

# **RAČUNALNIŠKI VID**

## **izr. prof. dr. Božidar Potočnik**

Fakulteta za elektrotehniko, računalništvo in  
informatiko

2. letnik R\_IT – MAG  
I. 2025/2026

# OSEBJE

- **Predavatelj:**
  - izr. prof. dr. Božidar Potočnik
    - Kabinet: G2-2N.37
    - E-pošta: bozidar.potocnik@um.si
    - Govorilne ure: sreda, 9.00 – 11.00
- **Asistent:**
  - dr. Martin Šavc (martin.savc@um.si)



# PREDVIDENI ŠTUDIJSKI REZULTATI

- **Cilji:**
  - Posredovati izpopolnjene tehnike računalniškega vida, ki so osnova za sledenje gibanju, za razpoznavanje objektov in scen, za nadzor in varovanje prostorov, imovine in oseb, za biometriko in podobno.
- **Znanje in razumevanje:**
  - Študent bo sposoben:
    - reproducirati izpopolnjene postopke za segmentacijo slik in za tvorbo globinskih slik,
    - razumeti in uporabiti algoritme za sledenje gibanju in za rekonstrukcijo trdnih teles iz njihovega gibanja ali gibanja kamere okoli njih,
    - dojemati posnetke 3D prostorov, njihovo analizo in modeliranje,
    - oceniti zahtevnost postopkov in predvideti potrebno računalniško, komunikacijsko in video opremo.

# OBVEZNOSTI ŠTUDENTA

- ECTS: 6 točk  $\approx$  180 ur dela
  - 1 točko ECTS lahko enačimo z 30 urami dela
- Kontaktne ure:
  - Predavanja: 30 ur
  - Laboratorijske oz. računalniške vaje: 30 ur
- Samostojno delo študenta (npr. doma):  $\approx$  120 ur
  - Laboratorijske vaje:  $\approx$  70 ur
  - Priprava na izpit:  $\approx$  50 ur
  - Dejanske porabljenе ure seveda odvisne od študenta.

# NAČINI OCENJEVANJA

- Končno oceno tvorijo:
  - Opravljene laboratorijske vaje – 50 % oz. 500 točk (**>= 250 točk**)
  - Pisni izpit – 50 % oz. 500 točk (**>= 250 točk**)
- **Ne pozabite se prijavljati na izpitne roke!**

# PREDVIDEN RAZPORED PREDAVANJ

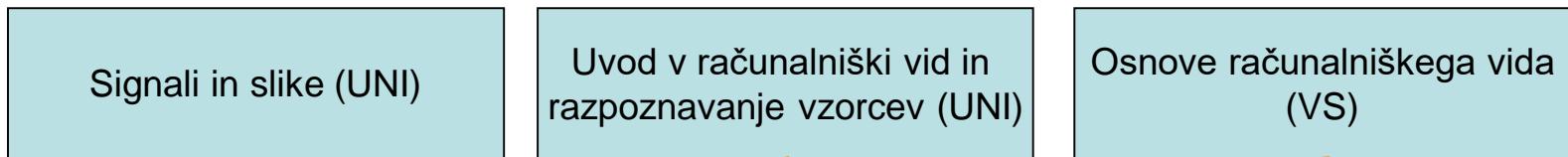
- |  |  |
|--|--|
| 1. teden (3.10.)                           | 9. teden (28.11.)                          |
| 2. teden (10.10.)                          | 10. teden (5.12.)                          |
| 3. teden (23.10.) – čet,<br>7-9, A-301     | 11. teden (12.12.)                         |
| 4. teden (24.10.)                          | 12. teden (19.12.)                         |
| 5. teden (31.10.) - praznik                | 13. teden (9.1.)                           |
| 6. teden (7.11.)                           | 14. teden (16.1.)                          |
| 7. teden (14.11.)                          | <b>15. teden (23.1.) – 2.<br/>kolokvij</b> |
| <b>8. teden (21.11.) – 1.<br/>kolokvij</b> |  |

# LITERATURA

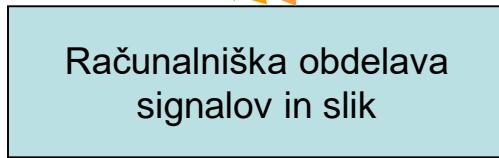
- Gradivo iz predavanj, zapiski in dodatni študijski material objavljen na spletni strani predmeta
- R. Szeliski: *Computer Vision: Algorithms and Applications*, Springer, 2010 ali novejša izdaja (<http://szeliski.org/Book>).
- D. A. Forsyth, J. Ponce: *Computer Vision: A Modern Approach*, Pearson Education International, Upper Saddle River, 2003.
- R. Hartley, A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.
- Spletni viri
- (Ostali temeljni študijski viri navedeni v učnem načrtu predmeta – spletna stran)

# POVEZANOST PREDMETOV S PODROČJA OBDELAVE SIGNALOV IN SLIK NA PROGRAMU R-IT

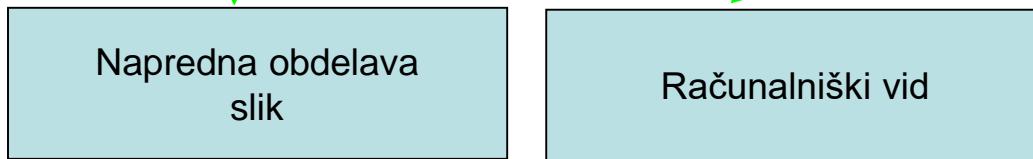
1. stopnja,  
osnove:



2. stopnja,  
osnove:



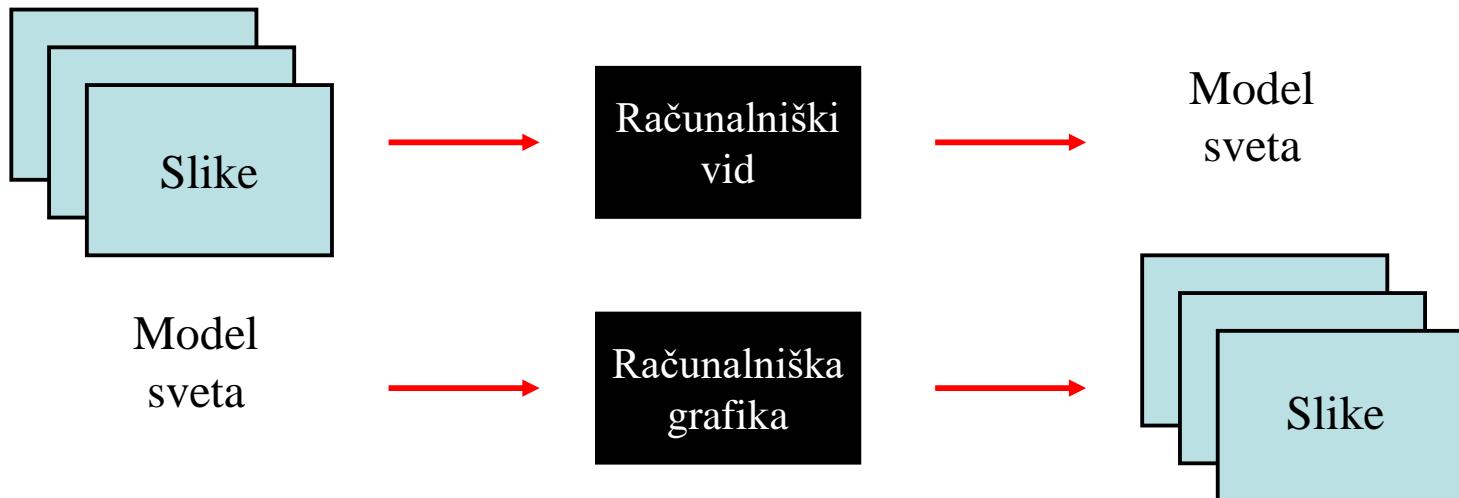
2. stopnja,  
nad-  
gradnja:



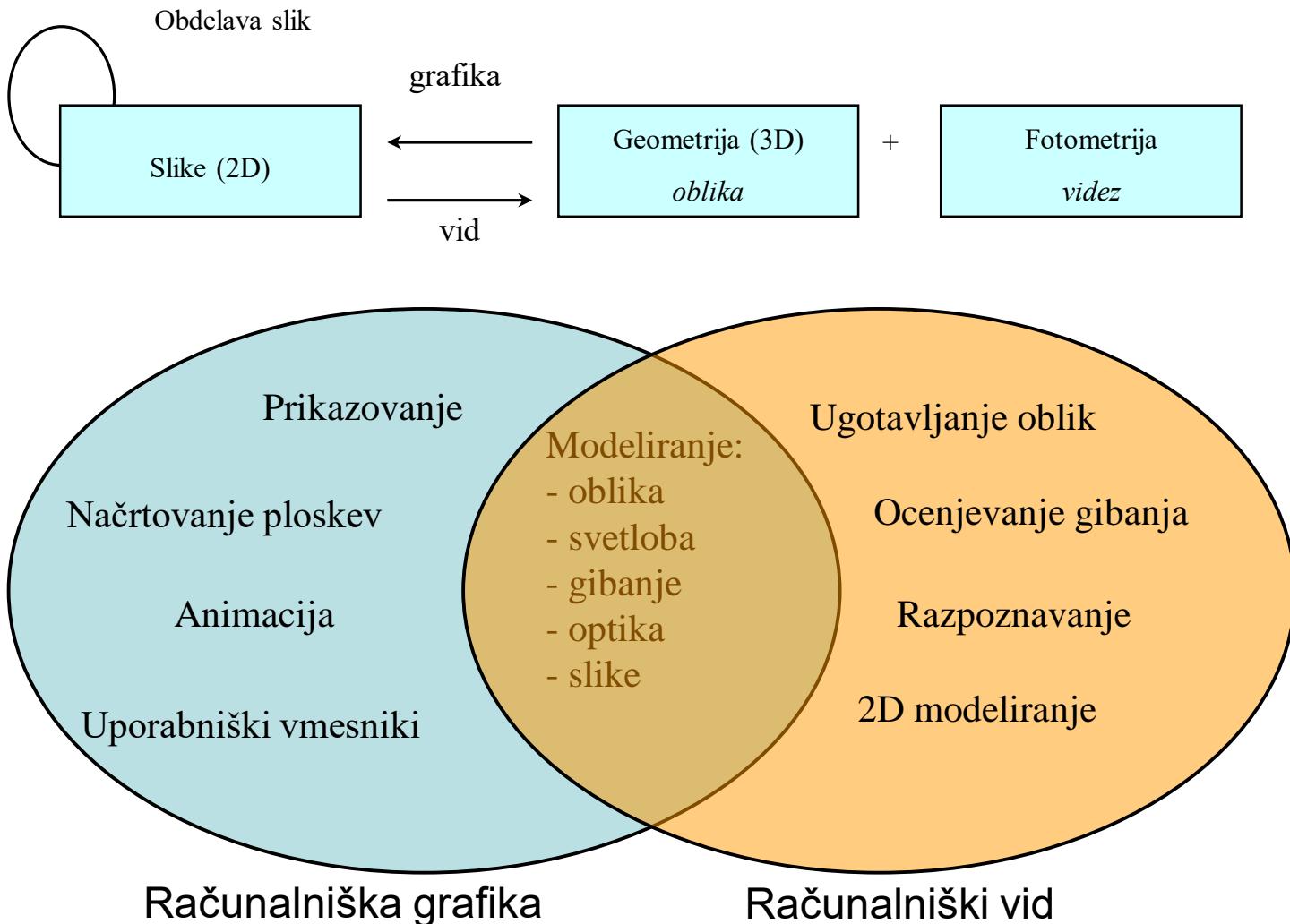
# Kaj je računalniški vid?

(po R. Szeliski, Univerza Washington, [www.cs.washington.edu/education/courses/cse576/05sp/](http://www.cs.washington.edu/education/courses/cse576/05sp/))

- Dojemanje, razumevanje slik (umetna inteligenco)
- Zaznavanje, tipala (senzorji) v robotiki
- Računalniška emulacija človeškega vida
- Obratno od računalniške grafike



# PRESEK MED RAČUNALNIŠKIM VIDOM IN GRAFIKO



# Druge vede, povezane z računalniškim vidom

- Obdelava slik
- Slikovni sistemi v znanosti in medicini
- Razpoznavanje vzorcev
- Računalniška grafika
- Strojno učenje
- Umetna inteligenco
- Vizualna nevroznanost
- Uporabna matematika
  - algoritmi za obdelavo signalov in slik
  - evklidska in projekcijska geometrija
  - računalniška geometrija
  - vektorski račun
  - optimizacije
  - statistične cenilke

# **Področja uporabe računalniškega vida**

1. Medicinske in znanstvene aplikacije
2. Nadzorni sistemi, zaznavanje gibanja
3. Razpoznavanje prstnih odtisov
4. Biometrika, razpoznavanje na osnovi očesne šarenice
5. Geometrično rekonstruiranje (modeliranje, forenzika, posebni efekti)
6. Varnost vožnje, zaznavanje ovir
7. Obogatena resničnost
8. Urejanje slik in videov
9. Spletno oddajanje in indeksiranje videov

# RAČUNALNIŠKI VID

## Učna snov

### 1. Osnove

- Uvod, motivacija, ravninske in prostorske transformacije
- Camera obscura, projekcijske transformacije
- Kalibracija kamer
- Epipolarna geometrija

### 2. Poravnava slik

- Toge poravnave – premik in zasuk
- Poravnave s pomočjo frekvenčnega prostora
- Afine poravnave, netoge poravnave
- Stereo vid, globina

### 3. Zaznavanje gibanja

- Gibanje, zaznano s pomočjo dveh slik
- Optični tok, polje gibanja

### 4. 3D rekonstrukcija

- Operatorji za iskanje korespondenčnih točk
- Določanje korespondenčnih točk s poravnavo slik
- Projekcijska 3D rekonstrukcija

### 5. Razpoznavanje (opcijsko)

- *Zaznavanje objektov*
- *Razpoznavanje obrazov*
- *Računalniški vid v vmesnikih človek-stroj*

# Kazalo

1.	Camera obscura in osnovne projekcije.....	15
2.	Kalibracija kamer.....	32
3.	Epipolarna geometrija.....	46
4.	Poravnava slik.....	58
5.	Stereo vid.....	91
6.	Optični tok.....	114
7.	Struktura iz gibanja.....	137
8.	3D-rekonstrukcija objektov.....	162

# Camera obscura in osnovne projekcije

(povzeto po R. Szeliski, Univerza Washington,  
[www.cs.washington.edu/education/courses/cse576/05sp/](http://www.cs.washington.edu/education/courses/cse576/05sp/))

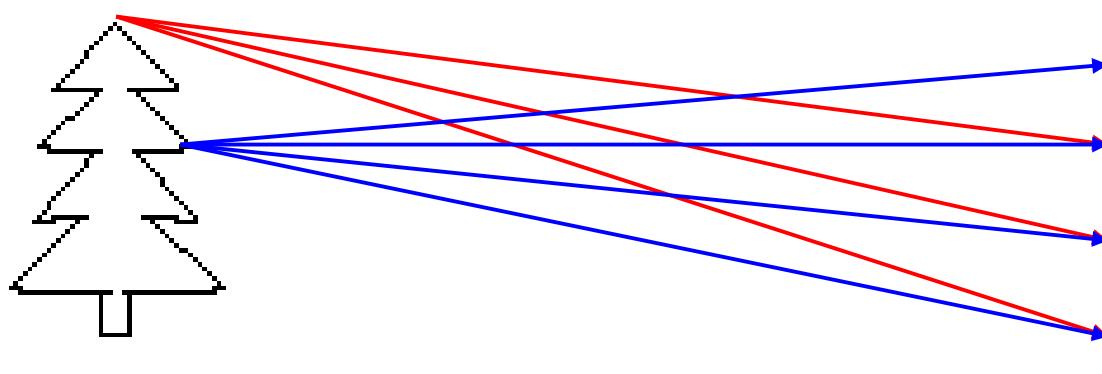
# NASTANEK Slike

## 1. Naravna svetloba

- ob odboju s površine predmetov se svetloba razsipa

predmet

film – slika?

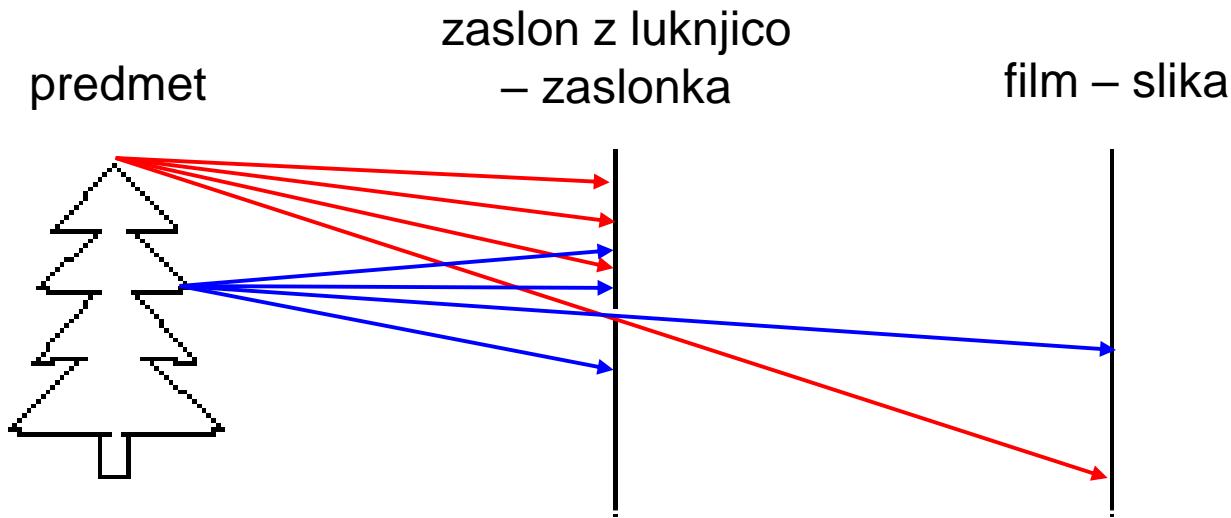


## 2. Prestrezanje ravninske slike

- svetlobni žarki, odbiti iz iste točke predmeta, osvetlijo slikovno ravnino (film) na več mestih
- slika je **nerazpoznavna**

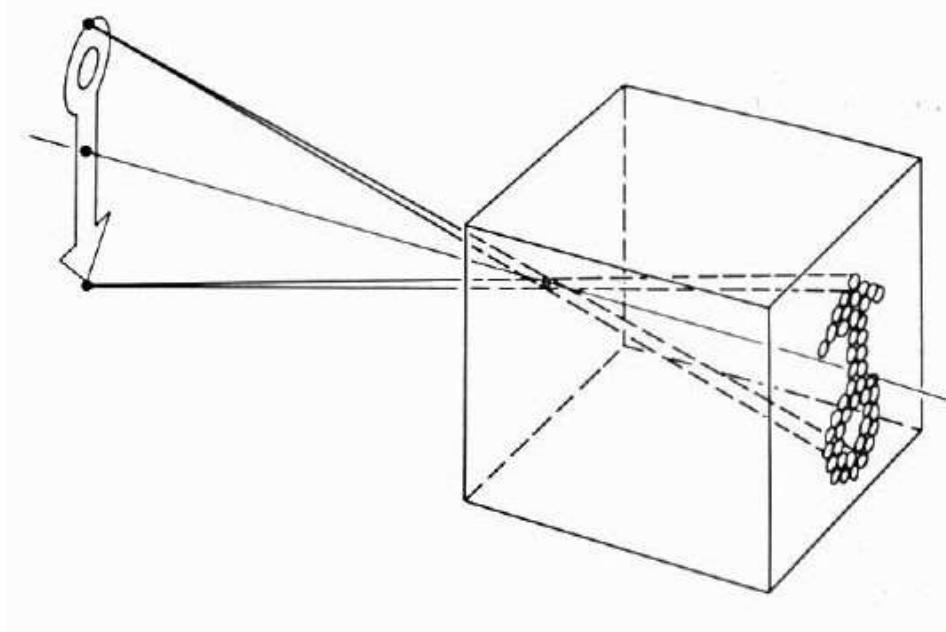
# NASTANEK SЛИKE (2)

## 1. Izločanje ozkega snopa svetlobe



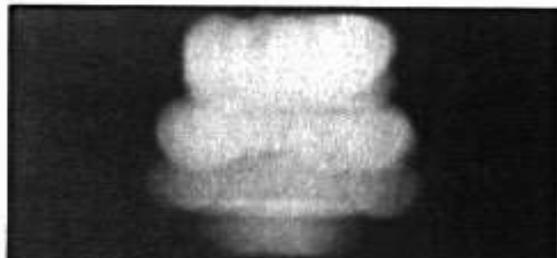
- svetlobni žarki so omejeni z odprtino (*aperture*) v zaslonki in osvetlijo za ločene točke predmeta slikovno ravnilo (film) na ločenih, a koncentriranih mestih
- slika je odvisna od odprtine v zaslonki
- slika je **razpoznavna**

# CAMERA OBSCURA



- Poznal jo je že Aristotel
- Kako velikost odprtine vpliva na sliko?

# CAMERA OBSCURA – VPLIV ZASLONKE



2 mm



1 mm



0.6mm

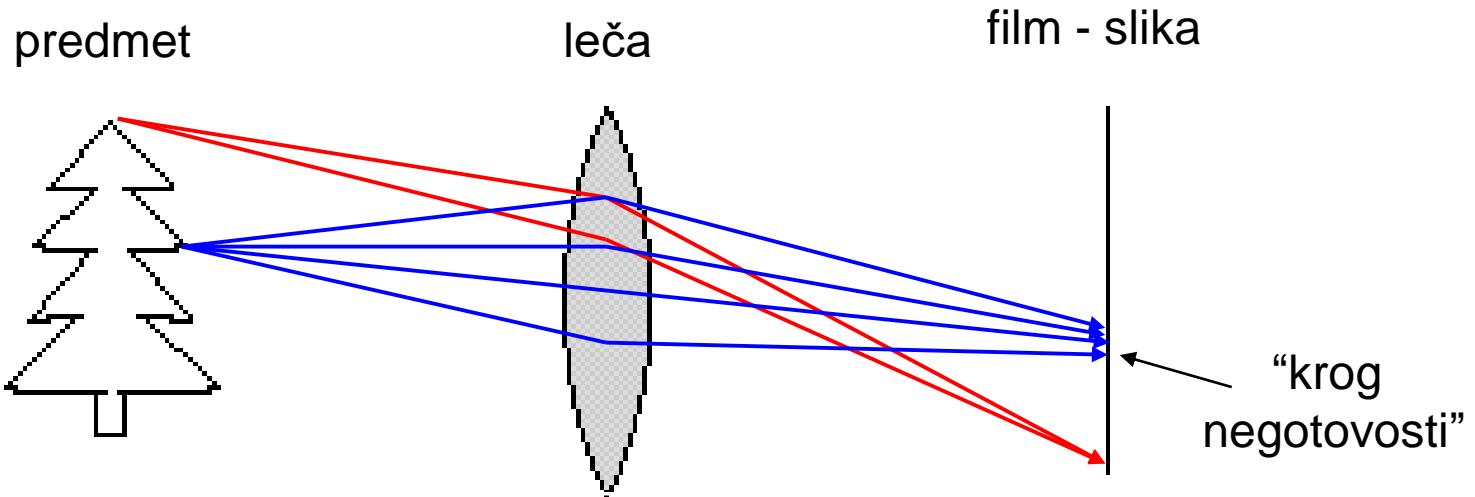


0.35 mm

- Čim manjša odprtina, tem bolje – ali res?
- Ne –
  - manjša odprtina prepusti manj svetlobe
  - pojavi se lom žarkov

# DODANA LEČA

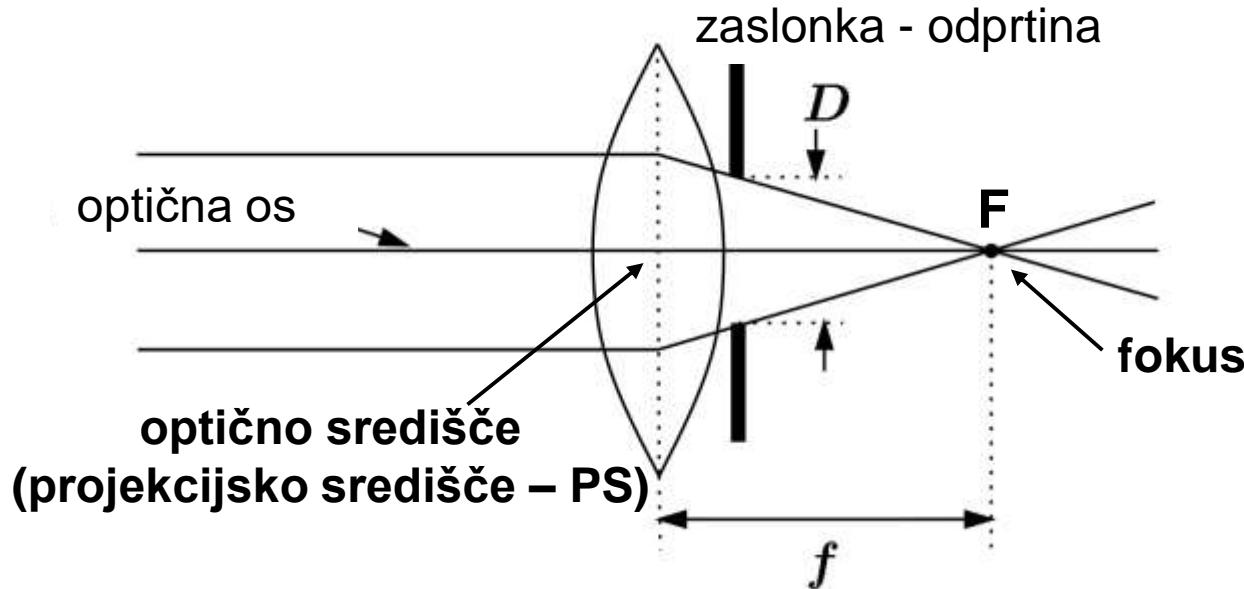
## 1. Usmerjanje svetlobe z dodano lečo



- samo pri določeni razdalji predmeta in slike od leče je slika ostra (fokusirana)
- pri drugih razdaljah nastane t. i. “krog negotovosti” (“circle of confusion”)
- oblika leče vpliva na razdaljo, pri kateri je slika fokusirana

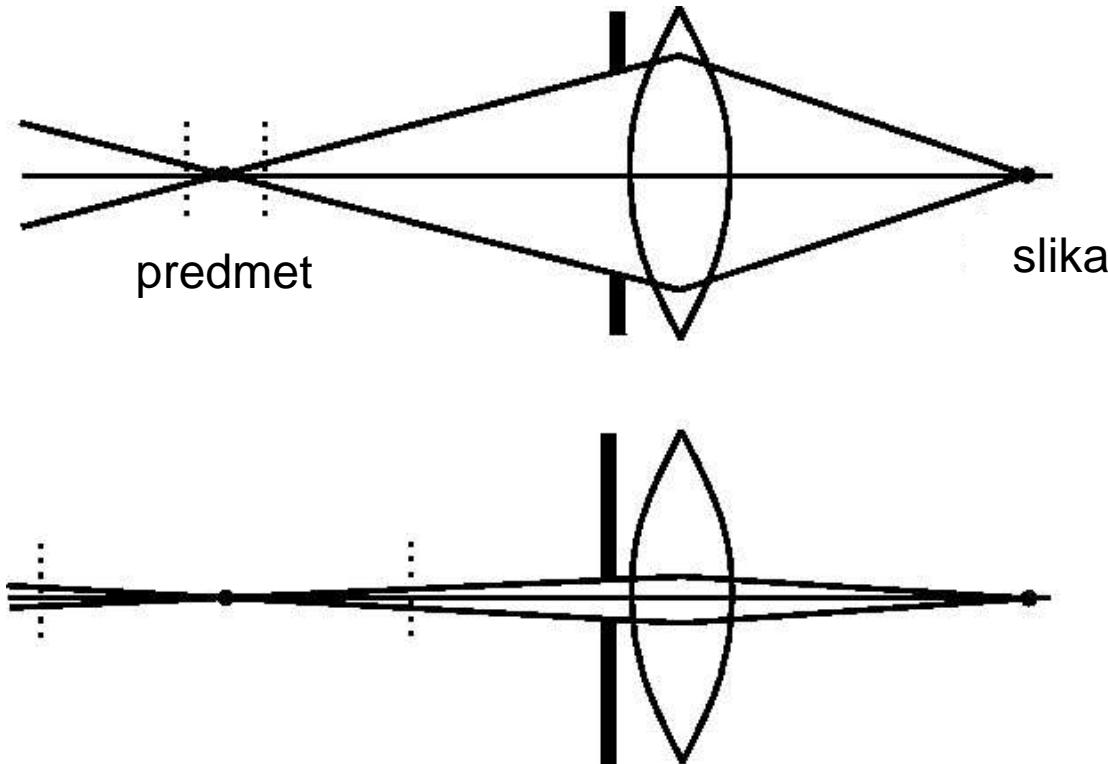
# OBRAVNAVA LEČE

## 1. Leča usmeri vzporedne žarke v isto zbirno, goriščno točko (fokus)



- vzporedni žarki se sekajo v zbirni točki F za lečo (gorišče, fokus)
- goriščna razdalja  $f$  je odvisna od oblike in lomnega količnika leče
- oblika leče vpliva na razdaljo, pri kateri je slika fokusirana
- središče leče pomeni projekcijsko središče – PS
- zaslonka je lahko na eni ali drugi strani leče – odprtina  $D$  omeji svetlobni snop
- leče so običajno sferične, ker se najlaže izdelajo

# GLOBINSKA OSTRINA

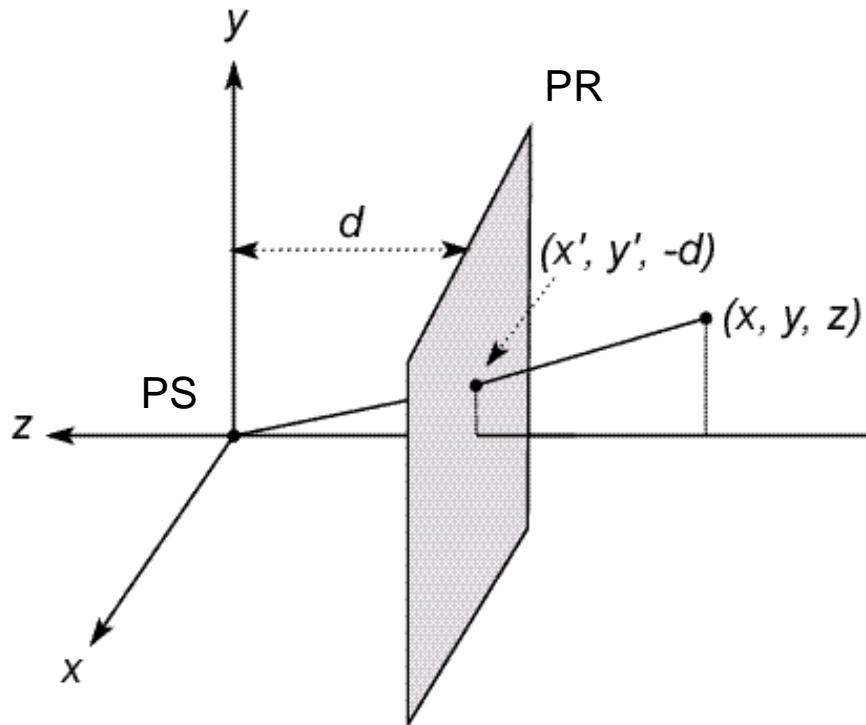


- Velikost odprtine v zaslonki vpliva na globinsko ostrino slike:
  - manjša odprtina poveča globinsko ostrino, torej podaljša odsek, na katerem je lahko predmet in je pri tem še vedno približno fokusiran (ima ostro sliko)

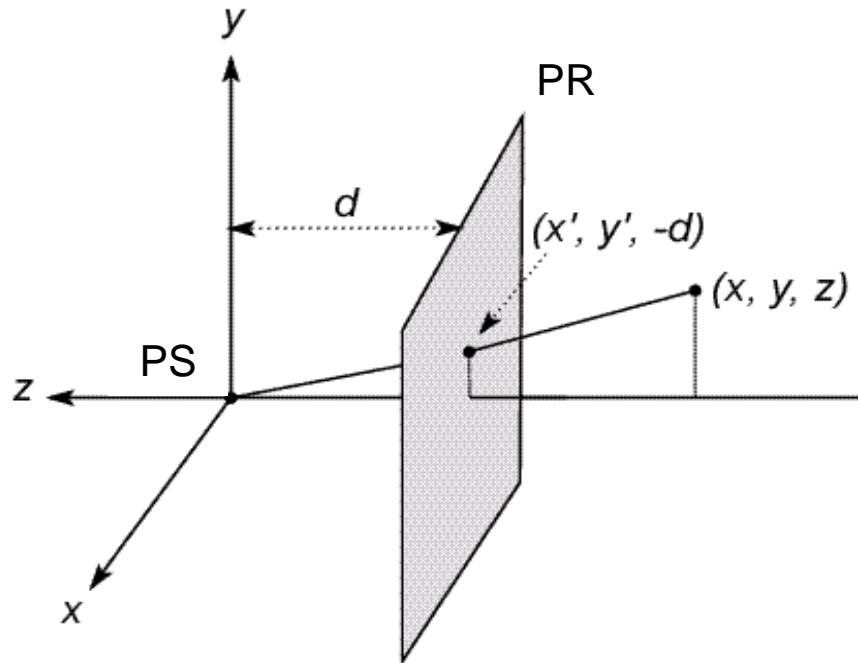
# PROJEKCIJSKI MODEL

## 1. Koordinatni sistem

- Uporabimo model s točkovno odprtino (*pin-hole model*)
- PS postavimo v koordinatno izhodišče
- Projekcijsko ravnino (PR) postavimo **pred** PS (na negativni del z-osi)
- Kamera gleda v negativno stran z-osi, kar pomeni, da je projekcija usmerjena proti pozitivnim z (desnosučni koordinatni sistem,  $x, y \rightarrow z$ )



# PROJEKCIJSKI MODEL (2)



## 2. Projekcijske enačbe

- Žarek iz opazovane točke  $(x, y, z)$  je usmerjen v PS – na PR dobimo točko:  
$$(x, y, z) \rightarrow (-d\frac{x}{z}, -d\frac{y}{z}, -d)$$
- Če se osredotočimo samo na PR (ne upoštevamo 3. dimenzije z):  
$$(x, y, z) \rightarrow (-d\frac{x}{z}, -d\frac{y}{z})$$

# HOMOGENE KOORDINATE

**Ali so projekcijske enačbe linearne?**

- Ne, ker deljenje z z vnaša nelinearnost.

**3. Rešitev ponujajo homogene koordinate (dodana še ena koordinata):**

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogene ravninske  
koordinate (2D)

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogene prostorske  
koordinate (3D)

**Preračun homogenih v nehomogene koordinate:**

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

# PERSPEKTIVNA PROJEKCIJA

4. Homogene koordinate omogočajo, da opišemo projekcijo z matriko:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix} \Rightarrow \left( -d\frac{x}{z}, -d\frac{y}{z} \right)$$

delimo s tretjo koordinato

Prišli smo do projekcijske matrike, ki jo lahko zapišemo še drugače:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ -z/d \end{bmatrix} \Rightarrow \left( -d\frac{x}{z}, -d\frac{y}{z} \right) \quad \text{← gledano na PR}$$

delimo s četrto koordinato

# PERSPEKTIVNA PROJEKCIJA (2)

Spreminjanje skale (povečava) ne vpliva na perspektivno projekcijo  
– je zanjo invariantna:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \\ 1 \end{bmatrix} \Rightarrow \left( -d\frac{x}{z}, -d\frac{y}{z} \right)$$

Projekcijsko matriko množimo z  $-d$ :

– predmet in njegovo razdaljo od projekcijske ravnine hkrati spremenimo za enak faktor, recimo ( $d \cdot x$ ,  $d \cdot y$ ,  $d \cdot z$ ):

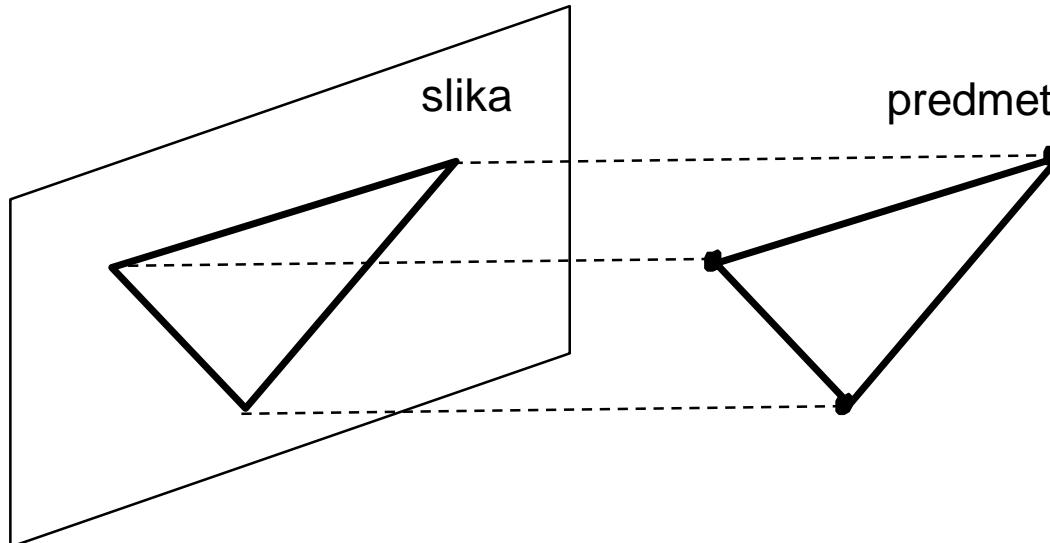
$$\begin{bmatrix} -d & 0 & 0 & 0 \\ 0 & -d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -dx \\ -dy \\ z \end{bmatrix} \Rightarrow \left( -d\frac{x}{z}, -d\frac{y}{z} \right)$$

- To potrjuje znano dejstvo, da so večji, a bolj oddaljeni predmeti videti manjši kot bližnji, a v resnici manjši predmeti.

# ORTOGRAFSKA PROJEKCIJA

Posebna vrsta perspektivne projekcije, imenovana tudi vzporedna:

- razdalja med PS in PR je neskončna



Projekcijska matrika:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

# DRUGE VRSTE PROJEKCIJ

Povečava (pomanjšava), imenovana tudi šibka perspektiva (*weak perspective*):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1/d \end{bmatrix} \Rightarrow (dx, dy)$$

Afina projekcija, imenovana tudi paraperspektiva:

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Ortografska projekcija in projekcija s šibko perspektivo sta izvedenki afine projekcije.

# 2D TRANSFORMACIJE

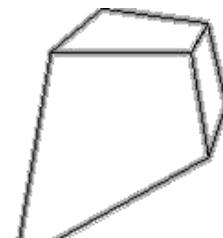
Transformation	Matrix	# DoF	Preserves	Icon
translation	$\left[ \begin{array}{c c} I & t \\ \hline \end{array} \right]_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\left[ \begin{array}{c c} R & t \\ \hline \end{array} \right]_{2 \times 3}$	3	lengths	
similarity	$\left[ \begin{array}{c c} sR & t \\ \hline \end{array} \right]_{2 \times 3}$	4	angles	
affine	$\left[ \begin{array}{c} A \\ \hline \end{array} \right]_{2 \times 3}$	6	parallelism	
projective	$\left[ \begin{array}{c} H \\ \hline \end{array} \right]_{3 \times 3}$	8	straight lines	

Povzeto po Richard Szeliski: Computer Vision: Algorithms and Applications, Springer, 2010.

