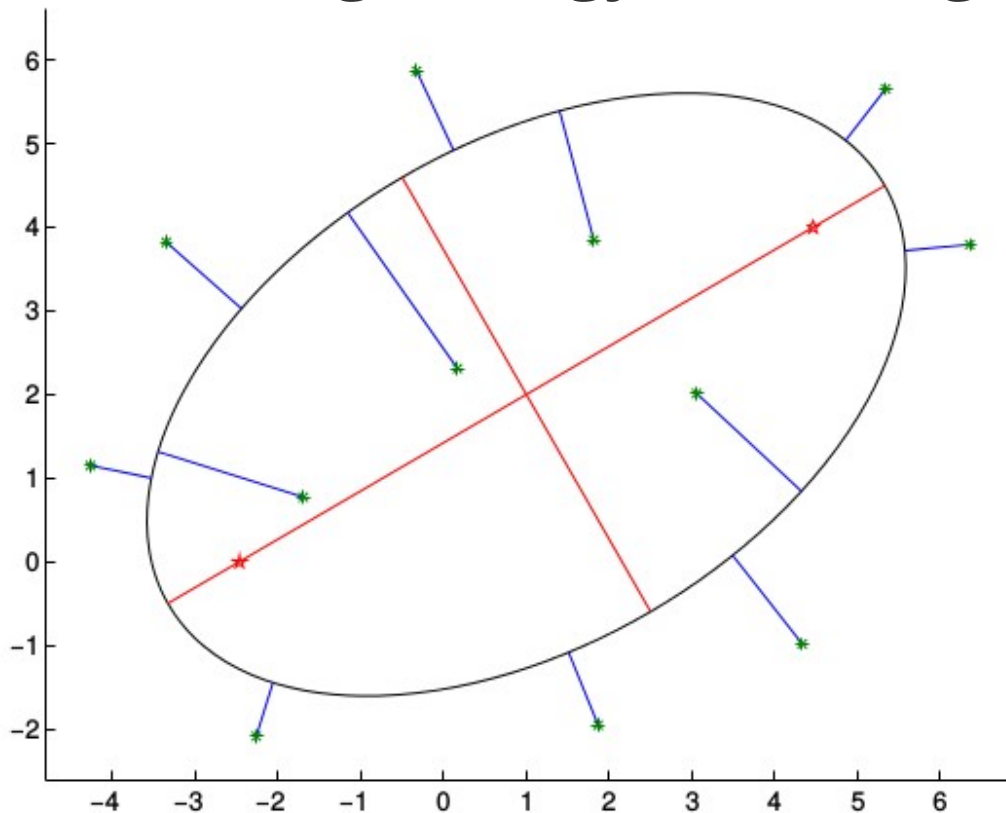
$$x^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

$$Bx^2 + 2(C - 1)xy - By^2 + (2y_i + E)x + \\ + (By_i - D - 2Cx_i)y + (Dy_i - Ex_i) = 0$$

- vagy egy negyedfokú algebrai egyenlet gyökei

Teljes LKN illesztés >5 pontra

- merőleges távolságok négyzetösszege minimális



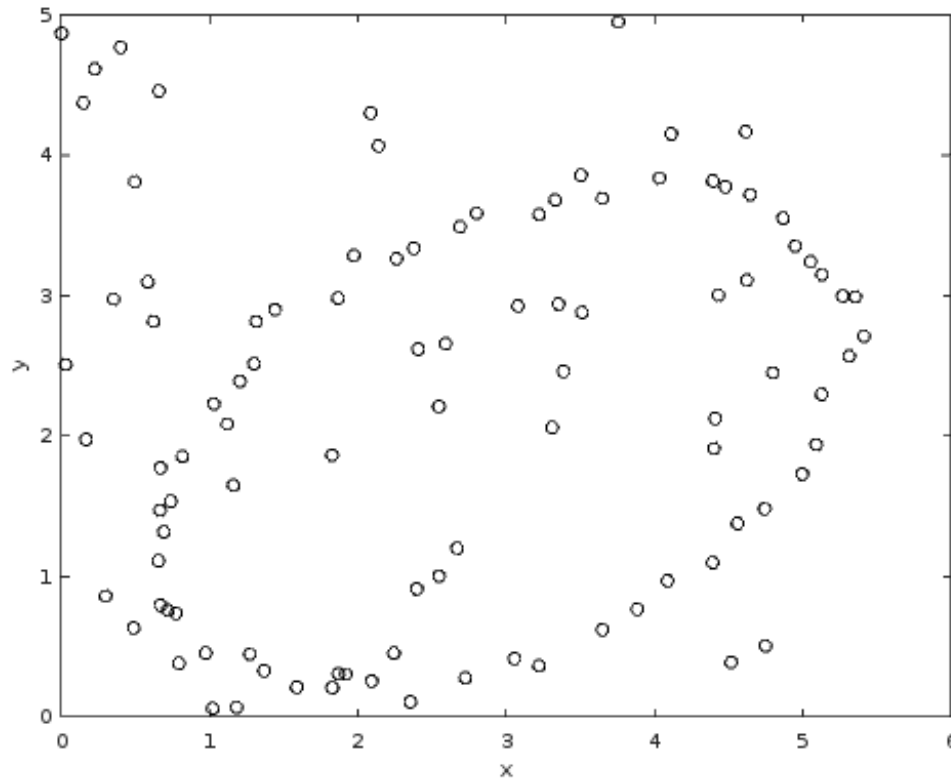
Példa ellipszis illesztésre

- A felvett ellipszis paramétereit:

középpont: $(3, 2)$, fél tengelyhosszak: $(2.5, 1.5)$

elfordulás szöge: 30°

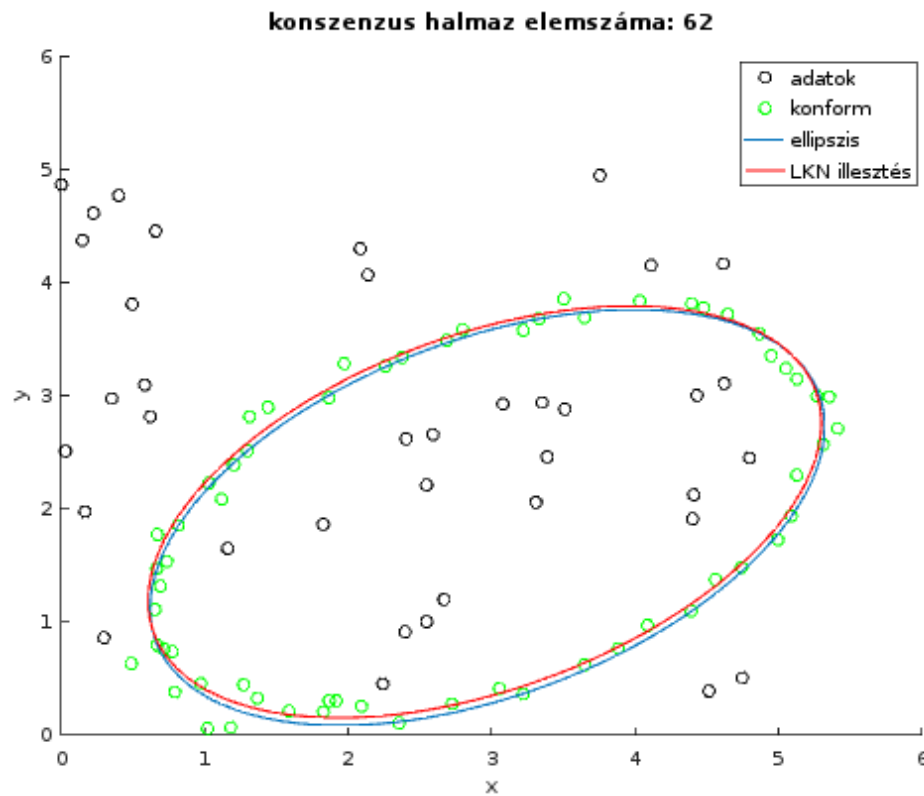
50% a kivágó
adatok aránya



Illesztés eredményei

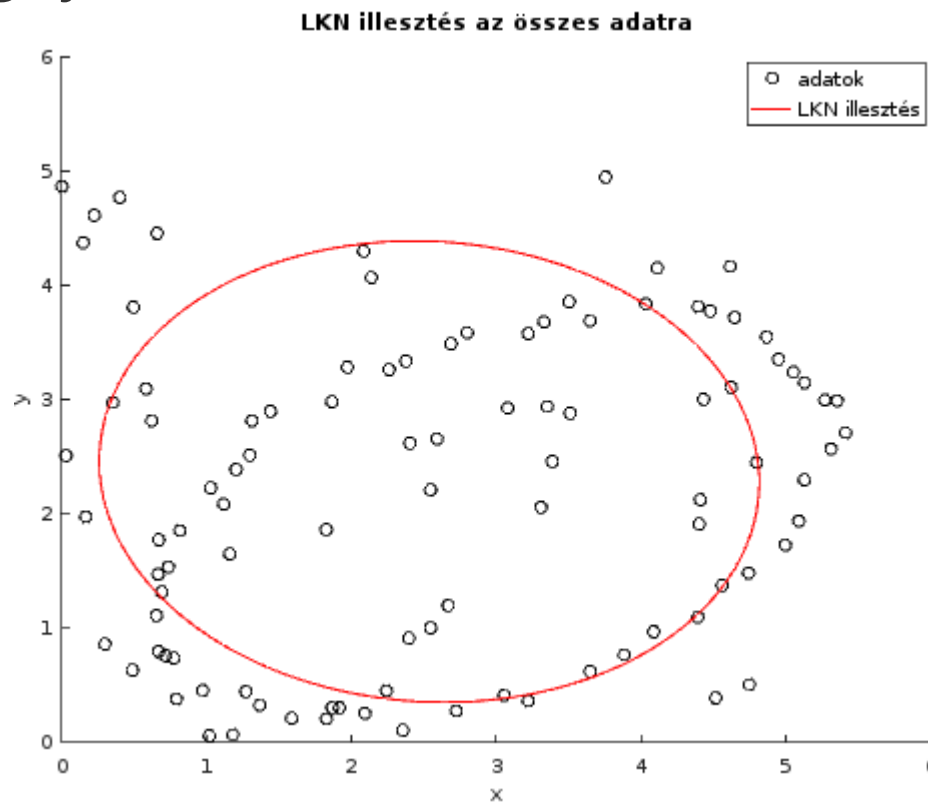
- A meghatározott ellipszis paramétereit (gradiens súlyozás):
középpont: (2.95, 1.97), fél tengelyhosszak: (2.56, 1.50)
elfordulás szöge: 29.7°

hiba küszöb: 0.25



Illesztés RANSAC nélkül

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





Gömb illesztése

- Lézerszkenner illesztőpontok
- Hány pont szükséges a feladat megoldásához?
- Hogyan lehet meghatározni a gömb helyzetét, sugarát?
- Próba számítások

A probléma szakirodalma

- Kåsa (1976) – körillesztés, gömbre is jó
- Pratt (1987) – Pratt módszere
- Taubin (1991) – Taubin módszere
- Lukács et al. (1997, 1998)
- Paláncz, Molnár (2012) – algebrai, geometriai, RANSAC megoldások

Pratt (1987)

- algebrai felületek (kör, kúpszelet, sík, gömb) illesztése
- nem igényel iterációt

Direct Least-Squares Fitting of Algebraic Surfaces

Vaughan Pratt

Sun Microsystems Inc.

and

Stanford University

April 30, 1987

Abstract In the course of developing a system for fitting smooth curves to camera input we have developed several direct (i.e. noniterative) methods for fitting a shape (line, circle, conic, cubic, plane, sphere, quadric, etc.) to a set of points, namely exact fit, simple fit, spherical fit, and blend fit. These methods are all dimension-independent, being just as suitable for 3D surfaces as for the 2D curves they were originally developed for.

