Univerza v Ljubljani

Fakulteta za elektrotehniko

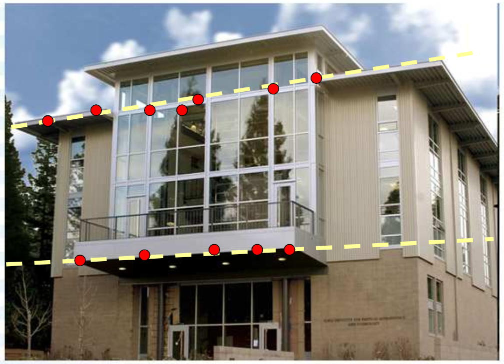
Prileganje in poravnava

Stanislav Kovačič

MACHINE VISION LABORATORY

http://vision.fe.uni-lj.si/

Primer: prileganje premic



Imamo detekcije točk vzdolž premice

Določamo parametre premice („fitamo“)

Slide from Silvio Savarese

Prileganje (in poravnava)

Prileganje:

Poišči parametre modela, ki najbolje opisujejo podatke (model se najbolje prilega podatkom oz meritvam)

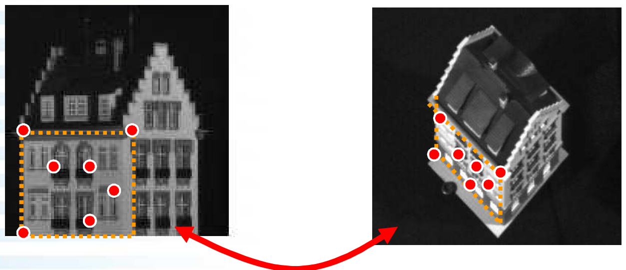
Poravnavanje (geometrijsko usklajevanje):

Poišči parametre transformacije, ki najbolje poravna podatke:

- Sliko s sliko
- Točke v sliki s korespondenčnimi točkami v drugi sliki
- Značilke ene slike z značilkami druge slike

V vsakem primeru ugotavljamo podobnost oz razdaljo

Primer: homografija

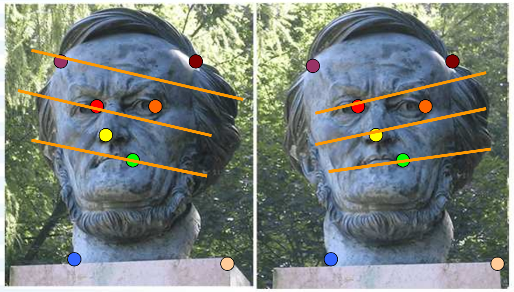


Imamo detektirane točk dveh pogledov ravnine.

Določamo parametre preslikave – matriko H_{3x3}

Slide from Silvio Savarese

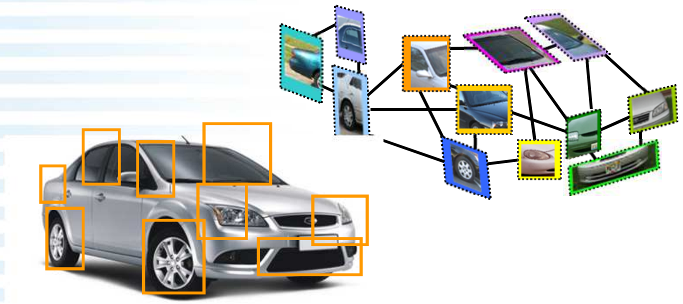
Primer: stereo geometrija



Imamo pare točk (korespondenčne točke)
Določamo „epipolarno geometrijo"
tj „fundamentalno matriko"