# 3D TRANSFORMACIJE

Projekcijska  
15 prost. st.

$$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$$



Presečišča in dotikališča

Afina  
12 prost. st.

$$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$$



Paralelnost ravnin,  
razmerja volumnov,  
težišča, ravnina v  
neskončnosti  $\pi_\infty$

Podobnost  
7 prost. st.

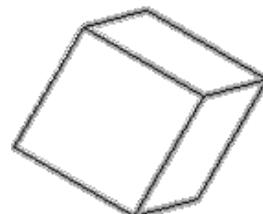
$$\begin{bmatrix} s R & t \\ 0^T & 1 \end{bmatrix}$$



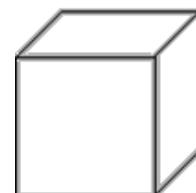
Koti, razmerja dolžin,  
absolutna stožnica  $\Omega_\infty$

Evklidska  
6 prost. st.

$$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$$

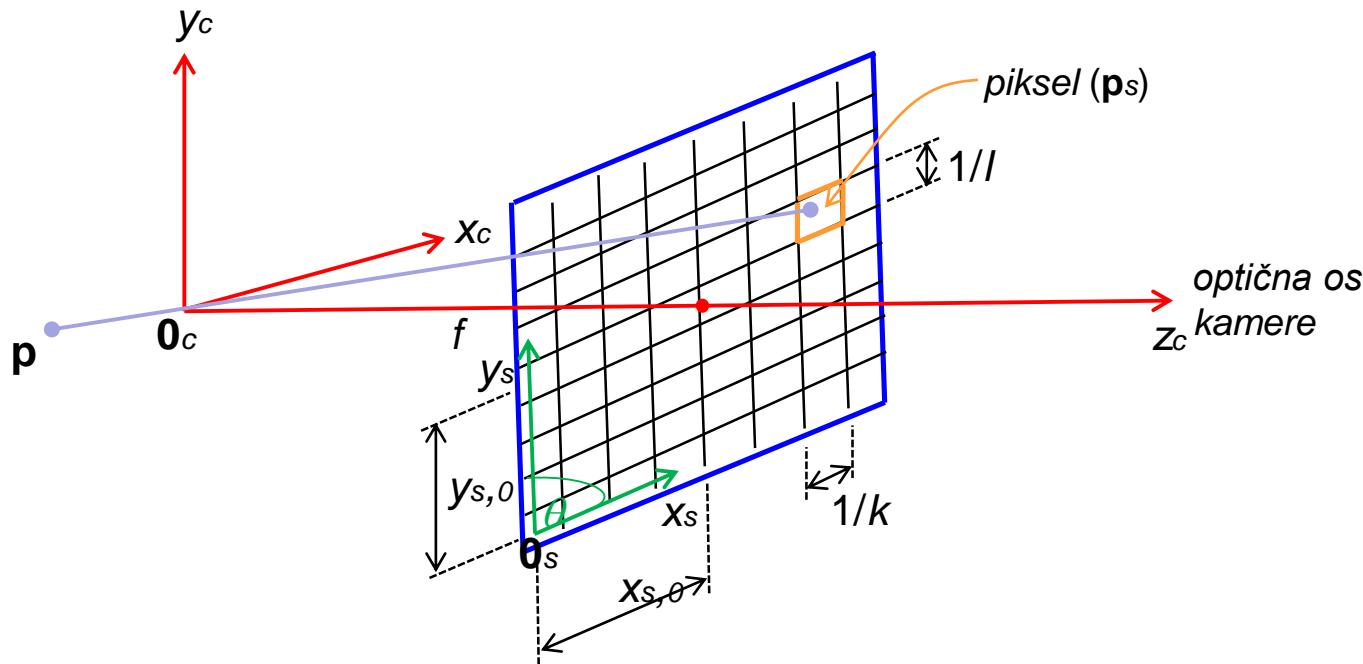


Volumen



# Kalibracija kamer

# LASTNOSTI CCD-KAMER



## Oznake:

$0_c$  – središče kamerinega koordinatnega sistema

$0_s$  – središče slikovnega koordinatnega sistema

$(x_{s,0};y_{s,0})$  – premik slikovnega središča od optične osi

$1/k$  – širina piksla

$1/l$  – višina piksla

$f$  – goriščna razdalja

$\theta$  – nagib slikovnih osi (optični element nima pravokotnih osi)

Notranji parametri kamere

# GEOMETRIJSKI MODEL KAMERE

## (notranji parametri)

- Slike, posnete s CCD-kamero, praviloma merimo s številom pikslov.
- Če je dimenzija piksla  $\frac{1}{k} \times \frac{1}{l}$ , potem lahko vzamemo  $k$  kot faktor povečave po širini in  $l$  kot faktor povečave po višini slike.
- Zato koordinate na sliki (v pikslih):

$$x_s = kf \frac{x}{z} \quad \text{in} \quad y_s = lf \frac{y}{z} \quad (k \text{ in } l \text{ imata enote piksel/m})$$

- Uvedimo nova faktorja povečave:  $f_x = kf$  in  $f_y = lf$
- Ko upoštevamo še premik središč, sledi:

$$x_s = f_x \frac{x}{z} + x_0 \quad \text{in} \quad y_s = f_y \frac{y}{z} + y_0$$

- Poševne osi slikovnega elementa vnesejo še dodatna odstopanja:

$$x_s = f_x \frac{x}{z} - f_x \operatorname{ctg} \theta \frac{y}{z} + x_0 \quad \text{in} \quad y_s = \frac{f_y}{\sin \theta} \frac{y}{z} + y_0$$

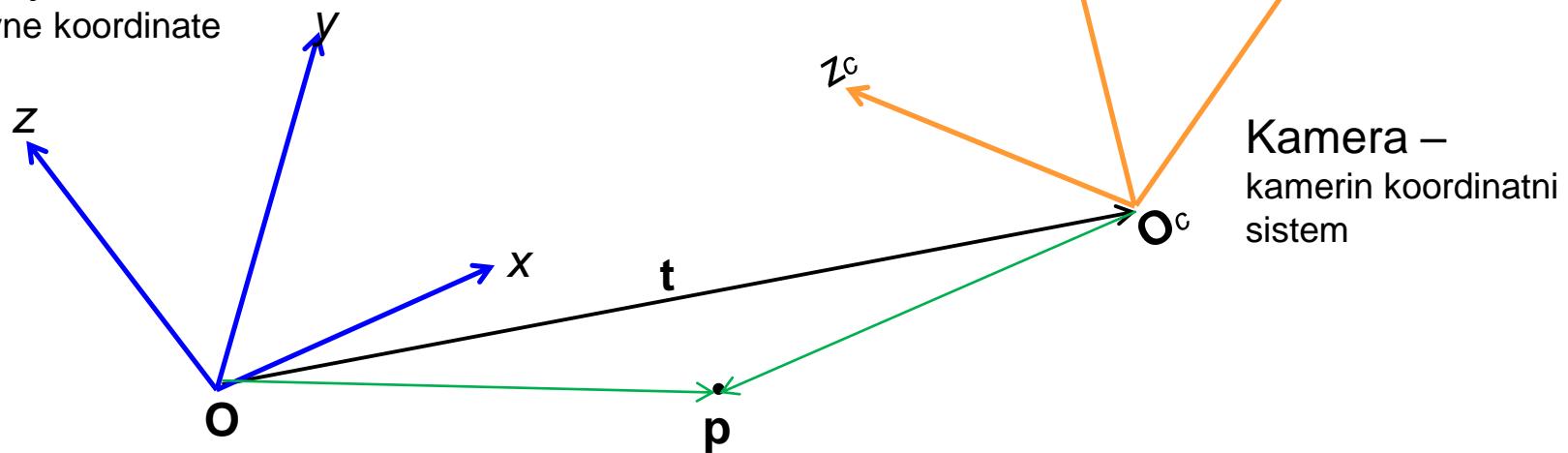
**K** – kalibracijska matrika kamere  
**M<sub>c</sub>** – projekcijska matrika kamere

- Matrični zapis:  $\mathbf{K} = \begin{bmatrix} f_x & -f_x \operatorname{ctg} \theta & x_0 \\ 0 & \frac{f_y}{\sin \theta} & y_0 \\ 0 & 0 & 1 \end{bmatrix}$ ,  $\mathbf{M}_c = [\mathbf{K} \ 0]$ ,  $\mathbf{p}_s = \frac{1}{z} \cdot \mathbf{M}_c \cdot \mathbf{p}_h \quad \mathbf{p}_h = [x \ y \ z \ 1]^T$

# GEOMETRIJSKI MODEL KAMERE

(zunanji parametri)

Zunanji svet –  
svetovne koordinate



Koordinatni sistem kamere ni poravnан s svetovnim koordinatnim sistemom:

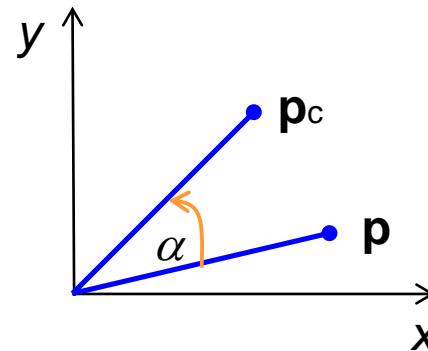
- izhodišče je premaknjeno za  $\overline{OO_c} = \mathbf{t}$  (vektor premika)
- središče kamere je v svetovnem koordinatnem sistemu  $(x_c, y_c, z_c) = (t_x, t_y, t_z)$
- kamera je v svetovnem koordinatnem sistemu zasukana za kote  $(\alpha, \beta, \gamma)$
- zasuk opišemo z matriko zasuka  $\mathbf{R}$

# GEOMETRIJSKI MODEL KAMERE

(toga transformacija koordinatnih sistemov)

Rotacijska matrika – zasuk okoli osi  $z$ :

$$\mathbf{p}_c = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \mathbf{p}$$



Zapis s homogenimi koordinatami:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}_c = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{R}_z \cdot \mathbf{p}_h$$

Zasuk in premik koordinatnega sistema:

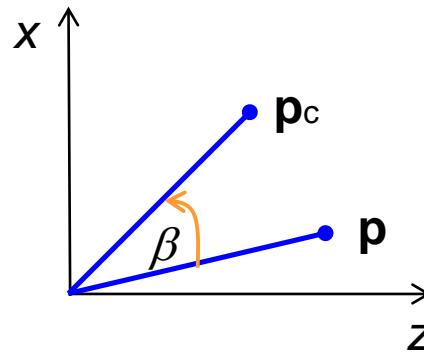
$$\mathbf{p}_{c,h} = \mathbf{R}_z \mathbf{p}_h + \mathbf{t}$$

# GEOMETRIJSKI MODEL KAMERE

(rotacijska matrika – zasuk okoli vseh osi)

Rotacijska matrika – zasuk okoli osi  $y$ :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{R}_y \cdot \mathbf{p}_h$$



Rotacijska matrika – zasuk okoli osi  $x$ :

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma & 0 \\ 0 & \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{R}_x \cdot \mathbf{p}_h$$

Skupna rotacijska matrika:

$$\mathbf{R} = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x$$

6 zunanjih parametrov za kamero:

- 3 koti za zasuk
- 3 odseki za premik

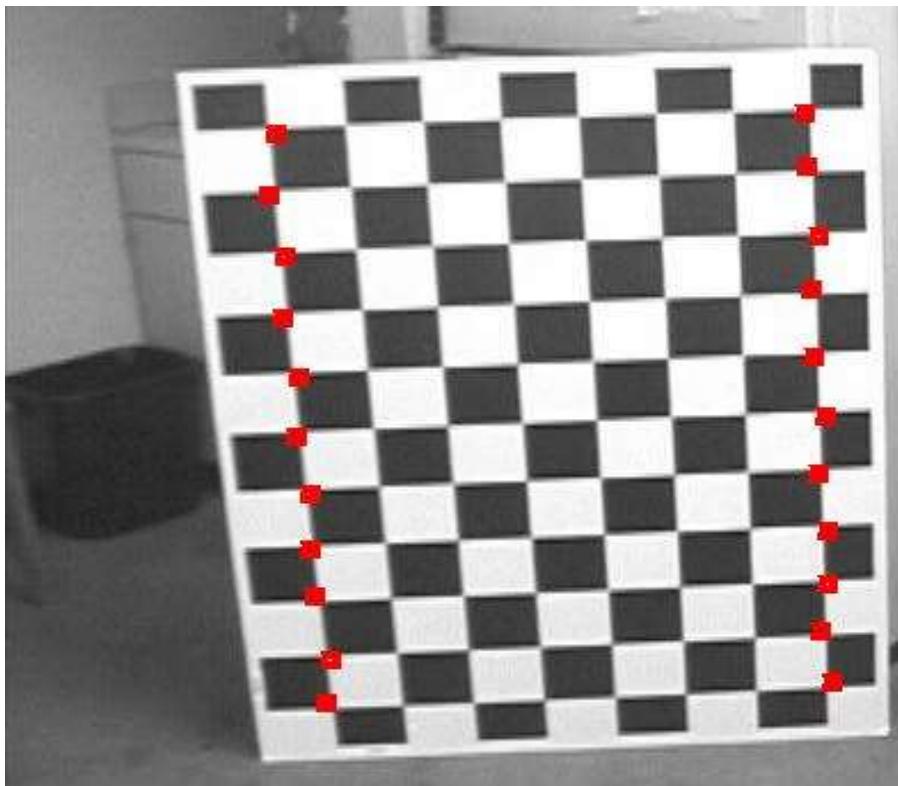
# DOLOČANJE PARAMETROV KAMERE

## (kalibracija)

1. Za kamero lahko določimo 5 notranjih in 6 zunanjih parametrov.
2. Notranji parametri:
  - $(x_{s,0}, y_{s,0})$  – premik slikovnega središča od optične osi
  - $f_x$  – goriščna razdalja  $f$ , deljena s širino piksla
  - $f_y$  – goriščna razdalja  $f$ , deljena z višino piksla
  - $\theta$  – nagib slikovnih osi (optični element nima pravokotnih osi)  $\Rightarrow \frac{f_x}{f_y}$  aspektno (projekcijsko) razmerje
3. Zunanji parametri:
  - kot zasuka okoli osi  $z$ :  $\alpha$
  - kot zasuka okoli osi  $y$ :  $\beta$
  - kot zasuka okoli osi  $x$ :  $\gamma$
  - premik vzdolž osi  $x$ :  $t_x$
  - premik vzdolž osi  $y$ :  $t_y$
  - premik vzdolž osi  $z$ :  $t_z$
4. Parametre kamere lahko izračunamo, če opazujemo sceno z zanimimi merami oz. razmerji.
5. Nastavimo sistem enačb – kalibracija.
6. Za kalibracijo se najpogosteje uporablja šahovnica.

# DOLOČANJE PARAMETROV KAMERE

(posnetek znanih točk – šahovnica)



Značilne točke na slikah lahko poiščemo na različne načine, recimo:

- s Cannyjevim postopkom opredelimo robove
- robove aproksimiramo z ravnimi črtami
- poiščemo presečišča teh črt

# SISTEM ENAČB ZA KALIBRACIJO KAMERE

Izhajamo iz (projekcijske) matrike kamere:

$$\mathbf{M} = [\mathbf{K} \ \mathbf{0}] [\mathbf{R} \ \mathbf{t}] = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_0^T \\ \mathbf{m}_1^T \\ \mathbf{m}_2^T \end{bmatrix}$$

Projekcija iz prostora na sliko za  $i$ -to točko:

$$x_s(i) = \frac{\mathbf{m}_0^T \mathbf{p}_h(i)}{\mathbf{m}_2^T \mathbf{p}_h(i)}; \quad \mathbf{p}_h(i) = \begin{bmatrix} x(i) \\ y(i) \\ z(i) \\ 1 \end{bmatrix}$$
$$y_s(i) = \frac{\mathbf{m}_1^T \mathbf{p}_h(i)}{\mathbf{m}_2^T \mathbf{p}_h(i)}$$

Rešiti je treba sistem enačb:

$$\begin{bmatrix} \mathbf{p}_h^T(1) & \mathbf{0}^T & -x_s(1)\mathbf{p}_h^T(1) \\ \mathbf{0}^T & \mathbf{p}_h^T(1) & -y_s(1)\mathbf{p}_h^T(1) \\ \vdots & & \\ \mathbf{p}_h^T(N) & \mathbf{0}^T & -x_s(N)\mathbf{p}_h^T(N) \\ \mathbf{0}^T & \mathbf{p}_h^T(N) & -y_s(N)\mathbf{p}_h^T(N) \end{bmatrix} \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix} = \mathbf{0} \quad \xrightarrow{\text{rešitev z lastnim vektorjem}}$$

Razcep na lastne vrednosti in vektorje (SVD) – rešitev:  
lastni vektor ob najmanjši lastni vrednosti

# RAZCEP MATRIKE KAMERE

1. Projekcijsko matriko kamere  $\mathbf{M}$  v splošnem zapišemo kot:

$$\mathbf{M} = \rho [\mathbf{A} \ \mathbf{b}] = \rho \begin{bmatrix} \mathbf{a}_0^T & b_0 \\ \mathbf{a}_1^T & b_1 \\ \mathbf{a}_2^T & b_2 \end{bmatrix},$$

pri čemer  $\rho$  pomeni neznano povečavo, saj ima rekonstruirana matrika  $\mathbf{M}$  vedno normo (povečavo) 1.

2. Nadalje velja:

$$\rho \begin{bmatrix} \mathbf{a}_0^T \\ \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} = \begin{bmatrix} f_x \mathbf{r}_0^T - f_x \operatorname{ctg} \theta \mathbf{r}_1^T + x_0 \mathbf{r}_2^T \\ \frac{f_y}{\sin \theta} \mathbf{r}_1^T + y_0 \mathbf{r}_2^T \\ \mathbf{r}_2^T \end{bmatrix}$$

# RAZCEP MATRIKE KAMERE

## (notranji parametri)

3. Notranje parametre kamere določimo iz:

$$\rho = \frac{\varepsilon}{|\mathbf{a}_2|}$$

$$\mathbf{r}_2 = \rho \mathbf{a}_2$$

$$x_0 = \rho^2 (\mathbf{a}_0^T \cdot \mathbf{a}_2)$$

$$y_0 = \rho^2 (\mathbf{a}_1^T \cdot \mathbf{a}_2)$$

pri čemer je  $\varepsilon = \pm 1$ .

Vrstice rotacijske matrike R imajo dolžino 1 in so ortogonalne. Iz tega sledijo relacije za razcep.

4.  $\theta$  je vedno blizu  $\pi/2$ , zato je  $\sin \theta$  pozitiven in velja:

$$\cos \theta = -\frac{(\mathbf{a}_0 \times \mathbf{a}_2) \cdot (\mathbf{a}_1 \times \mathbf{a}_2)}{|\mathbf{a}_0 \times \mathbf{a}_2| |\mathbf{a}_1 \times \mathbf{a}_2|}$$

$$f_x = \rho^2 |\mathbf{a}_0 \times \mathbf{a}_2| \sin \theta$$

$$f_y = \rho^2 |\mathbf{a}_1 \times \mathbf{a}_2| \sin \theta$$

# RAZCEP Matrike kamere

(zunanji parametri)

5. Zasuk kamere (dve možnosti glede na  $\varepsilon$ ):

$$\mathbf{r}_0 = \frac{\rho^2 \sin \theta}{f_y} (\mathbf{a}_1 \times \mathbf{a}_2) = \frac{1}{|\mathbf{a}_1 \times \mathbf{a}_2|} (\mathbf{a}_1 \times \mathbf{a}_2)$$

$$\mathbf{r}_1 = \mathbf{r}_2 \times \mathbf{r}_0$$

6. Premik kamere – velja:

$$\mathbf{K} \cdot \mathbf{t} = \rho \mathbf{b} \quad \Rightarrow \quad \mathbf{t} = \rho \mathbf{K}^{-1} \mathbf{b}$$

Predznak za premik je v praksi odvisen od tega, ali je izhodišče svetovnega koordinatnega sistema pred kamero ali za njo.

Minimalno število kalibracijskih točk:

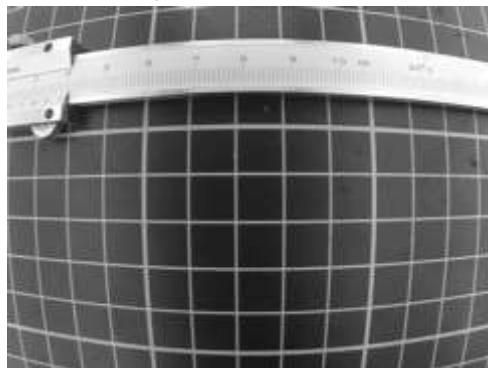
- ker imamo 11 neznank (5 notranjih in 6 zunanjih parametrov), je načelno dovolj 6 točk – vsaka da 2 enačbi (za x- in y-koordinato)
- v praksi več kot 6 – izkustveno pravilo za  $n$  neznank svetuje  $5n$  enačb

# KALIBRACIJA KAMERE – PROBLEM RADIALNIH POPAČENJ

1. Radialna popačenja nastanejo zaradi slabih (nesferičnih) leč:

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} x_{s,d} \\ y_{s,d} \end{bmatrix} (1 + k_1 d^2 + k_2 d^4 + \dots)$$

- $d$  pomeni razdaljo točke od središča slike,  $k_i$  pa so koeficienti popačenja (navadno upoštevamo samo prva dva)
- $x_{s,d}$  in  $y_{s,d}$  – koordinati popačene (premaknjene) točke



2. Kamero lahko kalibriramo tudi glede na radialno popačenje:

- matriko kamere določimo enako kot prej, a vsaj z 8 točkami
- podobno kot prej izračunamo notranje parametre (namesto  $\rho$  določimo aspekt)
- enako kot prej določimo zunanje parametre, razen premika v smeri  $z$
- premik v smeri  $z$  (in  $\rho$ ) določimo glede na popačenje s pomočjo dodatnih točk

# RAZLIČNE METODE ZA KALIBRACIJO KAMER

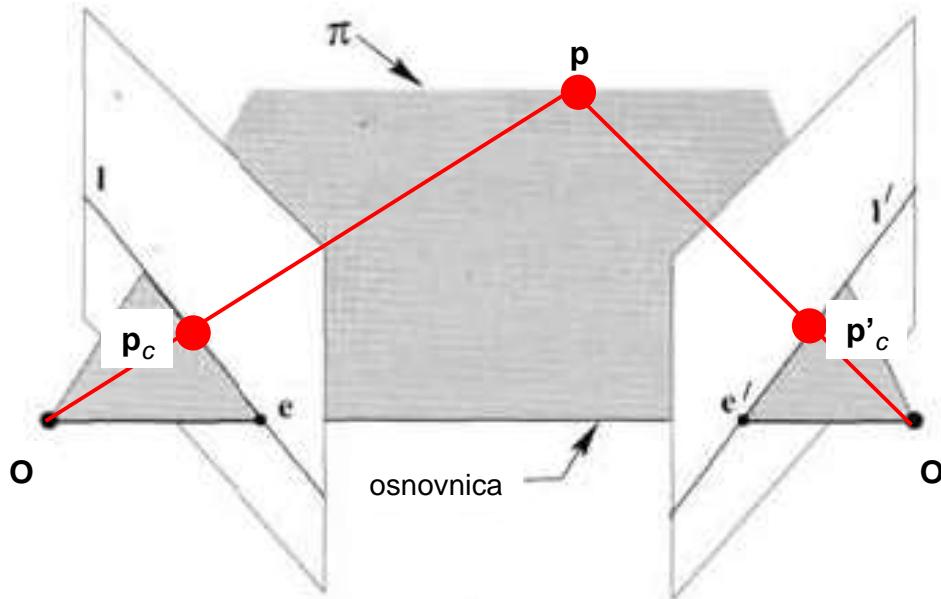
Kalibracija kamer je zelo široko raziskano področje z mnogimi metodami, kot na primer:

1. metoda z znanimi točkami (opisana tukaj)
2. metoda s točkami na premici
3. metoda s točkami na več ravninah (več kot 3 ravnine)
4. metoda s ponornimi točkami (*vanishing points*), točkami v neskončnosti

# Epipolarna geometrija

# DVA POGLEDA NA ISTO SCENO

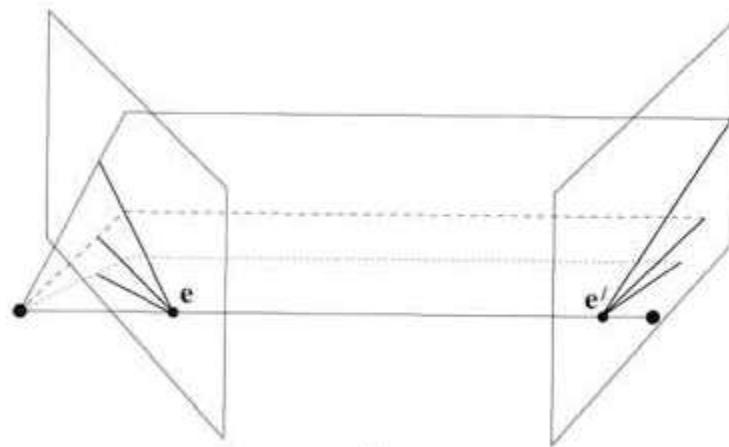
Imejmo dve kamери, ki snemata isto sceno iz dveh različnih zornih kotov:



- $O$  in  $O'$  – optični središči kamer, povezuje ju osnovnica
- $e$  in  $e'$  – epipola: epipol je točka, ki ustreza sliki (virtualnega) središča druge kamere
- $\pi$  - epipolarna ravnina, določena z optičnima središčema  $O$  in  $O'$  ter točko  $p$  v prostoru
- $I$  in  $I'$  – epipolarni premici, na kateri se projicirajo vse točke iz pripadajoče epipolarne ravnine

# EPIPOLARNE RAVNINE

1. Vse ravnine, ki gredo skozi osnovnico, so epipolarne ravnine – šop ravnin.
2. Šop epipolarnih ravnin vsebuje vse točke prostora.
3. Vsaka epipolarna ravnina seka slikovno ravnino v epipolarni premici.
4. Vse epipolarne premice ene slikovne ravnine se sekajo v epipolu in tvorijo šop premic.



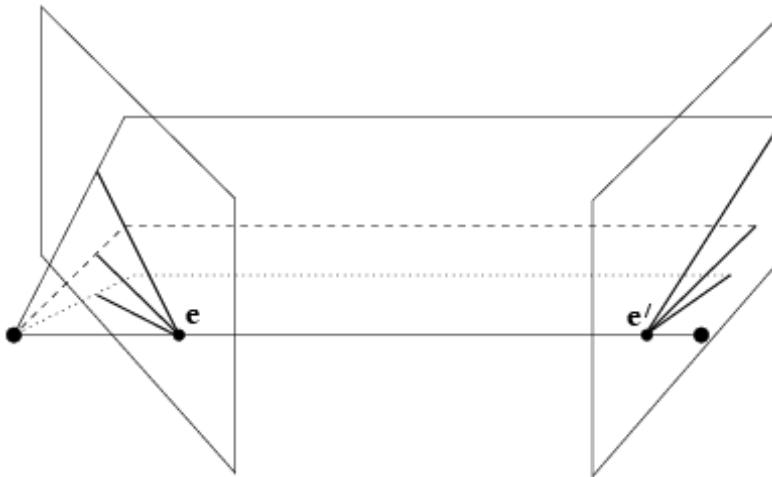
Epipol pomeni tudi ponorno točko pri premikih v smeri osnovnice – preprosto določljivo pri slikah kvadrov.

# POMEN EPIPOLARNE GEOMETRIJE

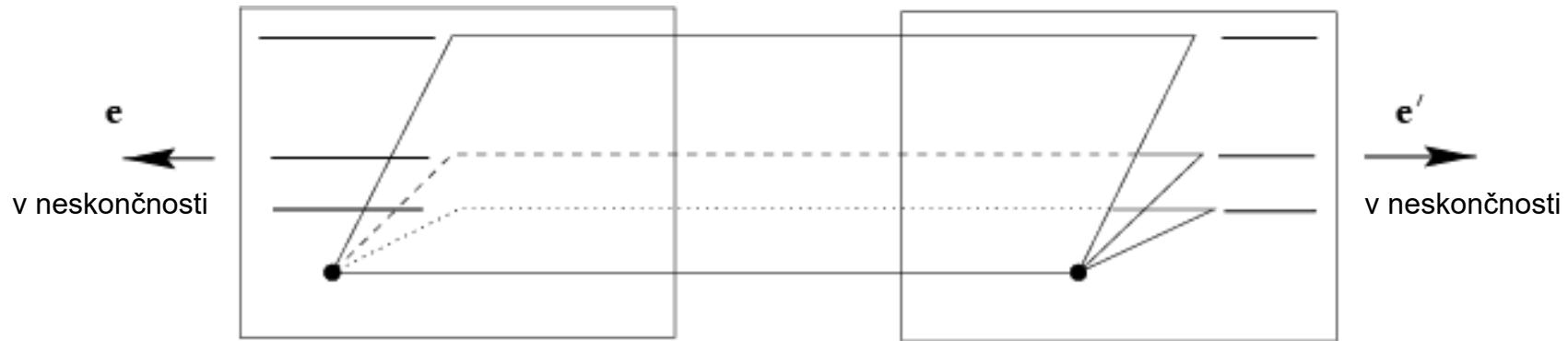
Z epipolarno geometrijo lahko odgovorimo na tri vprašanja:

1. Če poznamo točko  $\mathbf{p}_c$  na prvi sliki, kako to omejuje lego korespondenčne točke  $\mathbf{p}'_c$  na drugi sliki? (korespondenčna geometrija)
2. Če imamo množico korespondenčnih točk  $\mathbf{p}_{c,i} \leftrightarrow \mathbf{p}'_{c,i}$ ,  $i = 1, \dots, N$ , ali lahko iz njih določimo geometrijsko transformacijo med slikama oz. pogledoma? (geometrija kamere – gibanje)
3. Če imamo korespondenčni točki  $\mathbf{p}_c \leftrightarrow \mathbf{p}'_c$  in podatke o kamerah, ali lahko rekonstruiramo lego točke  $\mathbf{p}$  v prostoru? (scenska geometrija - struktura)

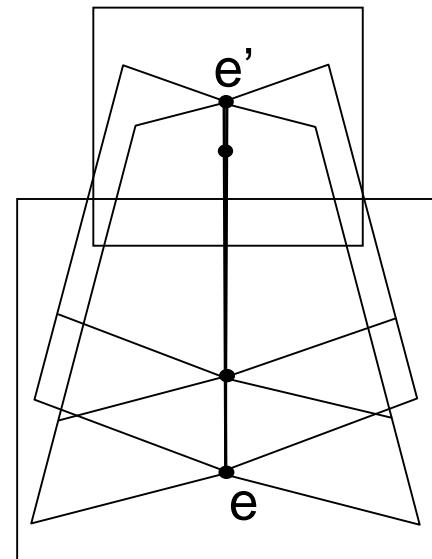
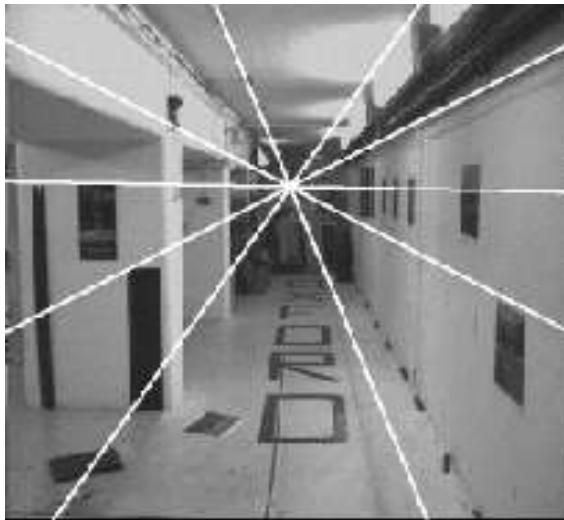
# Primer dveh konvergentnih kamer



# Primer kamer, premaknjenih vzporedno s slikovno ravnino



# Primer s premikanjem kamere naprej



# EPIPOLARNA OMEJITEV

1. Imejmo prostorsko točko  $\mathbf{p}$  in njeni sliki  $\mathbf{p}_c$  in  $\mathbf{p}'_c$ .
2. Osnovnica in ena od slik točke definirata epipolarno ravnino, na kateri zanesljivo leži tudi druga slika točke:

$$\overline{\mathbf{O}\mathbf{p}_c} \cdot (\overline{\mathbf{O}\mathbf{O}'} \times \overline{\mathbf{O}\mathbf{p}'_c}) = 0 \quad (\text{ortogonalnost})$$

3. Ta pogoj lahko zapišemo v koordinatnem sistemu prve kamere:

$$\mathbf{p}_c^T \cdot (\mathbf{t} \times \mathbf{R}\mathbf{p}'_c) = 0$$

4. Vektorski produkt

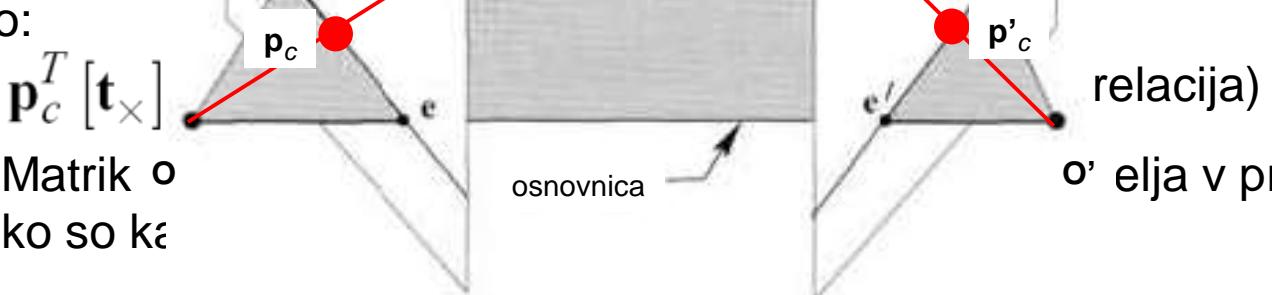
$$\mathbf{a} \times \mathbf{b} =$$

zato:

$$\mathbf{p}_c^T [\mathbf{t} \times]$$

5. Matrik o

ko so kā



# OSNOVNA MATRIKA

1. Osnovna matrika  $\mathbf{E}$  ima 9 koeficientov, vendar ima le 5 prostostnih stopenj: 3 za zasuk in 2 za premik, ker ima samo dve enaki singularni vrednosti.
  2. Matrika  $\mathbf{E}$  ne določa povečave in je od nje neodvisna.
  3. Rang matrike  $\mathbf{E}$  je 2.
  4. Produkt  $\mathbf{E}\mathbf{p}'_c$  določa epipolarno premico, ki gre skozi  $\mathbf{p}_c$  in epipol  $\mathbf{e}$  na prvi sliki:  $\mathbf{l} = \mathbf{E}\mathbf{p}'_c$ . Velja tudi:  $\mathbf{l}' = \mathbf{E}^T\mathbf{p}_c$ .
  5. Osnovna matrika  $\mathbf{E}$  torej preslikava točke z ene slike v epipolarne premice, na katerih lahko najdemo korespondenčne točke na drugi sliki.
  6. Matrika vrednos
  7. Singula vrednos
  8. Osnovno zato jih
- 
- enaki singularni  
icelno singularno  
vih parametrih kamere,

# TEMELJNA MATRIKA

1. Vzemimo matriko kamere  $\mathbf{K}$  ( $3 \times 3$ ), ki vsebuje notranje kamerine parametre.
2. Slikovni točki  $\mathbf{p}_s$  in  $\mathbf{p}'_s$  imata koordinate v pikslih in pripadajočim točkama  $\mathbf{p}_c$  in  $\mathbf{p}'_c$ , ki sta v koordinatah kamere (za vse  $\mathbf{p} = [x \ y \ 1]^T$ ).
3. Velja:  $\mathbf{p}_c = \mathbf{K}^{-1} \mathbf{p}_s$  in  $\mathbf{p}'_c = \mathbf{K}'^{-1} \mathbf{p}'_s$
4. In naprej:

$$\mathbf{p}_c^T \mathbf{E} \mathbf{p}'_c = \mathbf{p}_s^T \mathbf{K}^{-T} \mathbf{E} \mathbf{K}'^{-1} \mathbf{p}'_s = \mathbf{p}_s^T \mathbf{F} \mathbf{p}'_s = 0$$

5. Matriko  $\mathbf{F}$  imenujemo temeljna matrika in opisuje nekalibrirane razmere:

$$\mathbf{F} = \mathbf{K}^{-T} \mathbf{E} \mathbf{K}'^{-1}$$

6. Temeljna matrika ima 7 prostostnih stopenj (rang 2) in povezuje točke na eni sliki s pripadajočimi epipolarnimi premicami na drugi sliki:  
 $\mathbf{l}_s = \mathbf{F} \mathbf{p}'_s$  in  $\mathbf{l}'_s = \mathbf{F}^T \mathbf{p}_s$ .
7. Temeljna matrika  $\mathbf{F}$  določa s singularnim vektorjem, ki pripada singularni vrednosti 0, epipol  $\mathbf{e}'$ . Epipol  $\mathbf{e}$  določa matrika  $\mathbf{F}^T$ .

# RAČUNANJE TEMELJNE MATRIKE

1. Temeljno matriko lahko računamo iz korespondence točk na dveh slikah (pogledih na isto sceno), saj velja:

$$\mathbf{p}_{s,i}^T \mathbf{F} \mathbf{p}_{s,i}' = 0 \quad \text{za } i\text{-to točko}$$

2. Za par točk  $(x, y, 1)^T$  in  $(x', y', 1)^T$  torej velja:

$$xx' f_{00} + xy' f_{01} + xf_{02} + yx' f_{10} + yy' f_{11} + yf_{12} + x' f_{20} + y' f_{21} + f_{22} = 0$$

3. Za  $N$  ujemajočih se točk:

$$\begin{bmatrix} x_0 x_0' & x_0 y_0' & x_0 & y_0 x_0' & y_0 y_0' & y_0 & x_0' & y_0' & 1 \\ \vdots & & & & & & & & \\ x_{N-1} x_{N-1}' & x_{N-1} y_{N-1}' & x_{N-1} & y_{N-1} x_{N-1}' & y_{N-1} y_{N-1}' & y_{N-1} & x_{N-1}' & y_{N-1}' & 1 \end{bmatrix} \begin{bmatrix} f_{00} \\ f_{01} \\ f_{02} \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{20} \\ f_{21} \\ f_{22} \end{bmatrix} = \mathbf{A}\mathbf{f} = 0$$

4. Iščemo rešitev homogenega sistema enačb z 8 neodvisnimi spremenljivkami, zato potrebujemo najmanj 8 enačb oz. slikovnih parov točk – **algoritem 8 točk**.