- Paláncz B, Molnár B (2012): Fitting sphere to quantized depth information, Wolfram Library, 2012-09-20
algebrai, geometriai, iránymenti illesztés
neurális hálózat, Gröbner bázis, RANSAC

Fitting sphere to quantized depth information

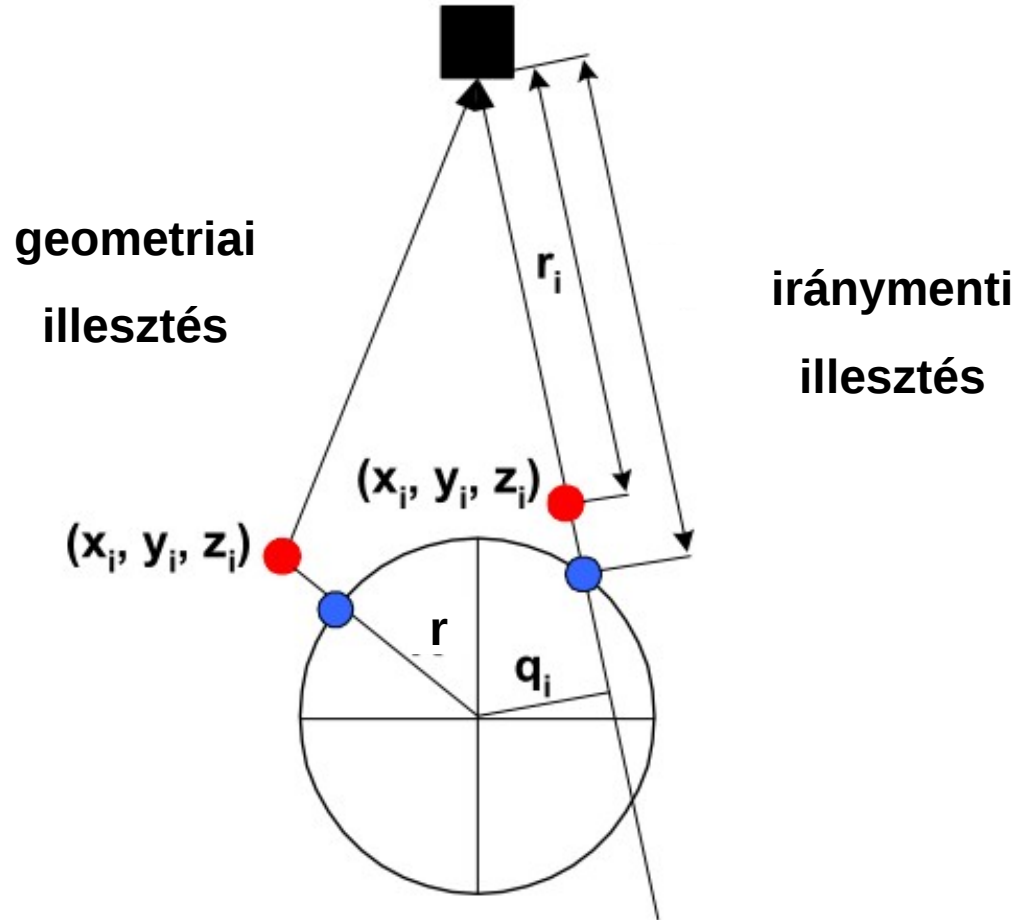
B. Paláncz and B. Molnár

Department of Photogrammetry and Geoinformatics,
Budapest University of Technology and Economy
1521 Budapest, Hungary

Abstract

This notebook presents different techniques to estimate radius and position of a sphere in case of quantized depth information obtained from low resolution sensors like Microsoft Kinect XBOX. First algebraic, geometric and directional least squares estimations were applied to the quantized data directly. Then two techniques Self-Organizing Map (SOM) and RANdom SAMple Consensus (RANSAC) were employed as preprocessing methods to smooth and reduce quantized data. To solve the resulted nonlinear algebraic systems Gröbner basis with Gauss-Jacobi method as global method as well as Newton method with pseudoinverse and direct minimization as local methods applying the result of the algebraic method as initial guess have been used. In order to decrease the computation time parallel computation on multi-core machine could be utilized. According to this case study all of these methods can be accepted from engineering point of view, although the geometrical approach with initial condition based on the solution of the algebraic method was proved to be the most effective.

Geometriai és iránymenti illesztés



Geometriai illesztés n pontra

- hiba i -edik pontban

$$v_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} - r, \quad i = 1, \dots, n$$

- ezért a minimalizációs feladat nemlineáris LKN feladatra vezet:

$$\sum_{i=1}^n v_i^2 = \left(\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} - r \right)^2 \Rightarrow \min$$

- A meghatározandó paraméterek: x_0, y_0, z_0, r^2
- Linearizálás és iteráció szükséges!

Geometriai illesztés Gröbner-bázissal

- „egyszerű” megoldás az a paraméterre (x_0)

$$aG = - \frac{\text{Coefficient}[grba[[1]], a, 0]}{\text{Coefficient}[grba[[1]], a, 1]} // \text{Simplify} \quad \{\mathbf{x} \rightarrow \eta_1, \mathbf{y} \rightarrow \xi_1, \mathbf{z} \rightarrow \chi_1\}$$

$$\begin{aligned} & (-\eta_2^2 \xi_3 \chi_1 - \xi_2^2 \xi_3 \chi_1 + \xi_2 \xi_3^2 \chi_1 + \eta_2^2 \xi_4 \chi_1 + \xi_2^2 \xi_4 \chi_1 - \xi_3^2 \xi_4 \chi_1 - \xi_2 \xi_4^2 \chi_1 + \xi_3 \xi_4^2 \chi_1 + \\ & \eta_1^2 \xi_3 \chi_2 + \xi_1^2 \xi_3 \chi_2 - \xi_1 \xi_3^2 \chi_2 - \eta_1^2 \xi_4 \chi_2 - \xi_1^2 \xi_4 \chi_2 + \xi_3^2 \xi_4 \chi_2 + \xi_1 \xi_4^2 \chi_2 - \xi_3 \xi_4^2 \chi_2 + \xi_3 \chi_1^2 \chi_2 - \\ & \xi_4 \chi_1^2 \chi_2 - \xi_3 \chi_1 \chi_2^2 + \xi_4 \chi_1 \chi_2^2 + \eta_2^2 \xi_1 \chi_3 - \eta_1^2 \xi_2 \chi_3 - \xi_1^2 \xi_2 \chi_3 + \xi_1 \xi_2^2 \chi_3 + \eta_1^2 \xi_4 \chi_3 - \eta_2^2 \xi_4 \chi_3 + \\ & \xi_1^2 \xi_4 \chi_3 - \xi_2^2 \xi_4 \chi_3 - \xi_1 \xi_4^2 \chi_3 + \xi_2 \xi_4^2 \chi_3 - \xi_2 \chi_1^2 \chi_3 + \xi_4 \chi_1^2 \chi_3 + \xi_1 \chi_2^2 \chi_3 - \xi_4 \chi_2^2 \chi_3 + \xi_2 \chi_1 \chi_3^2 - \\ & \xi_4 \chi_1 \chi_3^2 - \xi_1 \chi_2 \chi_3^2 + \xi_4 \chi_2 \chi_3^2 + \eta_4^2 (\xi_3 (\chi_1 - \chi_2) + \xi_1 (\chi_2 - \chi_3) + \xi_2 (-\chi_1 + \chi_3)) - \eta_2^2 \xi_1 \chi_4 + \\ & \eta_1^2 \xi_2 \chi_4 + \xi_1^2 \xi_2 \chi_4 - \xi_1 \xi_2^2 \chi_4 - \eta_1^2 \xi_3 \chi_4 + \eta_2^2 \xi_3 \chi_4 - \xi_1^2 \xi_3 \chi_4 + \xi_2^2 \xi_3 \chi_4 + \xi_1 \xi_3^2 \chi_4 - \xi_2 \xi_3^2 \chi_4 + \\ & \xi_2 \chi_1^2 \chi_4 - \xi_3 \chi_1^2 \chi_4 - \xi_1 \chi_2^2 \chi_4 + \xi_3 \chi_2^2 \chi_4 + \xi_1 \chi_3^2 \chi_4 - \xi_2 \chi_3^2 \chi_4 - \xi_2 \chi_1 \chi_4^2 + \xi_3 \chi_1 \chi_4^2 + \xi_1 \chi_2 \chi_4^2 - \\ & \xi_3 \chi_2 \chi_4^2 - \xi_1 \chi_3 \chi_4^2 + \xi_2 \chi_3 \chi_4^2 + \eta_3^2 (\xi_4 (-\chi_1 + \chi_2) + \xi_2 (\chi_1 - \chi_4) + \xi_1 (-\chi_2 + \chi_4)) \Big) / \\ & (2 (-\eta_2 \xi_3 \chi_1 + \eta_2 \xi_4 \chi_1 + \eta_1 \xi_3 \chi_2 - \eta_1 \xi_4 \chi_2 + \eta_2 \xi_1 \chi_3 - \eta_1 \xi_2 \chi_3 + \eta_1 \xi_4 \chi_3 - \eta_2 \xi_4 \chi_3 + \\ & \eta_4 (\xi_3 (\chi_1 - \chi_2) + \xi_1 (\chi_2 - \chi_3) + \xi_2 (-\chi_1 + \chi_3)) - \eta_2 \xi_1 \chi_4 + \eta_1 \xi_2 \chi_4 - \\ & \eta_1 \xi_3 \chi_4 + \eta_2 \xi_3 \chi_4 + \eta_3 (\xi_4 (-\chi_1 + \chi_2) + \xi_2 (\chi_1 - \chi_4) + \xi_1 (-\chi_2 + \chi_4))) \Big) \end{aligned}$$

Therefore the solution for parameter a is simple

Íránymenti illesztés n pontra

- i -edik pont távolsága a kamerától (x_c, y_c, z_c)

$$r_i = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2 + (z_c - z_i)^2}, \quad i = 1, \dots, n$$

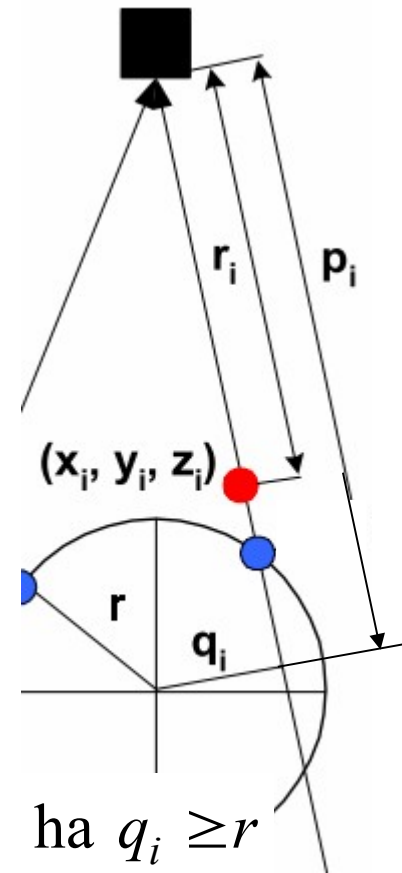
$$p_i = x_0 u_i + y_0 v_i + z_0 w_i$$

$$q_i = \sqrt{(v_i z_0 - w_i y_0)^2 + (w_i x_0 - u_i z_0)^2 + (u_i y_0 - v_i x_0)^2}$$

ahol: $u_i = \frac{x_i}{r_i}, v_i = \frac{y_i}{r_i}, w_i = \frac{z_i}{r_i}$

- A nemlineáris egyenletrendszer:

$$\left(p_i - \sqrt{r^2 - q_i^2} - r_i \right)^2 = 0, \quad \text{ha } q_i < r \quad (p_i - r_i)^2 + (q_i - r)^2 = 0, \quad \text{ha } q_i \geq r$$