Slide from Silvio Savarese

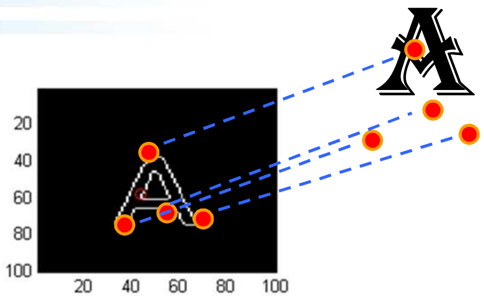
Primer: prileganje 3D modela



Ugotavljamo podobnost - kategoriziramo

Slide from Silvio Savarese

Primer: prileganje predloge



Ugotavljamo podobnost med predlogo in sliko - razpoznavamo

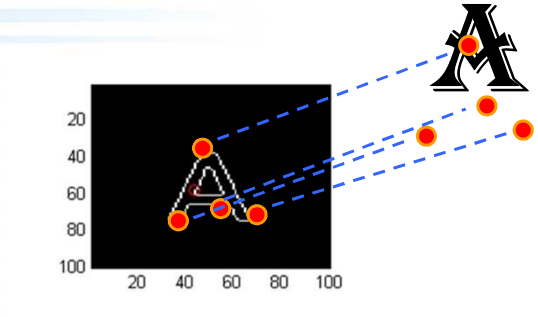
Slide from Silvio Savarese

Problem: šum

Slide from Silvio Savarese

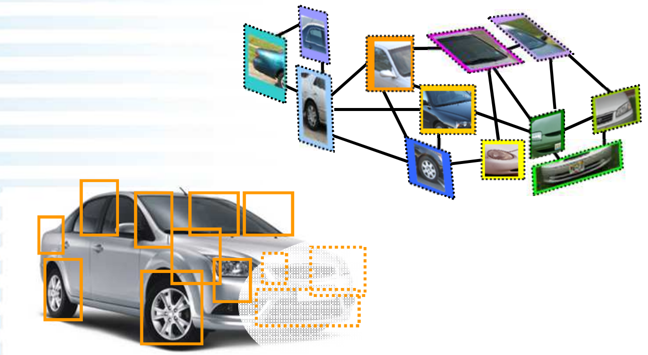
Problem: variabilnost

"All models are wrong, but some are useful." Box and Draper 1979



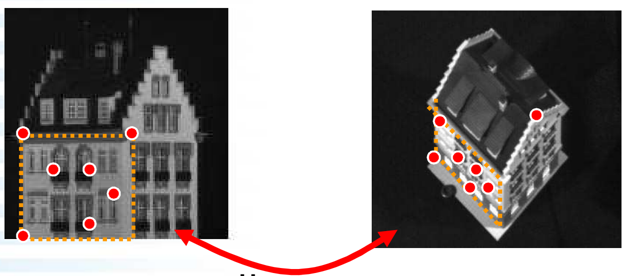
Slide from Silvio Savarese

Problem: nepopolni podatki



Slide from Silvio Savarese

Problem: grobe napake



Slide from Silvio Savarese

Prileganje in poravnava

Problem:

Nepopolni, napačni in negotovi podatki oz meritve

Naloge in izzivi:

- Izberi/izdelaj primerno mero kakovosti prileganja
  - Mera podobnosti – odvisno, kaj želimo
  - Odpornost na šum in grobe napake
- Izberi/izdelaj optimizacijski algoritem
  - Lokalni ekstremi
  - Hitrost, konvergenca

Prileganje in poravnava

(Globalna) optimizacija / iskanje parametrov modela
- LSM: metoda najmanjših kvadratov
- Robustna metoda kvadratov
- ICP: Iterativna metoda najbližje točke

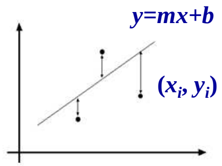
Glasovanje, preizkušanje in ocenjevanje hipotez
- Posplošeni Hough-ov transform
- RANSAC

Slide from Derek Hoiem

Prileganje premice – lin. regresija

$$\frac{\partial E}{\partial m} = -2\sum_{i=1}^{n}(y_i - x_i m - b)x_i = 0$$

$$\frac{\partial E}{\partial b} = -2\sum_{i=1}^{n}(y_i - x_i m - b)1 = 0$$



$$\sum_{i=1}^{n} x_i^2\, m + \sum_{i=1}^{n} x_i\, b = \sum_{i=1}^{n} y_i x_i$$

$$\sum_{i=1}^{n} x_i\, m + \sum_{i=1}^{n} 1\, b = \sum_{i=1}^{n} y_i$$

$$\Rightarrow \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

Prileganje premice – lin. regresija

Podatki: $(x_1, y_1), \ldots, (x_n, y_n)$

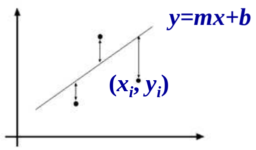
Model: Premica, $y_i = m x_i + b$

Naloga: poišči (m, b), ki minimizira

$$\boxed{E = \sum_{i=1}^{n}(y_i - m x_i - b)^2}$$

$$\frac{\partial E}{\partial m} = -2\sum_{i=1}^{n}(y_i - x_i m - b)x_i = 0$$

$$\frac{\partial E}{\partial b} = -2\sum_{i=1}^{n}(y_i - x_i m - b)1 = 0$$



Rešitev

$$\begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{bmatrix}$$

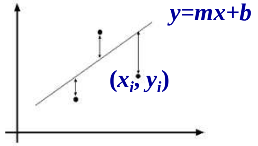
$$Det\left(\begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} 1 \end{bmatrix}\right) = \sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} 1 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i \neq 0$$