# Algoritem 8 točk

1. Po algoritmu 8 točk lahko temeljno matriko ocenimo do povečave (*scaling*) natančno, če razpolagamo vsaj z 8 korespondenčnimi točkami (praviloma vzamemo  $N>8$ ).
2. Algoritem:
  - vzamemo  $N$  korespondenčnih točk ( $N>8$ )
  - določimo matriko  $\mathbf{A}$
  - matriko razcepimo na singularne vrednosti (SVD):
$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$
  - elementi temeljne matrike  $\mathbf{F}$  so vsebovani v stolpcu, ki v matriki  $\mathbf{V}$  pripada najmanjši singularni vrednosti
  - ker zaradi šuma in nenatančnosti korespondenčnih točk matrika  $\mathbf{F}$  navadno ni singularna, singularnost vsilimo:
    - dobljeno matriko  $\mathbf{F}$  razcepimo na singularne vrednosti (SVD)
$$\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$
    - najmanjšo singularno vrednost na diagonali  $\mathbf{D}$  postavimo na  $0 \Rightarrow \mathbf{D}'$
    - izračunamo popravljenou oceno temeljne matrike:
$$\mathbf{F}' = \mathbf{U}\mathbf{D}'\mathbf{V}^T$$

# Poravnavo slik

# IZHODIŠČA

## 1. Definicija

**Poravnavna slik** (*image registration*) je postopek, pri katerem skušamo čim natančneje prekrivti dve sliki, ki sta bili posneti v različnih časih, iz različnih zornih kotov ali z različnimi senzorji, ali pa primerjamo posneto sceno z modelom.

Izvorno (source) ali opazovano sliko (*sensed*) geometrijsko poravnavamo z referenčno (*reference*) ali ciljno sliko (*target*).

## 2. Razdelitev algoritmov za poravnavo

### a) slikovni prostor

- temelječi na intenziteti pikslov
- temelječi na značilnicah slik
- temelječi na linearnih transformacijah (toge, affine)
- temelječi na nelinearnih transformacijah (netoge – elastične)

### b) frekvenčni prostor

- temelječi na faznem in amplitudnem spektru

# UPORABNOST PORAVNAVE SLIK

1. Poravnavo je potrebna, če hočemo primerjati ali združevati (spajati) slike, ki izhajajo iz različnih meritev (snemanj).

## 2. Področja uporabe

### a) daljinsko zaznavanje (*remote sensing*)

- multispektralna klasifikacija
- opazovanje okolja in sprememb
- tvorba mozaičnih slik
- napovedovanje vremena
- vključevanje v geografske informacijske sisteme (GIS)

### b) medicina

- združevanje posnetkov CT in NMR
- opazovanje rasti tumorjev
- preverjanje učinkov terapije
- primerjava posnetkov z anatomskimi atlasi
- usklajevanje operativnih modelov in pacienta med operacijo

### c) kartografija

- posodabljanje zemljevidov

### d) vojska

- razpoznavanje objektov
- slednje objektom

### e) šport

- razpoznavanje elementov igre in sledenje igri itd.

# Zgled za poravnavo slik



Opazovana  
slika



Referenčna  
slika



# PARAMETRIČNA PORAVNAVA SLIK

1. O parametrični poravnavi govorimo, če je transformacija izvirne (opazovane) slike predstavljena z manjšim številom parametrov, recimo z rotacijo in premikom cele slike. V nasprotnem primeru govorimo o neparametrični poravnavi slik, recimo s funkcijsko relacijo vseh slikovnih pikslov.
2. Parametrični model
  - a) s slikovnimi značilnicami
    - na izvorni in ciljni sliki določimo značilne ali nadzorne točke (*control points*)
    - odstranimo nezanesljive točke
    - točke z obeh slik transformiramo po parametričnem modelu
    - poiščemo korespondenco (ujemanje) med točkami izvirne in ciljne slike
    - optimiziramo ujemanje in glede na to določimo parametre za poravnavo
  - b) s slikovnimi pikslji (intenzivnostjo pikslov)
    - piksle izvirne slike premaknemo in spremenimo njihovo intenzivnost glede na parametrični model
    - po izbranem optimizacijskem kriteriju kar najbolje poravnamo transformirano izvorno sliko s ciljno sliko
    - pri optimalnem ujemanju določimo parametre za poravnavo

# PARAMETRIČNI MODEL ZA PORAVNAVO

1. Imejmo izvorno sliko  $I$  s piksli  $I(i,j)$  na koordinatah  $(i,j)$  in ciljno sliko s piksli  $I_r(i,j)$ .
2. Izvorno sliko želimo poravnati s ciljno sliko, zato jo transformiramo s parametri  $\Theta$  in dobimo transformirano sliko  $I_t$  s piksli  $I_t(i,j)$ .
3. Koordinati pikslov  $(i,j)$  v izvorni sliki parametrično pretvorimo v koordinati  $x_\Theta(i,j)$  in  $y_\Theta(i,j)$ .
4. Na transformirani sliki dobimo zato piksle

$$I_t(i,j) = I(x_\Theta(i,j), y_\Theta(i,j)).$$

5. Iščemo največje možno ujemanje med transformirano izvorno sliko  $I_t(i,j)$  in ciljno sliko  $I_r(i,j)$ .
6. Najboljše ujemanje iščemo z eno od optimizacijskim metod, recimo z najmanjšo vsoto kvadratnih razlik (*sum of squared differences - SSD*):

$$E(\Theta) = \sum_{\forall(i,j) \in \mathbf{I}} (I(x_\Theta(i,j), y_\Theta(i,j)) - I_r(i,j))^2$$

7. Transformirani koordinati  $(x_\Theta, y_\Theta)$  bi morali biti celoštevilski, a večinoma nista, zato je potreba korekcija – recimo z bilinearno interpolacijo.

# BILINEARNA INTERPOLACIJA

1. Vzemimo, da sta transformirani koordinati piksla  $(x_\Theta, y_\Theta)$  neceloštevilski, zato ju zapišemo kot vsoto celoštevilskega in frakcijskega dela:

$$x_\Theta = x_i + x_f \text{ in } y_\Theta = y_i + y_f$$

2. V okolici teh neceloštevilskih koordinat izberemo štiri točke s celoštevilskimi koordinatami:

$$(x_i, y_i), (x_i+1, y_i), (x_i, y_i+1), (x_i+1, y_i+1)$$

3. Vrednost piksla pri neceloštevilskih koordinatah interpoliramo po naslednjem postopku:

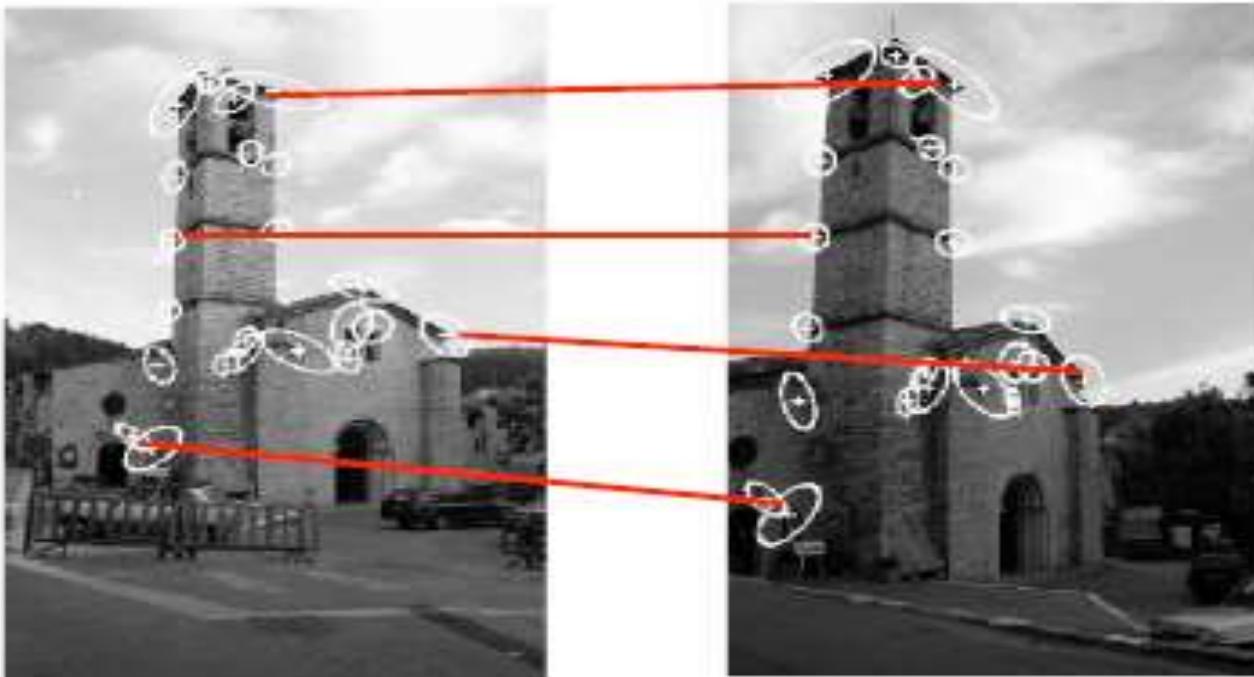
$$I_1(x_i, y_i + y_f) = (1 - y_f)I(x_i, y_i) + y_f I(x_i, y_i + 1)$$

$$I_2(x_i + 1, y_i + y_f) = (1 - y_f)I(x_i + 1, y_i) + y_f I(x_i + 1, y_i + 1)$$

$$I(x_i + x_f, y_i + y_f) = (1 - x_f)I_1(x_i, y_i + y_f) + x_f I_2(x_i + 1, y_i + y_f)$$

# PORAVNAVA SLIK Z NADZORNIMI TOČKAMI

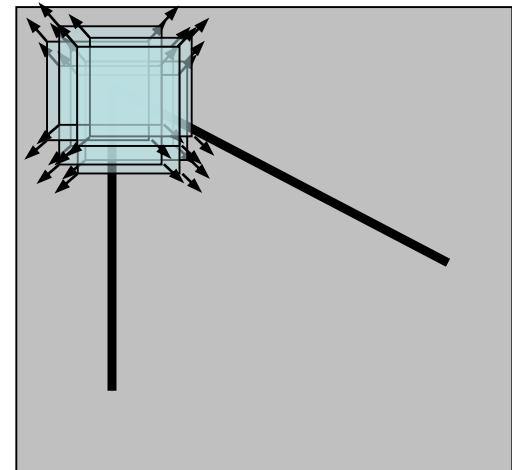
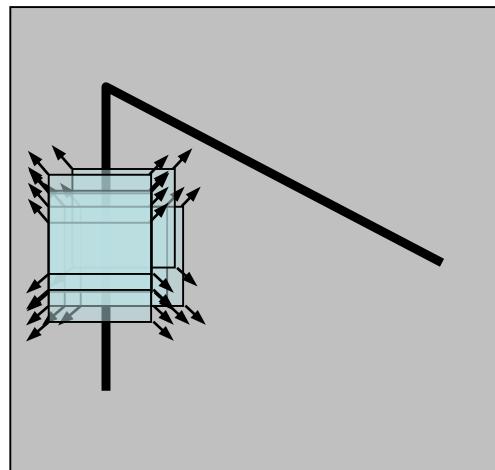
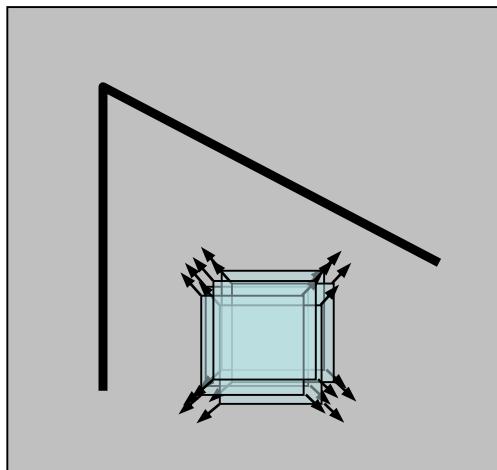
1. Na slikah iste scene, posnete iz dveh zornih kotov, najdemo korespondenčne nadzorne točke:



2. Izračunamo transformacijsko matriko, ki preslika točke ene slike v korespondenčne točke druge slike.
3. S to matriko lahko nato sliki medsebojno poravnamo (obdelamo vse točke).

# NADZORNE TOČKE – OGLIŠČA

1. Nadzorne točke za poravnavo slik so lahko maloštevilne, a morajo biti izrazite - na primer oglišča.
2. Za določanje oglišč se uporablja Harrisov detektor, ki uporablja ustrezeno majhen operator (okno).



**Enotno območje:**  
v nobeni smeri ne  
sme biti sprememb

**Rob:**  
ne sme biti sprememb  
v smeri roba

**Oglišče:**  
opazne spremembe v  
vseh smereh

# ZASNOVA HARRISOVEGA DETEKTORJA

1. Imejmo sliko  $I$  s pikslom  $I(x,y)$  na koordinatah  $(x,y)$ .
2. Premaknimo se za  $(\Delta x, \Delta y)$  in izračunajmo vsoto kvadriranih razlik v izbranem oknu:

$$E(\Delta x, \Delta y) = \sum_{x,y} w(x, y) [I(x + \Delta x, y + \Delta y) - I(x, y)]^2$$

pri čemer pomeni  $w(x,y)$  okensko funkcijo, recimo pravokotno ali Gaussovo.

3.  $I(x + \Delta x, y + \Delta y)$  izrazimo s prvim členom razvoja v Taylorjevo vrsto:

$$I(x + \Delta x, y + \Delta y) \approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y$$

$$= I(x, y) + \left[ I_x(x, y), I_y(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right]$$

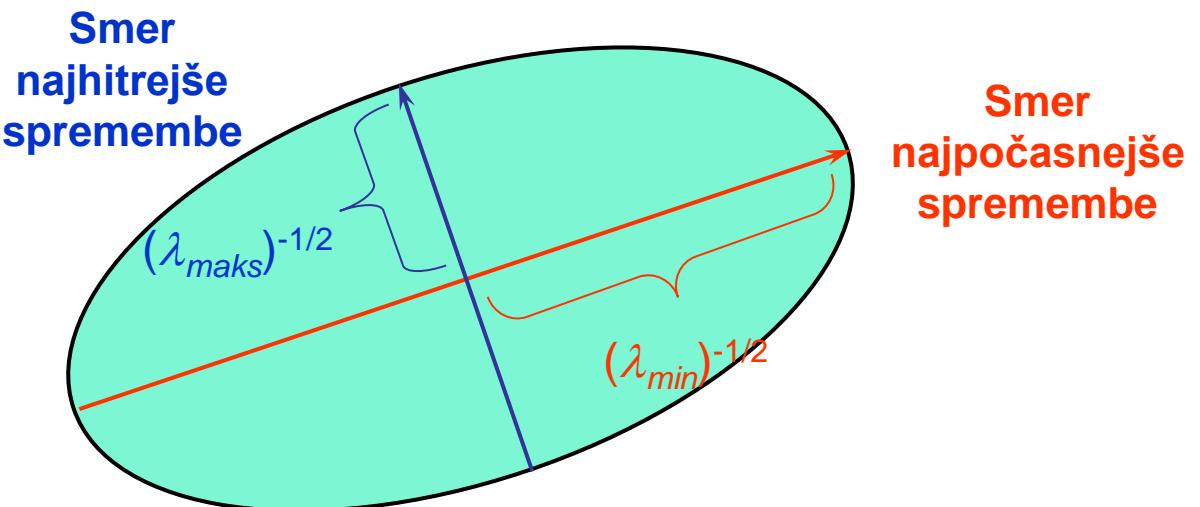
$$E(\Delta x, \Delta y) = \sum_{x,y} w(x, y) \left\{ \left[ I_x(x, y), I_y(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right] \right\}^2 \stackrel{\Downarrow}{=} [\Delta x, \Delta y] \mathbf{Q}(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

# ZASNOVA HARRISOVEGA DETEKTORJA (2)

1. Matrika  $\mathbf{Q}(x,y)$  vsebuje vsote odvodov po  $x$  in  $y$ , torej  $I_x(x,y)$  in  $I_y(x,y)$ , za vse piksle v oknu  $w(x,y)$ :

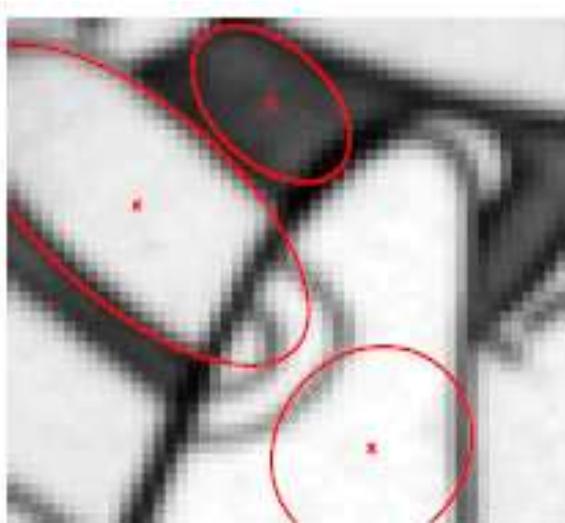
$$\mathbf{Q}(x, y) = \begin{bmatrix} \sum_{x, y} I_x^2(x, y) & \sum_{x, y} I_x(x, y)I_y(x, y) \\ \sum_{x, y} I_x(x, y)I_y(x, y) & \sum_{x, y} I_y^2(x, y) \end{bmatrix}$$

2.  $E(\Delta x, \Delta y) = \text{konst.}$  pomeni elipso, katere osi sta določeni z lastnima vrednostmi matrike  $\mathbf{Q}$ , to sta  $\lambda_{\text{maks}}$  in  $\lambda_{\text{min}}$ , medtem ko njeno rotacijo določajo lastni vektorji matrike  $\mathbf{Q}$ .

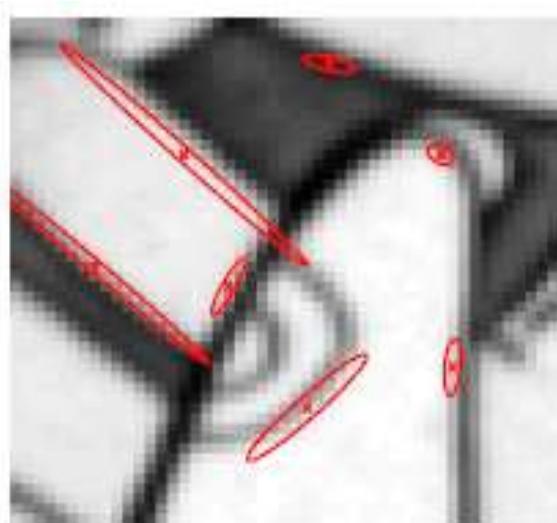


# ZASNOVA HARRISOVEGA DETEKTORJA (3)

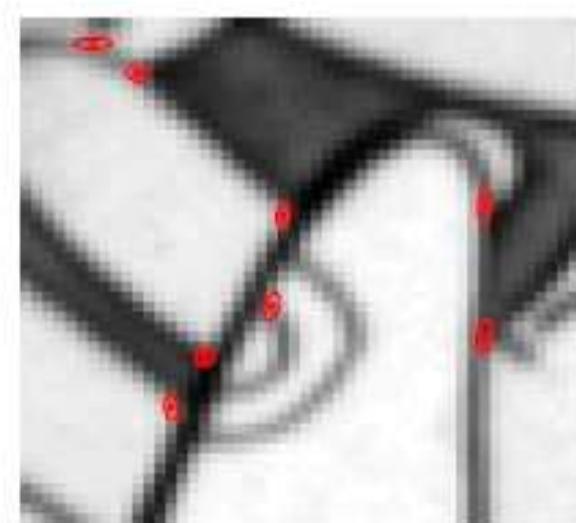
Lastni vrednosti matrike  $\mathbf{Q}$  opredeljujeta elipso, ki se prilega elementu slike: enotnemu območju, robu ali oglišču.



**Enotno območje:**  
obe lastni vrednosti  
majhni



**Robovi:**  
ena lastna vrednost  
majhna, ena velika



**Oglišča:**  
obe lastni vrednosti  
veliki

# HARRISOV DETEKTOR – MERA ZA OGLIŠČA

1. Determinanto in sled matrike  $\mathbf{Q}$  izkoristimo za mero, ki kaže na oglišča:

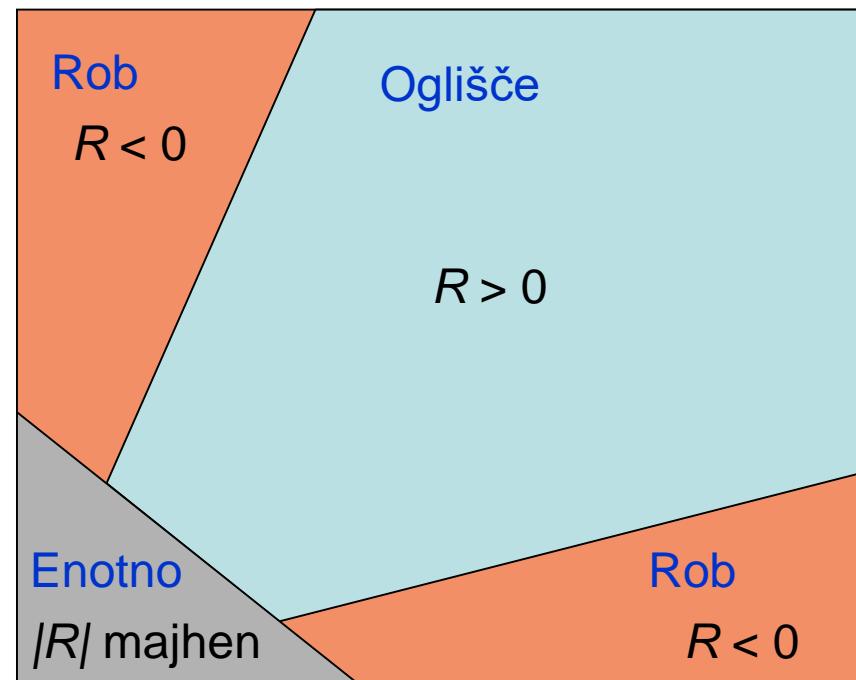
$$R = \det(\mathbf{Q}) - k(\text{sled}(\mathbf{Q}))^2$$

$$\det(\mathbf{Q}) = \lambda_1 \lambda_2$$

$$\text{sled}(\mathbf{Q}) = \lambda_1 + \lambda_2$$

( $k$  – izkustvena konstanta,  $k = 0,04 – 0,06$ )

- $R$  je velik za **oglišča**
- $R$  je zelo negativen za **robove**
- $|R|$  je majhen za **enotna območja**



# HARRISOV DETEKTOR – ALGORITEM IN LASTNOSTI

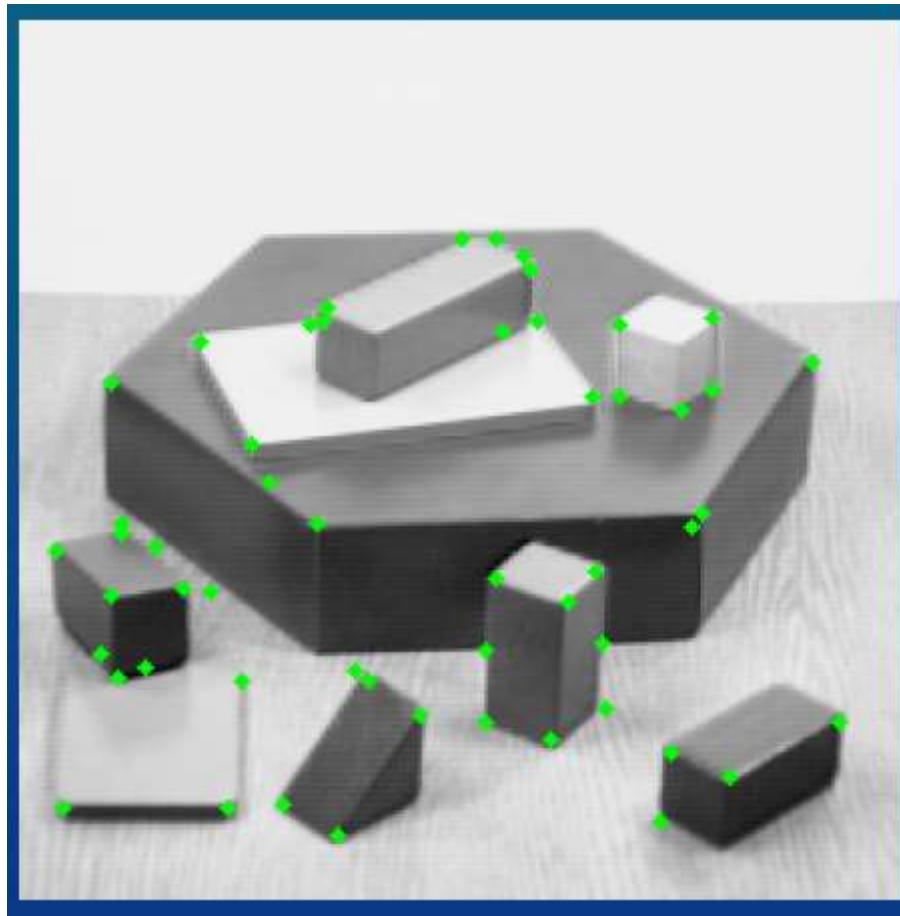
1. Povzetek Harrisovega algoritma za iskanje oglišč na slikah:

- v oknih  $3 \times 3$  ali  $5 \times 5$  s središči  $(x,y)$  izračunamo elemente matrike  $\mathbf{Q}(x,y)$
- z razcepom matrike  $\mathbf{Q}$  dobimo lastni vrednosti  $\lambda_1$  in  $\lambda_2$
- izračunamo  $R$
- največje vrednosti  $R$  (lokalni maksimumi) določajo položaje oglišč

2. Lastnosti Harrisovega detektorja:

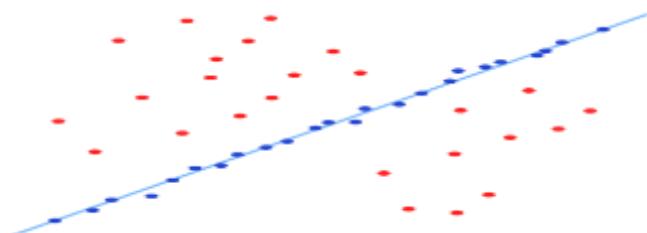
- rotacijsko neodvisen, neobčutljiv na zasuke slike
- neodvisen od globalnih premikov pikselskih intenzivnosti (sivin)
- odvisen od merila (skale):
  - neodvisnost od merila lahko dosežemo z uvedbo merilno neodvisne funkcije, na primer povprečja sivin, ki jo računamo v različno velikih območjih na slikah različnih meril
  - maksimum izbrane funkcije določa za vsako sliko merilo, pri katerem velja merilna neodvisnost med temi slikami

# PRIMER DELOVANJA HARRISOVEGA DETEKTORJA



# RANSAC

1. Kadar imamo opravka z množico vzorcev, ki opisujejo neki pojav ali obliko, se med njimi zaradi motenj, napak ali nenatančnosti pojavijo tudi prekomerno odstopajoče vrednosti, t. i. **osamelci (*outliers*)**.
2. Osamelce moramo odstraniti in upoštevati vzorce, ki z največjo verjetnostjo pripadajo iskani obliki ali modelu.
3. Zelo razširjen algoritem za to nalogo je imenovan **RANSAC (*random sample consensus*)**, prvič predlagan 1981 od avtorjev Fischlerja in Bollesa.
4. Algoritem uporablja tri vrste vhodnih podatkov:
  - množico izmerjenih vzorcev (ustreznih in odstopajočih osamelcev)
  - parametriziran model pojava ali oblike, za katerega predvidevamo, da mu pripadajo opazovani vzorci
  - kriterijsko funkcijo, s katero potrdimo ali zavrnemo izračunane modelne parametre
5. Primer za premico:



# RANSAC – ALGORITEM

Vhod:	<b>D</b>	vzorčeni podatki
	$k$	število modelnih parametrov
	$f_c$	kriterijska funkcija za preverjanje ustreznosti modelnih parametrov
	$N_{maks}$	največje število poskusov
Izhod:	<b>B</b>	vzorci, ki najbolje ustrezano iskanemu modelu
	<b>m</b>	modelni parametri, ki maksimirajo kriterijsko funkcijo

- 1: **ponavljam**
- 2:       naključno izberi  $k$  vzorcev iz nabora **D**
- 3:       izračunaj modelne parametre **m** iz izbranih  $k$  vzorcev
- 4:       s kriterijsko funkcijo  $f_c$  preveri podporo modelu **m** z ugotavljanjem, koliko odstopajo preostali elementi iz **D** od modela
- 5:       v **B** prenesi vzorce, ki v **D** ustrezano modelu **m** glede na funkcijo  $f_c$
- 6: **dokler**   modelni parametri **m** ne ustrezano največjemu številu vzorcev v **D**  
**ali** ni preseženo število poskusov  $N_{maks}$

# RANSAC – DOLOČANJE VHODNIH PARAMETROV

1. Kriterijska funkcija  $f_c$ , ki odredi, ali poljubni vhodni vzorec ustreza trenutnemu modelu (parametri  $\mathbf{m}$ ), mora biti določena izkustveno – na primer kot največja oddaljenost od iskane premice.
2. Število vzorcev, ki morajo glede na kriterijsko funkcijo  $f_c$  pripadati modelu, da izračunane parametre  $\mathbf{m}$  vzamemo kot verjeten pravilni rezultat, tudi določimo izkustveno.
3. Število ponovitev poskusa pa se da oceniti vnaprej z naslednjim sklepanjem:
  - naj bo  $\eta$  verjetnost, da z večkratnim naključnim poskušanjem zberemo skupke  $n$  vzorcev, ki vsi ustrezano pravemu modelu (v praksi vzamemo  $\eta = 0,99$ )
  - naj bo  $\mu$  verjetnost, da izberemo vzorec, ki ustreza pravemu modelu:

$$\mu = \frac{\langle \text{število vseh ustreznih vzorcev} \rangle}{\langle \text{število vseh vhodnih vzorcev} \rangle} \quad (\text{izkustveno ocenimo})$$

- verjetnost, da v  $N$  poskusih nikoli nismo uspeli najti  $n$  vzorcev, ki bi bili vsi ustrezni, je zato:

$$(1 - \mu^n)^N = 1 - \eta$$

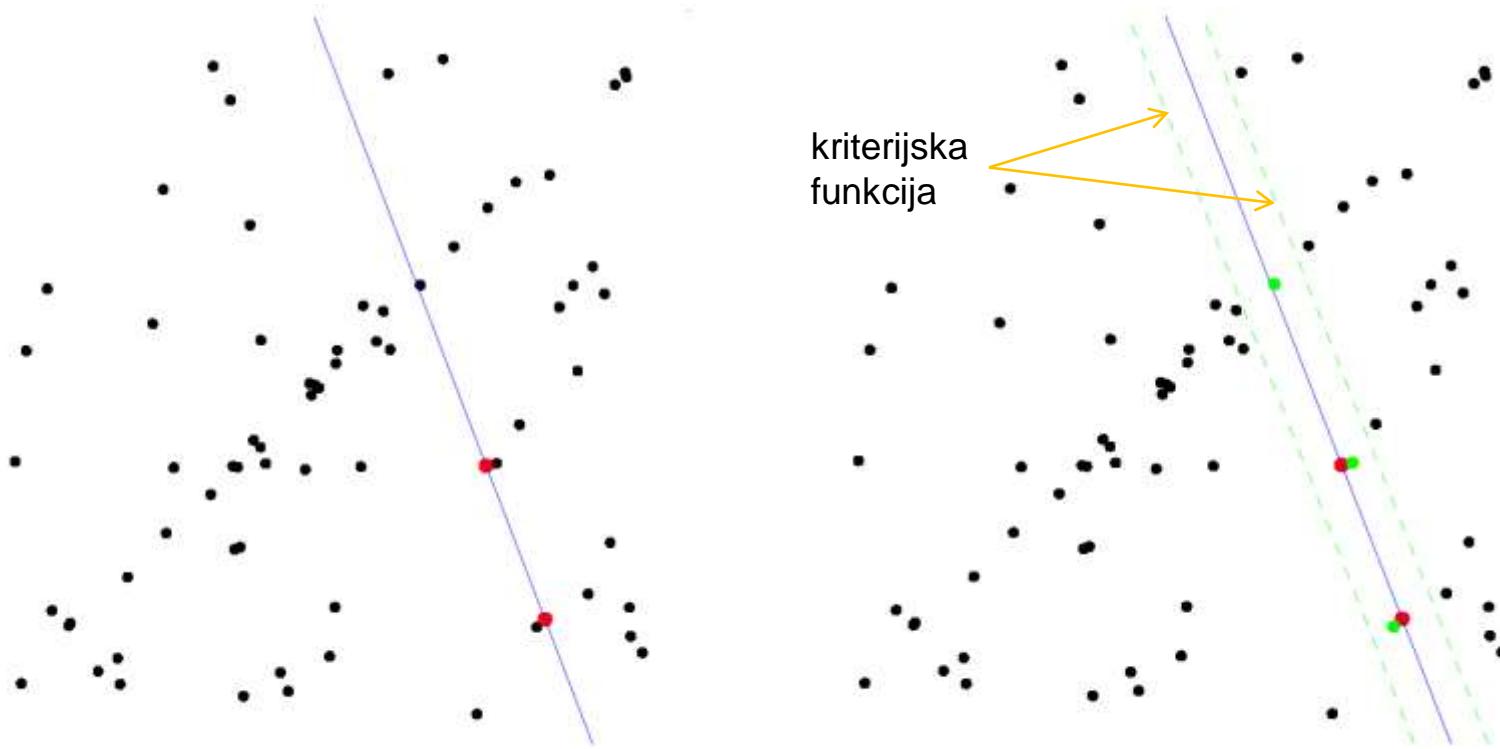
- sledi:

$$N = \frac{\ln(1 - \eta)}{\ln(1 - \mu^n)}$$

kar vzamemo kot največje potrebno število poskusov, tj.  $N_{\text{maks}}$

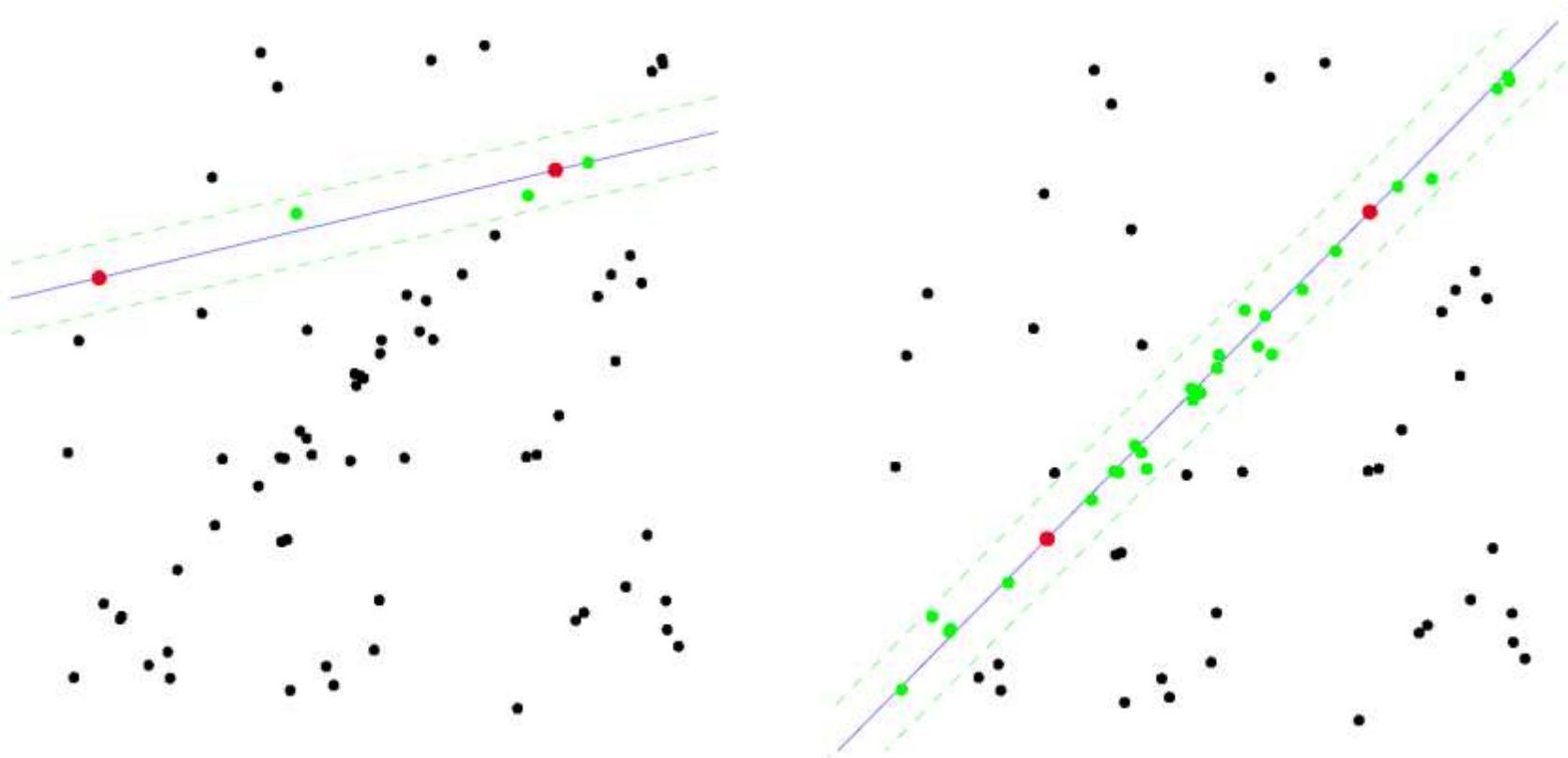
# Primer iskanja premice z RANSAC

1. Pri iskanju premice moramo naključno jemati po dva vzorca (točki) –  $k = 2$ .
2. Nato pogledamo, koliko ostalih vzorcev se pojavi v okolici, ki jo izberemo za preverjanje pravega modela (prav postavljene premice).



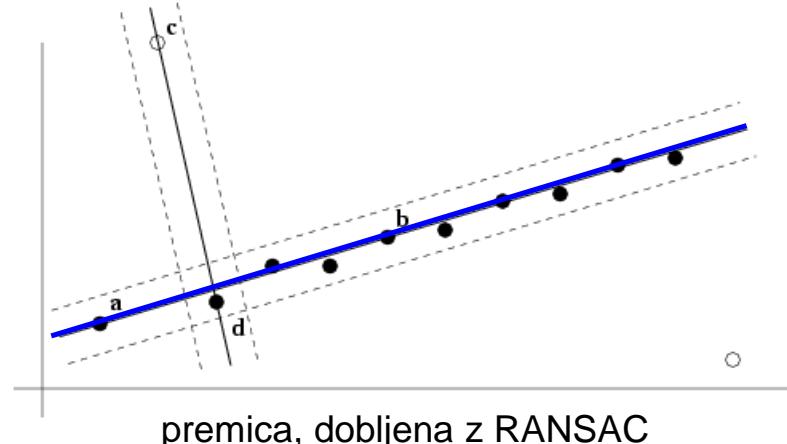
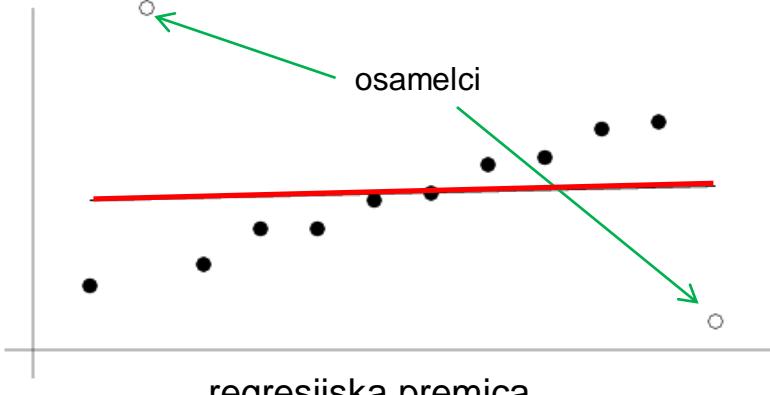
# Primer iskanja premice z RANSAC (2)

1. Poskušamo še za druge pare vzorcev (točk).
2. Izberemo model, ki mu glede na kriterijsko funkcijo ustreza največ vhodnih vzorcev ali pri katerem je preseženo vnaprej določeno število ustreznih vhodnih vzorcev.