Algebrai illesztés 4 pontra

- Mindegyik pontra illeszkedik az r sugarú gömb:

$$\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} - r = 0, \quad i = 1, \dots, 4$$

ezért mindegyik pontra

$$\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} + r = 2r = c = \text{áll.}$$

- A két egyenletet összeszorozva

$$(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 - r^2 = 0$$

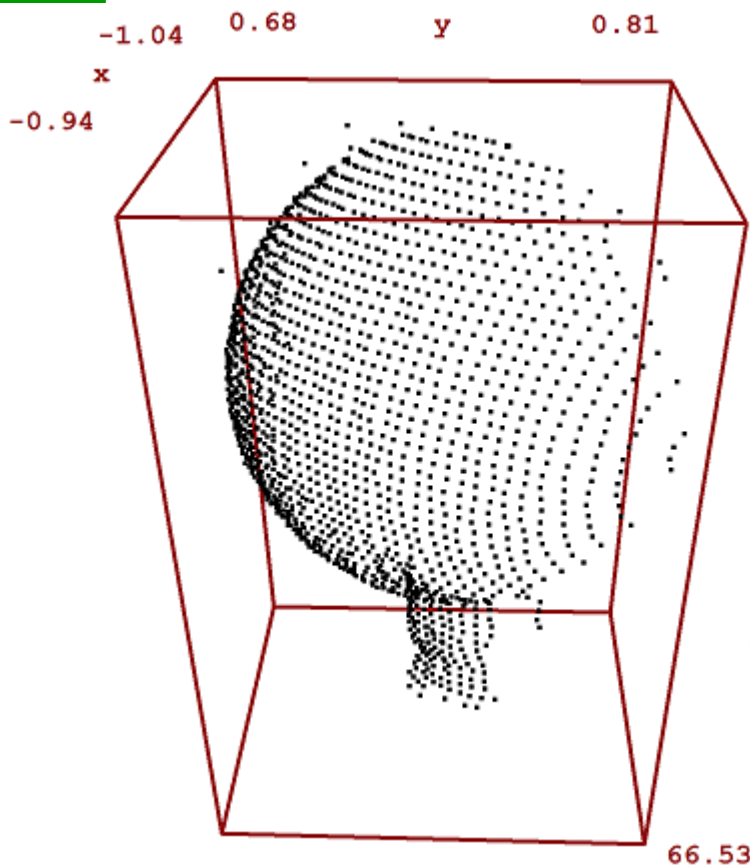
- Új α ismeretlent bevezetve r helyett *lineáris* $\alpha = x_0^2 + y_0^2 + z_0^2 - r^2$ egyenletrendszerrel kell megoldani:

$$x_i^2 + y_i^2 + z_i^2 - 2x_0x_i - 2y_0y_i - 2z_0z_i + \alpha = 0, \quad i = 1, \dots, 4$$

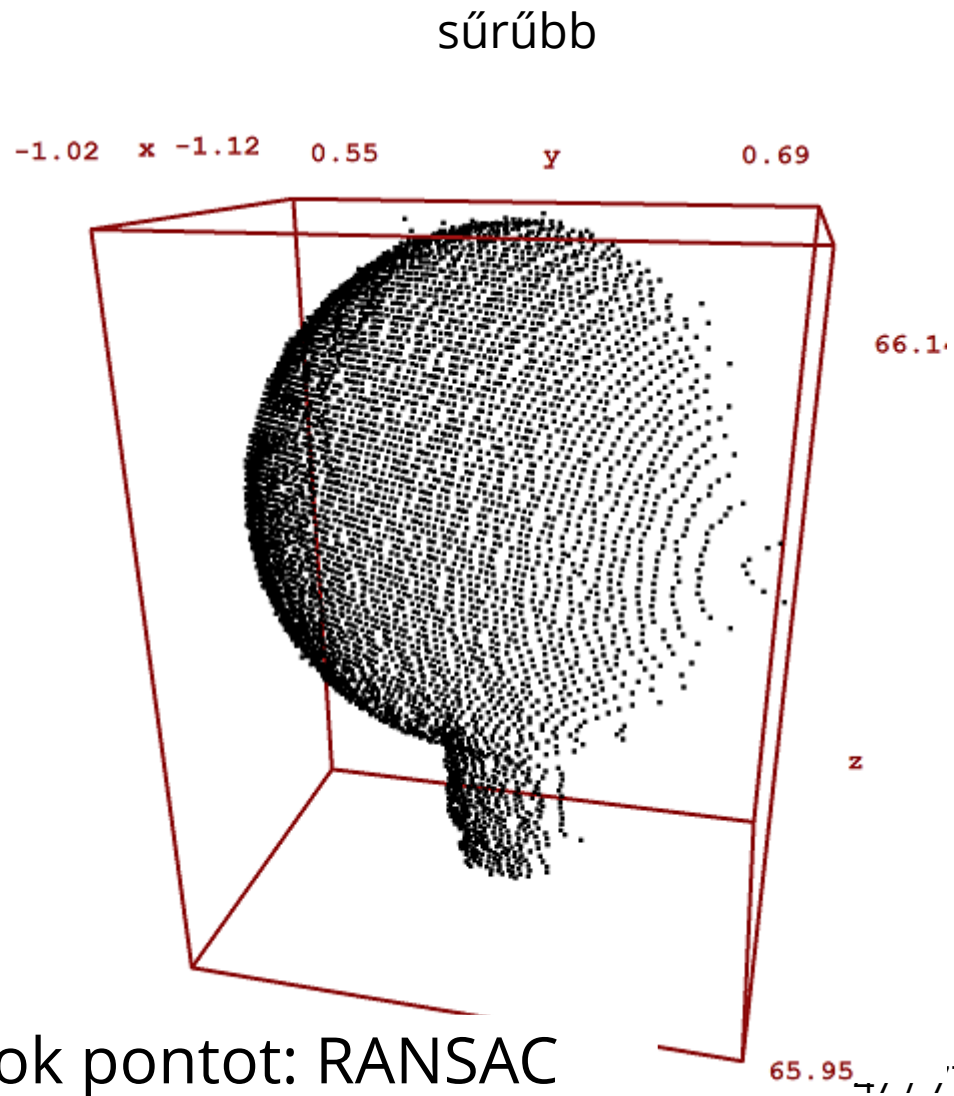
4 pont alapján szinguláris eset

- Ha a 4 pont egy síkba esik, nincs megoldás (r gömbsugár végtelen nagy)
- A gyakorlatban a lineáris egyenletrendszer kondíciós számát korlátozzuk, pl. 1000-re

Adatok



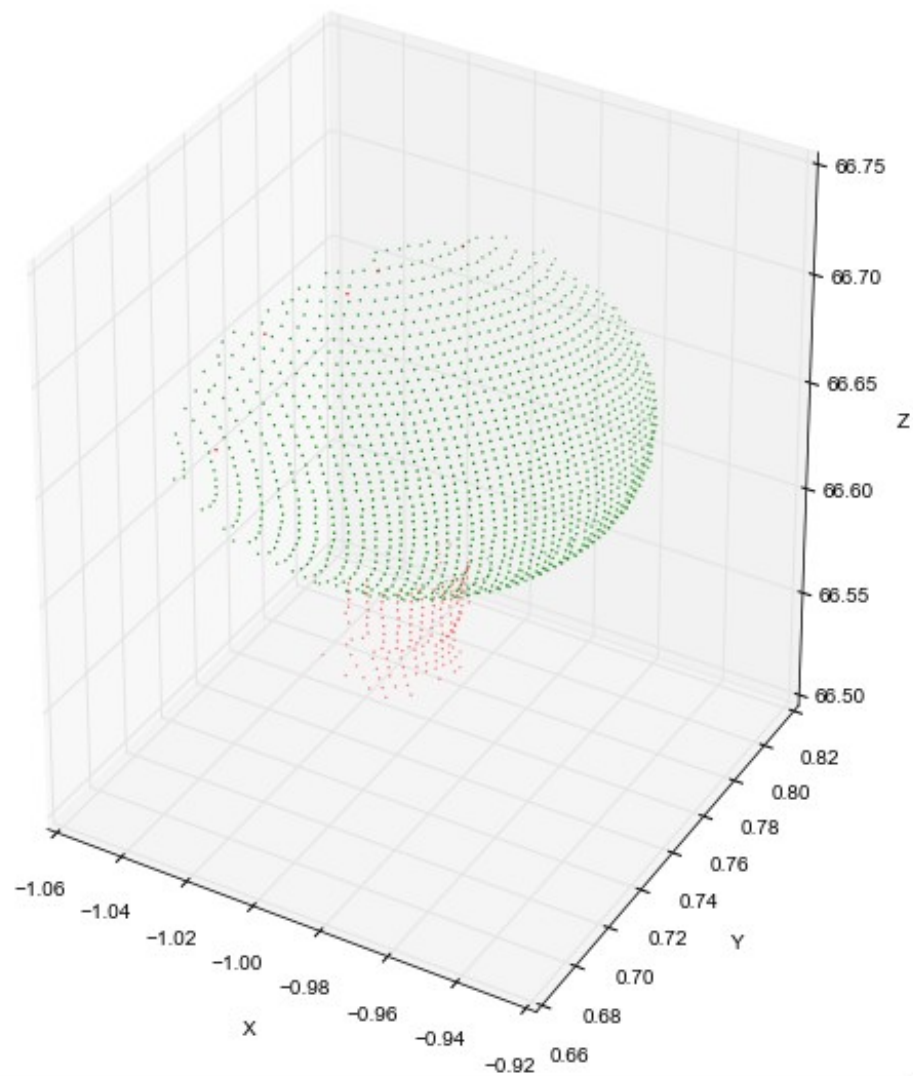
ritkább



sűrűbb

a „nyél” miatt eleve ki kell szűrni sok pontot: RANSAC

RANSAC becslés



Egyenes körhenger illesztése

- Hány pont szükséges a feladat megoldásához?
- Hogyan lehet meghatározni a henger paramétereit?
- Próba számítások

A probléma szakirodalma

- Chaperon (2002)
- Devillers et al. (2003)
- Lichtblau (2006)
- Beder, Förstner (2006)
- Zsombor-Murray, El Fashny (2006)
- Normálvektorokkal együtt:
Rabbani, Heuvel (2005)

Chaperon (2002)

- francia nyelven írt PhD értekezés
- algebrai megoldást ad: 1 harmadfokú, 2 másodfokú nemlineáris egyenlet a henger (x,y,z) normálvektorára
- páros számú, max. 6 megoldás 5 pontból

$$\begin{aligned}
 &x^2(M_{2345}(0,0)) \\
 &+xy(M_{2345}(0,1) + M_{2345}(1,0)) \\
 &+xz(M_{2345}(0,2) + M_{2345}(2,0)) \\
 &+y^2(M_{2345}(1,1)) \\
 &+yz(M_{2345}(1,2) + M_{2345}(2,1)) \\
 &+z^2(M_{2345}(2,2)) \\
 &+1(-c_{2345}) \\
 &= 0 \qquad x^2 + y^2 + z^2 - 1 = 0
 \end{aligned}$$

$$\begin{aligned}
 &x^3(M_{234}^{yz}(0,0)) \\
 &+x^2y(M_{234}^{yz}(0,1) + M_{234}^{yz}(1,0) + M_{234}^{xz}(0,0)) \\
 &+x^2z(M_{234}^{yz}(0,2) + M_{234}^{yz}(2,0) + M_{234}^{xy}(0,0)) \\
 &+xy^2(M_{234}^{yz}(1,1) + M_{234}^{xz}(0,1) + M_{234}^{xy}(1,0)) \\
 &+xyz(M_{234}^{yz}(1,2) + M_{234}^{yz}(2,1) + M_{234}^{xz}(0,2) + M_{234}^{xz}(2,0) + M_{234}^{xy}(0,1) + M_{234}^{xy}(1,0)) \\
 &+xz^2(M_{234}^{yz}(2,2) + M_{234}^{xy}(0,2) + M_{234}^{xy}(2,0)) \\
 &+y^3(M_{234}^{xz}(1,1)) \\
 &+y^2z(M_{234}^{xz}(1,2) + M_{234}^{xz}(2,1) + M_{234}^{xy}(1,1)) \\
 &+yz^2(M_{234}^{xz}(2,2) + M_{234}^{xy}(1,2) + M_{234}^{xy}(2,1)) \\
 &+z^3(M_{234}^{xy}(2,2)) \\
 &+x(-V_{234} \cdot X) \\
 &+y(-V_{234} \cdot Y) \\
 &+z(-V_{234} \cdot Z) \\
 &= 0
 \end{aligned}$$