$$n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2 = n\sum_{i=1}^{n} x_i^2 - n^2\bar{x}^2 = n^2\left(\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2\right)$$

$$n^2\left(\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2\right) = n^2\left(\overline{x^2} - \bar{x}^2\right) = n^2\left(\frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2\right) \neq 0$$

Rešitev obstaja (t.j. Det()≠0), če le niso vsi x_i enaki.

Prileganje premice – lin. regresija



$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

$$E = \sum_{i=1}^{n} \left(\begin{bmatrix} x_i & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} - y_i \right)^2 = \left\| \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right\|^2 = \|\mathbf{Ap} - \mathbf{y}\|^2$$

$$= \mathbf{y}^T\mathbf{y} - 2(\mathbf{Ap})^T\mathbf{y} + (\mathbf{Ap})^T(\mathbf{Ap}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{p}^T(\mathbf{A}^T\mathbf{y}) + \mathbf{p}^T(\mathbf{A}^T\mathbf{A})\mathbf{p})$$

$$\frac{dE}{d\mathbf{p}} = 2\mathbf{A}^T\mathbf{A}\mathbf{p} - 2\mathbf{A}^T\mathbf{y} = 0$$

$$\boxed{\mathbf{A}^T\mathbf{A}\mathbf{p} = \mathbf{A}^T\mathbf{y}} \Rightarrow \mathbf{p} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$$

Matlab: p = A \ y;

Modified from S. Lazebnik

SVD

$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ U je stolpično ortogonalna, V ortogonalna, Σ diagonalna

$\mathbf{Av}_i = \sigma_i \mathbf{u}_i$ (množici ortonormalnih vektorjev)

$\mathbf{A}[\mathbf{v}_1|\mathbf{v}_2|\cdots|\mathbf{v}_i|\cdots|\mathbf{v}_n] = [\sigma_1\mathbf{u}_1|\sigma_2\mathbf{u}_2|\cdots|\sigma_i\mathbf{u}_i|\cdots|\sigma_n\mathbf{u}_n]$

$\mathbf{AV} = \mathbf{U}\Sigma$

$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$

Nadalje:

$\mathbf{A}^T\mathbf{A} = [\mathbf{U}\Sigma\mathbf{V}^T]^T\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{V}(\mathbf{U}\Sigma)^T\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{V}\Sigma^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{V}\Sigma^2\mathbf{V}^T$

$\mathbf{A}\mathbf{A}^T = \mathbf{U}\Sigma\mathbf{V}^T[\mathbf{U}\Sigma\mathbf{V}^T]^T = \mathbf{U}\Sigma\mathbf{V}^T\mathbf{V}(\mathbf{U}\Sigma)^T = \mathbf{U}\Sigma\Sigma^T\mathbf{U}^T = \mathbf{U}\Sigma^2\mathbf{U}^T$

Rešitev normalne enačbe LS problema

$\mathbf{A}^T\mathbf{A}\mathbf{p} = \mathbf{A}^T\mathbf{y}$ Normalna enačna za linearni LS problem

$\mathbf{A}^T\mathbf{A}\mathbf{p} = \mathbf{A}^T\mathbf{y} \Rightarrow \mathbf{p} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$

$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ posplošeni oziroma "psevdo inverz" matrike A

Standardni pristop: QR razcep ali SVD razcep

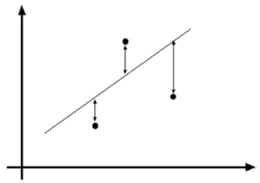
SVD: Razcep na singularne vrednosti in vektorje matrike A

$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ U je stolpično ortogonalna, V ortogonalna, Σ diagonalna

$$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = [(\mathbf{U}\Sigma\mathbf{V}^T)^T(\mathbf{U}\Sigma\mathbf{V}^T)]^{-1}(\mathbf{U}\Sigma\mathbf{V}^T)^T$$
$$= [\mathbf{V}\Sigma^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T]^{-1}\mathbf{V}\Sigma^T\mathbf{U}^T = \mathbf{V}\Sigma^{-2}\mathbf{V}^T\mathbf{V}\Sigma\mathbf{U}^T$$
$$= \mathbf{V}\Sigma^{-2}\Sigma\mathbf{U}^T = \mathbf{V}\Sigma^{-1}\mathbf{U}^T$$

Zapažanja?