# UPORABA ALGORITMA RANSAC

1. Z zgledom modela premice lahko prikažemo bistveno razliko med računanjem modela v obliki regresijske premice iz vseh vhodnih vzorcev in računanjem modela z metodo RANSAC, ki upošteva samo ustrezne vzorce.



2. RANSAC lahko uporabimo pri izbiri 8 (ali več) najustreznejših točk, s katerimi računamo temeljno matriko.
3. RANSAC pomaga splošneje tudi pri iskanju najustreznejših korespondenčnih točk, ki jih potrebujemo za tvorbo nekega modela (v 2D ali 3D).

# Afina poravnavo slik

# TOGA PORAVNAVA SLIK

1. Pri togi poravnavi slik lahko pride samo do premika ali zasuka cele slike.
2. Najpreprostejši postopek je naslednji:
  - sliki  $\mathbf{I}_1$  in  $\mathbf{I}_2$  binariziramo, tako da dobimo piksle  $b_i(x,y)$ , ki imajo vrednosti 0 ali 1,  $i \in \{1,2\}$ ; uporabimo enega od pragovnih pristopov
  - izračunamo težišči slik in sliki s težišči poravnamo – izničimo **premik**
  - premaknjeno opazovano (sivinsko, barvno) sliko  $\mathbf{I}_2$  primerjamo z referenčno  $\mathbf{I}_1$  – na primer z evklidsko razdaljo
  - nato opazovano sliko iterativno sučemo za izbrani (majhen) kot in po vsakem zasuku opravimo primerjavo z referenčno sliko
  - najustreznejši zasuk določa najmanjša razlika med slikama – od tega zasuka lahko iščemo še boljšo poravnavo tako, da iteriramo dalje z manjšimi koti zasuka – izničimo **zasuk**
3. Izračun težišča slike:

$$n_i = \sum_{x=1}^M \sum_{y=1}^N b_i(x, y) \quad x_{0,i} = \frac{\sum_{x=1}^M \sum_{y=1}^N x \cdot b_i(x, y)}{n_i} \quad y_{0,i} = \frac{\sum_{x=1}^M \sum_{y=1}^N y \cdot b_i(x, y)}{n_i}$$

# AFINA PORAVNAVA SLIK

1. Afina poravnava slik v splošnem ni toga, saj lahko vsebuje povečavo in strig.
2. Imejmo referenčno sliko  $\mathbf{I}_1$  in opazovano sliko  $\mathbf{I}_2$  in matriko  $\mathbf{M}$  za afino preslikavo  $\mathbf{I}_2$  v  $\mathbf{I}_1$ :

$$\mathbf{M} = \begin{bmatrix} m_1 & m_2 & m_5 \\ m_3 & m_4 & m_6 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Pri popolni poravnavi slik bi našli za piksle z vrednostmi  $p_1(x',y')$  na sliki  $\mathbf{I}_1$  piksle z enakimi vrednostmi  $p_2(x,y)$  na sliki  $\mathbf{I}_2$ :

$$p_2(x, y) = p_1(m_1x + m_2y + m_5, m_3x + m_4y + m_6)$$

4. Ker idealne poravnave večinoma nimamo, iščemo optimalno rešitev – na primer z najmanjšim odstopanjem kvadratov:

$$E(\mathbf{m}) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [p_2(x, y) - p_1(m_1x + m_2y + m_5, m_3x + m_4y + m_6)]^2$$

5. Parametri poravnave  $\mathbf{m}$  v kriterijski funkciji niso eksplisitno izraženi, zato vrednosti piksov na opazovani sliki zapišemo z linearimi členi Taylorjevega razvoja glede na referenčno sliko:

$$E(\mathbf{m}) \approx \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \{p_2(x, y) - [p_1(x, y) + (m_1x + m_2y + m_5 - x)p_{1,x}(x, y) + (m_3x + m_4y + m_6 - x)p_{1,y}(x, y)]\}^2$$

Parcialni odvod po x

Parcialni odvod (razlika) z opazovane na referenčno sliko

Parcialni odvod po y

# AFINA PORAVNAVA SLIK (2)

6. Vsoto kvadriranih razlik piksov na slikah razvijemo v:

$$E(\mathbf{m}) \approx \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left\{ p_{2,1}(x,y) - (m_1 x + m_2 y + m_5 - x) p_{1,x}(x,y) - (m_3 x + m_4 y + m_6 - y) p_{1,y}(x,y) \right\}^2$$

7. Pri odvodih izpustimo koordinate piksov in uvedimo naslednji funkciji:

$$k = p_{2,1} + x p_{1,x} + y p_{1,y}$$

$$\mathbf{c} = [xp_{1,x} \; yp_{1,x} \; xp_{1,y} \; yp_{1,y} \; p_{1,x} \; p_{1,y}]^T$$

8. Sledi:

$$E(\mathbf{m}) = \sum_{x,y} \left( k - \mathbf{c}^T \mathbf{m} \right)^2; \quad \mathbf{m} = [m_1, m_2, m_3, m_4, m_5, m_6]^T$$

9. Odvajamo po afinih parametrih  $\mathbf{m}$ :

$$\frac{dE(\mathbf{m})}{d\mathbf{m}} = \sum_{x,y} -2\mathbf{c} \left( k - \mathbf{c}^T \mathbf{m} \right)$$

10. Rezultat izenačimo z 0 in dobimo parametre  $\mathbf{m}$  pri minimalnem odstopanju med poravnanimi slikama  $E(\mathbf{m})$ :

$$\mathbf{m} = \left( \sum_{x,y} \mathbf{c} \mathbf{c}^T \right)^{-1} \left( \sum_{x,y} \mathbf{c} k \right)$$

# Primer za togo in afino poravnavo slik

1. Opazovana in referenčna slika:



2. Rezultat toge poravnave (premik in zasuk):



3. Rezultat po dodatni afini poravnavi:



# AFINA PORAVNAVA S SPREMEMBAMI INTENZIVNOSTI PIKSLOV

1. Doslej izpeljana afina poravnava slik temelji na predvidevanju, da sta intenzivnosti opazovane in referenčne slike konstantni. Gre za **kontrast in svetlost**.
2. Če se kontrast in svetlost spreminja, lahko v afino poravnavo slik uvedemo še dva parametra –  $m_7$  za kontrast in  $m_8$  za svetlost:  
$$m_7 \cdot p_2(x, y) + m_8 = p_1(m_1x + m_2y + m_5, m_3x + m_4y + m_6)$$
3. Z iskanjem optimuma pridemo do parametrov po enaki poti kot pri afini poravnavi:

$$E(\mathbf{m}) = \sum_{x,y} \left( k - \mathbf{c}^T \mathbf{m} \right)^2; \quad \mathbf{m} = [m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8]^T$$

$\Downarrow$

$$\mathbf{m} = \left( \sum_{x,y} \mathbf{c} \mathbf{c}^T \right)^{-1} \left( \sum_{x,y} \mathbf{c} k \right)$$

pri čemer velja:

$$k = p_{2,1} - p_2 + xp_{1,x} + yp_{1,y}$$

$$\mathbf{c} = [xp_{1,x} \; yp_{1,x} \; xp_{1,y} \; yp_{1,y} \; p_{1,x} \; p_{1,y} \; -p_2 \; -1]^T$$

# NETOGA, ELASTIČNA PORAVNAVA SLIK

1. Za elastično poravnavo slik je značilno, da transformacijska matrika ni enaka za celotno območje slike, temveč je lokalno spremenljiva.
2. Obstaja mnogo metod, med preprostejšimi je uporaba lokalne afine poravnave:
  - predstavljena metoda za afino poravnavo lahko ujemanje slik opazuje samo na lokalno omejenih območjih (vsote v izrazih ne tečejo prek vseh pikslov slike, ampak le v okviru manjših, lokalnih oken)
  - izračunani afini parametri za poravnavo (tudi za intenzivnost pikslov) veljajo samo na izbranem območju
  - lokalna območja se poravnavajo ločeno
  - pristop je iterativen:  
ko poravnamo vsa območja slike, začnemo nov ciklus lokalnih poravnav
  - da dosežemo gladko prehajanje med lokalno poravnanimi območji, uvedemo v optimizacijsko funkcijo za poravnavo dodatne parametre, ki zagotavljajo zvezne prehode med območji

(Senthil Priaswamy, Hany Farid: Elastic Registration in the Presence of Intensity Variations,  
*IEEE Transactions on Medical Imaging*, Vol. 22, No. 7, str. 865-874, 2003)

# TOGA PORAVNAVA SLIK S POMOČJO FREKVENČNEGA PROSTORA

1. Toga poravnava slik se v slikovnem prostoru izvede iterativno in je lahko časovno zahtevna.
2. Če izvorno in ciljno sliko najprej transformiramo v frekvenčni prostor, lahko uporabimo dva postopka, ki v frekvenčnem prostoru delujeta v zaključeni obliki, brez iterativnega poskušanja pri iskanju parametrov premika in zasuka.
3. 2D hitra Fourierova transformacija (FFT2) zagotovi, da je iskanje parametrov za poravnavo časovno optimalno.
4. Dejansko poravnava slik izvedemo v slikovnem prostoru, potem ko najdemo parametre premika in zasuka s pomočjo frekvenčnega prostora.
5. Imejmo sliko  $I$  s piksli  $x(n,m)$  in dimenzijami  $N \times N$ . 2D Fourierova transformacija zanjo se glasi:

$$X(k,l) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n,m) W_N^{kn+lm}$$

6. Vzemimo, da sliko  $I$  zasučemo in premaknemo, tako da se spremenijo koordinate:

$$\begin{bmatrix} n_1 \\ m_1 \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n \\ m \\ 1 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} n_1 \\ m_1 \\ 1 \end{bmatrix} = R \begin{bmatrix} n \\ m \\ 1 \end{bmatrix} + t$$

# TOGA PORAVNAVA SLIK S POMOČJO FREKVENČNEGA PROSTORA (2)

- Izrazimo koordinate izvorne slike s koordinatami transformirane slike:

$$\begin{bmatrix} n \\ m \end{bmatrix} = \mathbf{R}^{-1} \left( \begin{bmatrix} n_1 \\ m_1 \end{bmatrix} - \mathbf{t} \right); \quad \mathbf{R} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}, \quad \mathbf{R}^{-1} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

- 2D Fourierova transformacija premaknjene in zasukane slike  $\mathbf{I}_1$  je tako:

$$\begin{aligned} X_1(k, l) &= \sum_{n_1=0}^{N-1} \sum_{m_1=0}^{N-1} x_1(n_1, m_1) W_N^{kn_1+lm_1} \\ &= \sum_{n_1=0}^{N-1} \sum_{m_1=0}^{N-1} x(\mathbf{R}^{-1} \begin{bmatrix} n_1 \\ m_1 \end{bmatrix} - \mathbf{t}) W_N^{kn_1+lm_1} \\ &= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n, m) W_N^{\{\mathbf{R} \begin{bmatrix} n \\ m \end{bmatrix} + \mathbf{t}\}^T \begin{bmatrix} k \\ l \end{bmatrix}} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n, m) W_N^{\{\mathbf{R} \begin{bmatrix} n \\ m \end{bmatrix}\}^T \begin{bmatrix} k \\ l \end{bmatrix}} W_N^{\mathbf{t}^T \begin{bmatrix} k \\ l \end{bmatrix}} \end{aligned}$$

Privzemimo, da tudi zasukana slika ostane v istih dimenzijah, kot je bila originalna slika.

zasuk in premik zapišemo ločeno – to je osnova za določanje parametrov zasuka in premika

# TOGA PORAVNAVA SLIK S POMOČJO FREKVENČNEGA PROSTORA – ZASUK

- Izračunajmo amplitudni spekter, tj. absolutno vrednost za  $X_1(k,l)$ :

$$\begin{aligned}|X_1(k,l)| &= \left| \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n,m) W_N^{\{\mathbf{R}_m^n\}^T[k][l]} W_N^{[t_1,t_2][k][l]} \right| \\&= \left| \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n,m) W_N^{\{\mathbf{R}_m^n\}^T[k][l]} \right| \left| W_N^{[t_1,t_2][k][l]} \right|\end{aligned}$$

1

- Dobljeni izraz kaže, da je amplitudni spekter neodvisen od premika slike ( $\mathbf{t}$ ).
- Zasuk se da določiti iz:
- $$\Delta(k,l) = \frac{|X_1(k,l)|^2}{X_1^2(0,0)} - \frac{|X(k,l)|^2}{X^2(0,0)} = 0$$
       $X_1(k,l)$  in  $X(k,l)$  – Fourierovi transformiranki slik v poravnavi
- $\Delta(k,l)$  mora biti 0, ko  $X_1(k,l)$  in  $X(k,l)$  nista hkrati enaka 0.
- Izraz  $\Delta(k,l)=0$  določa dve premici, ki se sekata pod kotom  $\alpha/2$ , če je  $\alpha$  kot zasuka med slikama
- V razponu kotov med  $-\pi/4$  in  $\pi/4$  se vidi kot  $\alpha/2$  neposredno

# TOGA PORAVNAVA SLIK S POMOČJO FREKVENČNEGA PROSTORA – PREMIK

1. Amplitudni spekter je pri zasukani sliki  $X_1(k,l)$  odvisen samo od zasuka, medtem ko se premik vidi samo v faznem spektru.
2. Fazna spektra slik v poravnavi lahko primerjamo s pomočjo frekvenčne oblike prečne korelacije med izvorno sliko in njeno transformirano (premaknjeno in zasukano) inačico:

$$\Phi(k,l) = \frac{X_1(k,l) \cdot X^*(k,l)}{|X_1(k,l) \cdot X(k,l)|}$$

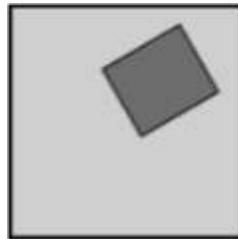
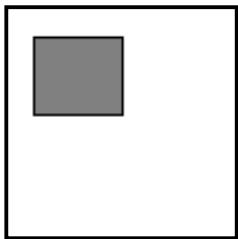
pri čemer ustreza  $X^*(k,l)$  konjugirani vrednosti Fourierove transformiranke.

3. Premik v obeh smereh slikovne ravnine določimo z iskanjem maksimalne vrednosti v frekvenčni transformiranki prečne korelacije slik:

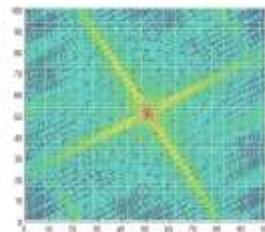
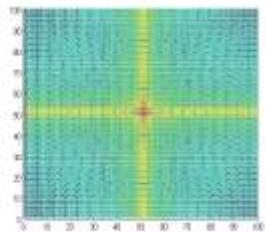
$$(t_1, t_2) = \arg \max_{t_1, t_2} \{\text{IFFT}[\Phi(k,l)]\}$$

# Primer poravnave slik s pomočjo frekvenčnega prostora

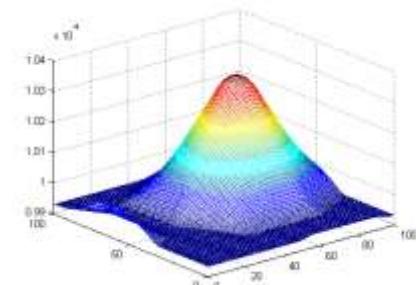
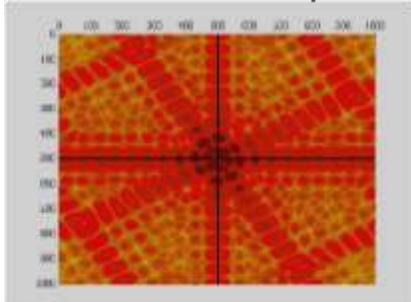
1. Referenčna in opazovana slika:



2. Amplitudni spekter referenčne in opazovane slike:



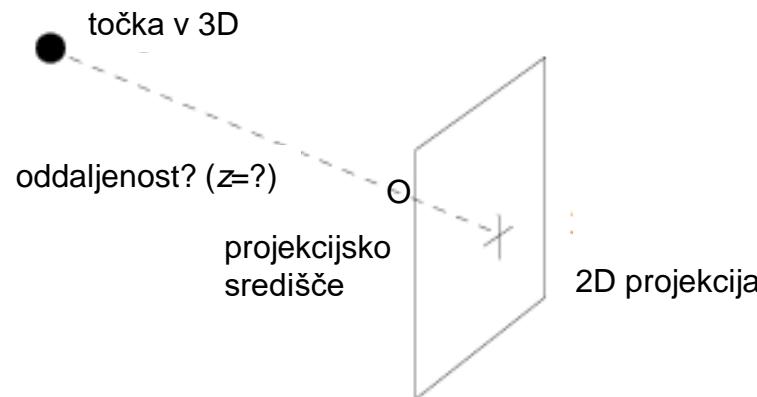
3. Razlika amplitudnih spektrov, slika  $\Delta(k, l)$  in frekvenčna slika prečne korelacije:



# Stereo vid

# IZHODIŠČA

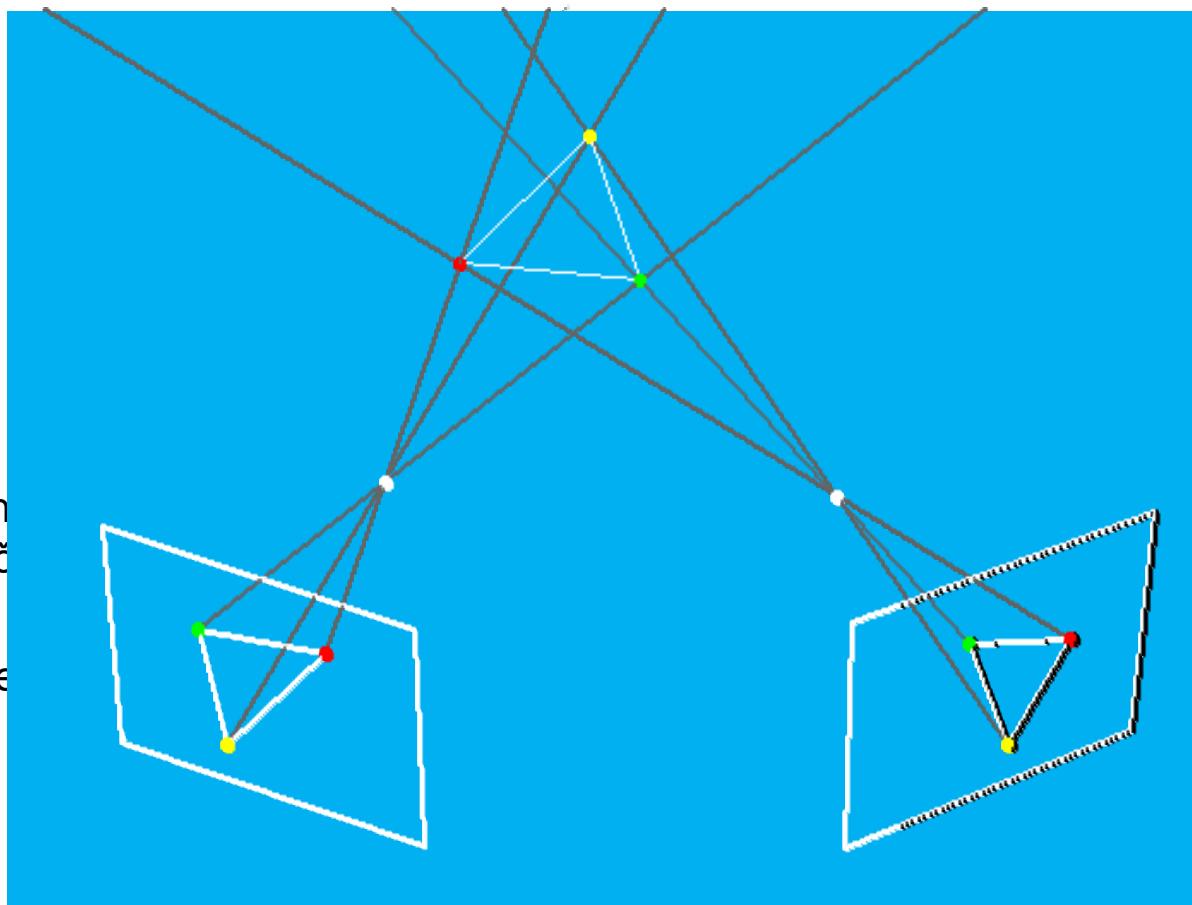
1. En sam pogled na tridimenzionalno sceno ne omogoča zaznave oddaljenosti predmetov na sceni.
2. Ljudje lahko iz enega samega pogleda le sklepamo o oddaljenosti predmetov, in sicer glede na medsebojna prekrivanja, razmerja v velikosti predmetov, sence itd.



3. Pri le eni sliki lahko trdimo samo, da je točka v 3D nekje na premici, ki povezuje projekcijsko središče in 2D projekcijo točke na sliki.

# DVA POGLEDA NA ISTO SCENO

- Če vidimo isto sceno v dveh pogledih iz različnih zornih kotov, lahko določimo oddaljenost 3D točk od projekcijskih središč kamer.



- Vzvratna matrika je določena s slikah.
- Sečišče

v prostoru  
točki na

# DVA POGLEDA NA ISTO SCENO – PROJEKCIJSKA GEOMETRIJA

1. Imenujmo slike "leva" in "desna".
2. Točko v prostoru označimo v koordinatnem sistemu leve kamere s koordinatami  $\mathbf{p}_l = [X_l, Y_l, Z_l]^T$ , v koordinatnem sistemu desne kamere pa s  $\mathbf{p}_d = [X_d, Y_d, Z_d]^T$
3. Projekciji na levi in desni slike opredelimo s koordinatami  $(x_l, y_l)$  in  $(x_d, y_d)$ .
4. Vzemimo, da imata obe kamere enako goriščno razdaljo  $f$ .
5. Tedaj velja:

$$x_l = f \frac{X_l}{Z_l} \Rightarrow X_l = \frac{x_l Z_l}{f}$$

$$x_d = f \frac{X_d}{Z_d} \Rightarrow X_d = \frac{x_d Z_d}{f}$$

6. Položaj desne kamere lahko izrazimo s položajem leve kamere, ki jo premaknemo in zasučemo:

$$\mathbf{p}_d = \mathbf{R}(\mathbf{p}_l - \mathbf{t}) ; \mathbf{R} - \text{rotacijska matrika}, \mathbf{t} - \text{vektor premika}$$

7. Če vzamemo identični kameri in popolnoma poravnamo njuni optični osi, velja:

$$\frac{x_l Z_l}{f} - t = \frac{x_d Z_d}{f} \Rightarrow Z = \frac{t \cdot f}{x_l - x_d} = \frac{tf}{d} ; t - \text{premik le vzdolž koordinatne osi } x$$

# DVA POGLEDA NA ISTO SCENO – STEREO POGLED

1. Razdalja prostorske točke od poravnanih projekcijskih središč obih kamer, imenovana globina, je torej:

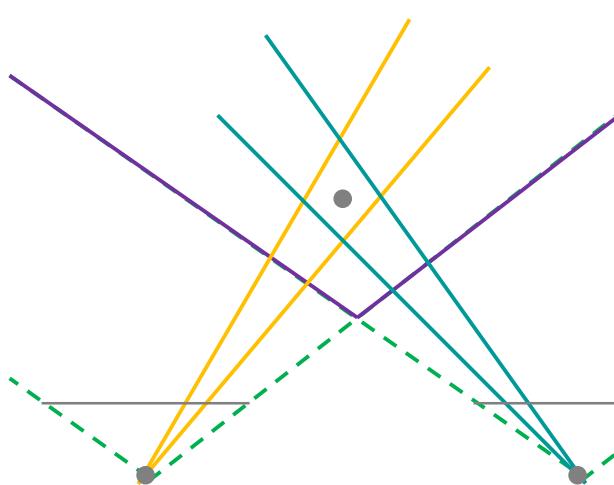
$$Z = \frac{t \cdot f}{x_l - x_d} = \frac{tf}{d}$$

pri čemer je razlika med koordinatama projiciranih točk (koordinate v smeri premika med kamerama) poimenovana **razlika oz. dispariteta (disparity)  $d$** .

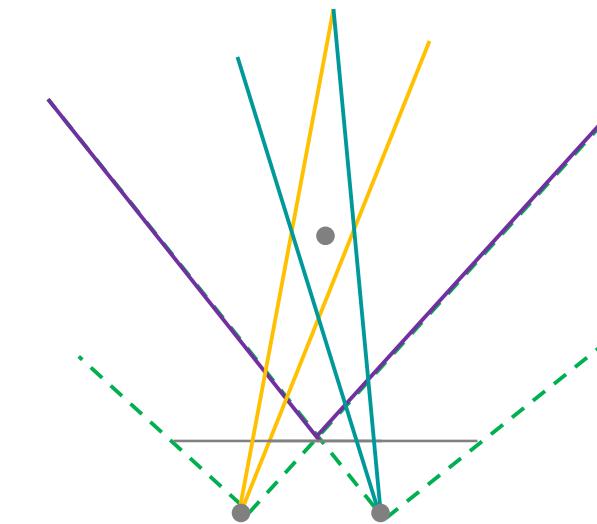
2. **Slikovna globina (range image)** je obratno sorazmerna s slikovno disparitetom.
3. Dispariteta  $d$  mora imeti dolžinske enote. Pri delu s slikami je običajno, da razmere opisujemo s številom pikslov, zato tako izrazimo tudi dispariteto, označeno z  $d_p$ :  
 $d = d_p/k$ , pri čemer je  $1/k$  širina pikslov.
4. Oddaljenost predmetov (globinsko sliko) izračunamo z disparitetom v pikslih:  
$$Z = \frac{t \cdot f \cdot k}{d_p}$$
5. Postavitev dveh identičnih kamer s popolnoma poravnanim optičnima osema imenujemo **stereo kamera**. Taka kamera zagotavlja **binokularni pogled**.

# STEREO POGLED – GLOBINSKA SLIKA

1. Računanje slikovne globine ne more biti neskončno natančno, ker je odvisno od ločljivosti slike oz. velikosti pikslov.
2. Čim nižja je ločljivost, tem manj je diskretiziranih oddaljenosti, ki jih lahko določimo iz slikovnih razlik.
3. Kakšna pa je odvisnost od premika med kamerama (pogledoma), torej od  $t$ ?



Večji premik – manjše globinske napake



Manjši premik – večje globinske napake

4. Za določanje oddaljenosti moramo točko videti v obeh pogledih, kar omejuje območje binokularnega pogleda.

# STEREO POGLED – EPIPOLARNA GEOMETRIJA

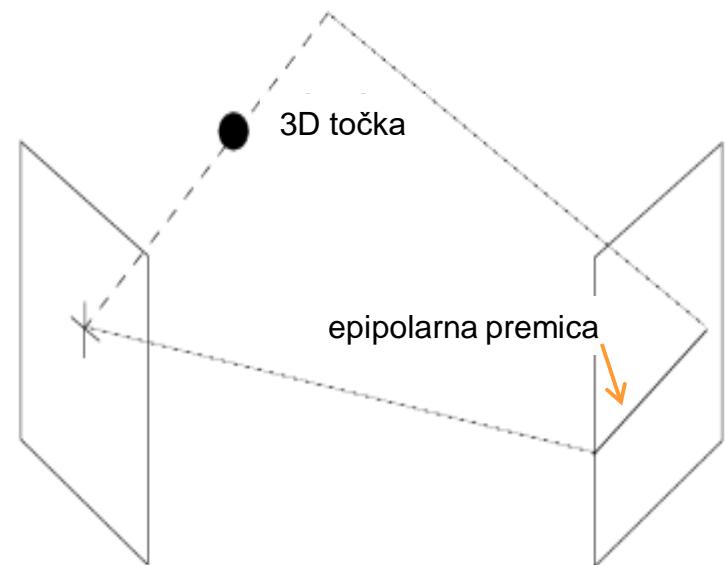
1. Epipolarna geometrija zagotavlja, da lahko za značilno točko na levi sliki najdemo epipolarno premico, na kateri zanesljivo leži korespondenčna točka na desni sliki.
2. Pri tem potrebujemo temeljno matriko  $\mathbf{F}$ :

$$\mathbf{l}_d = \mathbf{F}\mathbf{p}_l; \quad \mathbf{p}_l = \begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix}, \quad \mathbf{l}_d = \begin{bmatrix} a \\ b \\ w \end{bmatrix}$$

3. Če točka  $\mathbf{p}_d$  na desni sliki leži na epipolarni premici, velja epipolarna omejitev:

$$\mathbf{l}_d^T \mathbf{p}_d = ax_d + by_d + w = 0$$

4. Epipolarna omejitev je prvi pogoj, ki mora biti izpolnjen, da najdemo za točko z leve korespondenčno točko na desni sliki.
5. Drugi pogoj za korespondenčni točki je, da se ujemata po lastnostih:
  - po epipolarni premici na desni sliki iščemo točko, za katero pokaže neka primerjalna funkcija, da je z okolico najbolj podobna izbrani točki z leve slike
6. Razdalja med korespondenčnima točkama ( $d_p$ ) je obratno sorazmerna oddaljenosti predmeta, na katerem je ta točka, od kamere



# STEREO POGLED – ALGORITEM ZA DOLOČANJE GLOBINSKE SLIKE

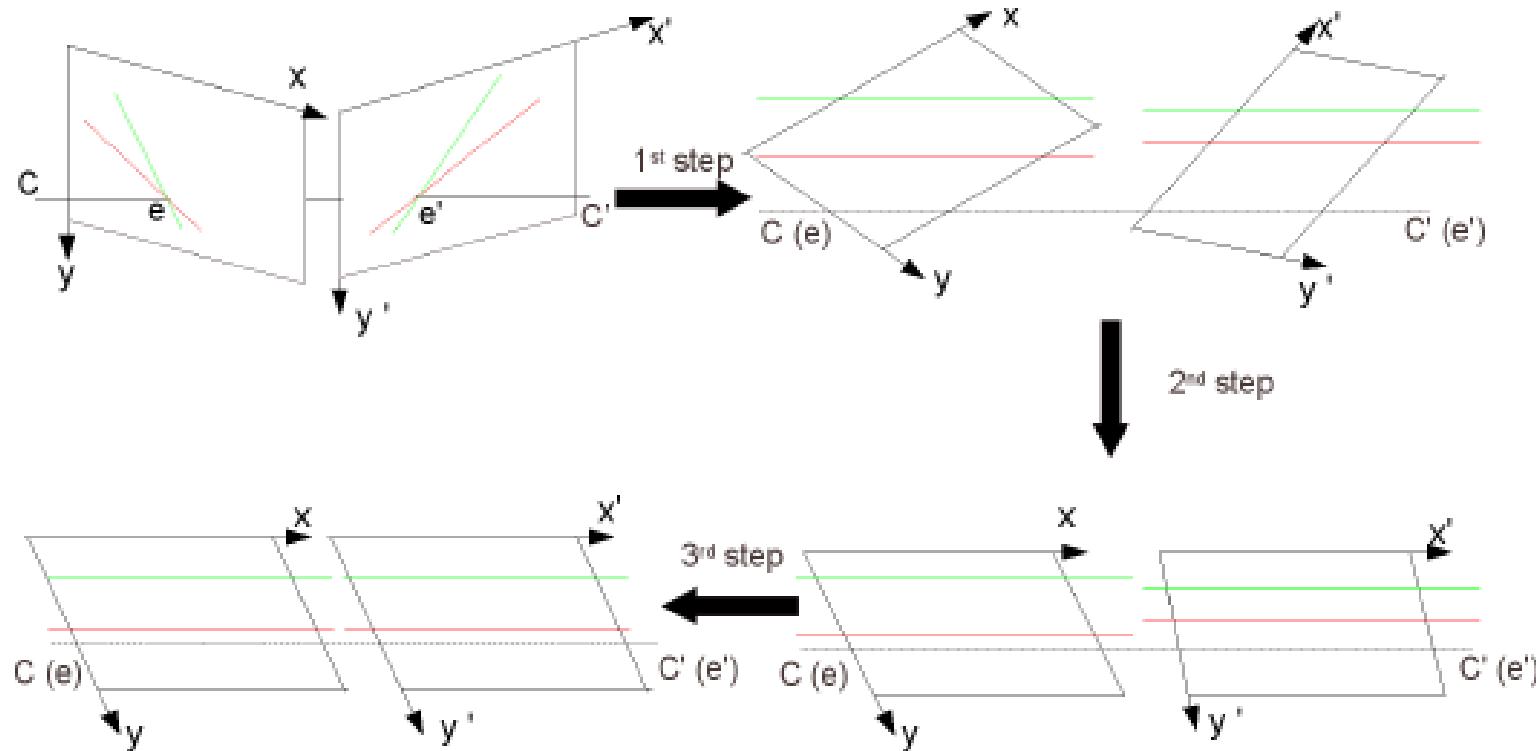
1. V praksi ne obstaja možnost stereo kamere brez kalibracije:
  - imamo dve kameri in vsaka ima svoje, različne notranje parametre
  - kameri ne moreta imeti popolnoma vzporednih svojih koordinatnih sistemov
2. Epipolarna geometrija s temeljno matriko  $\mathbf{F}$  omogoča stereo pogled z nekalibriranimi kamerama, vendar pri tem naletimo na nekaj težav:
  - ker epipolarne premice načelno niso vzporedne s koordinatnim sistemom slike, je “potovanje” po njih med iskanjem korespondence oteženo
  - odkrivanje korespondenčnih točk ni nujno zanesljivo in robustno
3. Algoritem za določanje globinske slike zato temelji na naslednjih korakih:
  - na levi in desni sliki poiščemo zanesljive nadzorne točke in izračunamo temeljno matriko  $\mathbf{F}$
  - sliki obdelamo s t. i. **izravnavo oz. popravljanjem (rectification)**, pri kateri “potisnemo” epipola na obeh slikah v neskončnost in tako naredimo epipolarne premice in koordinatna sistema slik vzporedne
  - z izravnavo slik zagotovimo, da lahko korespondenčne točke iščemo samo z istimi vertikalnimi koordinatami (bistveno olajšan postopek) – uporabimo enega od razpoložljivih postopkov za primerjavo slikovnih regij
  - med korespondenčnimi točkami določimo razliko v horizontalni smeri in izračunamo globino

# DOLOČANJE GLOBINSKE SLIKE – IZRAVNAVA SLIK

1. Za izravnavo slik obstaja več pristopov, ki temeljijo na optimizaciji 3 do 6 parametrov, a večinoma niso dovolj robustni.
2. V članku “Three-step image rectification” avtorjev P. Monassea, J.-M. Morela in Z. Tanga je bila leta 2010 razložena robustna metoda, ki temelji na rotacijah 2D pogledov (kamer) okoli projekcijskih središč s pomočjo **homografij** – transformacij med 2D slikami (transformacijska matrika  $3 \times 3$ ).
3. Pri izravnaji slik se rade pojavijo popačitve, ki se pojavijo pri napačnih zasukih para slik okoli osnovnice (zveznice med projekcijskima središčema). Velja namreč, da lahko izravnani sliki hkrati sučemo okoli osnovnice, pa še vedno ostaneta izravnani.
4. Metoda iz članka temelji na treh korakih:
  - epipola najprej potisne v neskončnost
  - nato epipola spravi v horizontalno lego
  - nazadnje sliki dokončno izravnamo z zasukom okoli osnovnice
5. Izhajamo iz zapisa 2D točk s koordinatami  $\mathbf{p} = (x, y, w)^T$  ozziroma nehomogenimi koordinatami  $(x/w, y/w)^T$ .
6. Kadar je  $w = 0$ , potisnemo točko v neskončnost po premici, ki gre skozi koordinati  $(x, y)$ .

# DOLOČANJE GLOBINSKE SLIKE – IZRAVNAV SLIK (2)

Izravnava slik v treh korakih:



(Povzeto po P. Monasse, J.-M. Morel in Z. Tang: Three-step image rectification, *British Machine Vision Conference 2010*, str. 89.1-89.10; <http://www.bmva.org/bmvc/2010/conference/paper89/index.html>.

# DOLOČANJE GLOBINSKE SLIKE – IZRAVNAVA (3)

## Korak 1

- Orientacijo kamere lahko spremenimo, če sliko transformiramo s homografijo  $\mathbf{H}$ :

$$\mathbf{H} = \mathbf{K} \cdot \mathbf{R} \cdot \mathbf{K}^{-1},$$

pri čemer je  $\mathbf{K}$  kalibracijska matrika kamere,  $\mathbf{R}$  pa matrika zasuka.

- Privzamemo, da kamera ni kalibrirana in da velja:

$$\mathbf{K} = \begin{bmatrix} f & 0 & \frac{w}{2} \\ 0 & f & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

$w$  – širina slike

$h$  – višina slike

$f$  – goriščna razdalja (neznana)

(privzamemo, da ni striga in da središče slike sovpada z optično osjo)

- Na levi in desni sliki določimo epipola  $\mathbf{e}_l = (e_{x,l}, e_{y,l}, 1)^T$  in  $\mathbf{e}_d = (e_{x,d}, e_{y,d}, 1)^T$ , tako da najprej izračunamo temeljno matriko  $\mathbf{F}$ . Velja:  $\mathbf{F}\mathbf{e}_l = 0$  in  $\mathbf{e}_d^T \mathbf{F} = 0$ .
- Epipola je treba spraviti v neskončnost:  $\mathbf{e}_l = (e_{x,l}, e_{y,l}, 0)^T$  in  $\mathbf{e}_d = (e_{x,d}, e_{y,d}, 0)^T$  in sicer s homografijama  $\mathbf{H}_{1,l}$  in  $\mathbf{H}_{1,d}$ :

$$\mathbf{H}_{1,l} \mathbf{e}_l = (e_{x,l}, e_{y,l}, 0)^T \text{ in } \mathbf{H}_{1,d} \mathbf{e}_d = (e_{x,d}, e_{y,d}, 0)^T$$

- Zasuk  $\mathbf{R}$ , ki ustreza, ni enoumno določen, zato se odločimo za najmanjšega med možnimi, saj tak najmanj popači sliko:

$$\mathbf{R}_l \mathbf{K}^{-1} \mathbf{e}_l = \mathbf{K}^{-1} (e_{x,l}, e_{y,l}, 0)^T \text{ in } \mathbf{R}_d \mathbf{K}^{-1} \mathbf{e}_d = \mathbf{K}^{-1} (e_{x,d}, e_{y,d}, 0)^T$$

# DOLOČANJE GLOBINSKE SLIKE – IZRAVNAVA SLIK (4)

- Gre za zasuk vektorja  $\mathbf{a} = \mathbf{K}^{-1}\mathbf{e}$  v vektor  $\mathbf{b} = \mathbf{K}^{-1}(e_x, e_y, 0)^T$ .
- Minimalni kot med temi vektorjema je:

$$\theta = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}\right)$$

os rotacije pa:

$$\mathbf{t} = \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$$

- Rodriguesova formula pravi:

$$\mathbf{R}(\theta, \mathbf{t}) = \mathbf{I} + \sin \theta [\mathbf{t}]_\times + (1 - \cos \theta) [\mathbf{t}]_\times^2$$

pri čemer pomeni

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

- Tako določimo matriki zasukov za levo in desno sliko, iz relacije  $\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$  pa še homografiji za obe slike:  $\mathbf{H}_{1,l}$  in  $\mathbf{H}_{1,d}$ . Obe homografiji vsebujeta še neznano vrednost goriščne razdalje, ki jo vzamemo kot parameter.