Chaperon (2002)

- működik, de algebrai rendszer (pl. Maxima) szükséges a nemlineáris egyenletrendszer megoldásához

Chaperon módszere
koordináták sorvektorokban

```
>function k(x,y):=(x+y).(x-y)'  
>function K(x,y):=(x+y)'.(x-y)  
>function D(Xi,Xj,Xk,Xl):=det((Xi-Xl)'|(Xj-Xl)'|(Xk-Xl)')  
>function M(X1,X2,X3,X4,X5):=-D(X3,X4,X5,X1)*K(X2,X1)+D(X2,X4,X5,X1)*K(X3,X1)  
>function c(X1,X2,X3,X4,X5):=-D(X3,X4,X5,X1)*k(X2,X1)+D(X2,X4,X5,X1)*k(X3,X1)  
>function mx(X1,X2,X3,X4):=d(X3[2:3],X4[2:3],X1[2:3])*K(X2,X1)-d(X2[2:3],X4[2:3],X1[2:3])*K(X3,X1)  
>function my(X1,X2,X3,X4):=-d(X3[[1,3]],X4[[1,3]],X1[[1,3]])*K(X2,X1)+d(X2[[1,3]],X4[[1,3]],X1[[1,3]])*K(X3,X1)  
>function mz(X1,X2,X3,X4):=d(X3[1:2],X4[1:2],X1[1:2])*K(X2,X1)-d(X2[1:2],X4[1:2],X1[1:2])*K(X3,X1)  
>function V(X1,X2,X3,X4)  
$ r1=[d(X3[2:3],X4[2:3],X1[2:3]),-d(X2[2:3],X4[2:3],X1[2:3]),d(X2[2:3],X3[2:3],X4[2:3],X1[2:3]),  
$ r2=[-d(X3[[1,3]],X4[[1,3]],X1[[1,3]]),d(X2[[1,3]],X4[[1,3]],X1[[1,3]]),-c(X1,X2,X3,X4,X5),  
$ r3=[d(X3[1:2],X4[1:2],X1[1:2]),-d(X2[1:2],X4[1:2],X1[1:2]),d(X2[1:2],X3[1:2],X4[1:2],X1[1:2]),  
$ return (r1_r2_r3).[k(X2,X1);k(X3,X1);k(X4,X1)];  
$endfunction  
  
>X1=pts[1];X2=pts[2];X3=pts[3];X4=pts[4];X5=pts[5];  
>M2345=M(X1,X2,X3,X4,X5)
```

```
[[X = 1.104464235956473 - 0.047484743681226 I,  
y = 0.39956281153479 I + 0.053889193637358,  
z = - 0.27164404158493 I - 0.11379996869441],  
[x = 0.047484743681226 I + 1.104464235956473,  
y = 0.053889193637358 - 0.39956281153479 I,  
z = 0.27164404158493 I - 0.11379996869441],  
[x = - 0.047484743681234 I - 1.104464235956473,  
y = 0.39956281153478 I - 0.053889193637363,  
z = 0.11379996869443 - 0.27164404158494 I],  
[x = 0.047484743681234 I - 1.104464235956473,  
y = - 0.39956281153478 I - 0.053889193637363,  
z = 0.27164404158494 I + 0.11379996869443],  
[x = 0.3216473791695, y = 0.015438368408223, z = - 0.94673366834171],  
[x = 0.16785038083414 - 0.28955016160991 I,  
y = 0.02315896492574 I - 0.99252867360514,  
z = - 0.21066862498019 I - 0.3398087477115],  
[x = 0.28955016160991 I + 0.16785038083414,  
y = - 0.02315896492574 I - 0.99252867360514,  
z = 0.21066862498019 I - 0.33980874771149],  
[x = - 0.28955016160991 I - 0.16785038083414,  
y = 0.02315896492574 I + 0.99252867360514,  
z = 0.3398087477115 - 0.21066862498019 I],  
[x = 0.28955016160991 I - 0.16785038083414,  
y = 0.99252867360514 - 0.02315896492574 I,  
z = 0.21066862498019 I + 0.3398087477115],  
[x = - 0.3216473791695, y = - 0.015438368408223,  
z = 0.94673366834171], [x = - 0.011383768705944,  
y = - 0.36145926589078, z = - 0.93231841526045],  
[x = 0.011383768705944, y = 0.36145926589078, z = 0.93231841526045]]
```

- Devillers et al. (2003)
 - Lichtblau (2006)
- számítógépes
algebrai megoldást
adnak
(Mathematica)

Solving simultaneously for all roots of the cylinder parameter equations

An obvious drawback to the methods seen thus far is the need for good initial guesses. We may take advantage of the fact that the equations are all polynomial and instead use a global solver suitable for such systems. We demonstrate below the utility of this approach. In order to have simpler equations for visual purposes will work with an example comprised of integer coordinates in the range $(-10,10)$. To further simplify matters we will solve for the square of the radius (this will avoid solutions with negative values for r as well as cut in half the number of complex valued solutions). An example problem with pseudorandom coordinates in the indicated range gave rise to the polynomials shown below. The points we chose to lie on the cylinder(s) are:

$$(7,9,8), (8,-4,-10), (-4,1,4), (-9,-9,-10), (-7,-10,-10) \quad (5)$$

The five corresponding polynomials we set to zero are as below.

$$\begin{aligned} & (145 - 126a + 113a^2 - 18b + 14ab + b^2 - 112c - 144ac + 16abc + 130c^2 - 18bc^2 + \\ & \quad b^2c^2 - 16d - 16a^2d + 14cd + 18acd - 2abcd + d^2 + a^2d^2 - rsqr - a^2rsqr - c^2rsqr, \\ & 116 + 64a + 164a^2 + 8b + 16ab + b^2 + 160c - 80ac - 20abc + 80c^2 + 8bc^2 + b^2c^2 + \\ & \quad 20d + 20a^2d + 16cd - 8acd - 2abcd + d^2 + a^2d^2 - rsqr - a^2rsqr - c^2rsqr, \\ & 17 + 8a + 32a^2 - 2b - 8ab + b^2 + 32c - 8ac + 8abc + 17c^2 - 2bc^2 + b^2c^2 - \\ & \quad 8d - 8a^2d - 8cd + 2acd - 2abcd + d^2 + a^2d^2 - rsqr - a^2rsqr - c^2rsqr, \\ & 181 - 162a + 181a^2 + 18b - 18ab + b^2 - 180c - 180ac - 20abc + 162c^2 + 18bc^2 + \\ & \quad b^2c^2 + 20d + 20a^2d - 18cd - 18acd - 2abcd + d^2 + a^2d^2 - rsqr - a^2rsqr - c^2rsqr, \\ & 200 - 140a + 149a^2 + 20b - 14ab + b^2 - 140c - 200ac - 20abc + 149c^2 + 20bc^2 + \\ & \quad b^2c^2 + 20d + 20a^2d - 14cd - 20acd - 2abcd + d^2 + a^2d^2 - rsqr - a^2rsqr - c^2rsqr) \end{aligned} \quad (6)$$

We mention that this system is not substantially simpler to solve numerically than the preceding one; its main virtue for our purposes is that it is more concise to print. In contrast to local methods, which, as we saw, may fail to get a particular solution, it turns out to be computationally straightforward to obtain all solutions to this system. We do this in *Mathematica* with the `NSolve` function. It uses a hybrid symbolic-numeric technique to efficiently find all roots. Details of this technology are discussed in [10] [11] [27]. The basic idea is to compute a numeric Gröbner basis and then do an eigendecomposition of a certain matrix formed therefrom. Our solution set is as below.

$$\begin{aligned} & \{a \rightarrow -1.03253 + 0.760393i, b \rightarrow 6.11349 - 3.37419i, c \rightarrow -0.322931 - 1.37768i, \\ & \quad d \rightarrow -0.295427 + 6.8709i, rsqr \rightarrow 344.25 + 23.8554i\}, \{a \rightarrow -1.03253 - 0.760393i, \\ & \quad b \rightarrow 6.11349 + 3.37419i, c \rightarrow -0.322931 + 1.37768i, d \rightarrow -0.295427 - 6.8709i, rsqr \rightarrow 344.25 - 23.8554i\}, \\ & \{a \rightarrow 0.151635, b \rightarrow -1.25748, c \rightarrow 1.58897, d \rightarrow -6.45046, rsqr \rightarrow 83.0554\}, \\ & \{a \rightarrow 30.9362, b \rightarrow 93.172, c \rightarrow 37.1186, d \rightarrow 92.7034, rsqr \rightarrow 198.258\}, \\ & \{a \rightarrow 0.613253 - 0.359335i, b \rightarrow -4.49777 - 3.77132i, c \rightarrow 0.102934 + 0.159852i, \\ & \quad d \rightarrow -1.56979 + 2.23275i, rsqr \rightarrow 57.5606 + 13.7534i\}, \{a \rightarrow 0.613253 + 0.359335i, \\ & \quad b \rightarrow -4.49777 + 3.77132i, c \rightarrow 0.102934 - 0.159852i, d \rightarrow -1.56979 - 2.23275i, rsqr \rightarrow 57.5606 - 13.7534i\} \end{aligned}$$

Beder, Förstner (2006)

- 8,7,6,5 pontos megoldás
- az 5 pontos megoldás két hatodfokú polinom *közös gyökeinek* meghatározását kívánja

Direct Solutions for Computing Cylinders from Minimal Sets of 3D Points

Christian Beder and Wolfgang Förstner

Institute for Photogrammetry
Bonn University, Germany
{ `beder,wf` } @ipb.uni-bonn.de

Beder, Förstner (2006)

- (a,b) -re két hatodfokú egyenlet (forgatási kvaternió)

$$\begin{bmatrix} X_1'^2(a,b) + Y_1'^2(a,b) & -2X_1'(a,b) & -2Y_1'(a,b) & 1 \\ X_2'^2(a,b) + Y_2'^2(a,b) & -2X_2'(a,b) & -2Y_2'(a,b) & 1 \\ X_3'^2(a,b) + Y_3'^2(a,b) & -2X_3'(a,b) & -2Y_3'(a,b) & 1 \\ X_4'^2(a,b) + Y_4'^2(a,b) & -2X_4'(a,b) & -2Y_4'(a,b) & 1 \\ X_5'^2(a,b) + Y_5'^2(a,b) & -2X_5'(a,b) & -2Y_5'(a,b) & 1 \end{bmatrix} \begin{bmatrix} 1 \\ s \\ t \\ u \end{bmatrix} = H(a,b) \begin{bmatrix} 1 \\ s \\ t \\ u \end{bmatrix} = \mathbf{0} \quad (9)$$

Each of the five 4×4 -submatrices of $H(a,b)$ must therefore be singular, i.e. have a zero determinant. The numerators of this five determinants are bivariate polynomials of 6-th degree in the two variables a and b , hence are expressible as

$$p_l(a,b) = [1 \ a \ a^2 \ a^3 \ a^4 \ a^5 \ a^6] \mathbf{G}_l [1 \ b \ b^2 \ b^3 \ b^4 \ b^5 \ b^6]^\top = 0, \quad l = 1, \dots, 5 \quad (10)$$

Their common roots need to be calculated in order to obtain the cylinder axis direction.