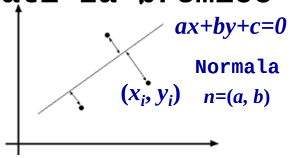
- **Rotacijska občutljivost**
- **Navpičnice**



- **V resnici sploh ne minimiziramo vsote (kvadratov) razdalj točk od premice**

Slide from S. Lazebnik

Najmanjši kvadrati za premico



Za ($a^2+b^2=1$) je pravokotna razdalja do točk (x_i, y_i)

$|ax_i + by_i + c|$

a,b sta potem komponenti enotinega vektorja, ki je pravokoten na premico, enotin vektor normale.

Naloga: poišči (a, b, c), ki minimizirajo vsoto kvadratov pravokotnih razdalj

$$\boxed{E = \sum_{i=1}^{n}(ax_i + by_i + c)^2 \text{pri pogoju } (a^2 + b^2) = 1}$$

$$\min\{\sum_{i=1}^{n}(ax_i + by_i + c)^2 - \lambda(a^2 + b^2 - 1)\}$$

LS (za premico)

$$E = \sum_{i=1}^{n}(a(x_i - \bar{x}) + b(y_i - \bar{y}))^2 = \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = \| \mathbf{Ap} \|^2$$

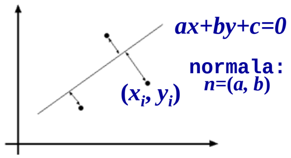
$$E = \| \mathbf{Ap} \|^2 = (\mathbf{Ap})^T \mathbf{Ap} = \mathbf{p}^T (\mathbf{A}^T \mathbf{A})\mathbf{p}$$

$$\min \mathbf{p}^T \mathbf{A}^T \mathbf{Ap} \quad \text{pri} \quad \mathbf{p}^T \mathbf{p} = 1 \qquad \min\{L(\mathbf{p})\} = \{\mathbf{p}^T \mathbf{A}^T \mathbf{Ap} - \lambda(\mathbf{p}^T \mathbf{p} - 1)\}$$

$$\frac{dE}{d\mathbf{p}} = \frac{d(L(\mathbf{p}))}{d\mathbf{p}} = 2\mathbf{A}^T \mathbf{Ap} - \lambda \mathbf{p} = (2\mathbf{A}^T \mathbf{A} - \lambda \mathbf{I})\mathbf{p} = \mathbf{0}$$

Rešitev: lastni vektor, ki pripada najmanjši lastni vrednosti A^TA

LS (za premico)



$$\frac{\partial E}{\partial c} = \sum_{i=1}^{n}(ax_i + by_i + c) = 0$$

$$\sum_{i=1}^{n} ax_i + \sum_{i=1}^{n} by_i + \sum_{i=1}^{n} c = 0$$

$$c = -\frac{a}{n}\sum_{i=1}^{n} x_i - \frac{b}{n}\sum_{i=1}^{n} y_i = -(a\bar{x} + b\bar{y})$$

$$E = \sum_{i=1}^{n}(ax_i + by_i + c)^2 = \sum_{i=1}^{n}(ax_i + by_i - (a\bar{x} + b\bar{y}))^2$$

$$E = \sum_{i=1}^{n}(a(x_i - \bar{x}) + b(y_i - \bar{y}))^2$$

LS - rešitev

$$\frac{dE}{d\mathbf{p}} = \frac{d(L(\mathbf{p}))}{d\mathbf{p}} = 2\mathbf{A}^T \mathbf{Ap} - \lambda \mathbf{p} = (2\mathbf{A}^T \mathbf{A} - \lambda \mathbf{I})\mathbf{p} = \mathbf{0}$$

$$2\mathbf{A}^T \mathbf{Ap} = \lambda \mathbf{p} \Rightarrow \mathbf{Cp} = \mu \mathbf{p};$$