# DOLOČANJE GLOBINSKE SLIKE – IZRAVNAVA (5)

## Korak 2

1. Epipola v neskončnosti moramo poravnati z osjo  $x$ :  $\mathbf{e}_l = (1, 0, 0)^T$  in  $\mathbf{e}_d = (1, 0, 0)^T$ .
2. Potrebujemo dodatni homografiji  $\mathbf{H}_{2,l}$  in  $\mathbf{H}_{2,d}$ , ki ju izračunamo na enak način kot homografiji  $\mathbf{H}_{1,l}$  in  $\mathbf{H}_{1,d}$ , le da sedaj izhajamo iz epipolov v neskončnosti.

## Korak 3

1. Po 2. koraku koordinatne osi leve in desne slike še niso vzporedne. Oba koordinatna sistema je mogoče izravnati z dodatnim zasukom ene od slik okoli osnovnice.
2. Za dve dokončno izravnani sliki mora imeti temeljna matrika v kalibriranih razmerah naslednjo obliko (samo premik v  $x$ -smeri):

$$[\mathbf{i}]_x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

3. Če upoštevamo, da nas dodatni zasuk  $\mathbf{R}_3$  privede do dokončno izravnanih slik, smemo temeljno matriko v nekalibriranih razmerah pred 3. korakom zapisati kot:

$$\mathbf{F}_3 = \mathbf{K}^{-T} [\mathbf{i}]_x \mathbf{R}_3 \mathbf{K}^{-1}$$

# DOLOČANJE GLOBINSKE SLIKE – IZRAVNAVA (6)

## Korak 3

4. Iz zapisa za temeljno matriko  $\mathbf{F}_3$  sledi, da je v tem primeru osnovna matrika  $\mathbf{E}_3 = [\mathbf{i}]_{\times} \mathbf{R}_3$ .
5. Dodaten zasuk v 3. koraku lahko torej dobimo, če poznamo osnovno matriko  $\mathbf{E}_3$ .
6. Prej pa moramo prav določiti kalibracijsko matriko  $\mathbf{K}$ , ki je v dosedanji razlagi še vedno odvisna od neznane goriščne razdalje  $f$ .
7. Za izbrani nabor korespondenčnih točk  $\mathbf{p}_{n,l}$  na levi in  $\mathbf{p}_{n,d}$  desni sliki izračunamo vsoto razdalj točk od epipolarnih premic:

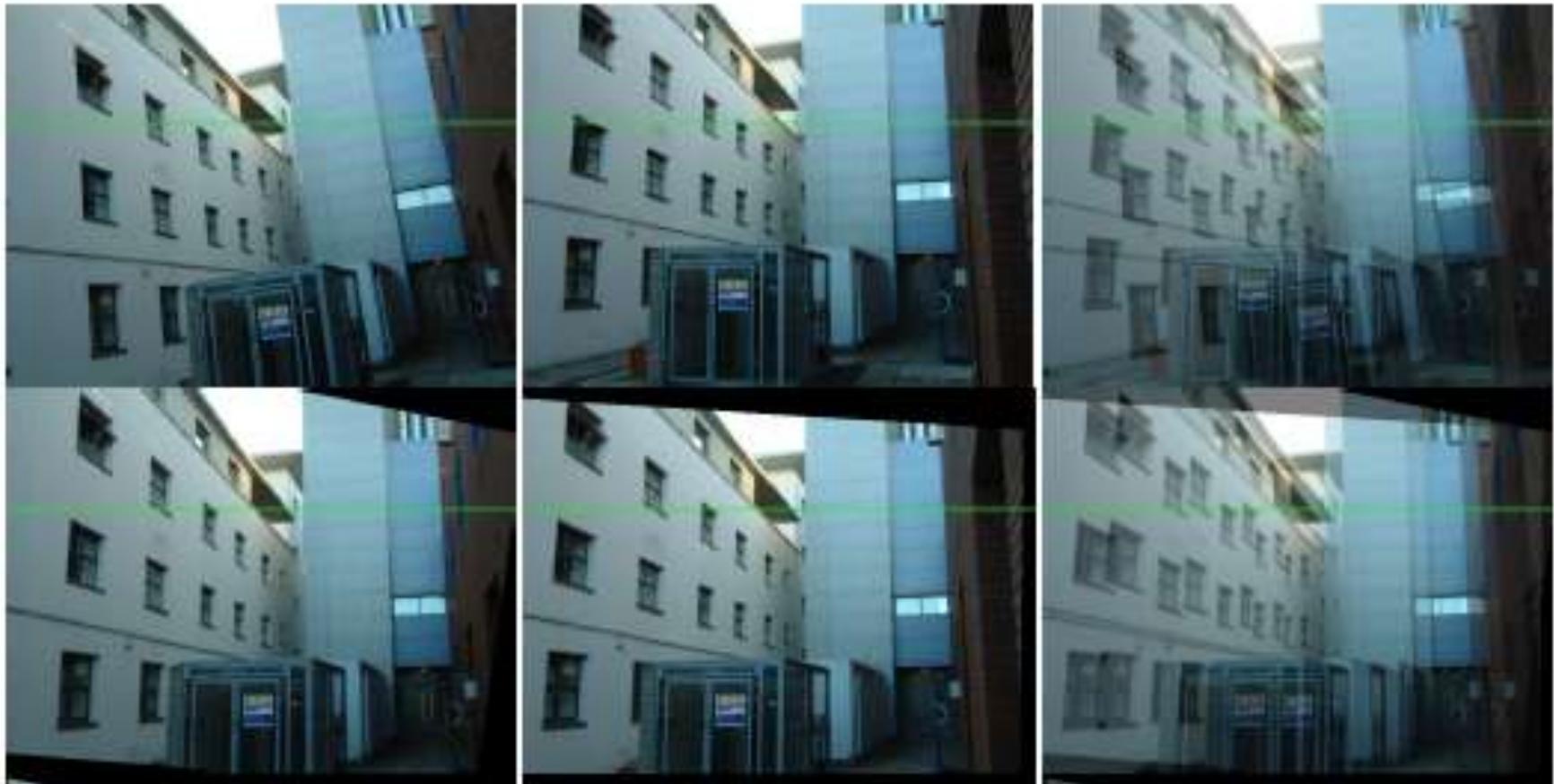
$$S(f) = \sum_{n=1}^N \left\{ d(\mathbf{p}_{n,d}, \mathbf{F}_3 \mathbf{e}_l) + d(\mathbf{p}_{n,l}, \mathbf{F}_3^T \mathbf{e}_d) \right\}$$

8. Optimalni  $f$  in  $\mathbf{K}$  dobimo, če vsoto  $S(f)$  minimiziramo v odvisnosti od  $f$  s pomočjo nelinearnega Levenberg-Marquardtovega minimizacijskega algoritma.
9. Iz optimizirane temeljne matrike  $\mathbf{F}_3$  izluščimo osnovno matriko  $\mathbf{E}_3$ , ki pa mora imeti pravilno strukturo, tj. dve singularni vrednosti enaki, tretjo pa 0.
10. Do takšne oblike pridemo tako, da prvo oceno za  $\mathbf{E}_3$  razcepimo na singularne vrednosti:  $\mathbf{E}_3 = \mathbf{U} \mathbf{D} \mathbf{V}^T$ . Nato dobljeno diagonalno matriko  $\mathbf{D}$  preuredimo v  $\mathbf{D}_3$  tako, da ima diagonalo  $(1, 1, 0)$  in z njo pridemo do prave ocene za  $\mathbf{E}_3$ :

$$\mathbf{E}'_3 = \mathbf{U} \mathbf{D}_3 \mathbf{V}^T$$

11. Iz  $\mathbf{E}'_3$  izluščimo dodaten zasuk  $\mathbf{R}_3$ .

# Primer za izravnavo stereo slik



Zgoraj: originalni sliki, spodaj: izravnani sliki

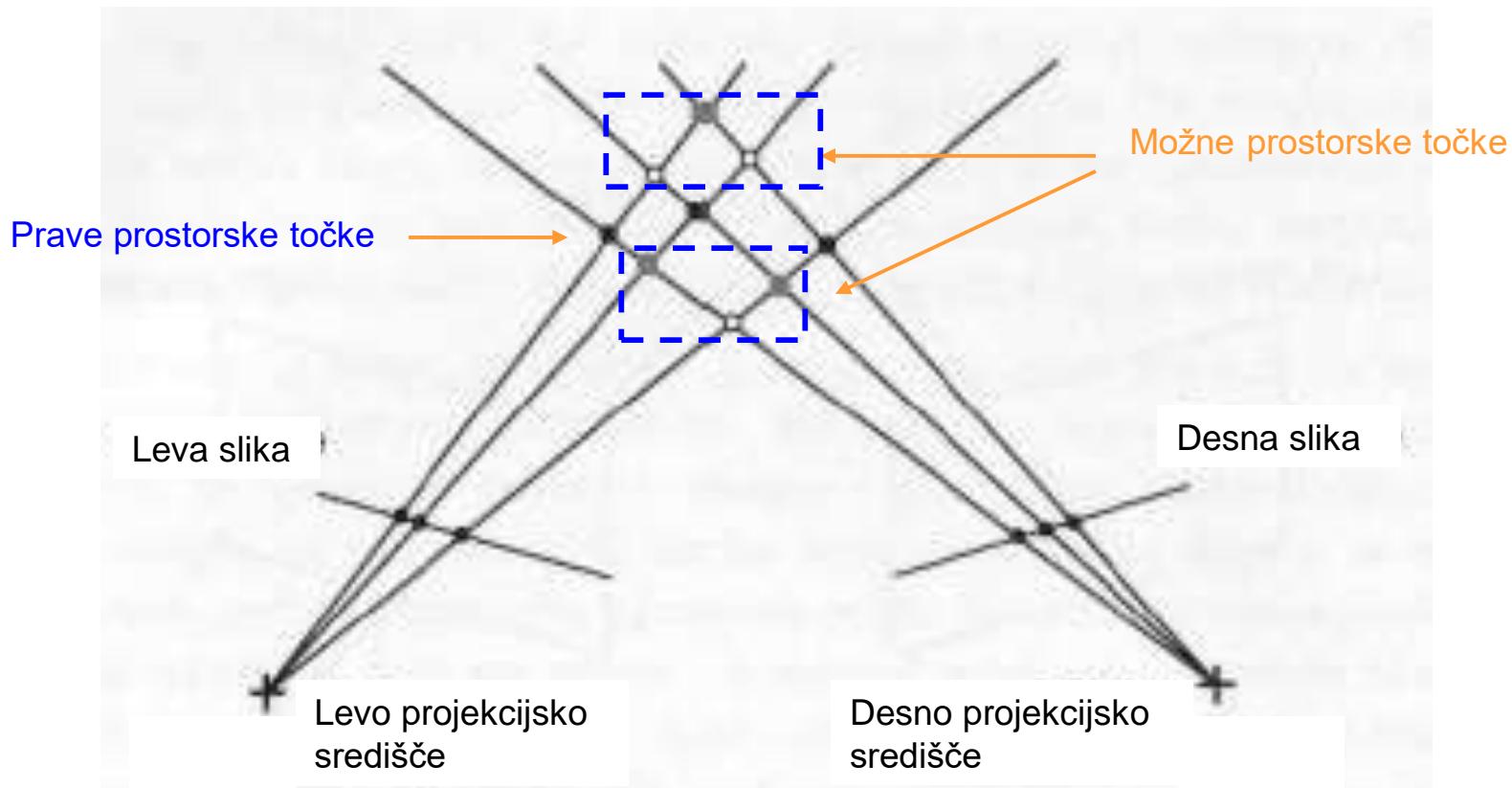
# DOLOČANJE GLOBINSKE SLIKE – ISKANJE KORESPONDENCE

1. Ko so epipolarne premice vzporedne, potrebujemo postopek za iskanje korespondenčnih točk, ki sovpadajo po lastnostih. Uveljavljene so:
  - metode na osnovi pikslov oz. slikovnih regij
  - metode, temelječe na značilnicah
2. Metode na osnovi vrednosti pikslov
  - primerjava podobnosti posameznih pikslov
  - iskanje podobnosti s korelacijo slikovnih regij (majhna okna)
  - iskanje podobnosti med lastnostmi okenskih funkcij (SAD – vsota absolutnih razlik, SSD – vsota kvadriranih razlik)
3. Metode, temelječe na značilnicah:
  - iskanje sovpadajočih premic (črt)
  - primerjava razpoznavnih robov
  - primerjava odkritih oglišč

# DOLOČANJE GLOBINSKE SLIKE – ISKANJE KORESPONDENCE S KORELACIJO

Pri iskanju korespondenčnih točk lahko povežemo napačne prostorske točke:

- vse točke iste projekcijske premice se namreč preslikajo v isto slikovno točko
- treba je najti podobne sosedstvne točke – na primer s korelacijo



# DOLOČANJE GLOBINSKE SLIKE – ISKANJE KORESPONDENCE S KORELACIJO (2)

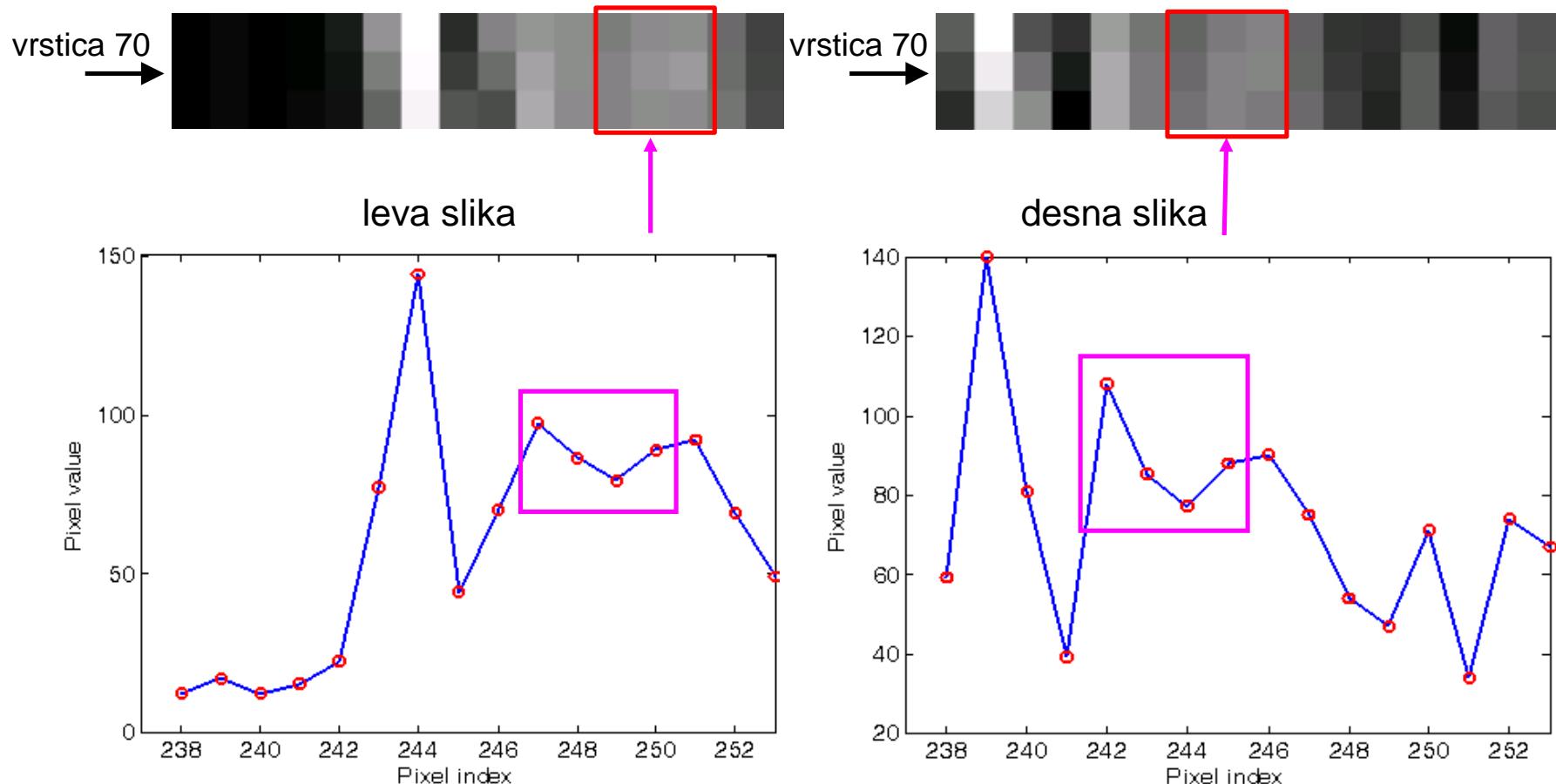
1. Imamo levo  $I_l$  in desno sliko  $I_d$
2. Za vsak piksel  $p_l(x,y)$  na levi sliki izračunamo korelacijo na desni sliki
3. Korelacijo računamo v oknu  $(2M+1) \times (2M+1)$ , ki ga premikamo po desni sliki glede na izbrano dispariteto  $\mathbf{d} = (d_x, d_y)$ :

$$c(\mathbf{d}) = \sum_{k=-M}^{M} \sum_{l=-M}^{M} I_l(x+k, y+l) I_d(x+k-d_x, y+l-d_y)$$

4. Dispariteto opredelimo z največjo izračunano prečno korelacijo:  
$$\mathbf{d} = \arg \max_{\mathbf{d}} [c(\mathbf{d})]$$
5. Če smo slike najprej izravnali:
  - dispariteto iščemo le po tisti vrstici (s spremenjanjem  $d_y$ ) na desni sliki, v kateri smo izbrali izhodiščno točko  $p(x,y)$  na levi sliki.
6. Namesto prečne korelacije lahko iščemo disparitetu tudi z:
  - normalizirano prečno korelacijo (delimo s standardnim odklonom)
  - vsoto kvadriranih razlik med pikslji
  - vsoto absolutnih razlik med pikslji

# DOLOČANJE GLOBINSKE SLIKE – ISKANJE KORESPONDENCE S KORELACIJO (3)

Profil pikselskih intenzivnosti in iskanje podobnosti s prečno korelacijo:



# DOLOČANJE GLOBINSKE SLIKE – ISKANJE KORESPONDENCE S PODOBNOSTJO ZNAČILNIC

1. Imamo levo  $I_l$  in desno sliko  $I_r$
2. Za vsak nabor značilnic  $f$ , na primer oglišč, na levi sliki ocenimo mero podobnosti z značilnicami na desni sliki
3. Podobnost iščemo v oknu  $(2M+1) \times (2M+1)$ , ki ga premikamo po desni sliki glede na izbrano dispariteto  $\mathbf{d} = (d_x, d_y)$ :

$$s(\mathbf{d}) = \frac{1}{\sum_{i=1}^{N_f} w_i (f_{i,l} - f_{i,d}(\mathbf{d}))^2}$$

pri čemer pomenijo  $w_i$  ustrezne uteži,  $N_f$  pa število značilnic.

4. Dispariteto opredelimo z največjo izračunano podobnostjo:

$$\mathbf{d} = \arg \max_{\mathbf{d}} [s(\mathbf{d})]$$

5. Če smo slike najprej izravnali:

- dispariteto iščemo le vzdolž tiste vrstice (s spremenjanjem  $d_y$ ) na desni sliki, v kateri smo izbrali izhodiščne značilnice  $f$ , na levi sliki.

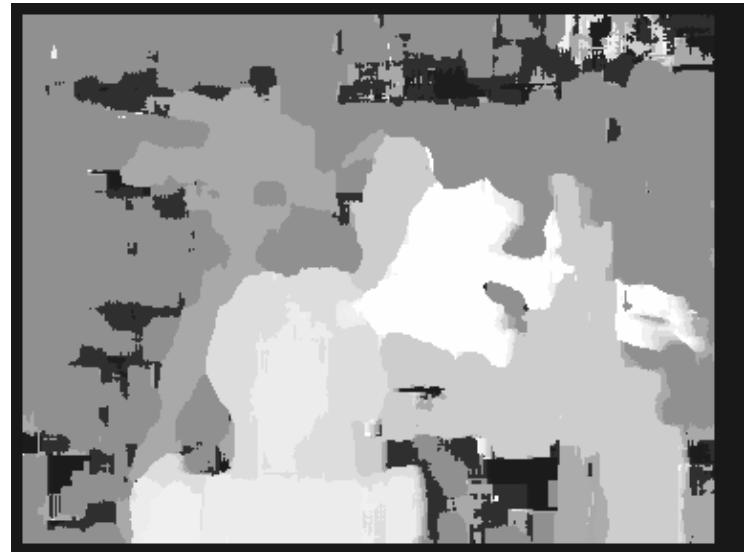
# PRIMERJAVA MED ISKANJI KORESPONDENCE S KORELACIJO IN Z ZNAČILNICAMI

- Metode s korelacijo:
  - laže se izvedejo
  - tvorijo **gostejše** mreže disparitet (koristno pri rekonstrukciji površin)
  - potrebujemo slike z izraženimi teksturami (drugače imamo mnogo napačnih ujemanj)
  - če se zorni koti zelo razlikujejo, ne deluje dobro, ker:
    - zaradi sprememb v smereh osvetlitve
      - predmeti nimajo idealnih odbojnih površin
    - zaradi spremenjanja dolžin
      - problem perspektive, ker površine predmetov niso ravne in pravokotne na optično os kamere
- Metode z značilnicami:
  - primerne, če lahko v sceni najdemo zanesljive značilnice
  - so hitrejše od korelacijskih metod
  - pripeljejo do **redkih** disparitetnih mrež
    - sprejemljivo za navigacijo
  - so precej neobčutljive na spremenljivo osvetlitev

# Primer globinske slike pri iskanju korespondence s prečno korelacijo



Originalna slika



Globinska slika: korespondenca dobljena s prečno korelacijo; temnejša območja pomenijo večjo oddaljenost od kamere, črno označena pa predele, kjer se globina ni dala določiti.

# UPORABA STEREO VIDA

1. Letalsko kartiranje
2. Forenzika – scene zločinov, prometne nesreče
3. Rudarstvo – meritve odkopa
4. Gradbeništvo – opazovanje struktur
5. Izogibanje trkom
  - aktivne metode zaznavanja (radar, sonar) so primerne le za nenatrpano, “redke” scene
  - potrebno je realnočasovno delovanje
  - natančno določanje globine je kritično
6. Proizvodnja
  - nadzor procesov
  - opazovanje procesov
7. Splošna fotogrametrija
  - kakršnekoli brezkontaktne meritve

# Optični tok

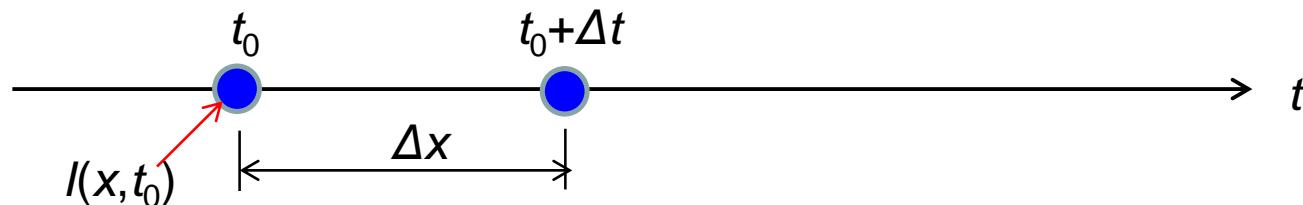
(povzeto po Sebastian Thrun, Dan Maynes-Aminzade,  
Mitul Saha, Greg Corrado: Optical Flow. Stanford  
University, 2006)

# OPTIČNI TOK - IZHODIŠČA

1. Kadar imamo več zaporednih posnetkov, ki so nastali med relativnim premikanjem kamere glede na sceno, lahko na zaporednih parih slik zaznamo spremembe pikselske intenzivnosti.
2. Lokalno omejeno gibanje (premike) slike imenujemo **optični tok** (*optical flow*):
  - optični tok je opredeljen z lokalnimi odvodi med zaporednimi slikami
  - glede na smeri premikanja govorimo o 1D, 2D in 3D optičnem toku
  - predvidevamo, da so vse spremembe intenzivnosti na slikah odvisne samo od gibanja:
    - objekti na sceni se ne smejo deformirati oz. morajo biti vsaj lokalno togi
    - osvetlitev scene se med snemanjem gibanja ne sme spremenjati
    - scena in kamera se ne smeta premikati z isto hitrostjo v isto smer
    - gibanje objektov mora povzročiti zaznavne spremembe v intenzivnosti pikslov (vrteča se, gladka kovinska krogla ne povzroča takšnih sprememb, čeprav se giblje)
3. Izhodišče za računanje diferencialnega optičnega toka je t. i. **enačba omejenega gibanja** (*motion constraint equation*).

# OMEJENO GIBANJE V 1D

1. Imejmo točko z intenzivnostjo  $I(x)$ , ki se premika v smeri  $x$ .
2. Premik za  $\Delta x$  se naj zgodi v času  $\Delta t$ , kar pomeni, da se točka premika v smeri  $x$  s hitrostjo  $v_x = \Delta x / \Delta t$ .
3. Vzemimo, da sta tudi dve zaporedni sliki točke v času razmaknjeni ravno za  $\Delta t$ .



4. Označimo zaporedne slike kar s časovnimi trenutki, ob katerih so nastale: intenzivnost točke na položaju  $x$  bo tako v času  $t$  označena kot  $I(x, t)$
5. Privzeli smo, da se intenzivnost (sivine oz. barve) točke  $I(x, t)$  med premikanjem ne spremeni; zato velja:

$$I(x, t) = I(x + \Delta x, t + \Delta t)$$

# ENAČBA OMEJENEGA GIBANJA V 1D

- Desno stran enačbe  $I(x,t) = I(x+\Delta x, t+\Delta t)$  razvijmo v Taylorjevo vrsto (upoštevajmo samo prvi člen, saj je zahtevano, da so premiki čim manjši, in zato so členi višjih redov zanemarljivi):

$$I(x,t) = I(x,t) + \frac{\partial I(x,t)}{\partial x} \Delta x + \frac{\partial I(x,t)}{\partial t} \Delta t$$



$$\frac{\partial I(x,t)}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I(x,t)}{\partial t} = 0$$

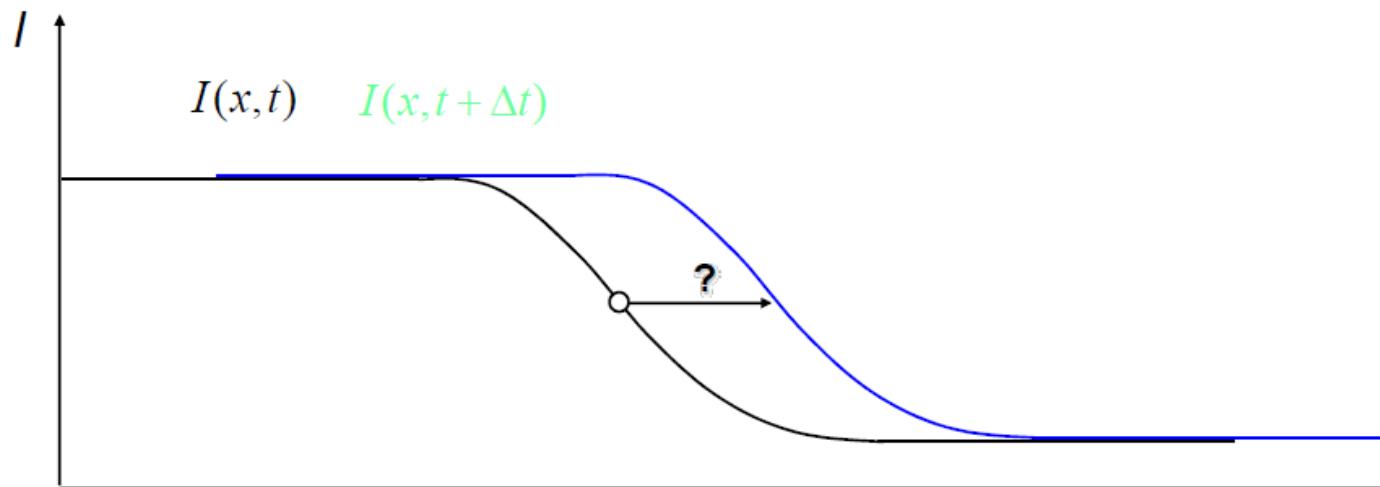
- Dobljena enačba se imenuje enačba omejenega gibanja.
- Parcialni odvod intenzivnosti točke  $I(x,t)$  po  $x$  pomeni slikovni gradient, tj. spremenjanje intenzivnosti točk v smeri  $x$ ; krajše ga označimo z  $I_x$ .
- Parcialni odvod  $I(x,t)$  po  $t$  pomeni gradient v času oz. spremenjanje intenzivnosti točke s slike na sliko pri isti koordinati  $x$ ; krajše ga označimo z  $I_t$ .

# OPTIČNI TOK OZ. HITROST GIBANJA V 1D

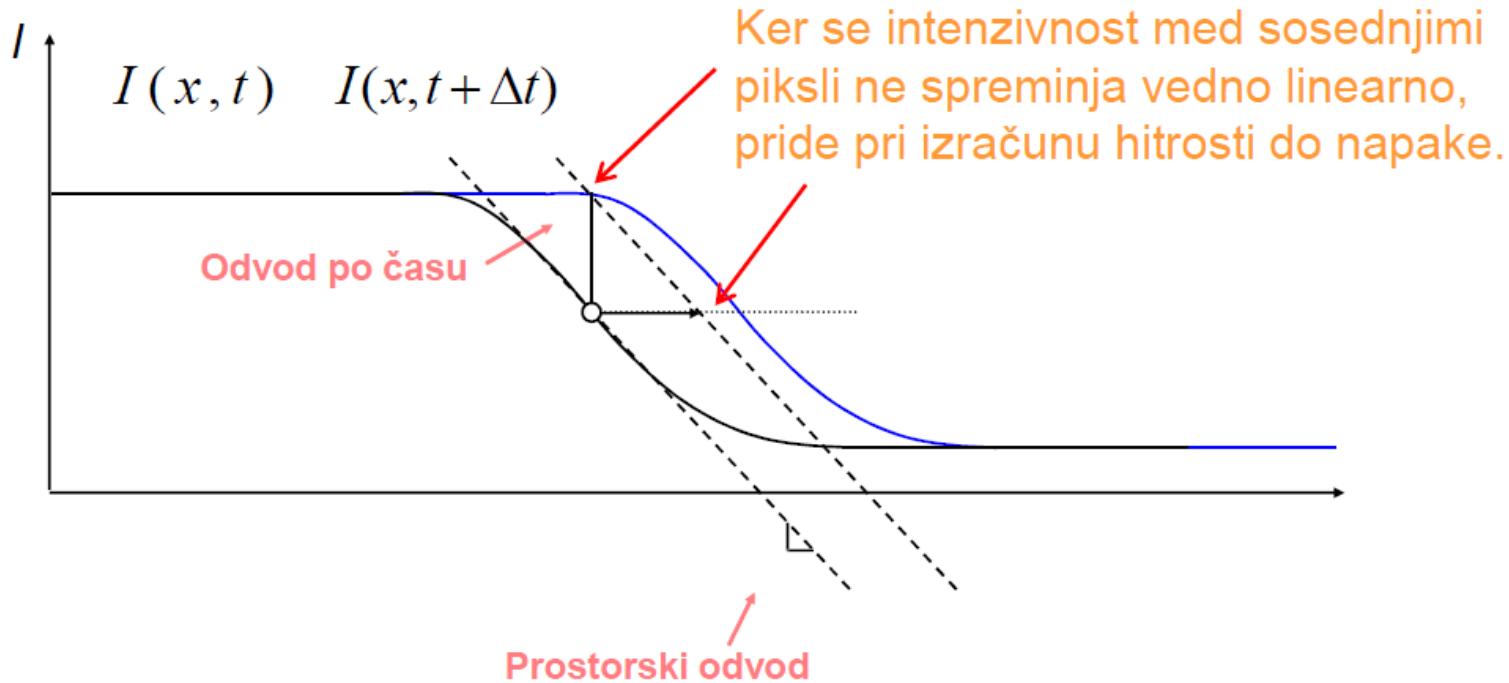
- Iz enačbe omejenega gibanja lahko določimo hitrost gibanja  $v_x$ :

$$v_x = -\frac{\frac{\partial I(x,t)}{\partial t}}{\frac{\partial I(x,t)}{\partial x}} = -\frac{I_t}{I_x}$$

- Če poznamo hitrost premikanja točke, ji lahko sledimo skozi čas (s slike na sliko).



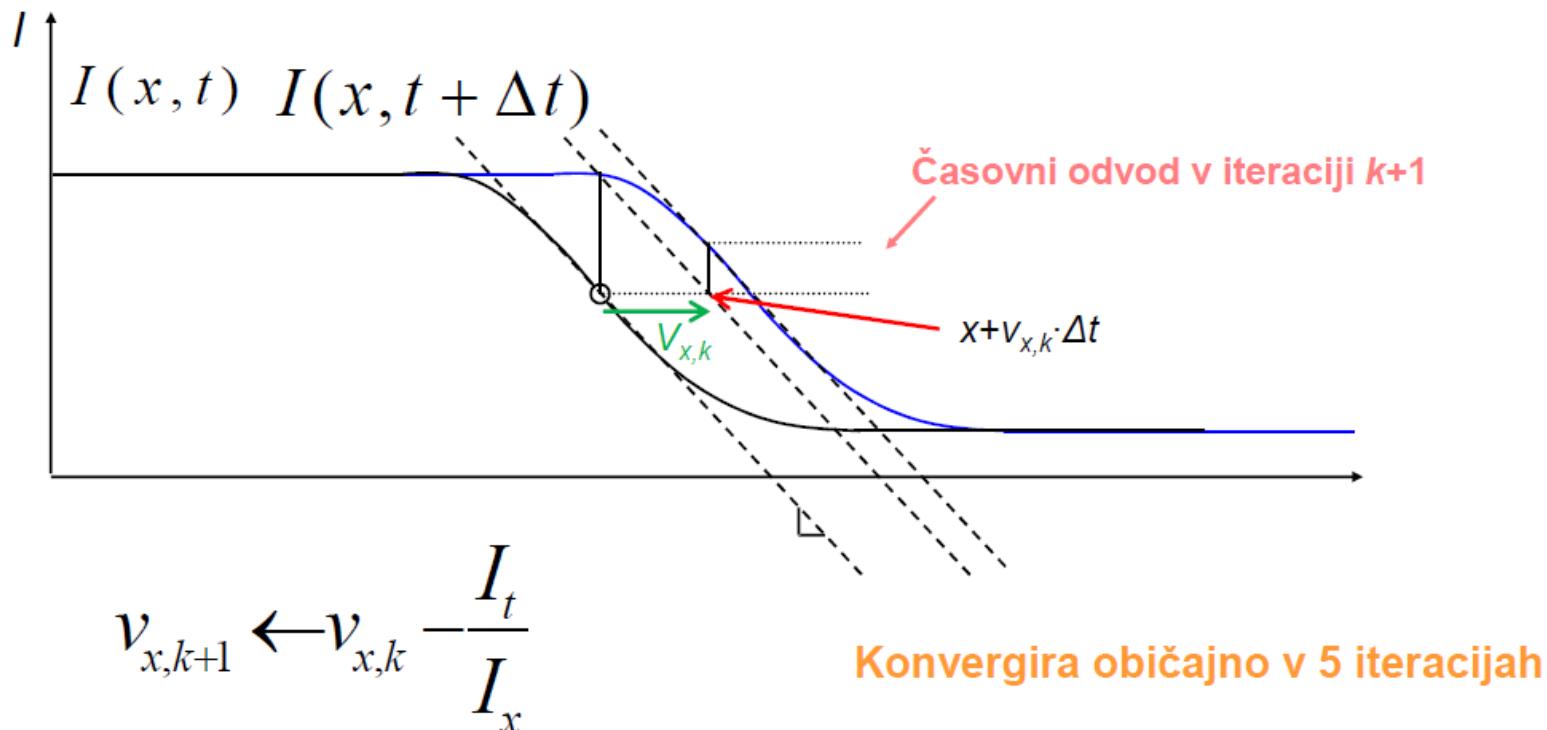
# PRIKAZ ZA IZRAČUN HITROSTI GIBANJA V 1D



$$I_x = \frac{\partial I}{\partial x} \Big|_t \quad I_t = \frac{\partial I}{\partial t} \Big|_{x=p} \quad \rightarrow v_x \approx -\frac{I_t}{I_x} \quad \left\{ \begin{array}{l} \text{Privzeto:} \\ \bullet \text{nespremenljiva osvetlitev} \\ \bullet \text{majhni premiki} \end{array} \right.$$

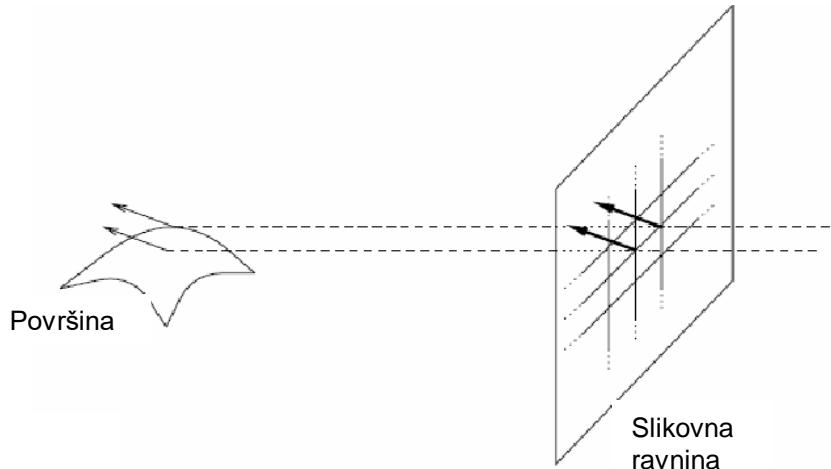
# ITERATIVNI POPRAVKI ZA IZRAČUN HITROSTI GIBANJA V 1D

1. Označimo izračun hitrosti v  $k$ -tem koraku z  $v_{x,k}$ .
2. V koraku  $k+1$  vzamemo na prvi sliki koordinato  $x+v_{x,k}\cdot\Delta t$  in izračunamo razliko intenzivnosti do istoležne točke na drugi sliki (uporabimo lahko linearno interpolacijo)



# OMEJENO GIBANJE V 2D

1. Imejmo točko z intenzivnostjo  $I(x,y)$ , ki se lahko premika v smereh  $x$  in  $y$ .
2. Premik za  $\Delta x$  in  $\Delta y$  se naj zgodi v času  $\Delta t$ , kar pomeni, da se točka premika v smeri  $x$  s hitrostjo  $v_x = \Delta x / \Delta t$ , v smeri  $y$  pa s hitrostjo  $v_y = \Delta y / \Delta t$ .
3. Privzamemo, da sosednje točke na sceni pripadajo isti površini in imajo zato enako gibanje.
4. Te točke se projicirajo kot sosednje tudi na sliki, zato pričakujemo, da bo 2D optični tok zanje prostorsko koherenzen (skladen).