Beder, Förstner (2006)

- gyökmeghatározás Bernstein-polinomokkal)

The efficiency of the root-finder: The performance of the five-point-method mainly depends on the efficiency for finding the common roots of the five polynomial equations yielding the axis direction of the cylinder. In figure 1, left, the logarithm of the sum of the five squared polynomials is shown for a typical point configuration. The standard Gauss-Newton-Method for finding the four roots would search this cost function.

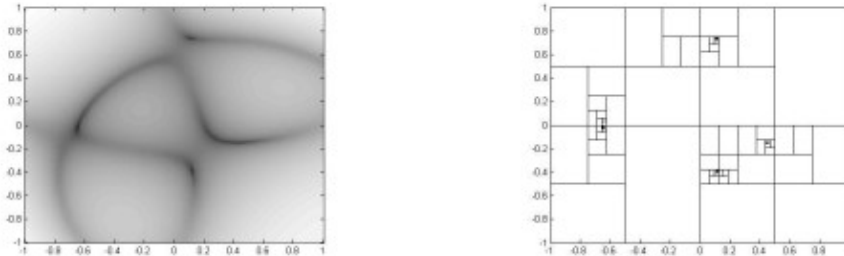


Fig. 1. Left: Logarithm of the sum of the five squared polynomials for a typical point configuration. The minima of this function would be searched with the standard Gauss-Newton-Method. Right: The bisections required with the Bernstein-Method for tracking down the roots of the same five polynomials as depicted in the figure left.

Zsombor-Murray, El Fashny (2006)

- ez a megoldás a legmegfelelőbb számunkra
- adott koordináta-rendszerbe transzformálja az 5 pontot
- alapegyenletből kiindulva levezet végül egy hatodfokú polinomot a henger irányvektorának egyik komponensére
- a másik komponens lineáris egyenletből kiszámítható

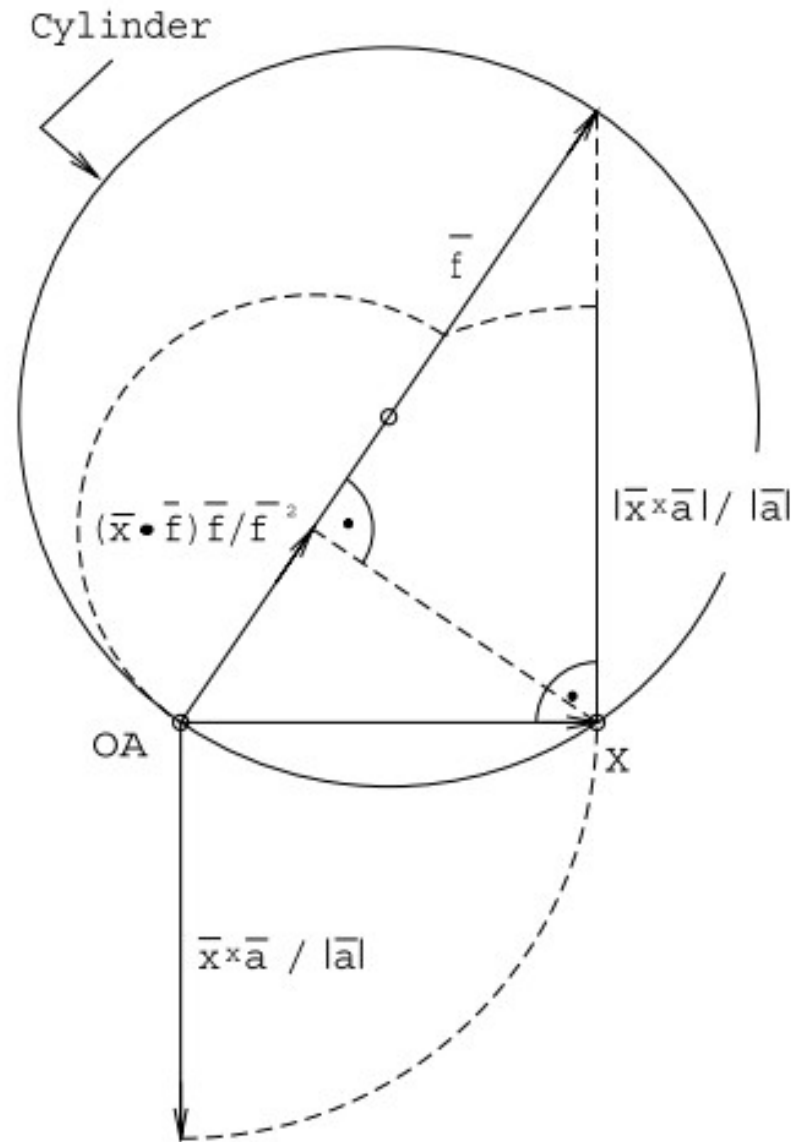
MECH 576 Geometry in Mechanics

November 18, 2009

A Cylinder of Revolution on Five Points

Alapegyenlet

$$(\mathbf{x} \times \mathbf{a})^2 - 2a^2(\mathbf{x} \cdot \mathbf{f}) = 0$$



- további egyenletek: $\mathbf{a} \cdot \mathbf{f} = 0, \mathbf{a}^2 = 1$

- 5 pont koordinátái:

$$\mathbf{o} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} p_1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ 0 \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}$$

- ismeretlenek:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

- egyenletrendszer:

$$a_1 f_1 + a_2 f_2 + a_3 f_3 = 0$$

$$p_1^2 a_3^2 + p_1 a_2^2 - 2p_1(a_1^2 + a_2^2 + a_3^2) f_1 = 0$$

$$q_2^2 a_3^2 + q_1^2 a_3^2 + (q_1 a_2 - q_2 a_1)^2 - 2(a_1^2 + a_2^2 + a_3^2)(q_1 f_1 + q_2 f_2) = 0$$

$$(r_2 a_3 - r_3 a_2)^2 + (r_3 a_1 - r_1 a_3)^2 + (r_1 a_2 - r_2 a_1)^2 - 2(a_1^2 + a_2^2 + a_3^2)(r_1 f_1 + r_2 f_2 + r_3 f_3) = 0$$

$$(s_2 a_3 - s_3 a_2)^2 + (s_3 a_1 - s_1 a_3)^2 + (s_1 a_2 - s_2 a_1)^2 - 2(a_1^2 + a_2^2 + a_3^2)(s_1 f_1 + s_2 f_2 + s_3 f_3) = 0 \quad 17$$

- hatodfokú polinom:

$$D_1 a_2^6 + D_2 a_2^5 + D_3 a_2^4 + D_4 a_2^3 + D_5 a_2^2 + D_6 a_2 + D_7 = 0$$

- együtthatók:

$$D_1 = d_1 d_2 - d_3^2, \quad D_2 = d_4 d_2 + d_1 d_5 - 2d_8 d_3, \quad D_3 = d_9 d_2 + d_4 d_5 + d_1 d_6 - 2d_7 d_3 - d_8^2$$

$$D_4 = d_9 d_5 + d_4 d_6 + d_1 d_{10} - 2d_{11} d_3 - 2d_7 d_8, \quad D_5 = d_9 d_6 + d_4 d_{10} + d_1 d_{12} - 2d_{11} d_8 - d_7^2$$

$$D_6 = d_9 d_{10} + d_4 d_{12} - 2d_{11} d_7, \quad D_7 = d_9 d_{12} - d_{11}^2$$

$$d_1 = c_1 b_2 - b_6 c_6, \quad d_2 = c_6 b_7 - b_2 c_2, \quad d_3 = c_1 b_7 - b_6 c_2, \quad d_4 = c_1 b_1 - b_8 c_6 - b_6 c_5$$

$$d_5 = c_5 b_7 + c_6 b_9 - b_1 c_1 - b_2 c_4, \quad d_6 = c_5 b_9 + c_6 b_5 - b_3 c_2 - b_1 c_4 - b_2 c_3, \quad d_7 = c_1 b_5 - b_8 c_4 - b_6 c_3$$

$$d_8 = c_1 b_9 - b_8 c_2 - b_6 c_4, \quad d_9 = c_1 b_3 - b_8 c_5, \quad d_{10} = c_5 b_5 + c_6 b_4 - b_3 c_4 - b_1 c_3$$

$$d_{11} = c_1 b_4 - b_8 c_3, \quad d_{12} = c_5 b_4 - b_3 c_3$$

- további együtthatók:

$$b_1 = 2q_2r_2(q_1 - r_1), \quad b_2 = q_2r_3(p_1 - 2q_1), \quad b_3 = q_2r_3(p_1 - 2r_1)$$

$$b_4 = q_2[r_1(r_1 - p_1) + r_2(r_2 - q_2)] + q_1r_2(p_1 - q_1)$$

$$b_5 = r_3[q_2(q_2 - 2r_2) + q_1(q_1 - p_1)], \quad b_6 = q_2^2r_3$$

$$b_7 = q_1r_3(q_1 - p_1), \quad b_8 = q_2[r_2(r_2 - q_2) + r_3^2], \quad b_9 = q_2[r_1(r_1 - p_1) + r_3^2] + q_1r_2(p_1 - q_1)$$

$$c_1 = q_2[r_2s_3(r_2 - q_2) + r_3s_2(q_2 - s_2) + r_3s_3(r_3 - s_3)]$$

$$c_2 = q_2r_3s_3(r_3 - s_3) + q_1^2(r_3s_2 - r_2s_3) + p_1[q_2(r_3s_1 - r_1s_3) + q_1(r_2s_3 - r_3s_2)] + q_2(r_1^2s_3 - r_3s_1^2)$$

$$c_3 = q_2[p_1(s_1r_3 - s_3r_1) - r_3(s_1^2 + s_2^2) + s_3(r_1^2 + r_2^2)] + (r_2s_3 - r_3s_2)[q_1(p_1 - q_1) - q_2^2]$$

$$c_4 = 2q_2r_3s_3(s_2 - r_2), \quad c_5 = 2q_2r_3s_3(s_1 - r_1), \quad c_6 = 2q_2[r_3s_2(s_1 - q_1) + r_2s_3(q_1 - r_1)]$$

- lineáris egyenlet a_1 -re:

$$[c_1(b_2a_2^2 + b_1a_2 + b_3) - (b_6a_2 + b_8)(c_6a_2 + c_5)]a_1 + [c_1(b_7a_2^3 + b_9a_2^2 + b_5a_2 + b_4) - (b_6a_2 + b_8)(c_2a_2^2 + c_4a_2 + c_3)] = 0$$

- számpélda:

The five given points

$$O(0, 0, 0), \quad P(3, 0, 0), \quad Q(2, 2, 0), \quad R(0, 2, 4), \quad S(2, 0, 3)$$

were coded into Eqs. 11, 10 and 9, in that order, and solved for a_2 with Eq. 8. The two real roots were used in Eq. 12 to get the corresponding values of a_1 . Of course $a_3 = 1$ in all cases.

a_3	a_2	a_1
1	0.3876994216	0.01221017201
1	$2.079377226 + 1.357288502i$...
1	$-1.32199443 + 0.3554442093i$...
1	-0.01630698077	-0.3397443430
1	$-1.32199443 - 0.3554442093i$...
1	$2.079377226 - 1.357288502i$...

- Hibásak a képletek! A példa a közölt képletekkel nem jön ki!
- Hol lehet a hiba?