$\mathbf{C}$ je simetrična, pozitivno (semi)definitna matrika
$\mathbf{p}$ je lastni vektor in μ je pripadajoča lastna vrednost

$$E = \mathbf{p}^T \mathbf{Cp} = \mathbf{p}^T \mathbf{M} \boldsymbol{\Lambda} M^T \mathbf{p} = (\mathbf{M}^T \mathbf{p})^T \boldsymbol{\Lambda} (\mathbf{M}^T \mathbf{p}) = \mathbf{e}^T \boldsymbol{\Lambda} \mathbf{e}$$

$\mathbf{M}$ je ortogonalna matrika lastnih vektorjev matrike $\mathbf{C}$ in
$\boldsymbol{\Lambda}$ je diagonalna matrika njenih lastnih vrednosti
$\mathbf{e}$ je enotin vektor z eno samo komponento različno od nič in enako 1

LS - rešitev

$$E = \mathbf{p}^T \mathbf{C} \mathbf{p} = \mathbf{p}^T \mathbf{M} \Lambda M^T \mathbf{p} = (\mathbf{M}^T \mathbf{p})^T \Lambda (\mathbf{M}^T \mathbf{p}) = \mathbf{e}^T \Lambda \mathbf{e}$$

$$\min\{E\} = \min\{\mathbf{e}^T \Lambda \mathbf{e}\} = \lambda_{\min}$$

E je najmanjši za p z najmanjšo pripadajočo lastno vrednostjo.

Rešitev je torej:
Lastni vektor z najmanjšo lastno vrednostjo matrike $A^T A$

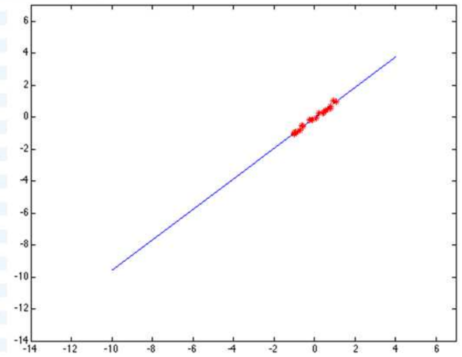
LS – rešitev - SVD razcep

$$\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T$$
$$\mathbf{C} = \mathbf{A}^T \mathbf{A} = (\mathbf{U} \Sigma \mathbf{V}^T)^T (\mathbf{U} \Sigma \mathbf{V}^T) = \mathbf{V} \Sigma^T \mathbf{U}^T \mathbf{U} \Sigma \mathbf{V}^T = \mathbf{V} \Sigma^2 \mathbf{V}^T$$
$$\mathbf{V} = \mathbf{M}$$
$$\Lambda = \Sigma^2$$

Rešitev je torej:
Singularni vektor z najmanjšo singularno vrednostjo matrike A

Najmanjši kvadrati:odpornost na šum

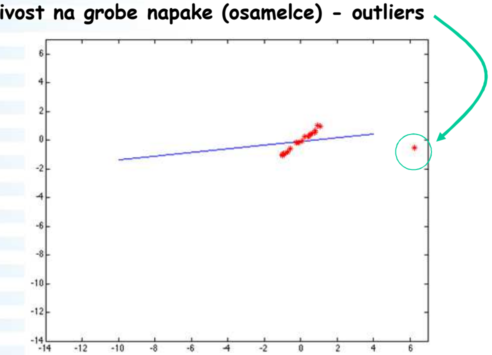
Prileganje k rdečim točkam:



Slides from Svetlana Lazebni

Najmanjši kvadrati

Občutljivost na grobe napake (osamelce) - outliers



Očitna težava: kvadrat velikih odstopanj je velik
Že ena sama groba napaka močno pokvari ocenjene parametre modela

Robustna LS

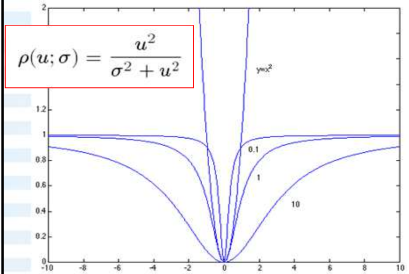
Koncept:

minimiziraj $\sum_{\mathrm{i}} \rho(\mathrm{u_i}(\mathrm{x_i}, \theta); \sigma)$ $\quad u = \sum_{i=1}^{n} (y_i - m x_i - b)^2$