# ENAČBA OMEJENEGA GIBANJA V 2D

1. Izračenje intenzivnosti piksla  $I(x, y, t)$  z njegovo premaknjeno inačico  $I(x + \Delta x, y + \Delta y, t + \Delta t)$  da:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

2. Razvoj v Taylorjevo vrsto (upoštevajmo samo prvi člen, saj je zahtevano, da so premiki čim manjši, in zato so členi višjih redov zanemarljivi):

$$I(x, y, t) = I(x, y, t) + \frac{\partial I(x, y, t)}{\partial x} \Delta x + \frac{\partial I(x, y, t)}{\partial y} \Delta y + \frac{\partial I(x, y, t)}{\partial t} \Delta t$$



$$\frac{\partial I(x, y, t)}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I(x, y, t)}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I(x, y, t)}{\partial t} = 0$$

3. Prišli smo do enačbe omejenega gibanja v 2D.
4. Parcialna odvoda intenzivnosti piksla  $I(x, y, t)$  po  $x$  in  $y$  označimo z  $I_x$  in  $I_y$ , parcialni odvod po času pa z  $I_t$ .

# **ENAČBA OMEJENEGA GIBANJA V 2D (2)**

- ## 1. Enačba omejenega gibanja v vektorski obliki:

$$I_x v_x + I_y v_y + I_t = 0$$

$$[I_x, I_y][v_x, v_y]^T + I_t = 0$$

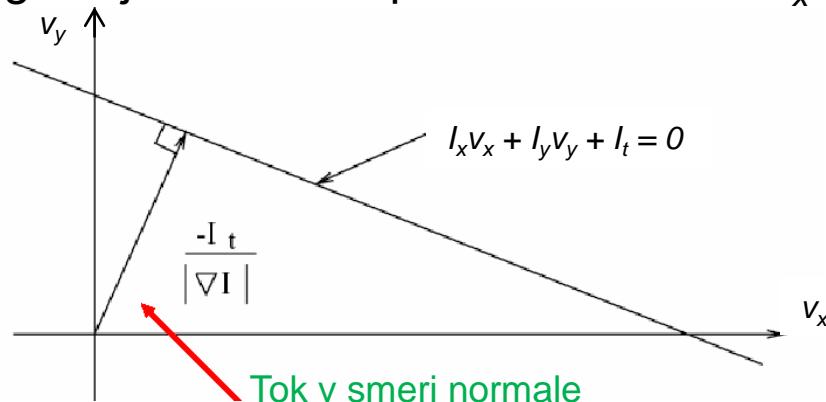
$$\nabla \mathbf{I}^T \mathbf{v} = -I_t; \quad \nabla \mathbf{I} = \begin{bmatrix} I_x \\ I_y \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

↓

2D gradient

vektor hitrosti gibanja

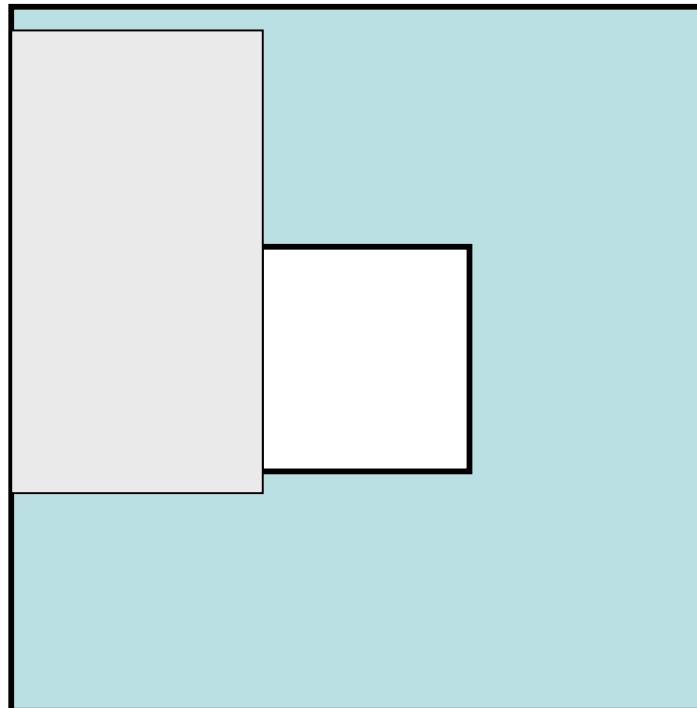
2. Enačba  $I_x v_x + I_y v_y = -I_t$  vsebuje dve neznani komponenti hitrosti gibanja in določa premico v ravni  $v_x - v_y$ :



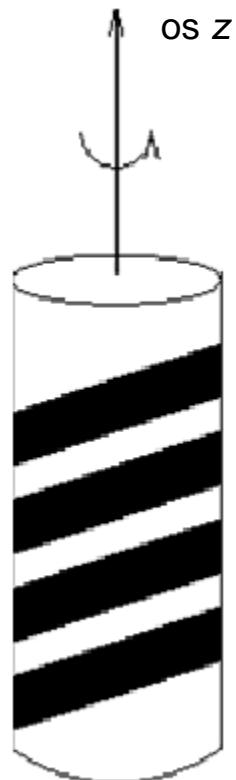
Z eno samo točko lahko določimo le optični tok pravokotno na svetlobni (intenzivnostni) gradient. Rešitev: vzeti več sosednjih točk skupaj.

# PROBLEM ZASLONKE (OKNA)

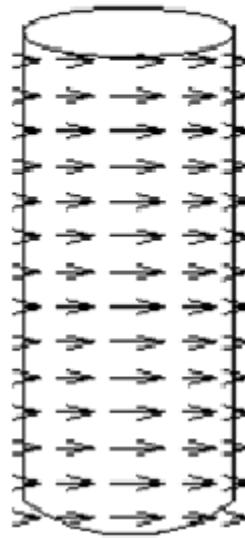
1. Če skušamo oceniti gibanje samo z opazovanjem omejenega območja, naletimo na t. i. **problem zaslонke oz. okna** (*aperture problem*):
  - če pri pomikanju vidimo skozi zaslonko (okno) le homogeno območje, se ne bodo pojavile razlike, ki bi nakazale smer gibanja
  - če skozi zaslonko ne vidimo tekture (robov) v vseh smereh, bomo lahko ocenili samo tisto komponento optičnega toka, ki je pravokotna na vidni rob



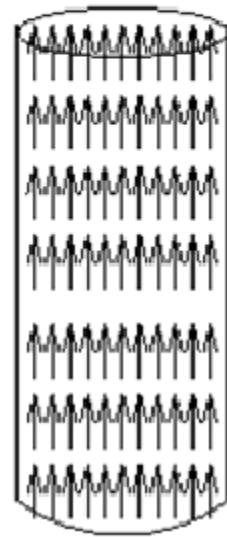
# Primer za problem zaslonek



Vrteči se valj



Smer gibanja



Ocenjeni optični tok

# REŠITEV PROBLEMA ZASLONKE

1. Za vsak piksel je treba dobiti več enačb, iz katerih pravilneje ocenimo vektor hitrosti:
  - prva možnost: upoštevamo, da mora biti polje optičnega toka v okolini nekega piksla gladko (brez hitrih sprememb vektorjev hitrosti)
  - druga možnost: privzamemo, da imajo vsi piksli v ožji okolici opazovanega piksla enak vektor hitrosti (izberemo manjše okno)

2. Primer sistema enačb za okno  $5 \times 5$ :

$$\begin{bmatrix} I_x(\mathbf{p}_1), I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2), I_y(\mathbf{p}_2) \\ I_x(\mathbf{p}_3), I_y(\mathbf{p}_3) \\ \vdots \\ I_x(\mathbf{p}_{25}), I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = -\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ I_t(\mathbf{p}_3) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix} \Rightarrow \nabla \mathbf{I}^T \cdot \mathbf{v} = -\mathbf{I}_t$$

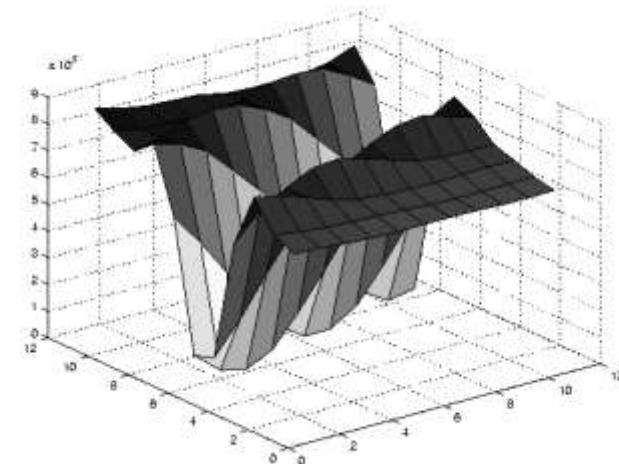
3. Rešitev:

$$\mathbf{v} = -[(\nabla \mathbf{I})(\nabla \mathbf{I})^T]^{-1} (\nabla \mathbf{I}) \mathbf{I}_t \quad \text{← Lucas-Kanadejeva rešitev}$$

# POGOJI ZA LUCAS-KANADEJEVO REŠITEV

1. Lucas-Kanadejeva enačba vodi do zadovoljivih rezultatov, če:
  - matrika  $(\nabla I)(\nabla I)^T$  se da invertirati
  - matrika  $(\nabla I)(\nabla I)^T$  zaradi vpliva šuma ne sme biti premajhna, zato njeni lastni vrednosti  $\lambda_1$  in  $\lambda_2$  ne smeta biti premajhni
  - matrika  $(\nabla I)(\nabla I)^T$  mora biti dobro pogojena, tako da razmerje  $\lambda_1/\lambda_2$  ni veliko ( $\lambda_1$  je večja lastna vrednost)
2. Lastne vrednosti matrike  $\mathbf{Q}(x,y) = (\nabla I)(\nabla I)^T$  smo spoznali že pri Harrisovem detektorju oglišč:
  - če smo na homogenem območju brez izrazitih tekstur, sta obe lastni vrednosti ( $\lambda_1$  in  $\lambda_2$ ) majhni
  - če smo na robu, je ena lastna vrednost majhna, druga velika (odvisno od usmerjenosti roba)
  - če smo na oglišču, sta obe lastni vrednosti veliki
3. Pogoji za ugodno rešitev Lucas-Kanadejeve enačbe določajo, da optični tok najnatančneje določimo, če ga računamo v okolici oglišč.
4. S Harrisovim detektorjem torej lahko poiščemo točke oz. območja, kjer bo izračun optičnega toka in s tem smeri in hitrosti gibanja objektov (ali kamere) optimalen.

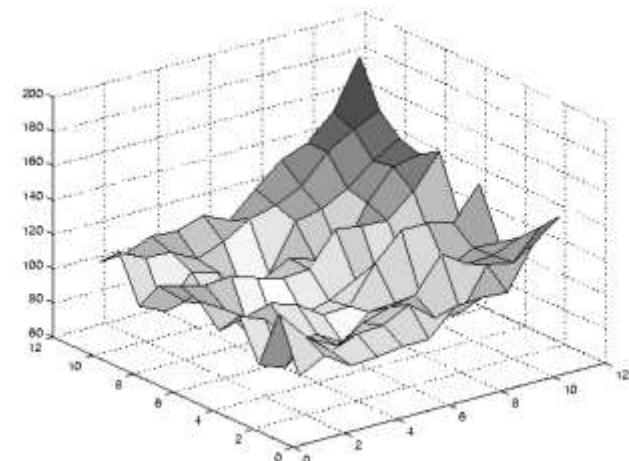
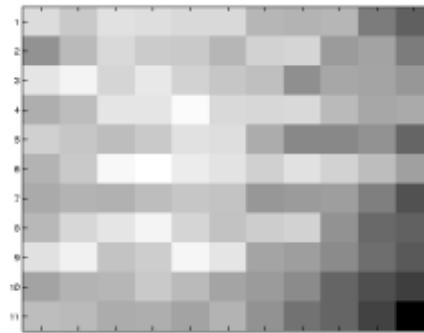
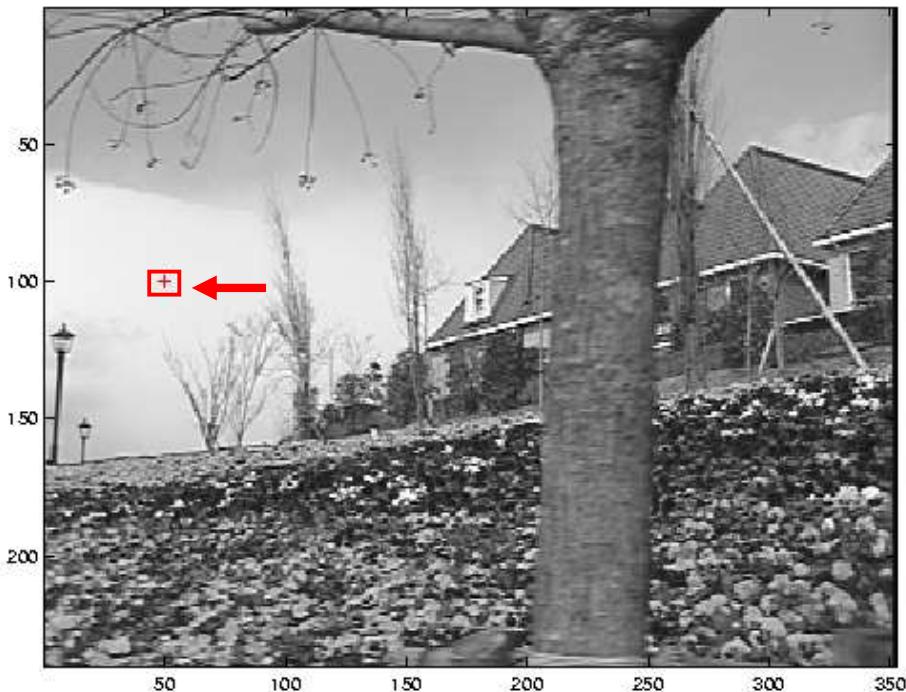
# Primer za izračun optičnega toka – rob



$$\Sigma (\nabla I)(\nabla I)^T$$

- veliki in enaki gradienti
- velik  $\lambda_1$ , majhen  $\lambda_2$

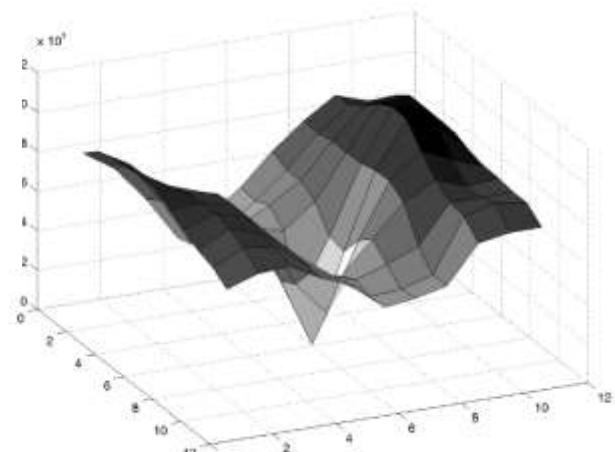
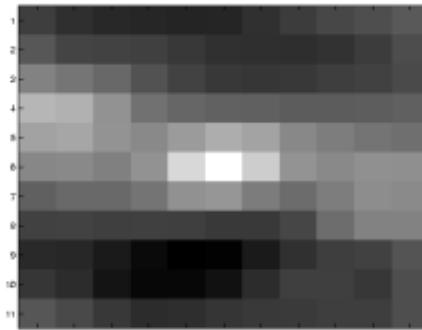
# Primer za izračun optičnega toka – homogeno območje (ozadje)



$$\Sigma (\nabla I)(\nabla I)^T$$

- gradienti imajo majhne vrednosti
- $\lambda_1$  in  $\lambda_2$  sta majhna

# Primer za izračun optičnega toka – območje z izrazito teksturo



$$\Sigma (\nabla I)(\nabla I)^T$$

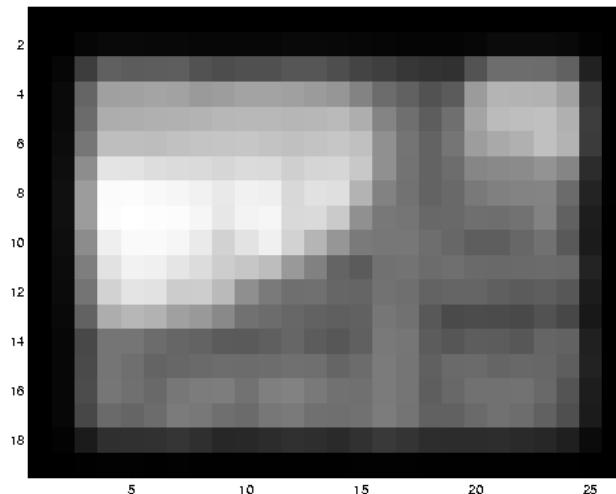
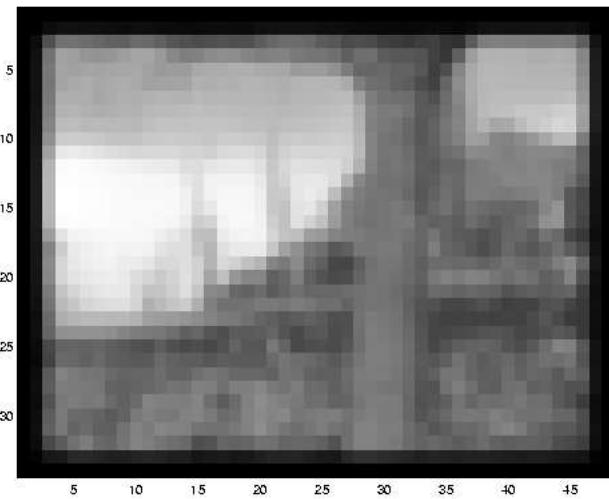
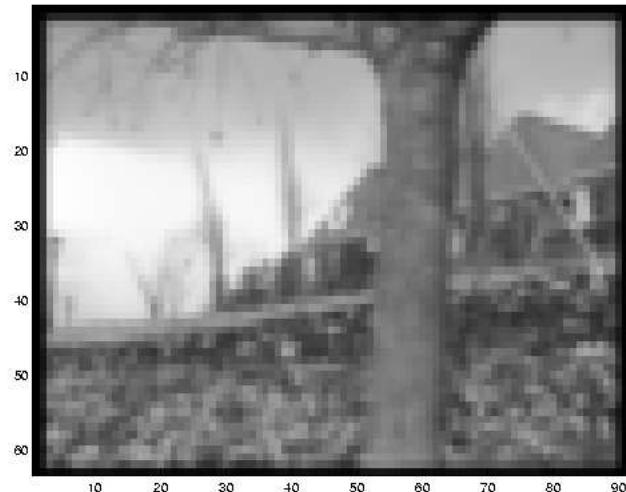
- gradienti so različni, a veliki
- velika  $\lambda_1$  in  $\lambda_2$

# ZAHTEVA PO MAJHNIH PREMIKIH – SPREMENBAH

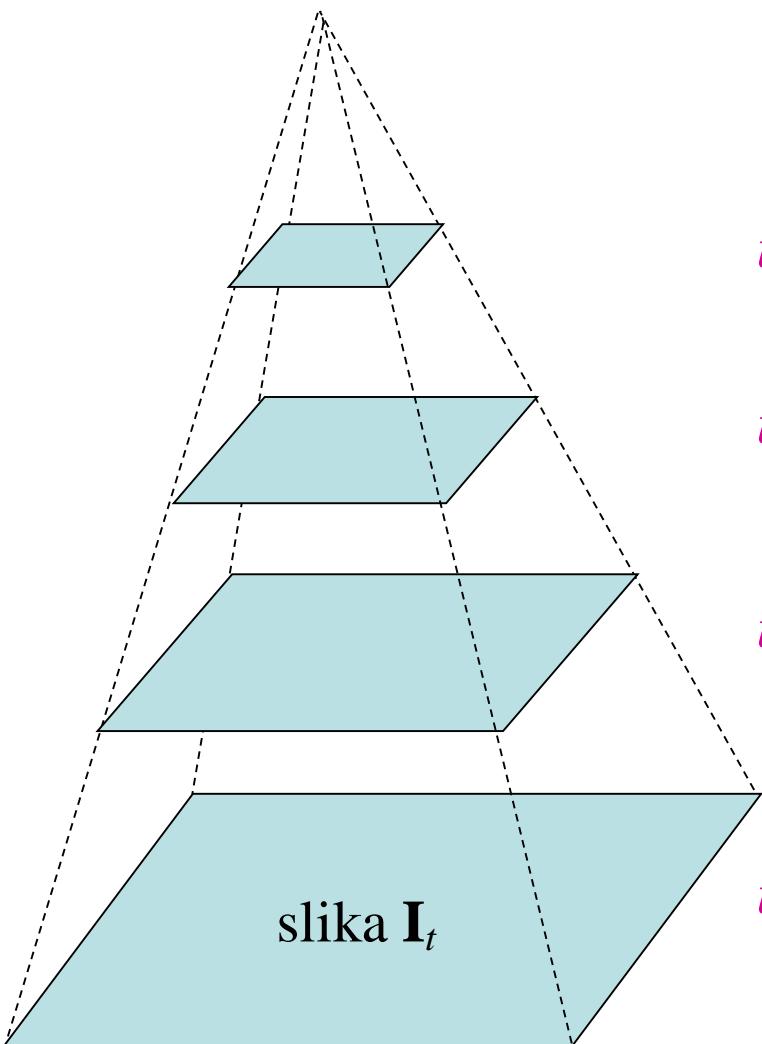
1. Spremembe zaradi gibanja ne smejo preseči enega piksla
2. Problem lahko obidemo, če zmanjšamo ločljivost slike, nato pa gibanje spremljamo nazaj proti višjim ločljivostim skozi večločljivostno shemo



# ZNIŽANJE LOČLJIVOSTI SLIK



# OCENA OPTIČNEGA TOKA V VEČLOČLJIVOSTNI SHEMI



Gaussova piramida za sliko  $I_t$

Predavanja iz Računalniškega vida, FERI Maribor

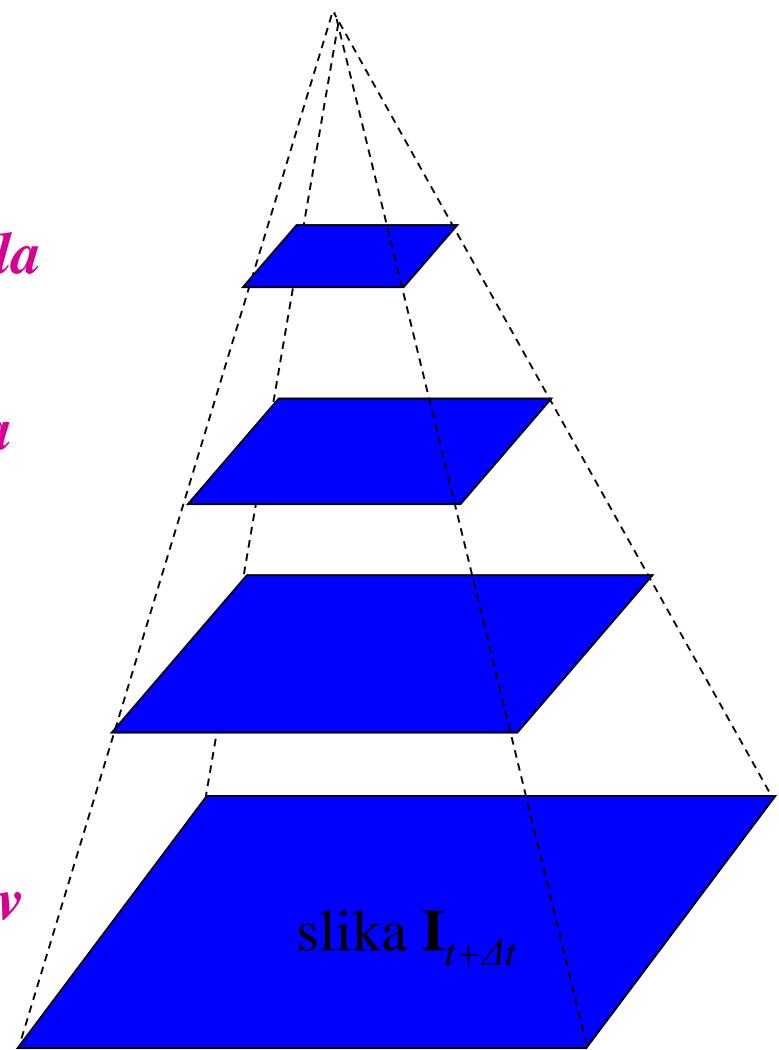
*Ločljivost:*

$u=1,25$  piksla

$u=2,5$  piksla

$u=5$  piksov

$u=10$  piksov

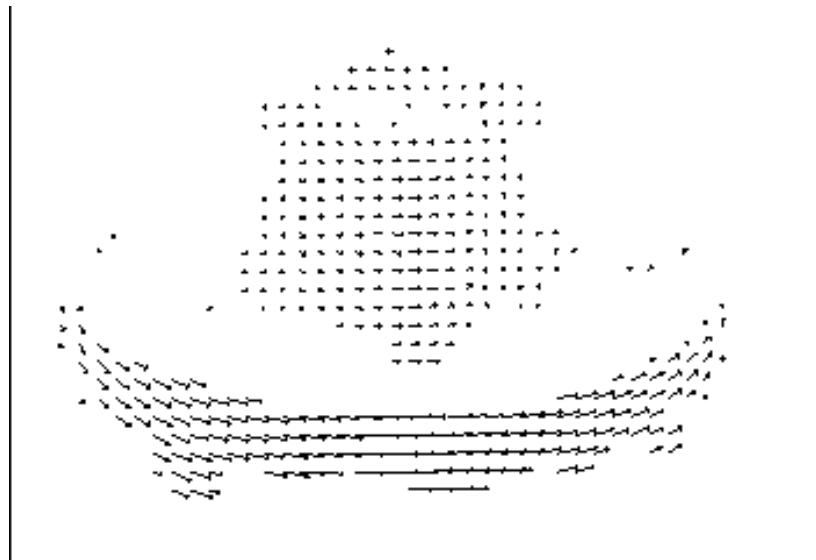
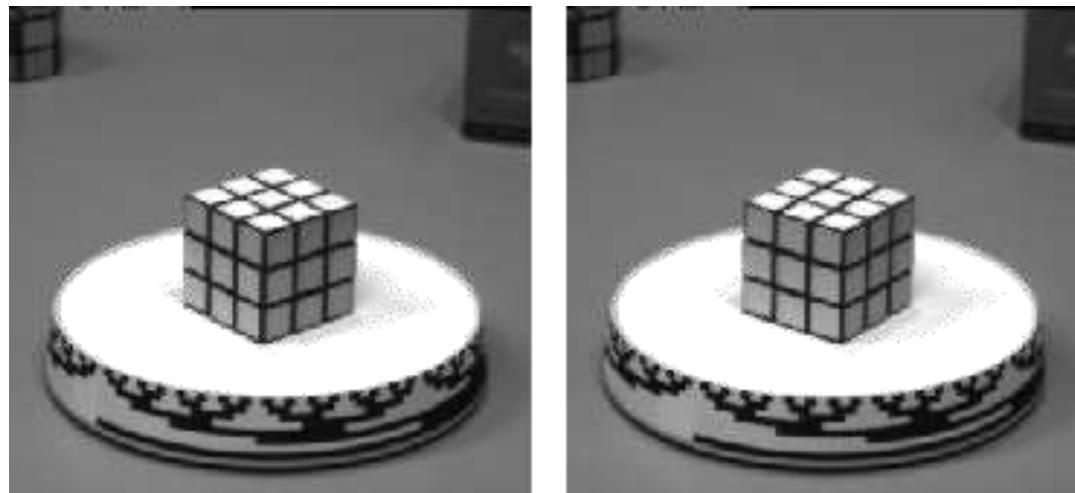


Gaussova piramida za sliko  $I_{t+\Delta t}$

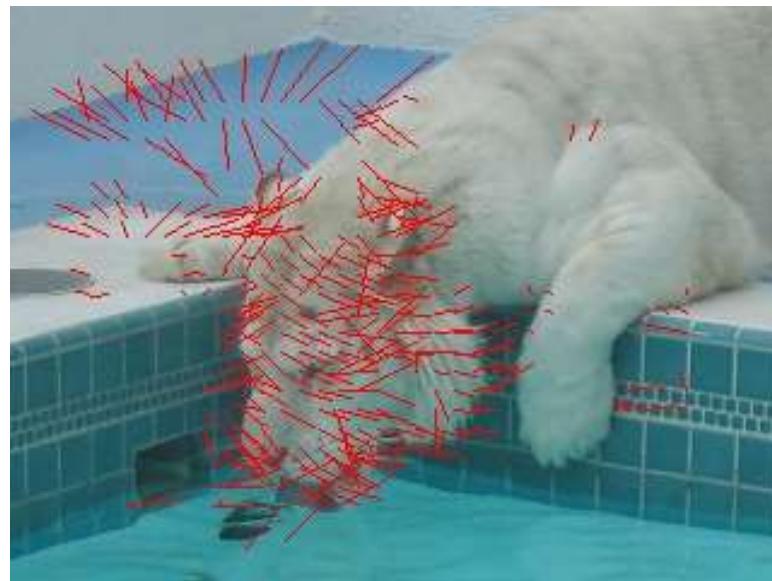
# ALGORITEM ZA VEČLOČLJIVOSTNO RAČUNANJE OPTIČNEGA TOKA PO LUCAS-KANADEJU

1. Izračunaj optični tok po Lucas-Kanadeju na najnižjo ločljivost
2. Za vsako višjo ločljivost  $k$ 
  - vzemi optični tok  $(v_{x,k-1}, v_{y,k-1})$  z nižje ločljivosti
  - izračunaj dvojno ločljivost toka z bilinearno interpolacijo na stopnji  $k$ :  $(v'_{x,k}, v'_{y,k})$
  - množi  $(v'_{x,k}, v'_{y,k})$  z 2
  - izračunaj odmik  $(\Delta x, \Delta y)$  glede na novo ocenjeno hitrost in se prestavi v nove koordinate pikslov
  - na tako premaknjeni mreži določi popravljen optični tok glede na sliki z višjo ločljivostjo (uporabi Lucas-Kanadeja ):  $(v''_{x,k}, v''_{y,k})$
  - za dokončno oceno hitrosti pri višji ločljivosti je treba sešteti začetno oceno hitrosti na stopnji  $k$  in popravke na tej stopnji:  
$$(v_{x,k}, v_{y,k}) = (v'_{x,k}, v'_{y,k}) + (v''_{x,k}, v''_{y,k})$$

# Primer optičnega toka



## Primer optičnega toka (2)



# Struktura iz gibanja

(razdelitev na strukturo scene in gibanje kamere)

# STRUKTURA 3D SCENE IZ DVEH ALI VEČ POGLEDOV – IZHODIŠČA

## 1. Cilj:

V zaporedju slik določiti korespondenčne pare značilnic, recimo nadzornih točk, in oceniti **3D-položaj točk, projiciranih v korespondenčne pare**

2. Privzeta je toga scena in projekcijska geometrija pri kamerah, s katerimi so projicirane točke v 3D na posamezne slikovne ravnine
3. Za določitev 3D-strukture moramo imeti možnost, da iz več posnetkov izluščimo globinsko informacijo. Te ne moremo določiti v dveh primerih, ki vodita le do homografij – homografija pomeni le preslikavo neposredno z ene slike na drugo, ne pa skozi reprojekcijo in projekcijo 3D-točk:
  - če so različni pogledi posnetki scene, ki je ravninska (planarna), iz teh slik ni mogoče opredeliti globine, torej oddaljenosti scenske ravnine od kamere
  - če so različni pogledi dobljeni le s sukanjem kamere okoli njenega projekcijskega središča, kljub 3D-sceni ne moremo s pomočjo slik opredeliti globine točk v 3D

# RAVNINSKA SCENA – 3D OPREDELITEV NI MOŽNA

1. Imamo ravninsko sceno in na njej opazujemo točko  $\mathbf{p}$ . Ravnina, na kateri je (2D) scena, ima normalo  $\mathbf{n}$ . Oddaljenost od izhodišča svetovnega koordinatnega sistema naj bo  $d$ . Dva pogleda sta zasukana z matriko  $\mathbf{R}$  in premaknjena za  $\mathbf{t}$ . Projekcija 3D-točke na sliki sta  $\mathbf{p}_s$  in  $\mathbf{p}'_s$ .

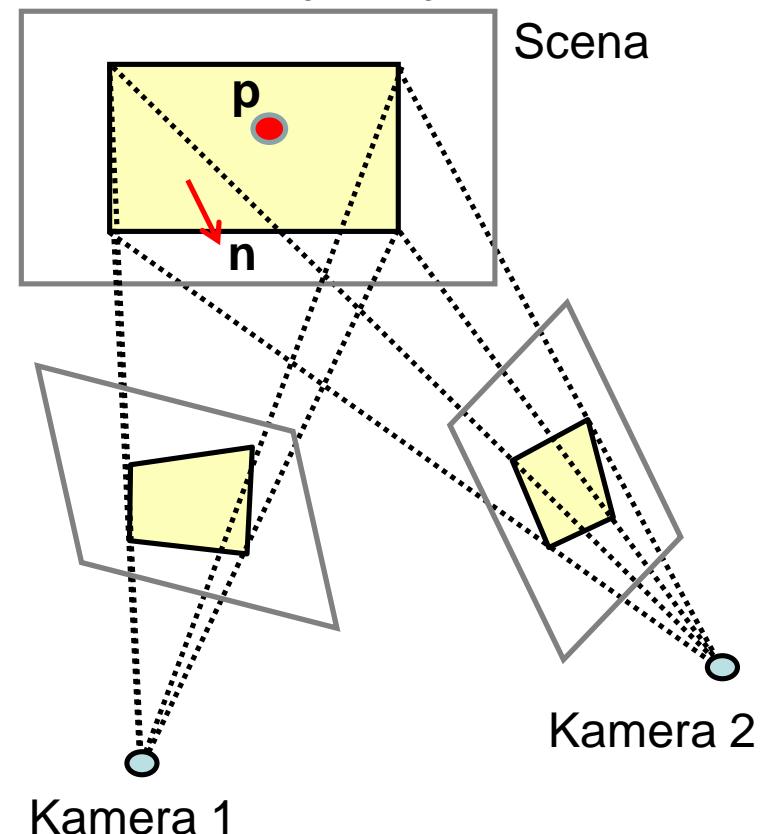
$$\mathbf{n}^T \mathbf{p} = d \quad (n_x x + n_y y + n_z z = d)$$

$$\begin{aligned}\mathbf{p}' &= \mathbf{Rp} + \mathbf{t} = \mathbf{Rp} + \mathbf{t} \left( \frac{\mathbf{n}^T \mathbf{p}}{d} \right) \\ &= \left( \mathbf{R} + \frac{\mathbf{tn}^T}{d} \right) \mathbf{p}\end{aligned}$$

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{tn}^T}{d}, \quad \mathbf{p}' = \mathbf{Hp}$$

$$\mathbf{p}'_s = \frac{z}{z'} \mathbf{Hp}_s$$

homografija med slikama



# ZASUK KAMERE OKOLI GORIŠČA – 3D OPREDELITEV NI MOŽNA

1. Imamo 3D-sceno in kamero, ki jo posname iz dveh zornih kotov. Med pogledoma obstaja samo zasuk, ki ga določa matrika  $\mathbf{R}$ . Na sceni imamo točko  $\mathbf{p}$ , ki se projicira na obe slike kot  $\mathbf{p}_s$  in  $\mathbf{p}'_s$ .
2. Preslikava scene med pogledoma:

$$\mathbf{p}' = \mathbf{Rp}, \quad \mathbf{t} = 0$$

3. Vzvratna projekcija slikovnih točk v 3D:

$$\mathbf{p} = \frac{z}{f} \mathbf{p}_s, \quad \mathbf{p}' = \frac{z'}{f} \mathbf{p}'_s$$

4. Ko oboje povežemo, dobimo:

$$\mathbf{p}'_s = \frac{z}{z'} \mathbf{Rp}_s$$

homografija med slikama

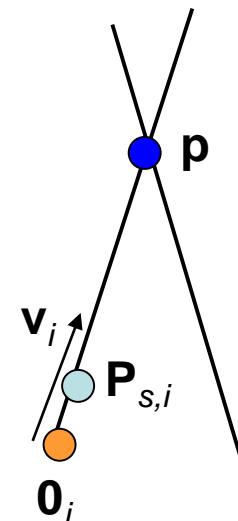
# STRUKTURA 3D SCENE IZ DVEH ALI VEČ POGLEDOV – TRIANGULACIJA

1. Imamo dva ali več pogledov na isto sceno. Z  $i$ -to kamero vidimo 3D-točko  $\mathbf{p}$  kot projekcijo  $\mathbf{p}_{s,i}$
2. Iz projekcijskih središč (gorišč) kamer potegnemo vzvratne projekcije (premice) skozi slikovne točke  $\mathbf{p}_{s,i}$ .
3. Sečišče teh premic pomeni položaj točke  $\mathbf{p}$  v prostoru.
4. Iščemo minimalno napako glede na položaj točke  $\mathbf{p}$ :

$$\arg \min_{\mathbf{p}} \sum_i \|\mathbf{0}_i + s_i \mathbf{v}_i - \mathbf{p}\|$$

- $\mathbf{p}$  – neznana točka v 3D
- $\mathbf{0}_i$  – projekcijsko središče kamere  $i$
- $\mathbf{v}_i$  – vektor, ki določa reprojekcijski “žarek” skozi slikovno točko  $\mathbf{p}_{s,i}$
- $s_i$  – neznana razdalja vzdolž  $\mathbf{v}_i$  od gorišča do točke  $\mathbf{p}$

5. Pogoj: medsebojne relacije (položaji) kamer in njihovi notranji parametri morajo biti znani.



# IZVEDBA TRIANGULACIJE S PROJEKCIJSKIMI Matrikami

1. Za  $j$ -to kamero (pogled) imamo znano projekcijsko matriko  $\mathbf{M}_j$ .
2. Točko  $\mathbf{p}_i$  projiciramo na  $j$ -to sliko v točko  $\mathbf{p}_{s_j,i}$ .
3. Za projekcijo velja:

$$\mathbf{p}_{s_j,i} = \mathbf{M}_j \mathbf{p}_i = \begin{bmatrix} m_{j,00} & m_{j,01} & m_{j,02} & m_{j,03} \\ m_{j,10} & m_{j,11} & m_{j,12} & m_{j,13} \\ m_{j,20} & m_{j,21} & m_{j,22} & m_{j,23} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

↓

$$x_{s_j,i} = \frac{m_{j,00}x_i + m_{j,01}y_i + m_{j,02}z_i + m_{j,03}}{m_{j,20}x_i + m_{j,21}y_i + m_{j,22}z_i + m_{j,23}}$$

$$y_{s_j,i} = \frac{m_{j,10}x_i + m_{j,11}y_i + m_{j,12}z_i + m_{j,13}}{m_{j,20}x_i + m_{j,21}y_i + m_{j,22}z_i + m_{j,23}}$$

4. Pridemo do sistema enačb - po dve za vsak pogled  $j$ . Z rešitvijo glede na najmanjšo kvadratno napako (SVD) lahko rekonstruiramo točke  $\mathbf{p}_i$  v 3D.