- A b és c együtthatók numerikus ellenőrzése, hogy kielégítik-e a példa pontjaival a (7) egyenleteket:

$$\begin{aligned}(b_6a_2 + b_8)a_1^2 + (b_2a_2^2 + b_1a_2 + b_3)a_1 + (b_7a_2^3 + b_9a_2^2 + b_5a_2 + b_4) &= 0 \\ c_1a_1^2 + (c_6a_2 + c_5)a_1 + (c_2a_2^2 + c_4a_2 + c_3) &= 0\end{aligned}\tag{7}$$

ez rendben van

- A hiba a d/D együtthatókban van

- a D együtthatókat algebrailag levezetve kiderült, hogy a hiba a d_5 együtthatóban volt:

where

ez helyesen c2

$$\begin{aligned}
 d_1 &= c_1 b_2 - b_6 c_6, & d_2 &= c_6 b_7 - b_2 c_2, & d_3 &= c_1 b_7 - b_6 c_2, & d_4 &= c_1 b_1 - b_8 c_6 - b_6 c_5 \\
 d_5 &= c_5 b_7 + c_6 b_9 - b_1 c_1 - b_2 c_4, & d_6 &= c_5 b_9 + c_6 b_5 - b_3 c_2 - b_1 c_4 - b_2 c_3, & d_7 &= c_1 b_5 - b_8 c_4 - b_6 c_3 \\
 d_8 &= c_1 b_9 - b_8 c_2 - b_6 c_4, & d_9 &= c_1 b_3 - b_8 c_5, & d_{10} &= c_5 b_5 + c_6 b_4 - b_3 c_4 - b_1 c_3 \\
 & & & & d_{11} &= c_1 b_4 - b_8 c_3, & d_{12} &= c_5 b_4 - b_3 c_3 \quad (10)
 \end{aligned}$$

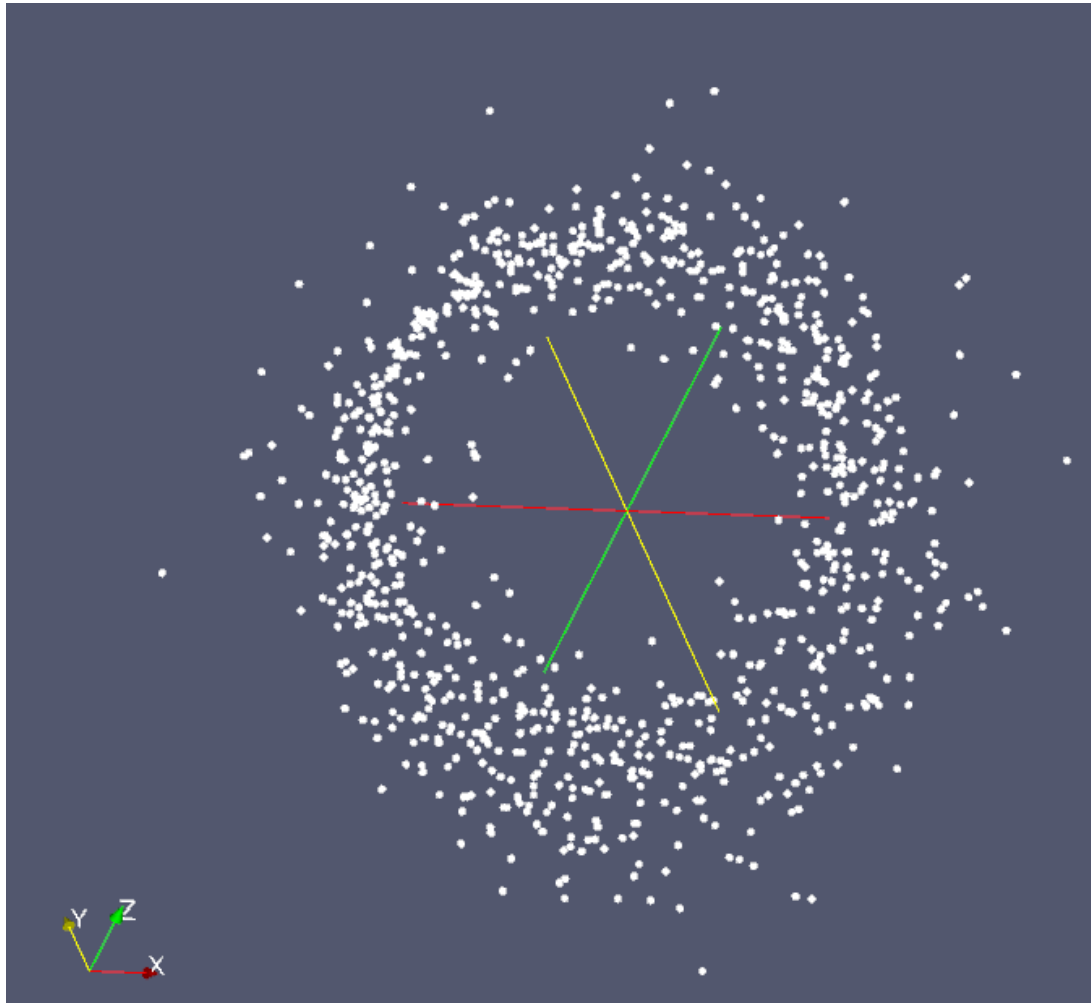
- A polinom gyökei most már kijöttek, de az f vektor még mindig nem volt jó!

- Újabb levezetés, most az (5)-ös egyenletekre
- A hiba a második egyenletben volt:

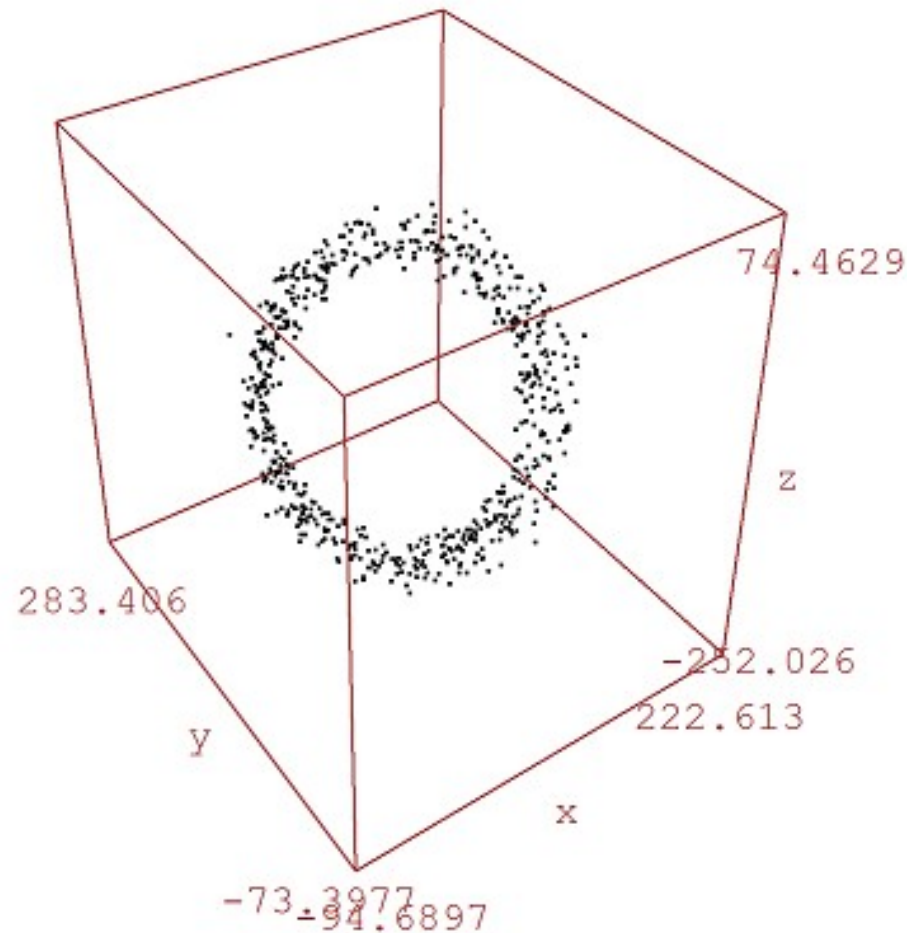
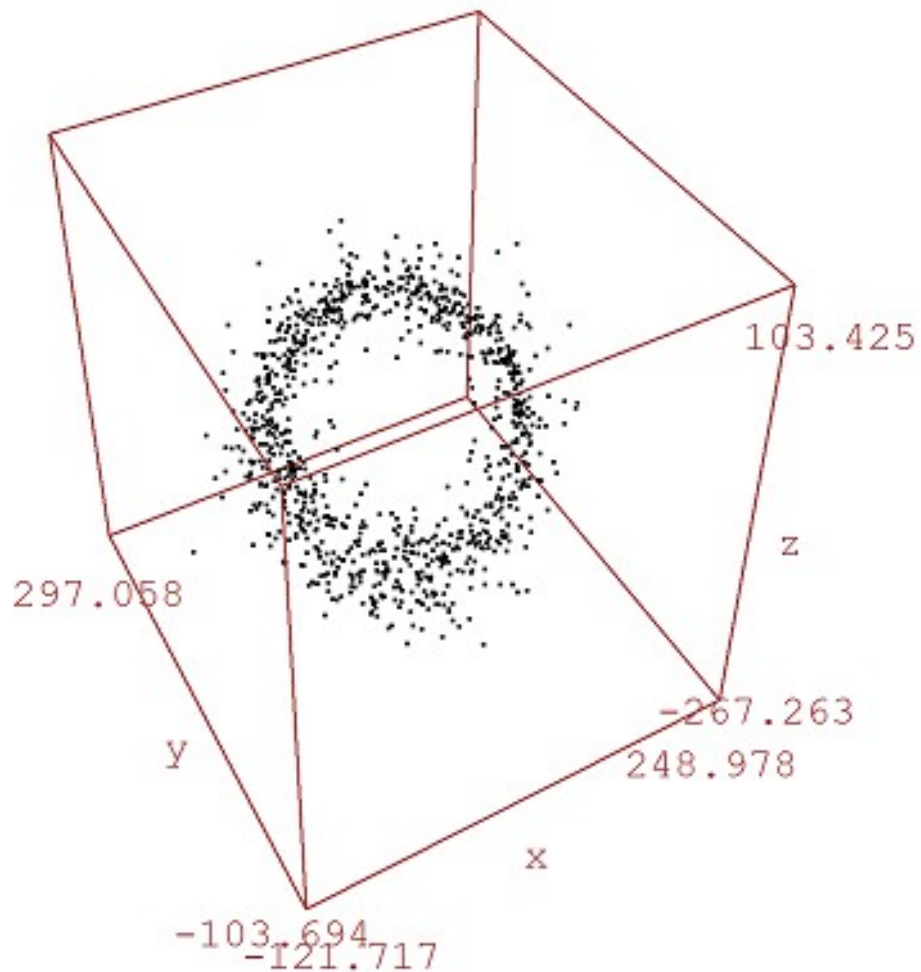
ez helyesen p1
négyzet

$$\begin{aligned}
 & a_1 f_1 + a_2 f_2 + a_3 f_3 = 0 \\
 & p_1^2 a_3^2 + p_1 a_2^2 - 2p_1(a_1^2 + a_2^2 + a_3^2) f_1 = 0 \\
 & q_2^2 a_3^2 + q_1^2 a_3^2 + (q_1 a_2 - q_2 a_1)^2 - 2(a_1^2 + a_2^2 + a_3^2)(q_1 f_1 + q_2 f_2) = 0 \\
 & (r_2 a_3 - r_3 a_2)^2 + (r_3 a_1 - r_1 a_3)^2 + (r_1 a_2 - r_2 a_1)^2 - 2(a_1^2 + a_2^2 + a_3^2)(r_1 f_1 + r_2 f_2 + r_3 f_3) = 0 \\
 & (s_2 a_3 - s_3 a_2)^2 + (s_3 a_1 - s_1 a_3)^2 + (s_1 a_2 - s_2 a_1)^2 - 2(a_1^2 + a_2^2 + a_3^2)(s_1 f_1 + s_2 f_2 + s_3 f_3) = 0 \quad (5)
 \end{aligned}$$

Teszt adatok Paraview-ben:



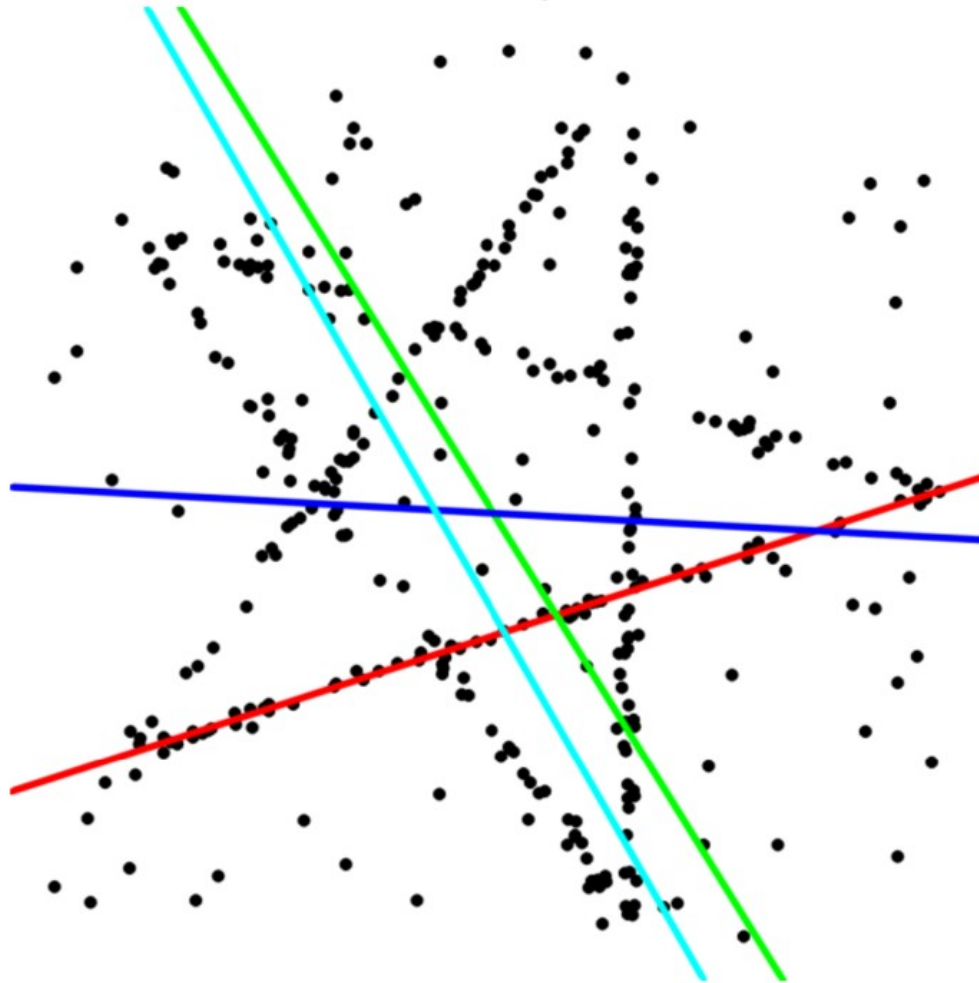
Eredeti és konszenzus adatok:



Több modell RANSAC becslése

- Az adatok nem csak *egy* modellt tartalmaznak:
több egyenes, ellipszis, kör, gömb, henger, stb.
található benne
- Hagyományos megközelítés nem működik:
LKN becslés
teljes LKN becslés
négyzetek legkisebb mediánja
RANSAC