$u_i(x_i, \theta)$ – odstopanje i^{-te} točke glede na parametre modela θ

ρ – funkcija robustnosti z merilom (skalo) σ

$$\rho(u;\sigma) = \frac{u^2}{\sigma^2 + u^2}$$



ρ

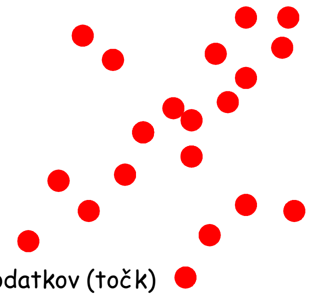
„nagradi" majhna odstopanja

„kaznuj" zelo velika odstopanja

Vprašanje:
kako izbrati „pravo" merilo

Slide from S. Savarese

RANSAC



Algoritem:

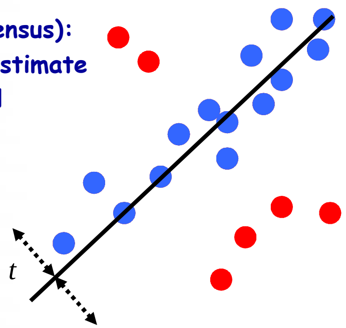
1. Naključno izberi ravno dovolj podatkov (točk) za določitev modela
2. Določi parametre modela
3. Oceni kakovost prileganja podatkov znotraj predpisanih mej

Ponavljaj 1-3

dokler ni z veliko gotovostjo izbran najboljši model

RANSAC

(RANdom SAmple Consensus):
Learning technique to estimate parameters of a model by random sampling of observed data

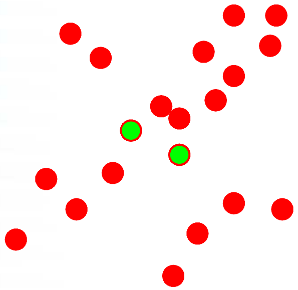


Fischler & Bolles in '81.

Source: Savarese

RANSAC

RANSAC za premico



Algoritem:

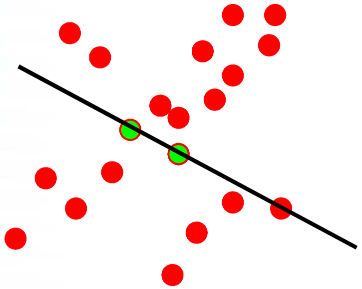
1. Naključno izberi ravno zadosti točk: 2
2. Izračunaj parametre modela: premice
3. Oceni kakovost modela (prileganje podatkov znotraj meja)

Ponavljaj 1-3

Illustration by Savarese

RANSAC

RANSAC za premico

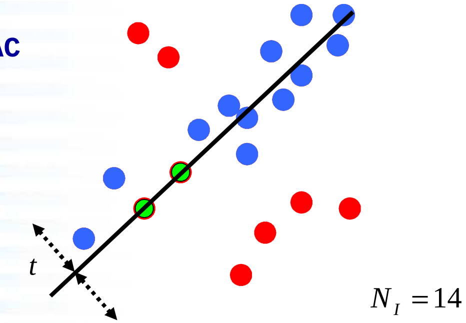


Algoritem:

1. Naključno izberi ravno zadosti točk: 2
2. Izračunaj parametre modela: premice
3. Oceni kakovost modela (prileganje podatkov znotraj meja)

Ponavljaj 1-3

RANSAC



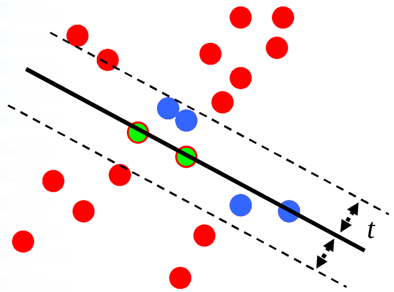
Algoritem:

1. Naključno izberi ravno zadosti točk: 2
2. Izračunaj parametre modela: premice
3. Oceni kakovost modela (prileganje podatkov znotraj meja)

Ponavljaj 1-3

RANSAC

$N_I = 6$



Algoritem:

1. Naključno izberi ravno zadosti točk: 2
2. Izračunaj parametre modela: premice
3. Oceni kakovost modela (prileganje podatkov znotraj meja)