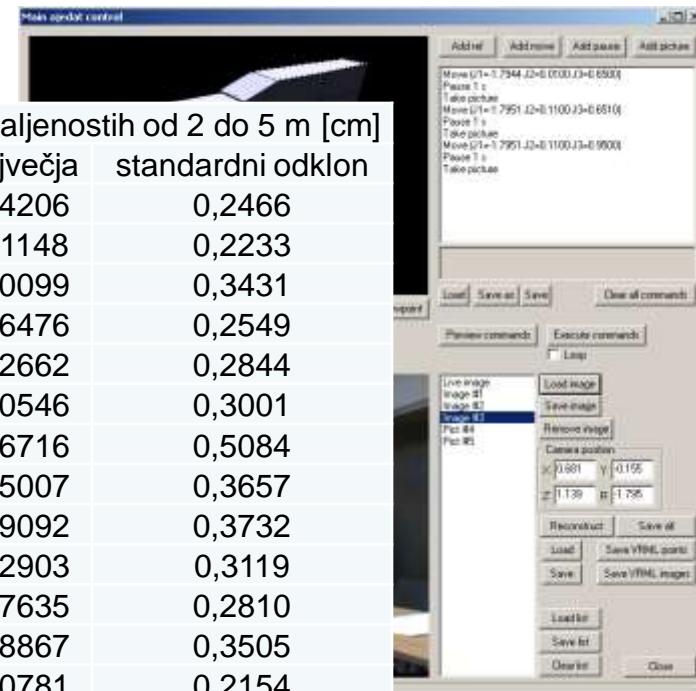
# Primer ocene strukture s triangulacijo



Vzeto iz: G. Pačnik,  
Magistrska naloga,  
FERI, 2007.

Napaka pri izmerjenih oddaljenostih od 2 do 5 m [cm]

Značilke	povprečna	najmanjša	največja	standardni odklon
1	0,7691	0,3326	1,4206	0,2466
2	0,5876	0,3232	1,1148	0,2233
3	1,2046	0,5828	2,0099	0,3431
4	0,8559	0,4128	1,6476	0,2549
5	0,7020	0,3180	1,2662	0,2844
6	1,0810	0,4252	2,0546	0,3001
7	1,4416	0,4362	2,6716	0,5084
8	1,5775	0,4889	2,5007	0,3657
9	1,3243	0,4421	2,9092	0,3732
10	1,1660	0,8002	2,2903	0,3119
11	1,3097	1,0090	1,7635	0,2810
12	0,7336	0,3392	1,8867	0,3505
13	0,7372	0,3750	1,0781	0,2154
14	0,6472	0,3130	1,3126	0,2221
15	0,4796	0,3542	0,7166	0,1557
16	0,4556	0,3232	1,0369	0,1523
17	0,3898	0,2945	0,5819	0,0814
18	0,3453	0,2329	0,5389	0,1077
Povprečna	0,8782			
Standardni odklon	0,3838			



# SOČASNO DOLOČANJE STRUKTURE IN GIBANJA

1. Imamo 2 ali več pogledov na isto sceno.
2. Na vseh pogledih poiščemo zanesljive korespondenčne točke – lahko za poljubne piksle, vendar je bolje, če uporabimo nadzorne točke (značilnice)
3. Naj bodo  $(x_{sj,i}, y_{sj,i})$  koordinate  $i$ -te točke na  $j$ -ti sliki; za vsako tako točko poznamo korespondenčne točke na ostalih slikah (ni nujno, da na vseh, bolje pa, da jih je čim več)
4. Privzemimo, da vse slike posnamemo z isto kamero oz. da pri vseh posnetkih velja enaka kalibracijska matrika  $\mathbf{K}$ .
5. Položaj kamere za  $j$ -ti pogled določata matrika zasuka  $\mathbf{R}_j$  in vektor premika  $\mathbf{t}_j$ .
6. Istočasno želimo oceniti položaj točk  $\mathbf{p}_i$  v prostoru (**struktura**) in parametre kamere (**gibanje**). Izhajamo iz naslednjih funkcijskih zvez:

$$\begin{aligned}\hat{x}_{sj,i} &= g_x(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{p}_i) \\ \hat{y}_{sj,i} &= g_y(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{p}_i)\end{aligned}$$

Funkcija  $g_x$  pomeni projekcijo točke  $\mathbf{p}_i$  v smeri x, funkcija  $g_y$  pa v smeri y

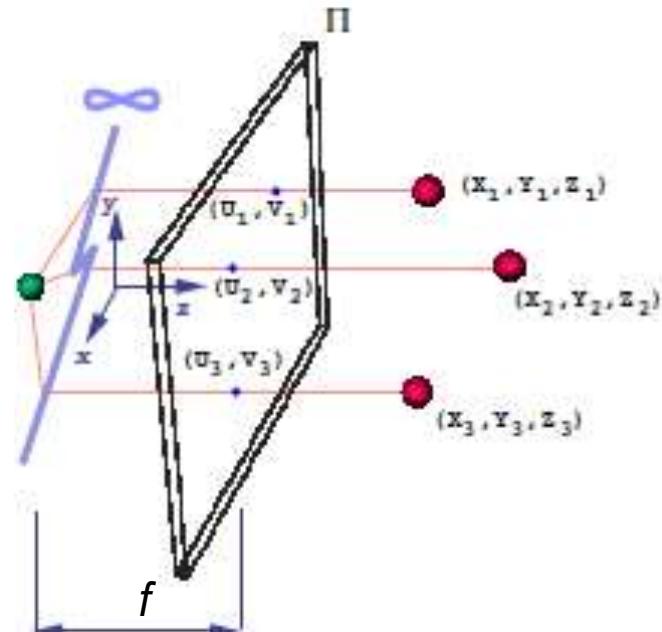
# STRUKTURA IZ GIBANJA – PROBLEM FAKTORIZACIJE

1. Izhajamo iz togih struktur (scene).
2. 3D-točke projiciramo na  $j$ -to sliko s projekcijsko matriko  $\mathbf{M}_j$ :

$$\mathbf{p}_{sj,i} = \mathbf{M}_j \mathbf{p}_i = \mathbf{K}[\mathbf{R}_j \ t_j] \mathbf{p}_i; \mathbf{p}_i = (x_i, y_i, z_i, 1)^T$$

Faktorizacija: **gibanje** in **struktura**

3. S faktorizacijo skušamo ločiti strukturo od gibanja.
4. V splošnem potrebujemo nelinearne in iterativne pristope, linearni izračun je možen le ob določenih omejitvah.
5. Znana je metoda Tomasi-Kanade, ki linearizira problem faktorizacije tako, da vzame namesto splošne projekcijske kamere ortografsko (pravokotno) projekcijo – omejitev na zelo veliko goriščno razdaljo oz. oddaljene scene.



# STRUKTURA IZ GIBANJA – FAKTORIZACIJA PO TOMASI-KANADEJU

- Pri ortografski kameri ima kalibracijska matrika naslednjo obliko:

$$\mathbf{K}_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

- Zato lahko posnetek  $i$ -te točke z  $j$ -to kamero zapišemo kot:

$$\begin{bmatrix} x_{s_j,i} \\ y_{s_j,i} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} [\mathbf{R}_j \ t_j] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} r_{j,00} & r_{j,01} & r_{j,02} \\ r_{j,10} & r_{j,11} & r_{j,12} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \begin{bmatrix} t_{j,x} \\ t_{j,y} \end{bmatrix}$$

- Iz matrik zasuka sta ostali prisotni samo prvi dve vrstici – kljub temu jih še naprej označujmo z  $\mathbf{R}_j$

# STRUKTURA IZ GIBANJA – FAKTORIZACIJA PO TOMASI-KANADEJU (2)

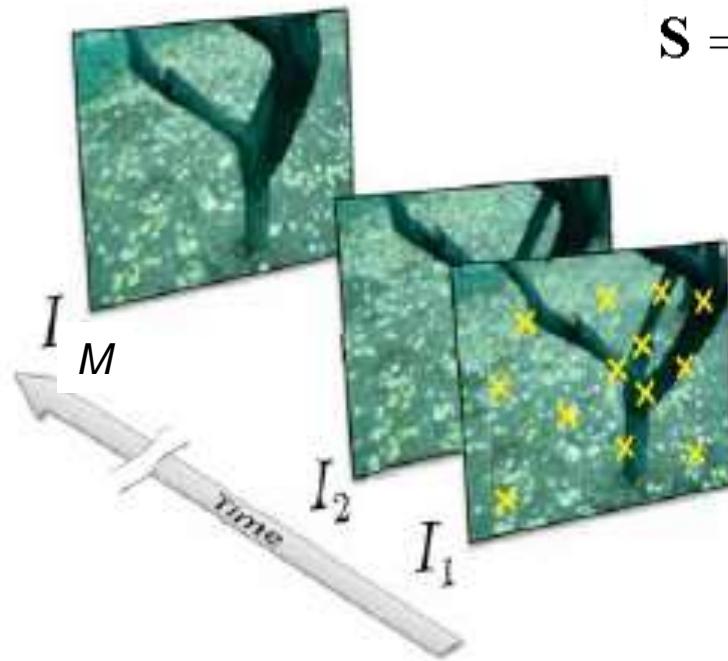
- Če imamo  $M$  slik iste scene in na vsaki  $N$  korespondenčnih točk, velja:

$$\begin{bmatrix} \mathbf{p}_{s_1,1} & \mathbf{p}_{s_1,2} & \dots & \mathbf{p}_{s_1,N} \\ \mathbf{p}_{s_2,1} & \mathbf{p}_{s_2,2} & & \mathbf{p}_{s_2,N} \\ \vdots & & & \\ \mathbf{p}_{s_M,1} & \mathbf{p}_{s_M,2} & & \mathbf{p}_{s_M,N} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \mathbf{R}_2 & \mathbf{t}_2 \\ \vdots & \\ \mathbf{R}_M & \mathbf{t}_M \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_N \end{bmatrix} \Rightarrow \mathbf{W} = [\mathbf{R} \ \mathbf{t}] \mathbf{S}$$

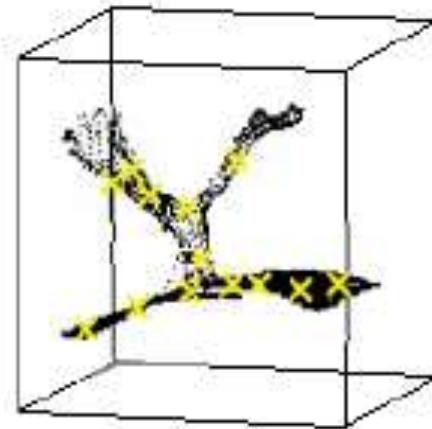
- Sedaj uporabimo trik: za vse točke iste slike, tj.  $\{\mathbf{p}_{sj,i}; i = 1, \dots, N\}$ , izračunamo povprečje koordinat in ga odštejemo od koordinat pri vsaki točki.
- S tem slike poravnamo in se znebimo premika med njimi ( $\mathbf{t}$ ).
- Hkrati pa postavimo izhodišče svetovnega koordinatnega sistema v aritmetično sredino (povprečje) vseh 3D-točk  $\mathbf{p}_i; i = 1, \dots, N$ .
- Na ta način dobimo:

$$\mathbf{W} = \mathbf{R} \cdot \mathbf{S}$$

# STRUKTURA IZ GIBANJA – FAKTORIZACIJA PO TOMASI-KANADEJU (3)



$$\mathbf{S} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \\ y_1 & y_2 & \cdots & y_N \\ z_1 & z_2 & \cdots & z_N \end{bmatrix}$$



$$\mathbf{W} = \begin{bmatrix} \mathbf{p}_{s_1,1} & \mathbf{p}_{s_1,2} & \dots & \mathbf{p}_{s_1,N} \\ \mathbf{p}_{s_2,1} & \mathbf{p}_{s_2,2} & & \mathbf{p}_{s_2,N} \\ \vdots & & & \\ \mathbf{p}_{s_M,1} & \mathbf{p}_{s_M,2} & & \mathbf{p}_{s_M,N} \end{bmatrix}$$

# STRUKTURA IZ GIBANJA – FAKTORIZACIJA PO TOMASI-KANADEJU (4)

1. Razcep na gibanje in strukturo naredimo sedaj s SVD:

$$\text{SVD}(\mathbf{W}) \Rightarrow \mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

2. Ker imata tako rotacijska matrika  $\mathbf{R}$  kot matrika 3D-točk  $\mathbf{S}$  po 3 prostostne stopnje, je v idealnih, brezšumnih razmerah matrika  $\mathbf{W}$  ranga 3.
3. Zaradi šuma in nenatančnosti se rang poveča, zato ga umetno reduciramo na 3, s tem ko upoštevamo samo 3 največje singularne vrednosti iz  $\mathbf{D}$ :

$$\mathbf{W}' = [\mathbf{U}_1 \ \mathbf{U}_2 \ \mathbf{U}_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \mathbf{V}_3^T \end{bmatrix} = \mathbf{U}' \mathbf{D}' \mathbf{V}'^T$$

4. Singularni vektorji  $\mathbf{U}_i$  in  $\mathbf{V}_i$  pripadajo singularnim vrednostim  $\lambda_i$ .
5. Prvi približek gibanju in strukturi ocenimo kot:

$$\hat{\mathbf{R}} = \mathbf{U}' \mathbf{D}' \quad \text{in} \quad \hat{\mathbf{S}} = \mathbf{V}'$$

6. Vendar pa rezultat ni enoumen, saj lahko zapišemo:

$$\hat{\mathbf{R}}\hat{\mathbf{S}} = (\hat{\mathbf{R}}\mathbf{Q})(\mathbf{Q}^{-1}\hat{\mathbf{S}})$$

# STRUKTURA IZ GIBANJA – FAKTORIZACIJA PO TOMASI-KANADEJU (5)

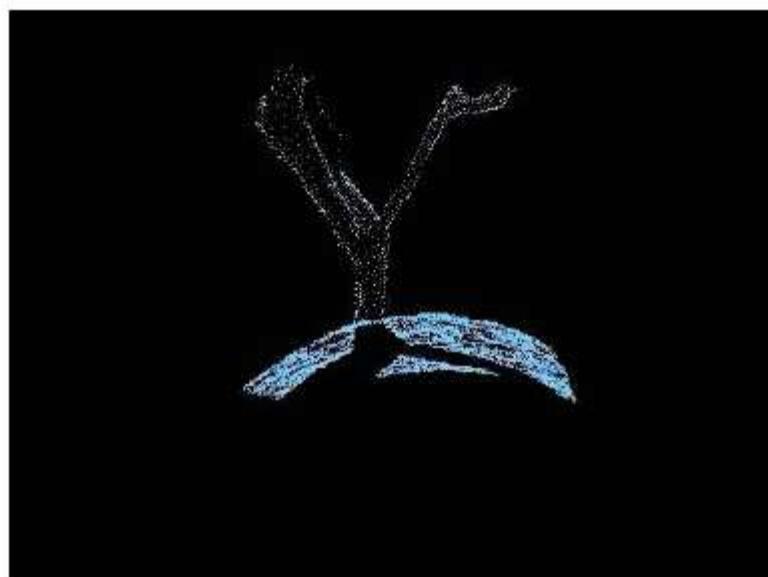
1. Za pravilno oceno gibanja ( $\mathbf{R}$ ) in strukture ( $\mathbf{S}$ ) je treba izračunati ustrezeno matriko  $\mathbf{Q}$ , ki je dimenzij  $3 \times 3$ .
2. V ocenjeni matriki  $\hat{\mathbf{R}}$  morajo biti sosednji pari vrstic ortogonalni, saj tvorijo matrike zasuka za posamezne poglede – to dosežemo z izbiro matrike  $\mathbf{Q}$ .
3. Iz tega sledi, da mora veljati:

$$(\hat{\mathbf{R}}_{2i-1:2i} \mathbf{Q}) (\hat{\mathbf{R}}_{2i-1:2i} \mathbf{Q})^T = \hat{\mathbf{R}}_{2i-1:2i} \mathbf{Q} \mathbf{Q}^T \hat{\mathbf{R}}_{2i-1:2i}^T = \mathbf{I}_{2 \times 2}$$

pri čemer matrika  $\mathbf{I}$  pomeni identiteto.

4. Matrika  $\mathbf{Q} \mathbf{Q}^T$  je simetrična in ima 6 različnih vrednosti.
5. Za izračun teh vrednosti potrebujemo najmanj 6 enačb – vsak pogled (slika) prispeva 2, po eno za koordinati  $x$  in  $y$  korespondenčnih točk.
6. Za končno ocenitev parametrov gibanja in strukture po metodi Tomasi-Kanade torej potrebujemo najmanj 3 slike!

# Primer ocene strukture po metodi Tomasi-Kanade



# STRUKTURA IN GIBANJE, DOLOČENA IZ LINIJSKIH SEGMENTOV

(povzeto po C.J. Taylor, D. J. Kriegman: Structure and Motion from Line Segments in Multiple Images)

## 1. Cilj:

V zaporedju slik so segmentirani odseki premic (robovi) in z njihovo pomočjo je treba najti relativne položaje kamer (gibanje) in 3D-položaje linijskih odsekov (struktura).

## 2. Privzeta je toga scena, robovi kot značilnice pa prinašajo še dodatne prednosti:

- ravni robovi so pogosti v okoljih, ki jih ustvarja človek
- postopki za določanje robov na slikah so učinkoviti
- robove je dokaj preprosto najti in jim slediti
- robovi (linijski odseki) dajejo boljše rezultate kot posamezne točke, saj vsebujejo več informacij

## 3. Določanje strukture scene in gibanja (položaja) kamer temelji na optimizaciji kriterijske funkcije.

# POSTOPEK OPTIMIZACIJE IN KRITERIJSKA FUNKCIJA

1. Imamo  $M$  slik in  $N$  črt (robov):

- $\mathbf{p}_i$  – 3D-položaj  $i$ -te črte (linije)
- $\mathbf{q}_j$  – položaj in usmeritev  $j$ -te kamere
- $\mathbf{u}_{ij}$  –  $i$ -ti rob na  $j$ -ti sliki
- $F$  – funkcija za vzvratno projekcijo (*reprojection*) črte  $\mathbf{p}_i$  na slikovno ravnino kamere  $\mathbf{q}_j$

2. Razpozname položaje linijskih segmentov in kamer iz 3D vzvratno projiciramo na slikovne ravnine kamer

3. Razliko med reprojiciranimi črtami in originalnimi črtami moramo minimizirati:

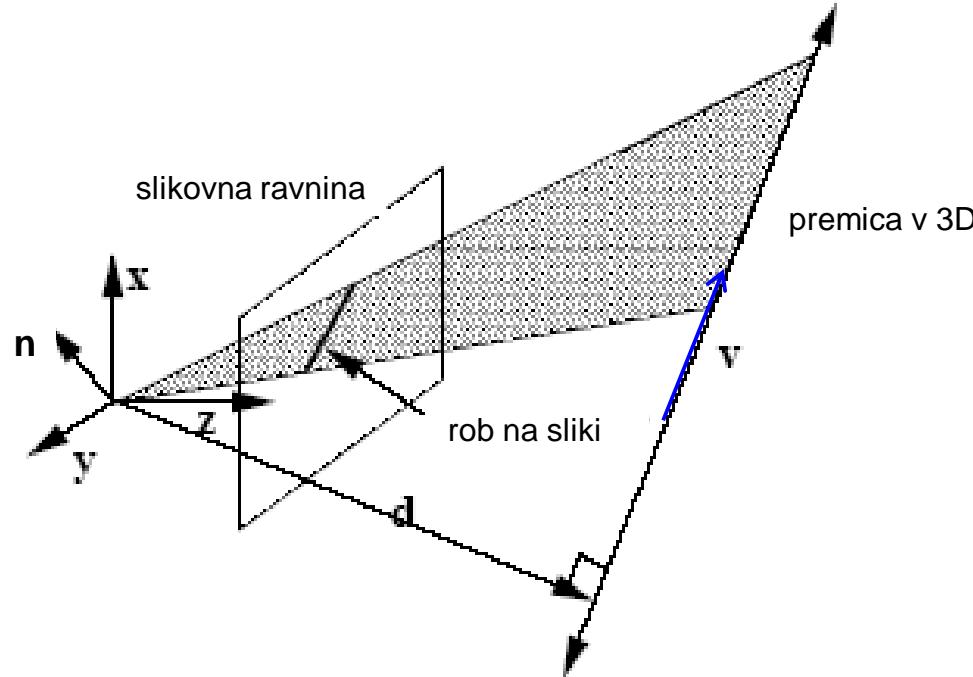
$$o = \sum_{j=1}^M \sum_{i=1}^N \text{razlika} \left[ F(\mathbf{p}_i, \mathbf{q}_j), \mathbf{u}_{ij} \right]$$

# PREDSTAVITEV PREMICE V 3D – SVETOVNE KOORDINATE

1. Opišimo premico v 3D z  $(\mathbf{v}, \mathbf{d})$ , pri čemer sta:

$\mathbf{v}$ : enotski vektor, ki kaže v smeri premice

$\mathbf{d}$ : vektor iz koordinatnega središča do najbližje točke na premici



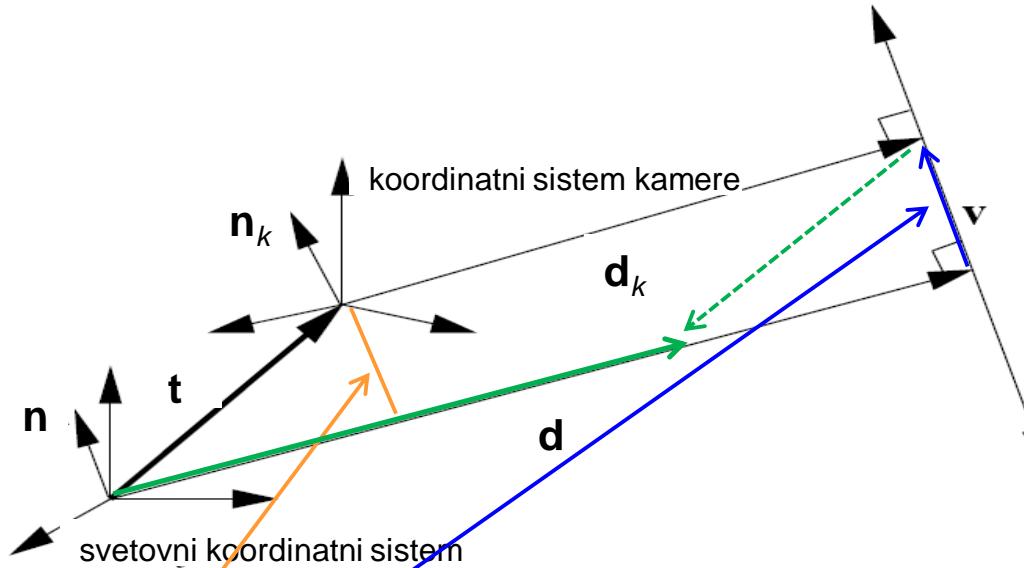
2.  $\mathbf{n}$ : normala na ravnino, ki jo določata projekcijsko središče in premica

3. Enačba za sliko premice (rob), ki leži na slikovni ravnini:

$$n_x x + n_y y + n_z = 0 \quad (\text{pesek s slikovno ravnino } z = 1)$$

# PREDSTAVITEV PREMICE V 3D – KOORDINATE KAMERE

- Postavimo kamero v svetovni (globalni) koordinatni sistem:  
 $t$ : premik kamerinega projekcijskega središča glede na svetovne koordinate  
 $R$ : rotacijska matrika kamere glede na svetovni koordinatni sistem



- Stik vektorja  $d_k$  s premico  $v$  izrazimo v svetovnih koordinatah:  
$$d_k = R[d - t + (t \cdot v)v]$$
- Enotski vektor  $v$  se preslika v vektor  $v_k$  v koordinatnem sistemu kamere:

$$v_k = R \cdot v$$

# PREDSTAVITEV PREMICE V 3D – KOORDINATE KAMERE (2)

1. Normala na ravnino, ki jo določata projekcijsko središče kamere in premica  $\mathbf{v}$ , je določena z vektorskim produktom med  $\mathbf{v}_k$  in  $\mathbf{d}_k$ :

$$\mathbf{n}_k = \mathbf{v}_k \times \mathbf{d}_k$$

2. To normalo izrazimo z vektorji v svetovnem koordinatnem sistemu:

$$\mathbf{n}_k = \mathbf{R}[\mathbf{v} \times (\mathbf{d} - \mathbf{t})],$$

saj je vektorski produkt enotskega vektorja  $\mathbf{v}$  samega s seboj 0.

3. Vektor  $\mathbf{n}_k$  še normiramo, da dobi dolžino 1 (normalo):

$$\mathbf{n}_k = \mathbf{n}_k / \|\mathbf{n}_k\|$$

4. Iz enačbe v točki 2 izhajata dve omejitvi, ker je normala pravokotna na ravnino, v kateri sta z  $\mathbf{R}$  zasukana vektorja  $\mathbf{v}$  in  $\mathbf{d}-\mathbf{t}$ :

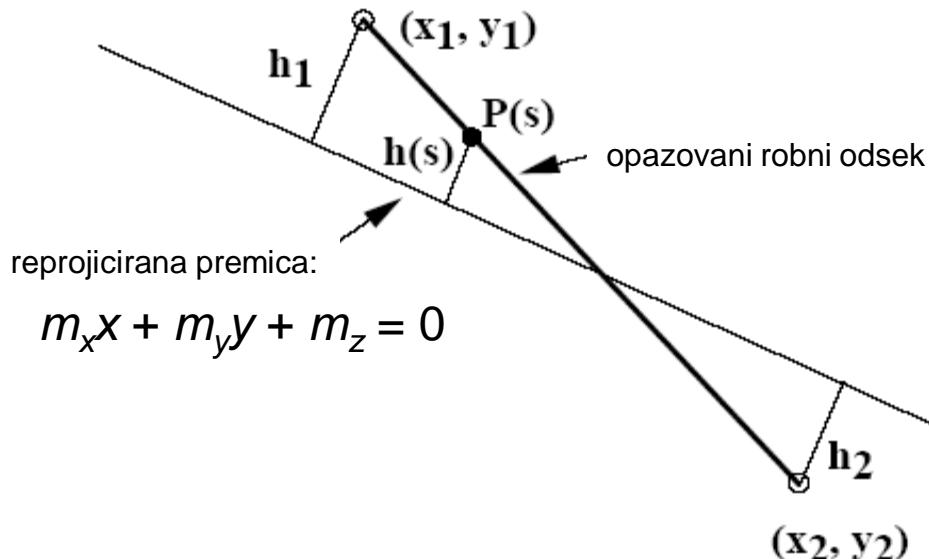
$$\mathbf{n}_k^T \mathbf{R}\mathbf{v} = \mathbf{0} \quad (\text{skalarni produkt pravokotnih vektorjev})$$

$$\mathbf{n}_k^T \mathbf{R}(\mathbf{d}-\mathbf{t}) = \mathbf{0} \quad (\text{skalarni produkt pravokotnih vektorjev})$$

# NEZNANI PARAMETRI IN KRITERIJSKA FUNKCIJA

1. Ko iščemo položaj premice v 3D in položaj ter usmeritev kamere, imamo naslednje neznane parametre:

- 3 za lokacijo  $i$ -te kamere ( $\mathbf{t}_i$ )
- 3 za usmeritev  $i$ -te kamere ( $\mathbf{R}_i$ )
- 4 za opis premice v 3D (položaj in usmeritev): 2 za vektor  $\mathbf{d}$  in 2 za vektor  $\mathbf{v}$  (upoštevano dejstvo, da sta  $\mathbf{v}$  in  $\mathbf{d}$  pravokotna)



2. Izračunamo odstopanje med točkami na robu s krajiščema  $(x_1, y_1)$  in  $(x_2, y_2)$  in točkami na premici, ki je reprojicirana na sliko

# SOČASNO DOLOČANJE GIBANJA IN STRUKTURE

1. Najprej določimo linijske odseke oziroma  $N$  ravnih odsekov izrazitih robov na  $M$  slikah. Uporabimo eno od metod za iskanje robov.
2. Nato najdemo korespondenčne robe – črte, ki na  $M$  slikah predstavljajo isto značilnost 3D-scene.
3. Za  $i$ -to črto na  $j$ -ti sliki, katere krajišči sta  $(x_{ij,1}, y_{ij,1})$  in  $(x_{ij,2}, y_{ij,2})$ , ocenimo noramlo  $\mathbf{n}_{ij}$ :  
$$\mathbf{n}_{ij} = (x_{ij,1}, y_{ij,1}, 1)^T \times (x_{ij,2}, y_{ij,2}, 1)^T \quad (\text{vektorski produkt točk na premici})$$
4. Za  $j$ -to sliko oz. kamero predvidimo omejitve zasuka in v teh mejah naključno tvorimo rotacijsko matriko  $\mathbf{R}_j$ .
5. V nadalnjem nelinearnem optimizacijskem postopku ločeno ocenimo matrike zasukov in vektorje premikov za kamere.

# SOČASNO OPTIMIRAJE GIBANJA IN STRUKTURE

1. Iz pogoja, ki ga določa enačba  $\mathbf{n}_{ij}^T \mathbf{R}_j \mathbf{v}_i = 0$ , ocenimo smerne vektorje premice  $\mathbf{v}_i$ . Za vsako oceno potrebujemo vsaj 2 slike s korespondenčnim parom robov.
2. Nato izračunamo optimizacijski kriterij

$$c_1 = \sum_{j=1}^M \sum_{i=1}^N (\mathbf{n}_{ij}^T \mathbf{R}_j \mathbf{v}_i)^2$$

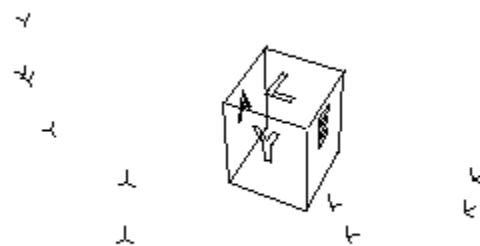
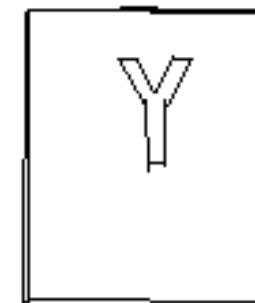
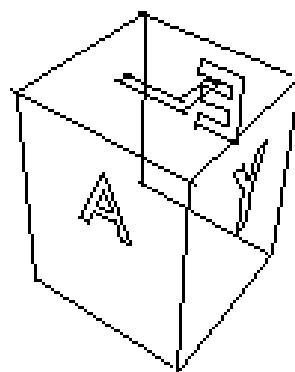
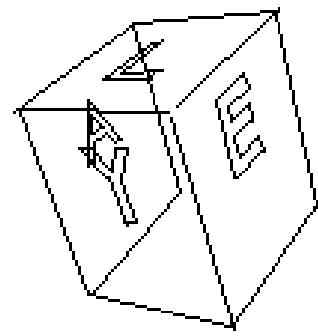
3. Kriterij  $c_1$  minimiziramo z gradientnim postopkom, dokler odstopanje ne pade pod izbrani prag, pri čemer iterativno posodabljam parametre zasuka ( $\mathbf{R}_j$ ) in usmerjenosti ( $\mathbf{v}_i$ ).
4. Sledi določanje premika  $\mathbf{t}_j$  za vsako kamero in 3D-položaja za vsako premico  $\mathbf{d}_i$ . Izhajamo iz drugega omejitvenega pogoja, ki ga določa enačba  $\mathbf{n}_{ij}^T \mathbf{R}_j (\mathbf{d}_i - \mathbf{t}_j) = 0$ , tako da minimiziramo odstopanje:

$$c_2 = \sum_{j=1}^M \sum_{i=1}^N [\mathbf{n}_{ij}^T \mathbf{R}_j (\mathbf{d}_i - \mathbf{t}_j)]^2$$

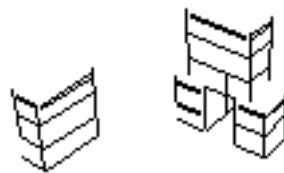
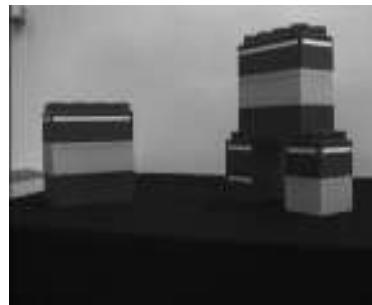
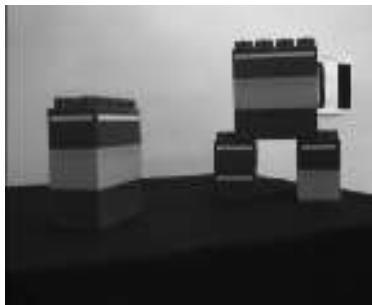
Potrebujemo vsaj 3 slike s po 6 korespondenčnimi robovi.

5. Končno izboljšavo dosežemo s popolno kriterijsko funkcijo, ko gradientno izboljšujemo vse parametre hkrati.

# Primer določanja strukture iz gibanja



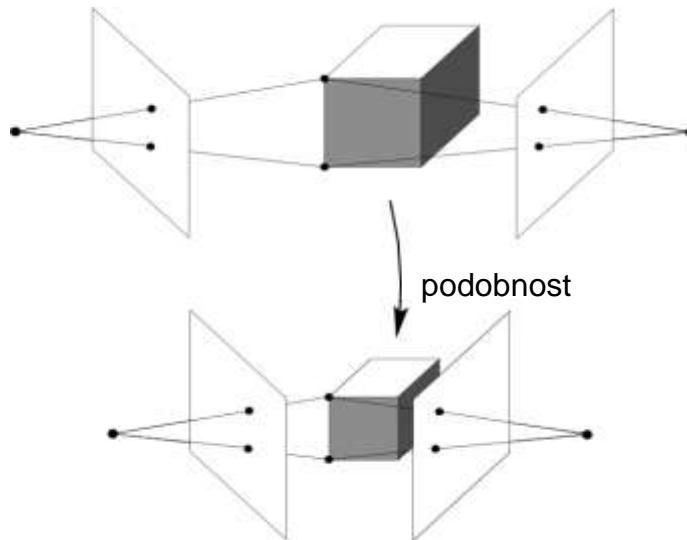
# Primer določanja strukture iz gibanja (2)



# 3D-rekonstrukcija objektov

# REKONSTRUKCIJSKA VEČLIČNOST: PODOBNOST

Imamo točke v prostoru,  $\mathbf{p}_i$ , in njihove projekcije na slikovno ravnino  $\mathbf{p}_{s,i}$ . Projekcijska matrika je  $\mathbf{M}$ .



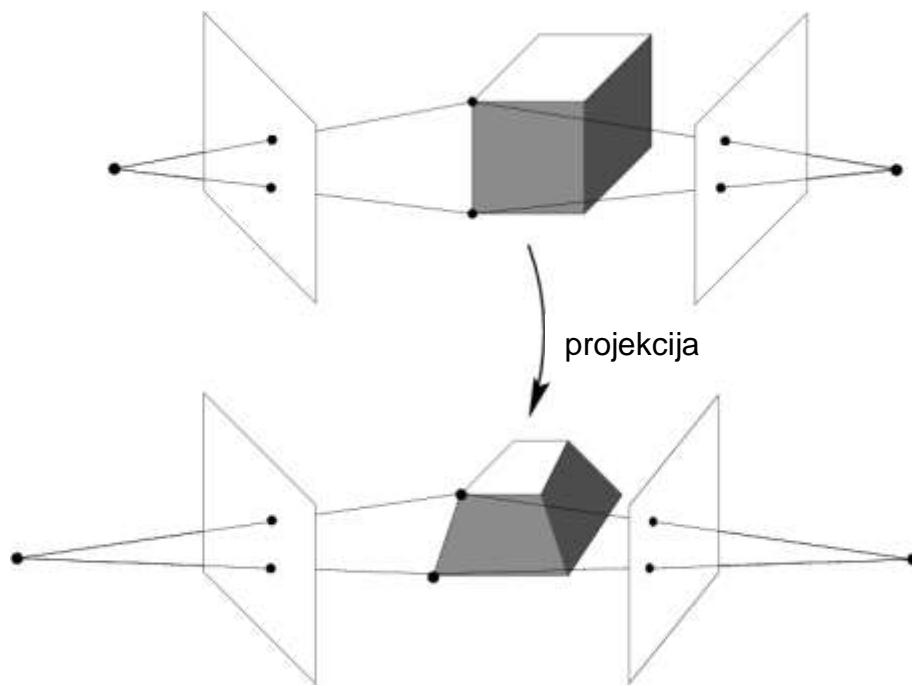
$$\mathbf{p}_{s,i} = \mathbf{M}\mathbf{p}_i = (\mathbf{M}\mathbf{H}_S^{-1})(\mathbf{H}_S\mathbf{p}_i)$$

$\mathbf{H}_s$  – homografija, ki  
ohrani podobnost

$$\mathbf{H}_s = \begin{bmatrix} \mathbf{R}' & \mathbf{t}' \\ \mathbf{0} & \lambda \end{bmatrix} \Rightarrow \mathbf{M}\mathbf{H}_S^{-1} = \mathbf{K}[\mathbf{R}] \mathbf{t} \begin{bmatrix} \mathbf{R}'^T & -\mathbf{R}'^T \mathbf{t}' \\ 0 & \lambda \end{bmatrix} = \mathbf{K}[\mathbf{R}\mathbf{R}'^T] - \mathbf{R}\mathbf{R}'^T \mathbf{t}' + \lambda \mathbf{t}$$

To dokazuje, da je s kalibrirano kamero možna rekonstrukcija do podobnosti natančno.