Több modell esete



LKN becslés

teljes LKN becslés

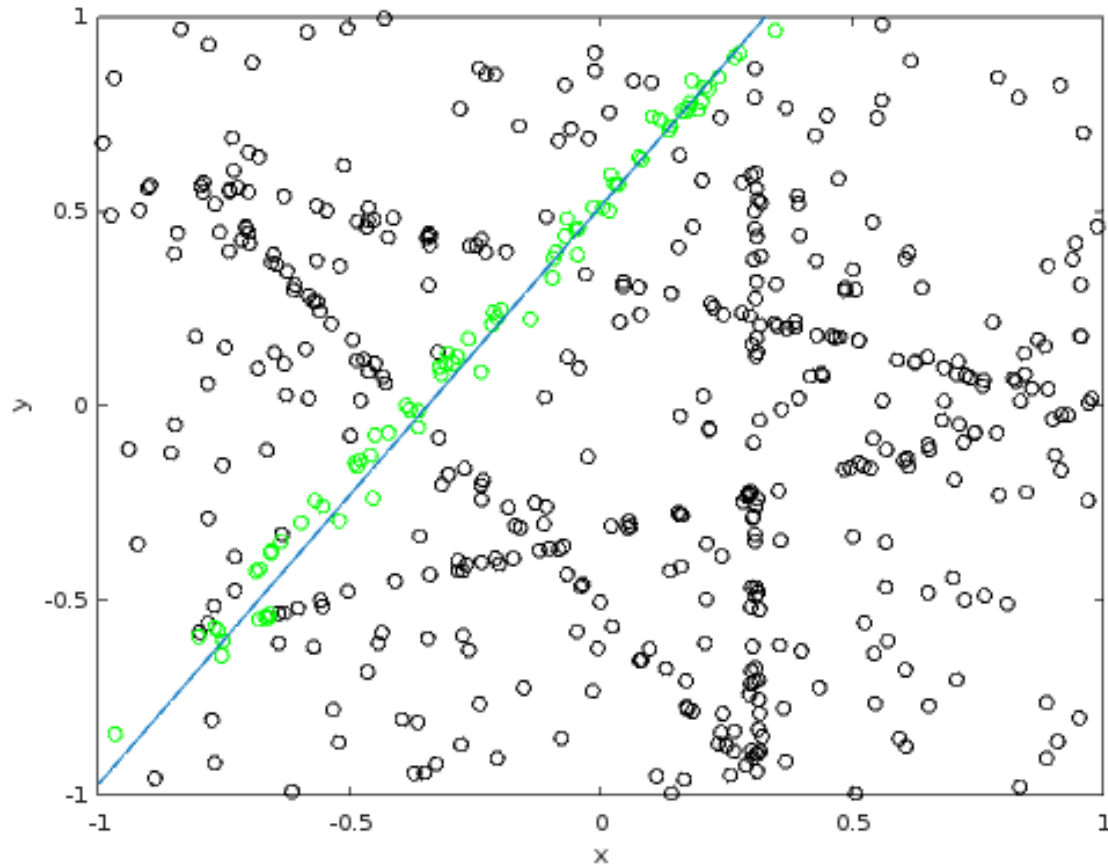
négyzetek

legkisebb mediánja

RANSAC

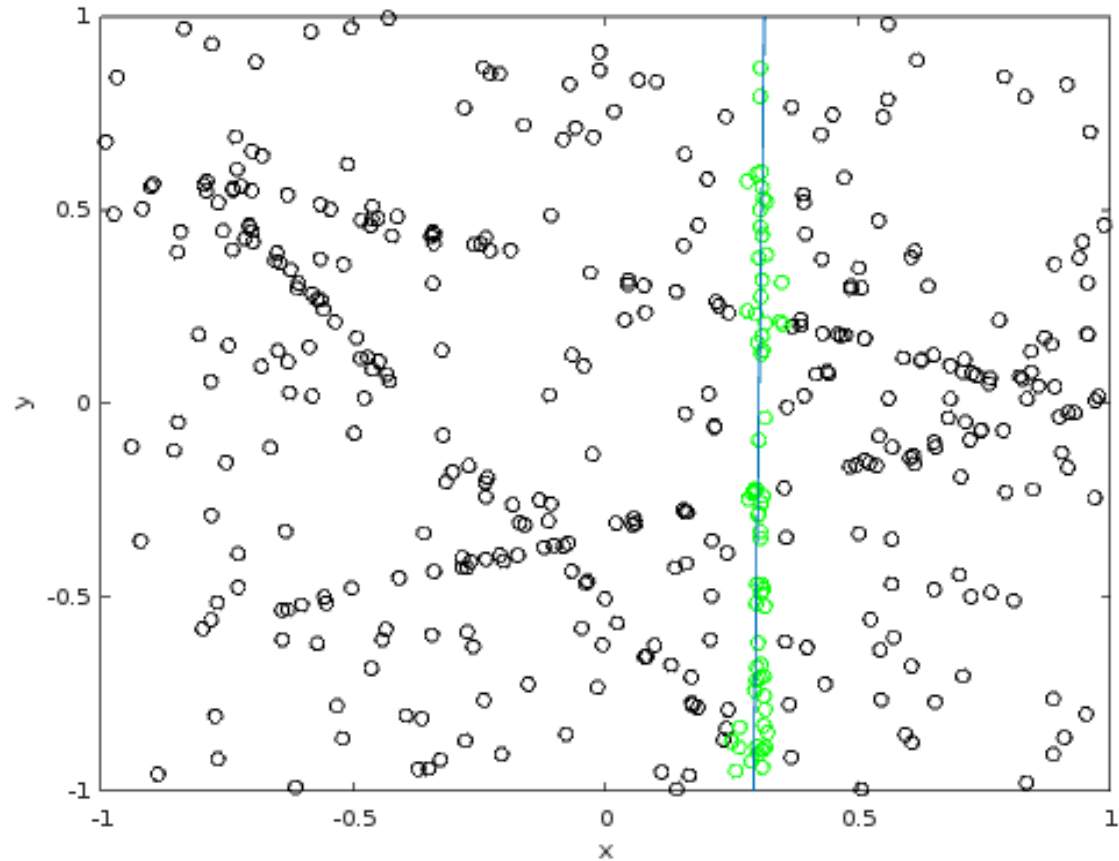
Több modell esete

LKN egyenes egyenlete: $y = 1.438x + 0.520$

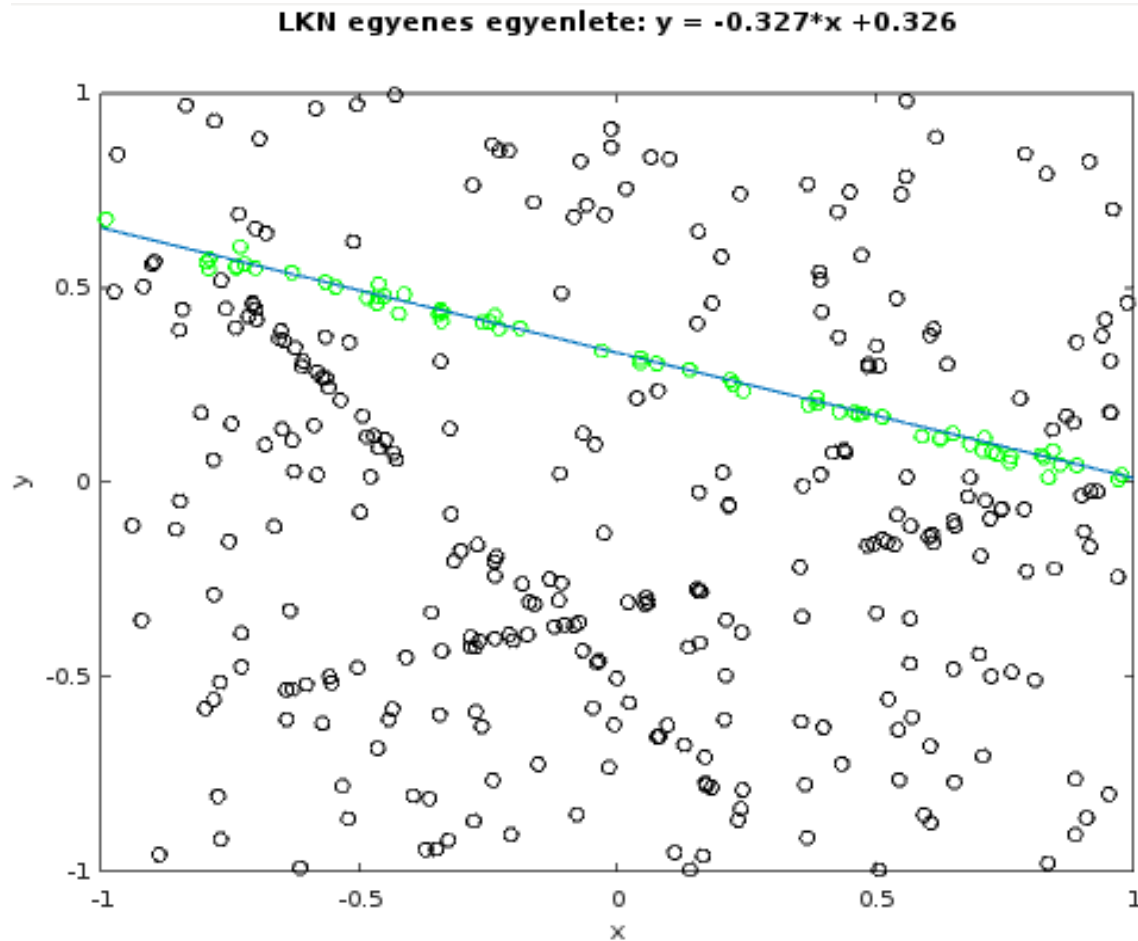


Több modell esete

LKN egyenes egyenlete: $y = 8.937 \cdot x - 2.938$

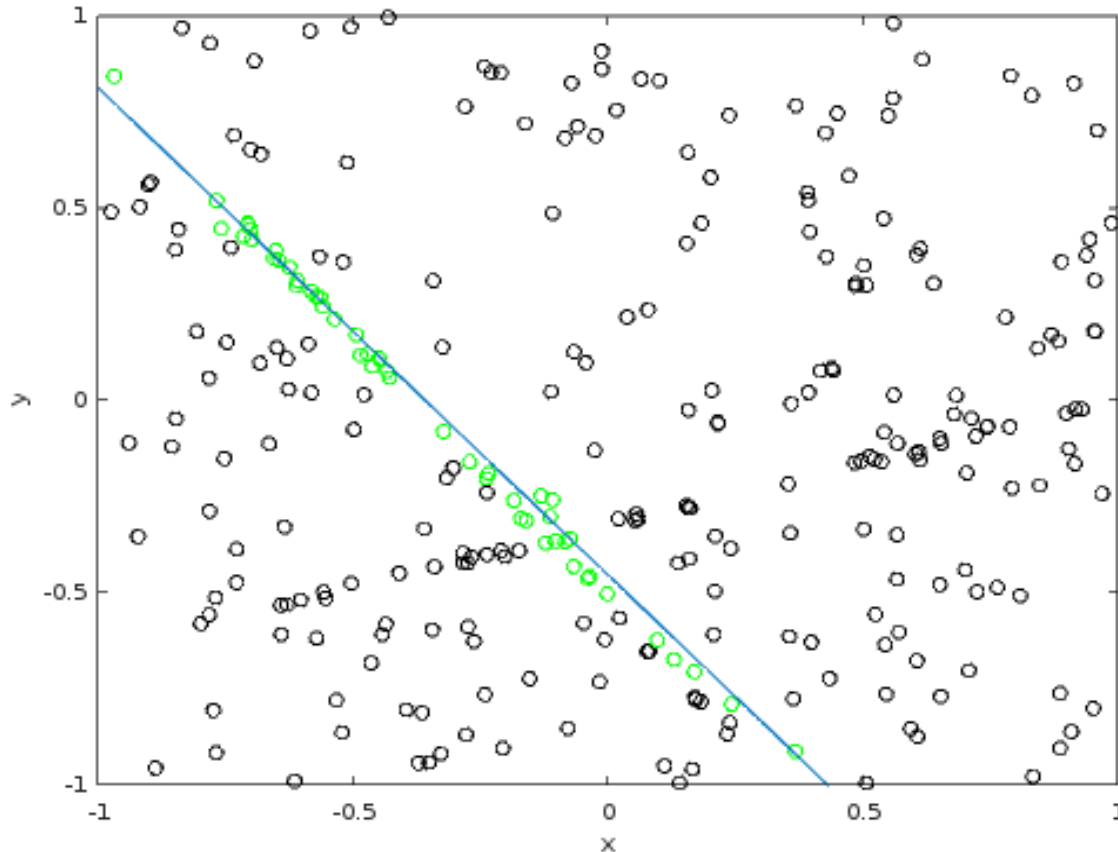


Több modell esete



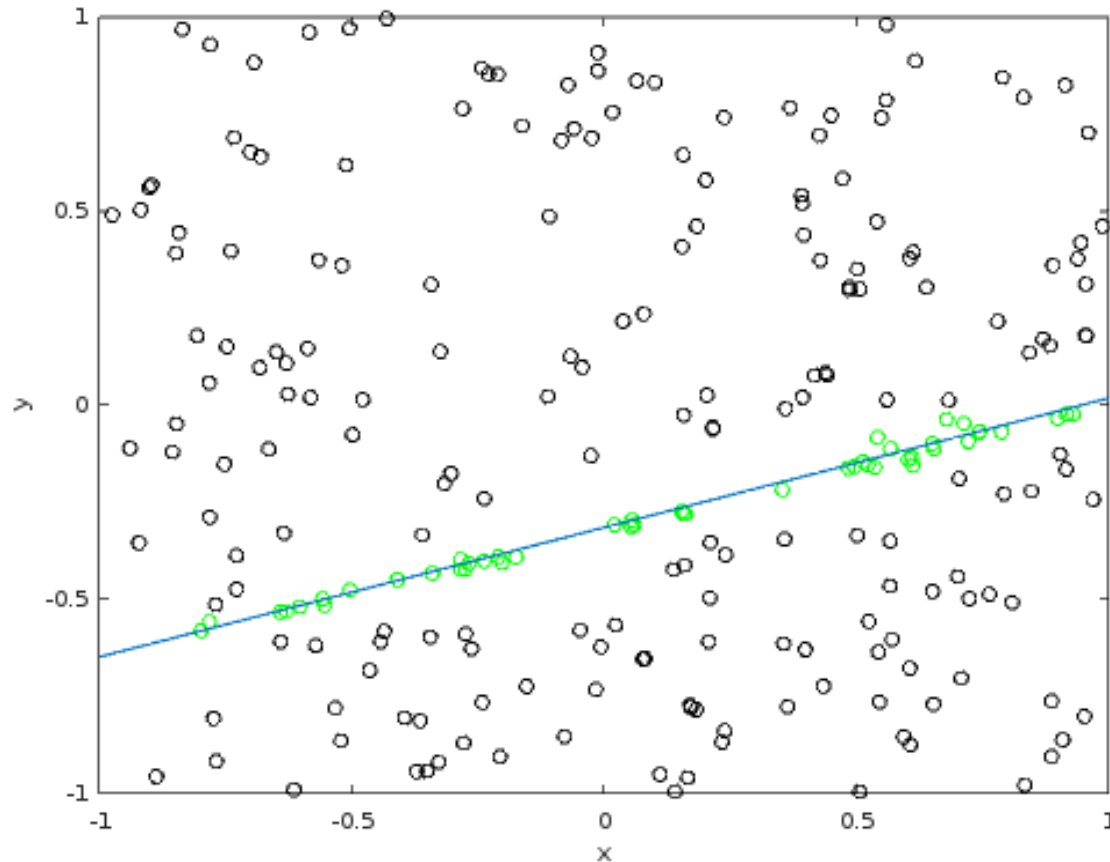
Több modell esete

LKN egyenes egyenlete: $y = -1.311 \cdot x - 0.488$

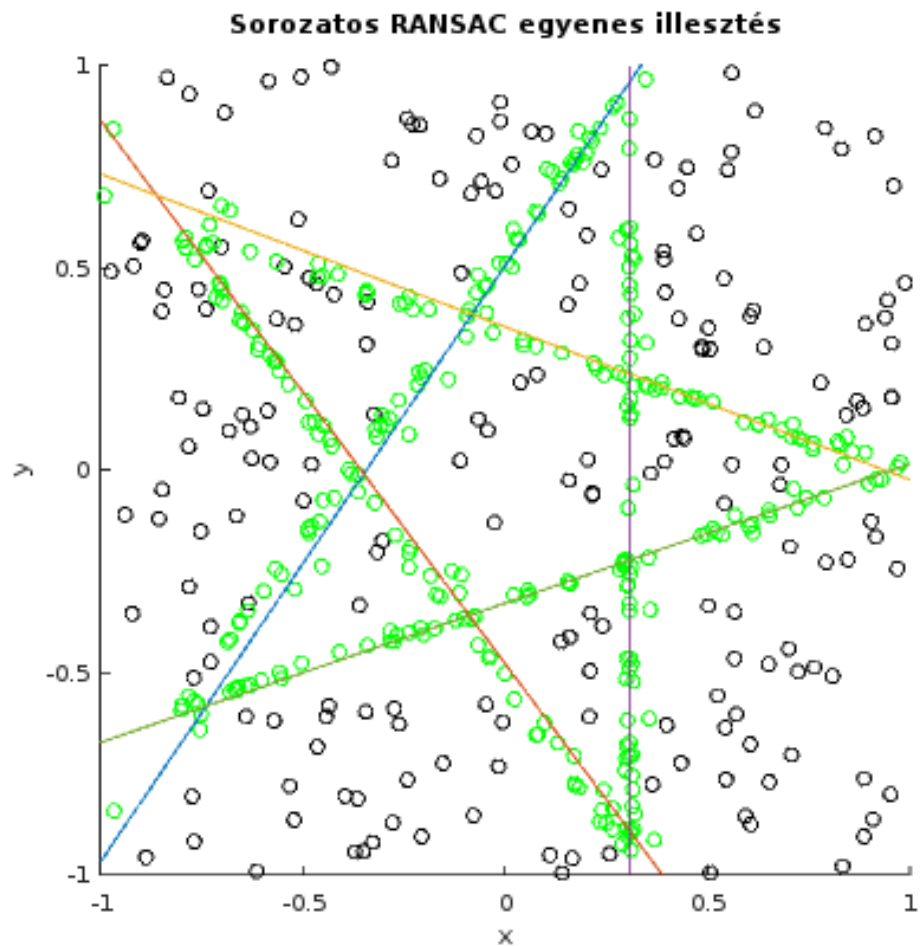


Több modell esete

LKN egyenes egyenlete: $y = 0.331*x - 0.321$



Sorozatos RANSAC



Több modell becslése

- Sorozatos RANSAC (Sequential RANSAC)
- párhuzamos RANSAC (MultiRANSAC, 2005)
- hisztogram elemzés (Residual Histogram Analysis, RHA, 2006)
- adatpontok másfajta leírása minimális minta elemszámú modellekkel (J-linkage, 2008, Merging J-Linkage, 2010, Kernel Fitting, 2009)
- energia minimalizáció (PEARL, 2010)

Szakirodalom

- Fischler M.A, Bolles R.C (1981): Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.
- Fitzgibbon A.W, Pilu M, Fisher R.B (1996): Direct Least Squares Fitting of Ellipses
- Zuliani M. (2014): RANSAC for Dummies. vision.ece.ucsb.edu/~zuliani, 101, GNU FDL
- Fouhey, D.F (2011): Multi-model estimation in the presence of outliers. PhD Thesis, Middlebury College