Ponavljaj 1-3

Izbira parametrov algoritma

- **Izbrano število točk: *s***
  - **Tipično ravno dovolj za izračun parametrov modela**
  - **za premico s = 2;**
- **Prag *t***
  - **Izberi *t* tako, da zelo verjetno izločimo osamelce (npr ≈ 0.95)**
  - **Za gaussov šum s srednjo vrednostjo nič in σ: t=2σ**
- **Število potrebnih podatkov znotraj praga t, ki potrjujejo model**
- **Število vzorčenj *N***
  - **Izberi *N* tako, da bo vsaj eno vzorčenje brez osamelcev z verjetnostjo p, (npr. *p*=0.99) (delež osamelcev: *e*)**

$$\boxed{\left(1-(1-e)^s\right)^N = 1-p}$$

Izbira parametrov algoritma

- $(1-e)^s$ *: verjetnost, da je vzorec s podatkov brez osamelcev*
- $1-(1-e)^s$ verjetnost, da vzorec vsebuje osamelce
- $(1-(1-e)^s)^N$: verjetnost, da v N poskusih nikoli ne izberemo vzorca brez osamelcev
- $p = 1-(1-(1-e)^s)^N$: verjetnost, da v N poskusih izberemo vsaj enkrat vzorec brez osamelcev

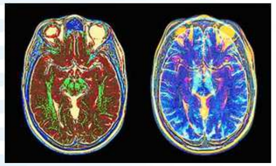
$$\boxed{\left(1-(1-e)^s\right)^N = 1-p} \Rightarrow N = \log(1-p)/\log\left(1-(1-e)^s\right)$$

- Na primer: s=2, e=0,5, p=0,999
- *N = 25*

Poravnavanje

Kako poravnati tedaj, ko nimamo a priori korespondenčnih točk?
LS, RANSAC ne moremo uporabiti, Hough tudi ne

Tipični primeri



Tipično področje uporabe:
poravnavanje medicinskih slik (Medical image registration)
Prileganje slik k slikam ali kontur k slikam

Slide from Derek Hoiem

RANSAC

+

- Robusten na grobe napake – „osamelce"
- Enostaven, parametre algoritma določimo analitično
- Primeren tudi za veliko število parametrov modela

–

- Računsko dokaj zahteven

Tipični primeri uporabe

- *Ocenjevanje homografije*
- *Ocenjevanje fundamentalne matrike (stereo)*

ICP - poravnavanje

Cilj:
oceni parametre transformacije (poravnave) med dvema (gostima) množicama točk S1 in S2.

Postopek:

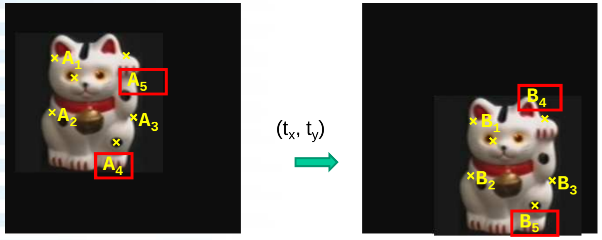
1. *Vsaki točki iz S1 priredi najbližjo iz S2*
2. *Oceni parametre transformacije*
   – *Na primer z LSM*
3. *Preslikaj točke iz S1 z ocenjeno transformacijo na S2*
4. *Ponavljaj 1-3 dokler ni sprememba transformacije majhna*

Primer: translacija

Za korespondenčne točke iz {A} in {B}, oceni translacijo

$$\begin{bmatrix} x_i^B \\ y_i^B \end{bmatrix} = \begin{bmatrix} x_i^A \\ y_i^A \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Primer: translacija



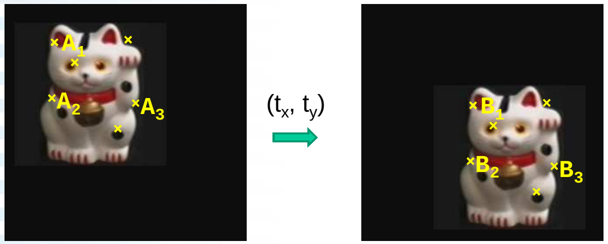
Problem: grobe napake

RANSAC

1. Vzorči korespondence (en par točk)
2. Izračunaj translacijo
3. Oceni kakovost translacije
4. Ponavljaj 1-3 N krat

$$\begin{bmatrix} x_i^B \\ y_i^B \end{bmatrix} = \begin{bmatrix} x_i^A \\ y_i^A \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Primer: translacija



$$\begin{bmatrix} x_i^B \\ y_i^B \end{bmatrix} = \begin{bmatrix} x_i^A \\ y_i^A \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Imamo tri korespondenčne točke

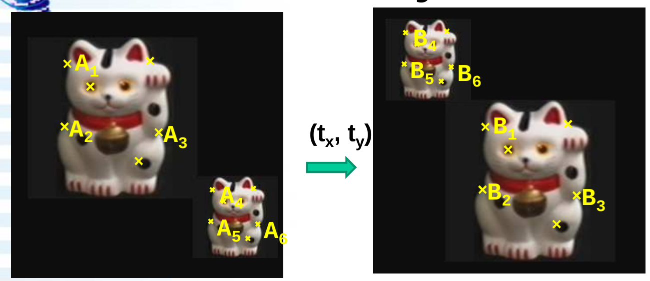
Vsaka prinese dve enačbi

Imamo dva parametra translacije

Rešimo z metodo najmanjših kvadratov

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x_1^B - x_1^A \\ y_1^B - y_1^A \\ \vdots \\ x_n^B - x_n^A \\ y_n^B - y_n^A \end{bmatrix}$$

Primer: translacija



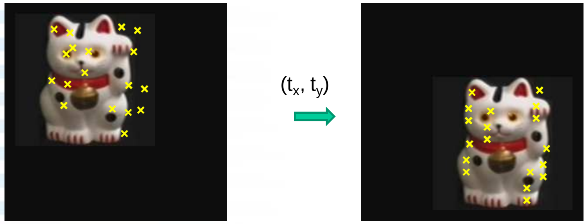
Problem: grobe napake, več objektov, zgrešena ujemanja

Hough

1. Inicializiraj prostor parametrov na nič
2. Vsak par korespondenčnih točk da glas za vrednosti parametrov
3. Poišči prametre z največ glasovi
4. Reši LS problem brez osamelcev

$$\begin{bmatrix} x_i^B \\ y_i^B \end{bmatrix} = \begin{bmatrix} x_i^A \\ y_i^A \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Primer: translacija



Problem: nimamo začetnih ujemanj - korespondenc

ICP

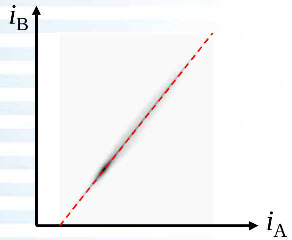
1. Poišči najbližjega soseda iz B k vsaki točki A
2. Izračunaj transformacijo
3. Transformiraj točke iz A
4. Ponavljaj 1-3 dokler ne skonvergira

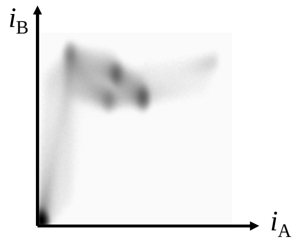
$$\begin{bmatrix} x_i^B \\ y_i^B \end{bmatrix} = \begin{bmatrix} x_i^A \\ y_i^A \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Medmodalno prileganje/poravnava

Svetlostna odvisnost slik:

Enaka modalnost: **Različna modalnost:**

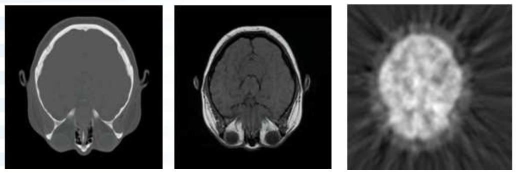




Linearna odvisnost **Nelinearna odvisnost**

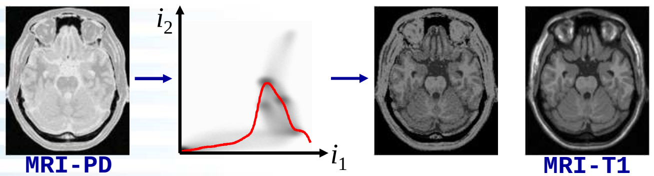
Medmodalno prileganje/poravnava

Različne modalnosti slik:



CT **MRI-T1** **PET**

Medmodalno prileganje/poravnava



Iščemo preslikavo, ki maksimizira

I(A,B) = H(A) + H(B) – H(A,B)

Na primer:

Minimiziramo H(A,B)v odvisnosti od parametrov poravnave

Viri

1. V glavnem povzeto po CV – Brown University (James Hays)
2. Nekaj po CV - A Modern Approach (Forsyth, Ponce)
3. Večmodalna poravnava iz doktorske disertacije (Peter Rogelj)
4. Zahvala Mateju Kristanu