# REKONSTRUKCIJSKA VEČLIČNOST: PROJEKCIJSKA



$$\mathbf{p}_{s,i} = \mathbf{M}\mathbf{p}_i = (\mathbf{M}\mathbf{H}_P^{-1})(\mathbf{H}_P\mathbf{p}_i)$$

$\mathbf{H}_P$  – homografija, ki ne spremeni projekcije

# 3D REKONSTRUKCIJA: POIMENOVANJA

1. Govorimo o rekonstrukcijah, ki so enake originalu do neke transformacije natančno.
2. Projekcijska, afina in podobna rekonstrukcija:
  - enake so originalu do projekcije, affine transformacije ali podobnosti natančno.
3. Podobna rekonstrukcija se v literaturi večkrat imenuje tudi metrična ali evklidska rekonstrukcija. Razlika med njima je le ta, da metrična rekonstrukcija ohrani prave proporce (razmerja dolžin), medtem ko evklidska rekonstrukcija doda dolžinske enote in kotne mere (metre, stopinje).
4. Obstaja zveza med rekonstrukcijami:  
projekcijska → afina → metrična → evklidska
  - projekcijska rekonstrukcija ohranja tip (točka, črta), pripadnost (točka na črti) in razmerja razdalj
  - afina rekonstrukcija dodaja paralelizem – ohranja paralelnost (ravnina v neskončnosti)
  - metrična rekonstrukcija ohranja kote (stožnica v neskončnosti)
  - ko se pomikamo od projekcijske proti evklidski rekonstrukciji, se manjša število možnih transformacij in veča število nespremenljivih parametrov – za evklidsko rekonstrukcijo sta možna le zasuk in premik

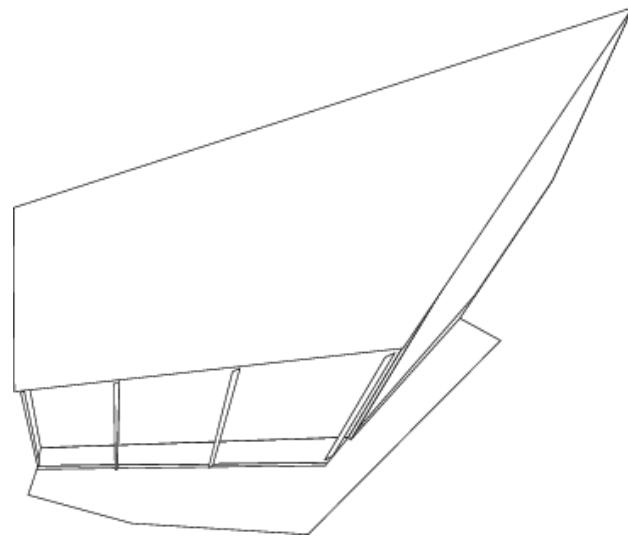
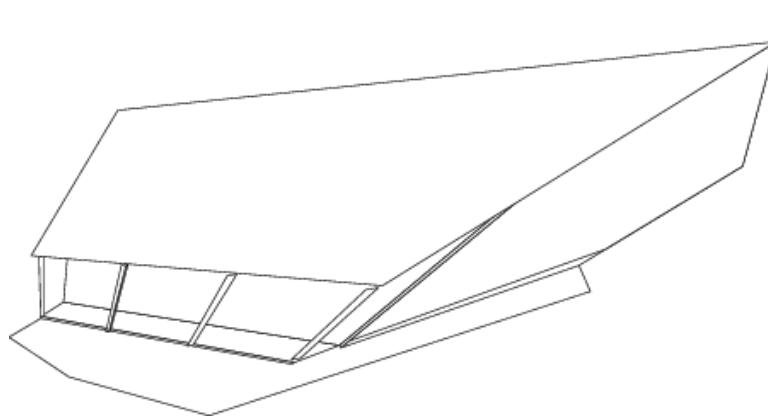
# TEOREM PROJEKCIJSKE REKONSTRUKCIJE

Če množica korespondenčnih točk na dveh slikah (pogledih) določa eno samo temeljno matriko, potem lahko rekonstruiramo parametre scene in kamere samo na osnovi teh korespondenčnih točk, pri tem pa sta katerikoli dve takšni rekonstrukciji projekcijsko ekvivalentni.

**Ključna ugotovitev:**

3D rekonstrukcija je možna iz para nekalibriranih slik

# Primer projekcijske rekonstrukcije



# HIERARHIJA TRANSFORMACIJ

Linearne projekcije

Afne transformacije – zadnja vrstica matrike  $(0,0,1)$

Evklidske transformacije – leva zgornja matrika  $2 \times 2$  mora biti ortogonalna

Usmerjene evklidske transformacije (zgornja leva matrika  $2 \times 2$  ima determinanto 1)

# IZOMETRIJE – OHRANJANJE MER (2D)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \varepsilon \cos \theta & -\sin \theta & t_x \\ \varepsilon \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \varepsilon = \pm 1$$

ohranja usmerjenost:  $\varepsilon = 1$

obrača usmerjenost:  $\varepsilon = -1$

$$\mathbf{p}' = \mathbf{H}_E \mathbf{p} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p} \quad \mathbf{R}^T \mathbf{R} = \mathbf{I}$$

3 prostostne stopnje: 1 za zasuk (rotation), 2 za premik (translation)

Se ne spreminja (invariantno): dolžine, koti, ploščina

# PODOBNOŠTI (2D)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_x \\ s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\mathbf{p}' = \mathbf{H}_S \mathbf{p} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p} \quad \mathbf{R}^T \mathbf{R} = \mathbf{I}$$

4 prostostne stopnje: 1 za povečavo, 1 za zasuk, 2 za premik

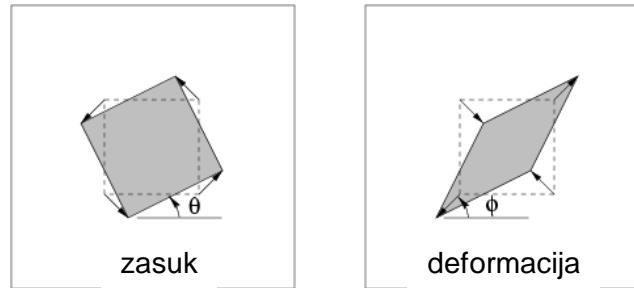
Ohranja oblike: *metrična struktura* = podobna struktura

Se ne spreminja (invariantno):  
razmerja dolžin, koti, razmerja ploščin, vzporednice

# AFINA TRANSFORMACIJA (2D)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\mathbf{p}' = \mathbf{H}_A \mathbf{p} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p}$$



$$\mathbf{A} = \mathbf{R}(\theta) \mathbf{R}(-\phi) \mathbf{D} \mathbf{R}(\phi) \quad \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

6 prostostnih stopenj: 2 za povečavo, 2 za zasuk, 2 za premik

Povečava ni enaka v vse smeri (ni izotropna)!

Se ne spreminja (invariantno):

vzporednice, razmerja vzporednih dolžin, razmerja ploščin

# PROJEKCIJSKA TRANSFORMACIJA (2D)

$$\mathbf{p}' = \mathbf{H}_P \mathbf{p} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \mathbf{p} \quad \mathbf{v} = (v_1, v_2)^T$$

8 prostostnih stopenj: 2 za povečavo, 2 za zasuk, 2 za premik, 2 za premico v neskončnosti)

Se ne spreminja (invariantno):  
medsebojno razmerje štirih točk na premici (razmerje razmerja)

# Primer vpliva afine in projekcijske transformacije (2D)

## Afina

$$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & v \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{pmatrix}$$

Linija v neskončnosti ostane v neskončnosti, točke na njej pa se premaknejo.

## Projekcijska

$$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ v_1 x_1 + v_2 x_2 \end{pmatrix}$$

Linija iz neskončnosti postane končna in omogoča, da določimo ponorne točke in horizont.

# DEKOMPOZICIJA PROJEKCIJSKE TRANSFORMACIJE (2D)

$$\mathbf{H} = \mathbf{H}_S \mathbf{H}_A^\sim \mathbf{H}_P^\sim = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^T & v \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \quad \mathbf{A} = s\mathbf{RK} + \mathbf{tv}^T$$

Dekompozicija je enoumna, če izberemo  $s>0$

$\mathbf{K}$  je zgornje trikotna,  $\det(\mathbf{K}) = 1$

$\mathbf{H}^\sim$  pomeni, da matrika prispeva k  $\mathbf{H}$

**Primer:**

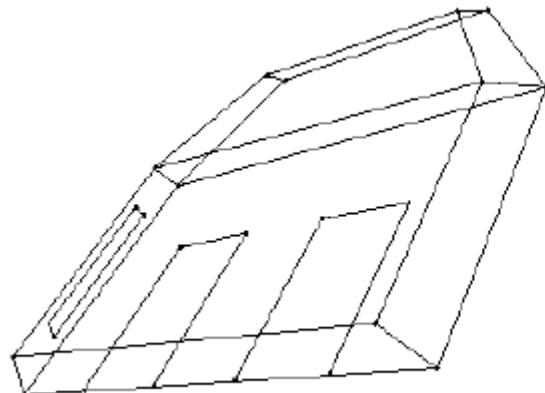
$$\mathbf{H} = \begin{bmatrix} 1,707 & 0,586 & 1,0 \\ 2,707 & 8,242 & 2,0 \\ 1,0 & 2,0 & 1,0 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 2\cos 45^\circ & -2\sin 45^\circ & 1,0 \\ 2\sin 45^\circ & 2\cos 45^\circ & 2,0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0,5 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

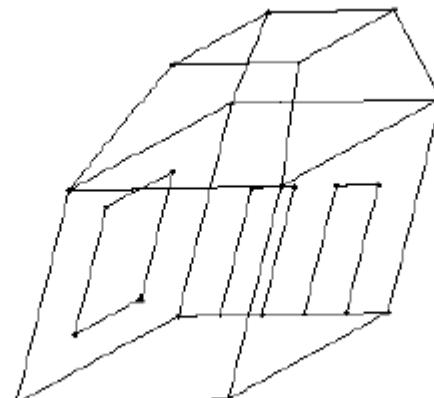
# SLOJEVIT PRISTOP K 3D REKONSTRUKCIJI

1. Kadar rekonstruiramo 3D sceno in položaj kamere iz dveh nekalibriranih slik, dobimo lahko samo projekcijsko rekonstrukcijo.
2. Prehod v metrični oz. evklidski prostor zahteva dodatne korake. Govorimo o slojeviti 3D rekonstrukciji (*stratified reconstruction*).
3. Slojevita rekonstrukcija se začne s projekcijsko, iz nje preidemo v afino, to pa potem nadgradimo z metrično rekonstrukcijo.

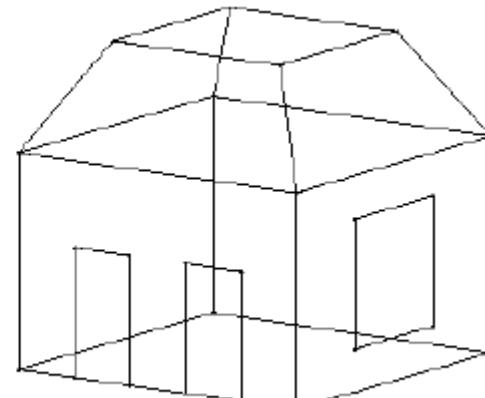
projekcijska



afina

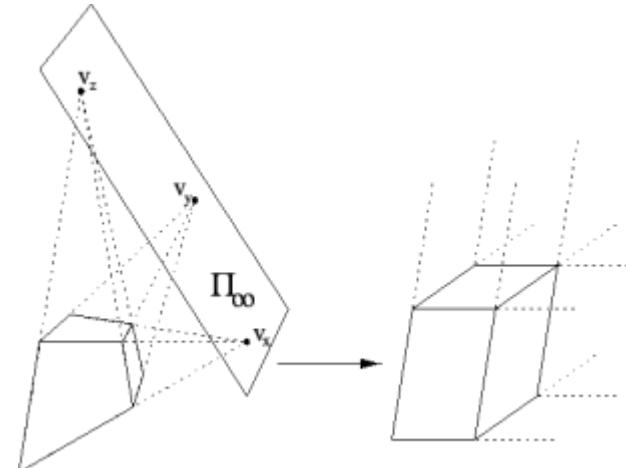


metrična



# PREHOD OD PROJEKCIJSKE K AFINI REKONSTRUKCIJI

1. Afina transformacija ohranja vzporednost:
  - dve premici, ki sta vzporedni, se sekata v točki na ravnini v neskončnosti
  - dve vzporedni ravnini se sekata v premici na ravnini v neskončnosti
2. Za afino transformacijo moramo torej poznati ravnino v neskončnosti  $\Pi_\infty$ .
3. Za ravnino v neskončnosti velja relacija:  
$$\Pi_\infty \mapsto (0, 0, 0, 1)^T$$
4. Ravnino v neskončnosti lahko določimo, če poznamo vsaj 3 točke na njej (te so neodvisne od translacije – recimo luna na nebu).
5. Točke v neskončnosti so **ponorne točke (vanishing points)**.
6. Za točke  $\mathbf{p} = [x, y, z, 0]^T$  na ravnini  $\Pi_\infty$  velja:  
$$[0, 0, 0, 1]\mathbf{p} = 0$$
7. Potrebujemo homografijo  $\mathbf{H}_P$ , s katero bomo preslikali projekcijsko rekonstrukcijo v afino.



# PREHOD OD PROJEKCIJSKE K AFINI REKONSTRUKCIJI (2)

1. Homografija za prehod od projekcijske k afini rekonstrukciji ima obliko:

$$\mathbf{H}_P^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^T & v_4 \end{bmatrix}$$

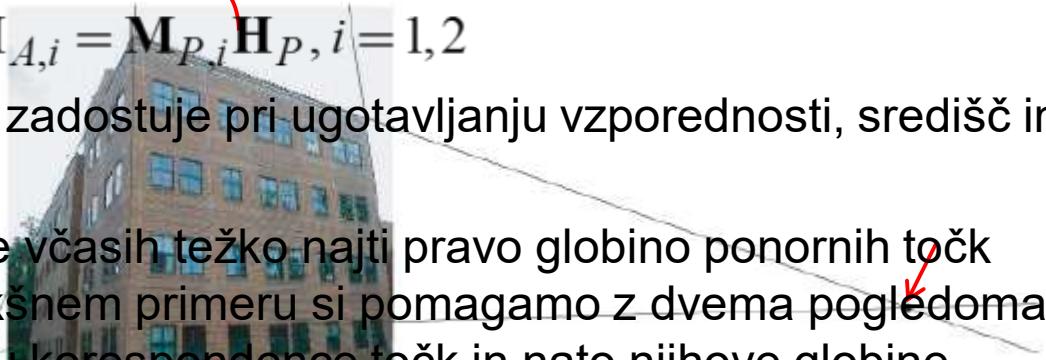
2. Zapis  $[\mathbf{v}^T, v_4]^T$  pomeni sliko ravnine v neskončnosti ( $\Pi_\infty$ ), projicirano na slikovno ravnino kamere.
3. Parametre slike za  $\Pi_\infty$ ,  $\mathbf{v} = (v_1, v_2, v_3)^T$  in  $v_4$ , določimo iz vsaj 3 ponornih točk na projekcijskih slikah (lahko samo eni sliki);  $\mathbf{p}_{sP,i}$ ,  $i = 1, 2, 3$ :

$$[v_1, v_2, v_3, v_4] \cdot \mathbf{p}_{sP,i} = 0$$

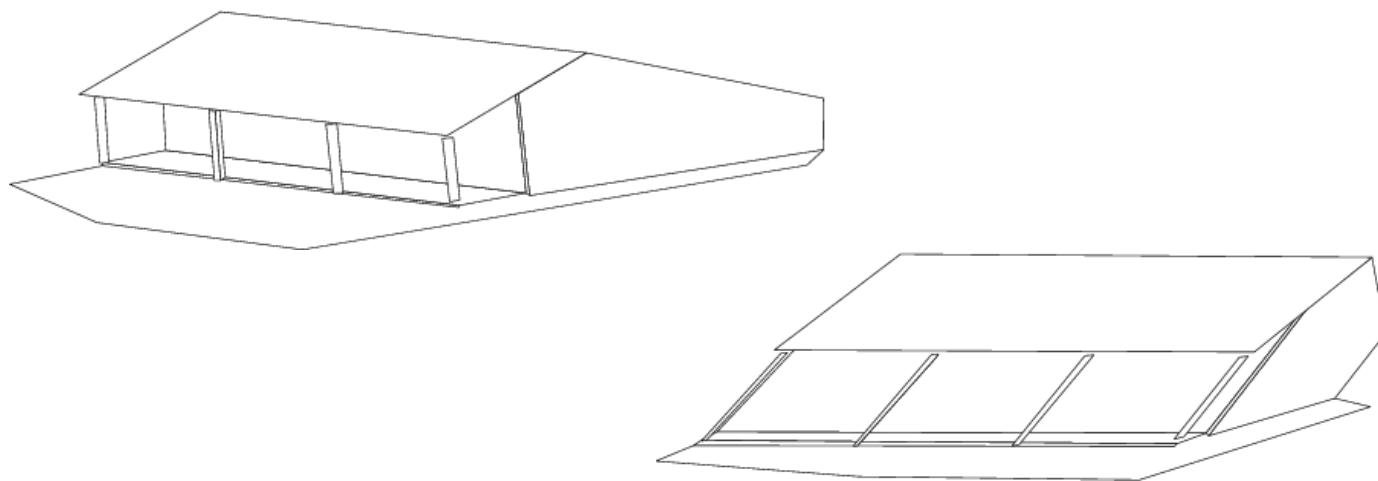
4. Za pretvorbo projekcijske v afino rekonstrukcijo velja:

$$\mathbf{p}_{sA} = \mathbf{H}_P^{-1} \mathbf{p}_{sP}; \quad \mathbf{M}_{A,i} = \mathbf{M}_{P,i} \mathbf{H}_P, i=1,2$$

5. Afina rekonstrukcija zadostuje pri ugotavljanju vzporednosti, središč in težišč
6. Zaradi šumnih slik je včasih težko najti pravo globino ponornih točk (koordinata  $z$ ). V takšnem primeru si pomagamo z dvema pogledoma, tako da pri določanju korespondence točk in nato njihove globine izkoristimo epipolarno omejitev.



# Primer pretvorbe v afino rekonstrukcijo



# PREHOD OD AFINE K METRIČNI REKONSTRUKCIJI

1. Za prehod od afine k metrični oz. evklidski rekonstrukciji se običajno uporablja pristop s t. i. **absolutno stožnico (absolute conic)**.
  2. Absolutna stožnica  $\Omega_\infty$  leži na ravnini v neskončnosti, vendar je na slikah ni mogoče videti kot posebno strukturo (je le navidezen element).
  3. Za točke na absolutni stožnici velja:
- $$x^2 + y^2 + z^2 = 0.$$
4. Za poljubno točko  $\mathbf{p} = (x, y, z)^T$  v neskončnosti in absolutno stožnico velja tudi:

$$\mathbf{p}^T \Omega_\infty \mathbf{p} = 0,$$

iz česar sledi, da absolutno stožnico zapišemo v idealnih razmerah kot identiteto:  $\Omega_\infty = \mathbf{I}$

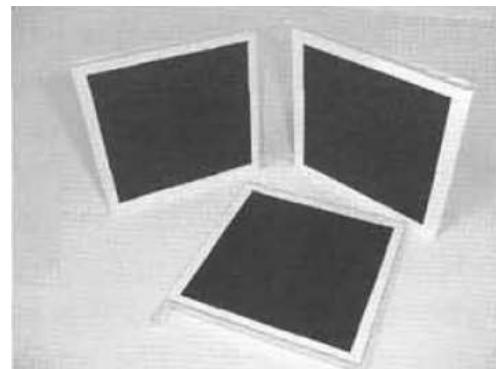
5. Za absolutno stožnico velja:
  - vse sfere sekajo ravnino v neskončnosti v obliki absolutne stožnice
  - vsi krogi sekajo absolutno stožnico v dveh točkah na ravnini v neskončnosti

# SLIKA ABSOLUTNE STOŽNICE IN KALIBRACIJSKA Matrika

1. Označimo projekcijo absolutne stožnice na slikovno ravnilo v kameri z  $\omega$ .
2. Dokazano je bilo, da je slika absolutne stožnice povezana s kalibracijsko matriko kamere  $\mathbf{K}$ :

$$\omega = (\mathbf{K} \cdot \mathbf{K}^T)^{-1}$$

3. Za določanje absolutne stožnice potrebujemo na sceni vsaj 3 kvadrate, ki ležijo v različnih ravninah – ni nujno, da so pravokotne.



4. Na slikah poiščemo te tri kvadrate in izračunamo homografijo, ki koordinate oglišč prestavi v točke  $(0, 0)$ ,  $(0,1)$ ,  $(1,0)$  in  $(1,1)$ .
5. S tem sicer premaknemo koordinatni sistem, a ker je to operacija podobnosti, nima vpliva na sliko absolutne stožnice.

# DOLOČANJE Slike ABSOLUTNE STOŽNICE

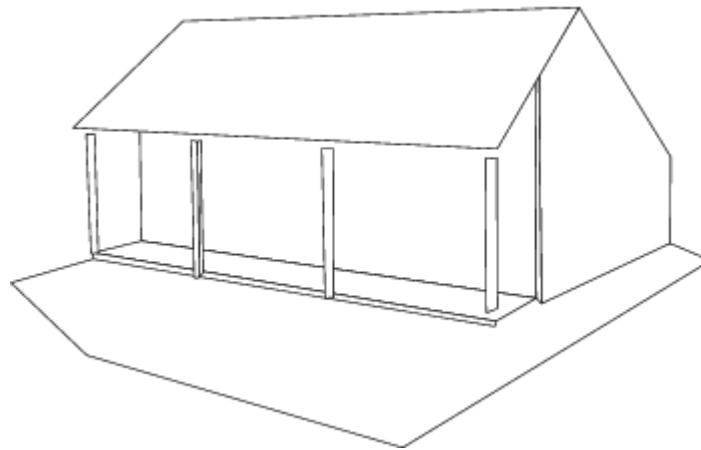
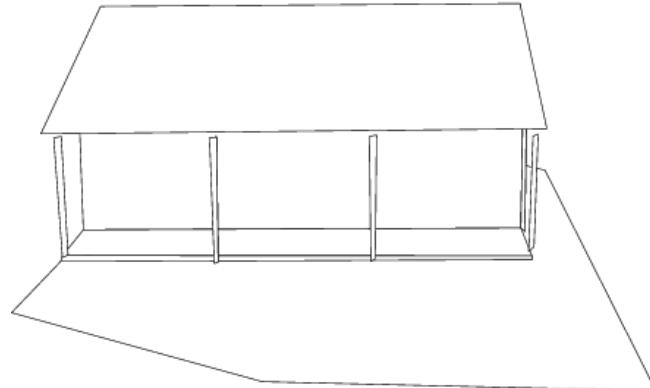
1. Homografija  $\mathbf{H}$  preslika koordinate oglišč kvadratov v točke  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$  in  $(1, 1)$ .
2. Homografijo zapišemo kot matriko s tremi stolpcji  $\mathbf{h}_i$ :  
$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$$
3. Za sliko absolutne stožnice  $\omega$  velja:  
$$\mathbf{h}_1^T \omega \mathbf{h}_2 = 0 \text{ in } \mathbf{h}_1^T \omega \mathbf{h}_1 = \mathbf{h}_2^T \omega \mathbf{h}_2$$
4. Iz teh dveh enačb izračunamo sliko absolutne stožnice  $\omega$  in iz nje s faktorizacijo določimo kalibracijsko matriko kamere  $\mathbf{K}$ .
5. Kadar tak postopek ponovimo za množico slik, ki jih dobimo ob premikih kamere, govorimo o samokalibraciji kamere (*self-calibration*).
6. Ko imamo na razpolago rezultate afine rekonstrukcije, jih lahko pretvorimo v metrične, če poznamo homografijo

$$\mathbf{H}_S = \begin{bmatrix} \mathbf{K}^{-1} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

7. Končni rezultat:

$$\mathbf{p}_{sS} = \mathbf{H}_S^{-1} \mathbf{p}_{sA}; \quad \mathbf{M}_{S,i} = \mathbf{M}_{A,i} \mathbf{H}_S, i=1,2$$

# Primer pretvorbe v metrično rekonstrukcijo

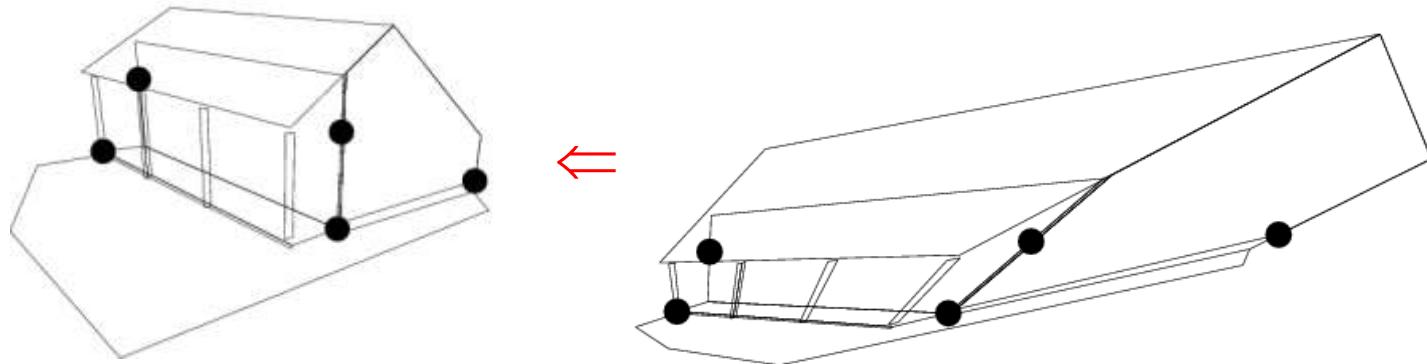


Dodane tekture



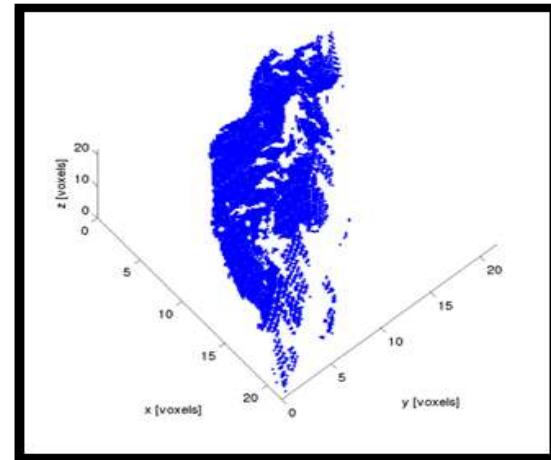
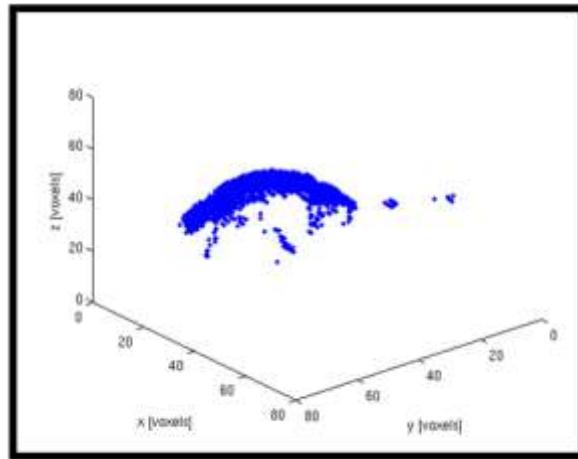
# MOŽNOST NEPOSREDNE METRIČNE REKONSTRUKCIJE IZ PROJEKCIJSKE

1. Ko dobimo 3D rekonstrukcijo iz projekcije (točke  $\mathbf{p}_i$ ), jo lahko pretvorimo neposredno v metrično, če poznamo dejanske koordinate za nadzorne točke  $\mathbf{p}_{E,i}$  v naravi
2. Potrebujemo vsaj 5 takšnih točk
3. Izračunati moramo homografijo  $\mathbf{H}$ , ki ima 15 prostostnih stopenj.
4. Za vsako rekonstruirano točko lahko nastavimo 3 linearne enačbe:  
$$\mathbf{p}_{E,i} = \mathbf{H}\mathbf{p}_i$$
5. Z dobljeno matriko  $\mathbf{H}$  preslikamo vse točke v metrični prostor:



# Primer projekcijske 3D rekonstrukcije

(dobljeno s faktorizacijo na posnetkih breskev)

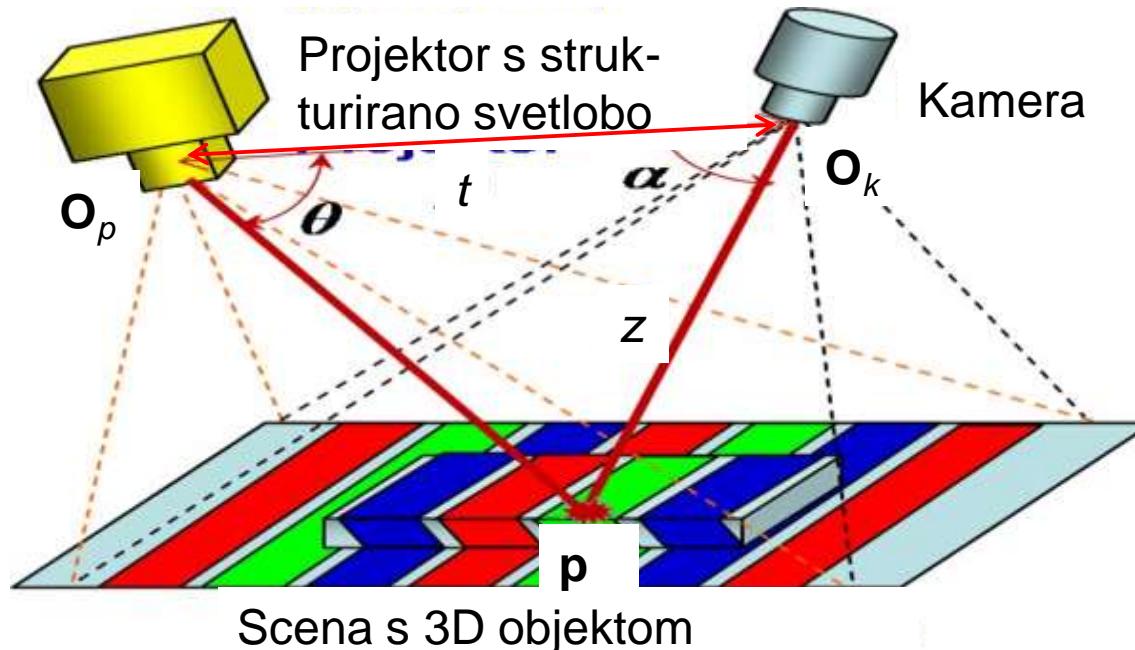


# 3D REKONSTRUKCIJA OBJEKTOV S POMOČJO STRUKTURIRANE SVETLOBE

1. 3D rekonstrukcija, ki temelji na *strukturirani svetlobi (structured light)*, določi le **položaj točk na površini objektov**.
2. Drugačno vrsto 3D rekonstrukcije srečamo v medicini, kjer je treba določiti tudi lego vokslov v notranjosti objektov – “presvetljevanje” tkiv z rentgenom (računalniška tomografija – CT), ultrazvokom ali magnetno resonanco (MRI).
3. Uporaba strukturirane svetlobe omogoča samo površinsko rekonstrukcijo.
4. Dejansko gre za **globinsko rekonstrukcijo**, podobno kot pri stereo pogledu, le da tukaj eno od kamer zamenja **projektor**.
5. Projektor mora imeti čim večjo svetilnost in oddajati strukturirano svetljivo, ki na sceni naredi lahko zaznavne svetlobne vzorce.
6. **Prednosti metode:** rekonstruiramo lahko predmete, ki nimajo tekstur, saj brez strukturirane svetlobe na njih ne bi mogli najti korespondenčnih točk.
7. **Slabosti metode:** deluje le v nadzorovanem okolju in je odvisna od osvetlitvenih pogojev.

# DOLOČANJE GLOBINE S STRUKTURIRANO SVETLOBO

1. Sklop projektorja in kamere mora biti kalibriran:
  - poznati moramo razdaljo med objektivoma:  $t$
  - poznati moramo kot, pod katerim projicira projektor določeno osvetlitveno značilnost:  $\theta$
2. Iz slike scene, osvetljene s strukturirano svetlobo, razberemo projicirano značilnost in določimo kot, pod katerim jo kamera vidi:  $\alpha$



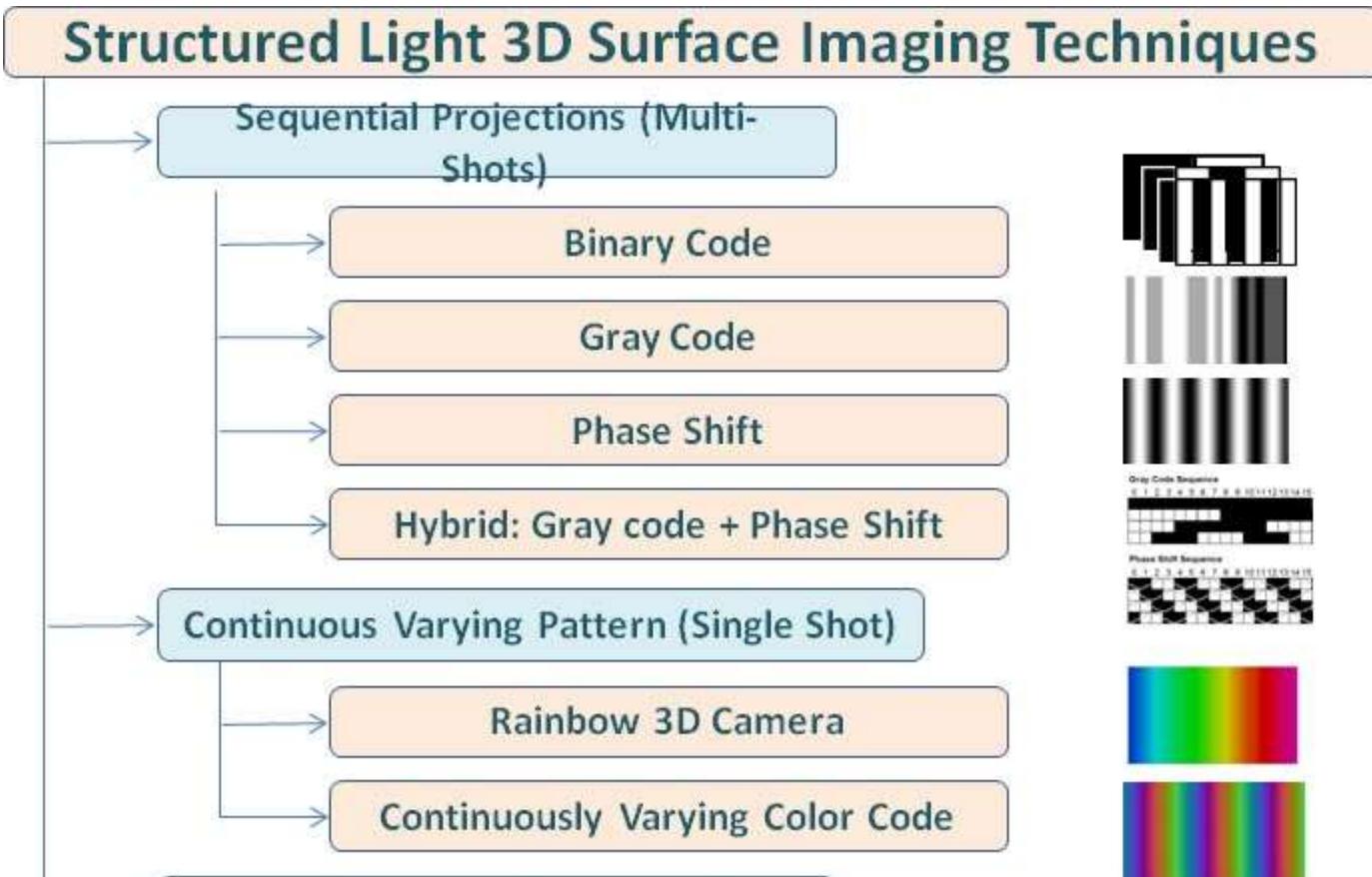
Globina z točke  $p$ :

$$z = t \frac{\sin \theta}{\sin(\theta + \alpha)}$$

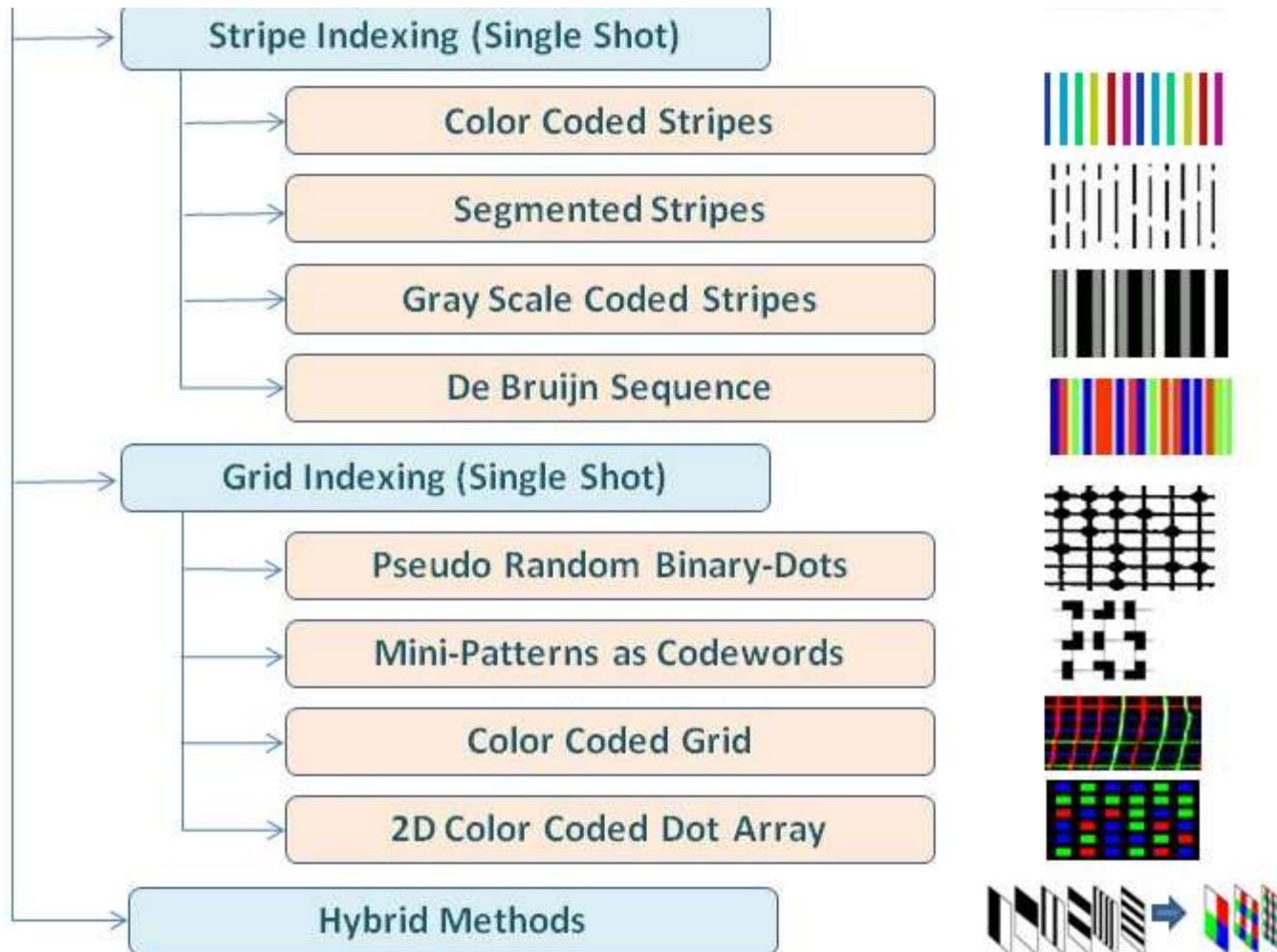
(pomagamo si z višino na zveznico med  $O_p$  in  $p$ )

# VRSTE STRUKTURIRANE SVETLOBE

(vzeto iz: Jason Geng, Structured-light 3D surface imaging: a tutorial,  
*Advances in Optics and Photonics*, Vol. 3, No. 2, str. 128-160)



# VRSTE STRUKTURIRANE SVETLOBE (2)



# REŠITEV PRI MICROSOFTOVEM KINECTU

1. Kinect projicira infrardečo (IR) svetlobo v obliki pegastega šuma.
2. Svetlobo strukturira tako, da IR laser sveti skozi plastično lečo z vgravirano mikrostrukturo (difuzer) za nastanek peg.
3. Kamera vsebuje monokromatski CMOS-senzor in zaznava globino z algoritmom, ki ga je razvilo izraelsko podjetje PrimeSense. Sledi obrazom in gibom telesa.
4. Primer scene s knjigo in brez nje:

