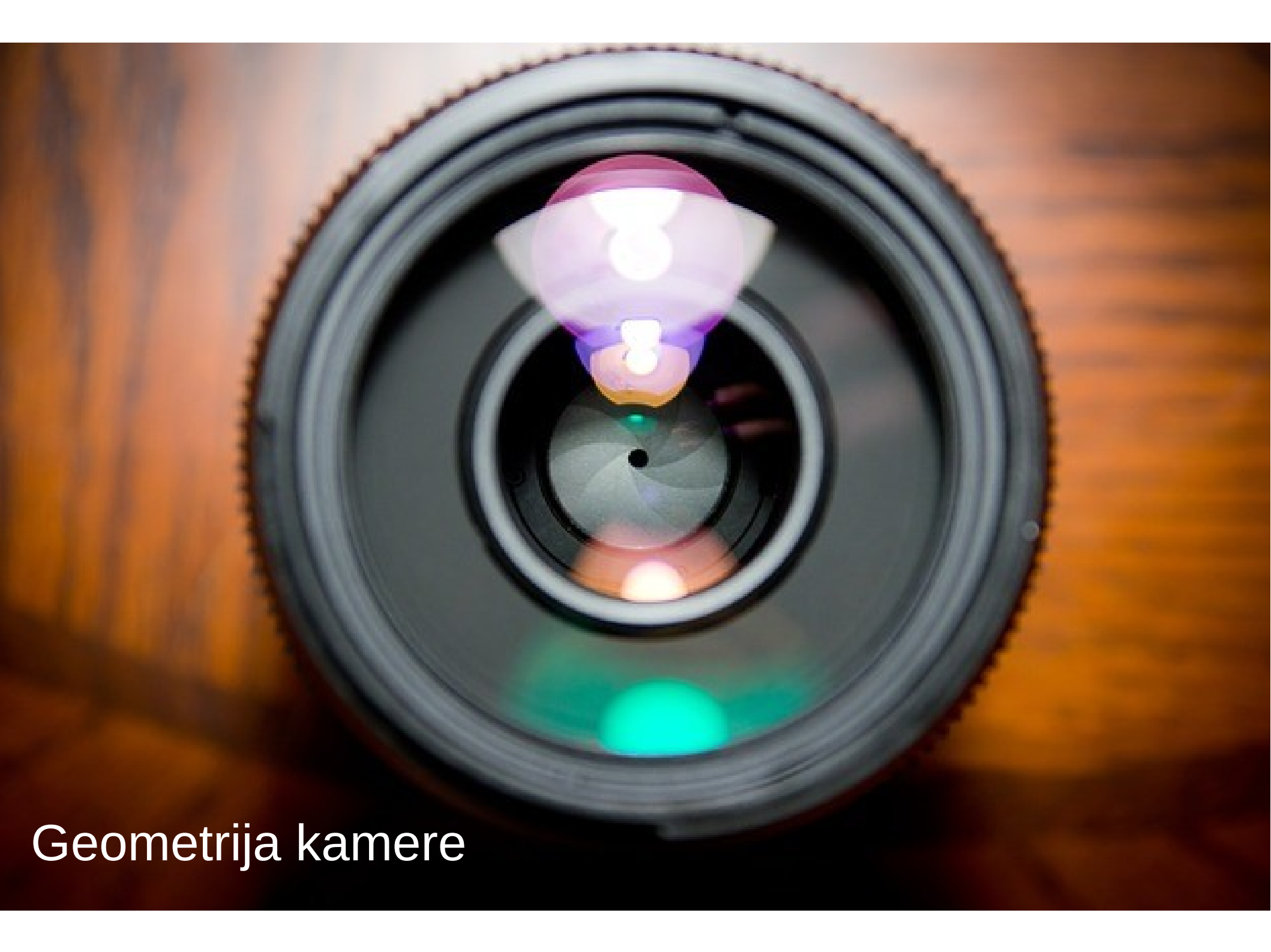


Projektivna geometrija

Pregled

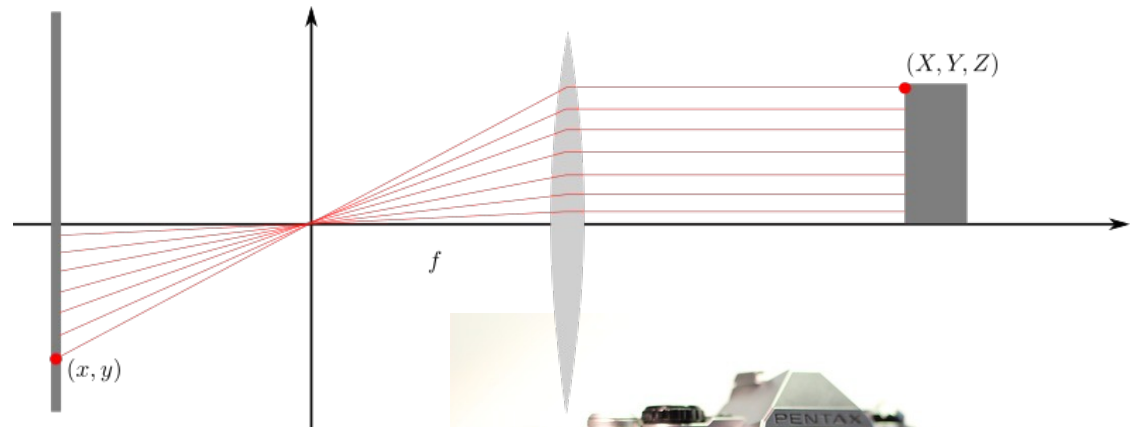
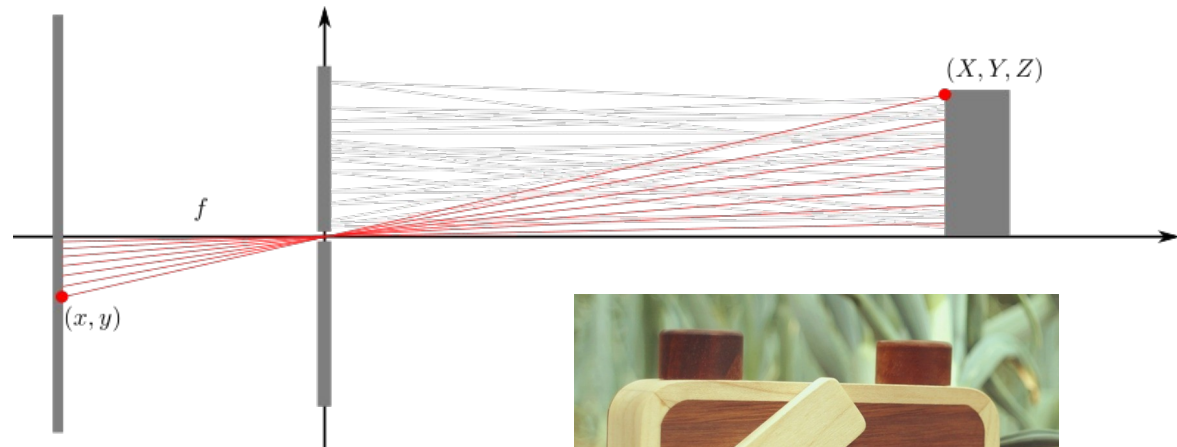
- Kamera
 - Kamera z luknjico ali lečo
 - Kalibracija kamere
- Značilne regije
 - Detekcija
 - Opis
 - Ujemanje
- Preslikave
 - Homografija



Geometrija kamere

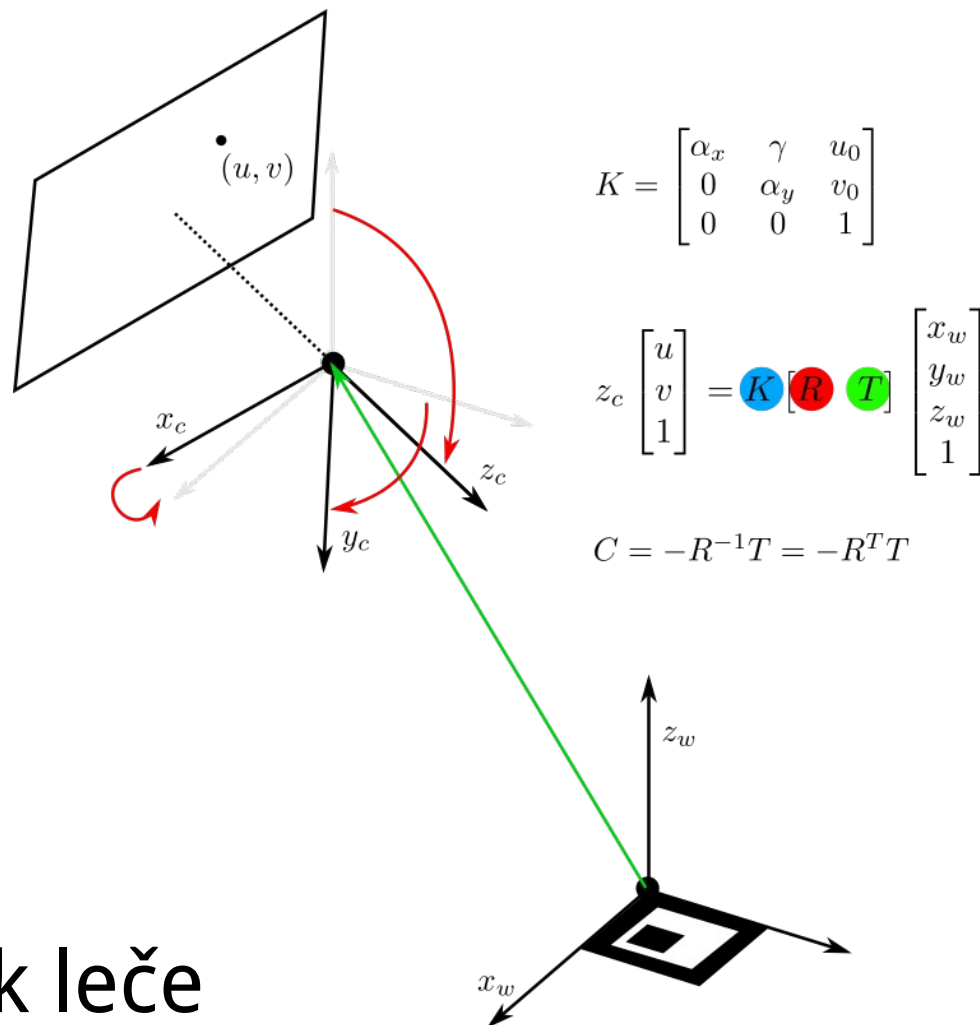
Kamera

- Kamera z luknjico
 - Malo svetlobe
 - Dolg čas osvetlitve
 - Luknjica neskončno majhna
- Kamera z lečo
 - Več svetlobe
 - Krajši čas osvetlitve
 - Popačenja



Parametri kamere

- Ekstrinzični parametri
 - Položaj
 - Rotacija
- Intrinzični parametri
 - Goriščna razdalja
 - Format senzorja
 - Izhodiščna točka
- Popačenje zaradi napak leče



Kalibracija kamere

- Modeliranje parametrov

- Intrinzični parametri

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

- Radialna distorzija

$$x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

$$y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

- Tangentalna distorzija

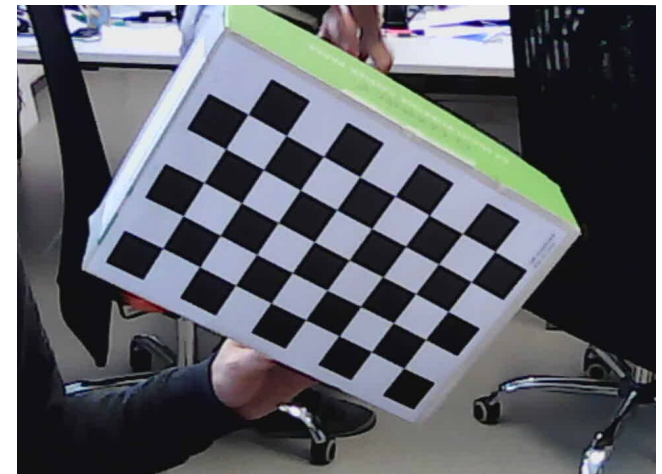
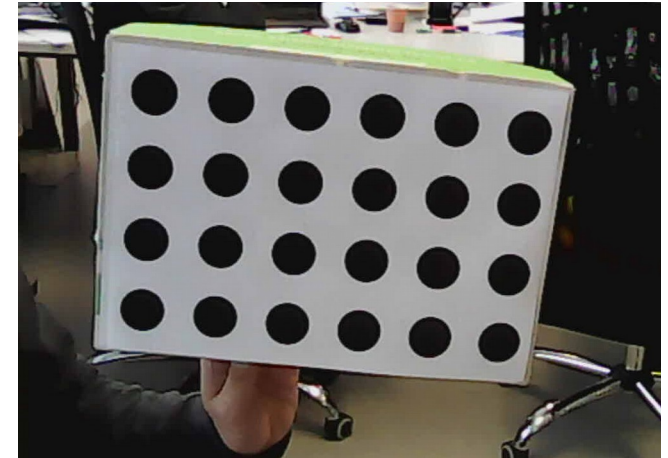
$$x_{corrected} = x + [2p_1xy + p_2(r^2 + 2x^2)]$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2xy]$$



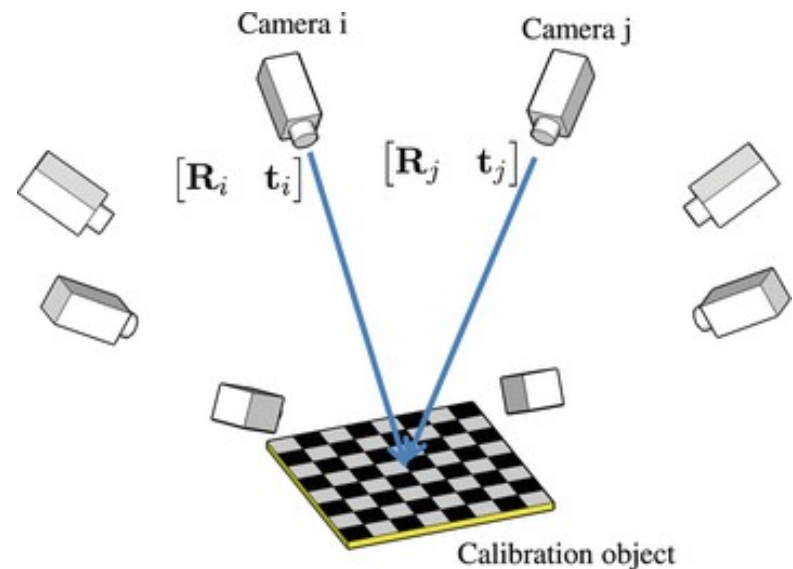
Potek kalibraicije

- Uporaba predloge
 - Šahovnica
 - Mreža krogov
- Zajem slik znane predloge
 - Več slik
 - Različni koti
 - Pokritost vidnega polja
- Zaznavanje predloge
 - Detekcija kotov ali krogov
 - Poznana velikost v realnem svetu
- Reševanje sistema enačb

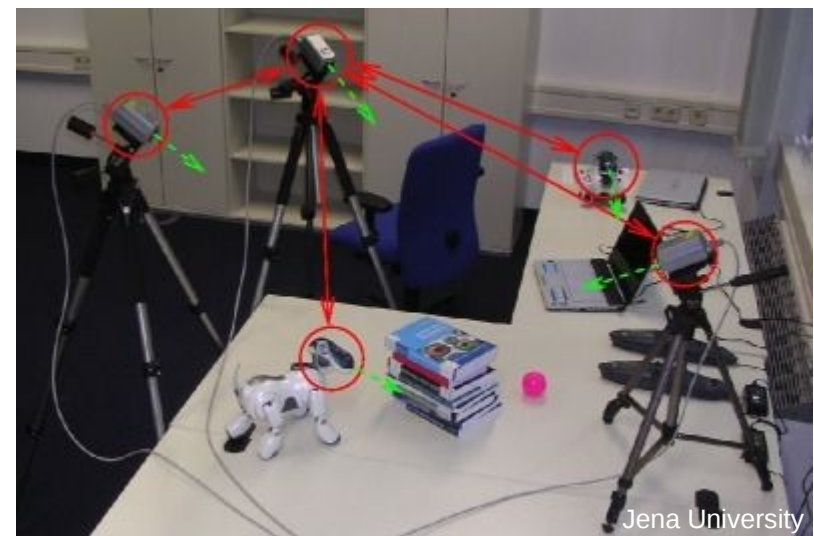


Ocenjevanje zunanjih parametrov

- Poznani vzorec
 - Podobno kot pri kalibraciji
 - Poznamo velikost
- Iskanje ujemanj med dvema kamerama
 - Iskanje istih točk v sceni (enak izgled)
- Sistem večih kamer
 - Kalibriramo pare kamer
 - Preslikava na en referenčni prostor



Jun-Sik Kim





Značilne točke

Kako rešimo sestavljanke?



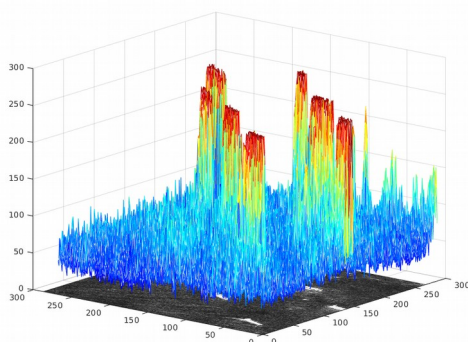
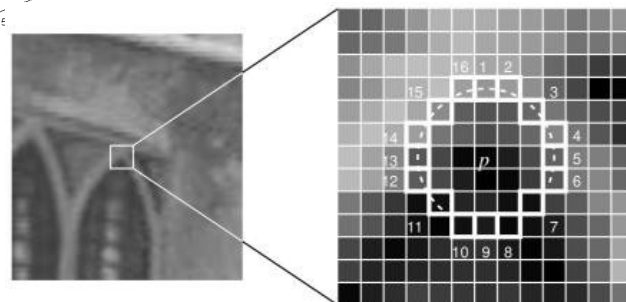
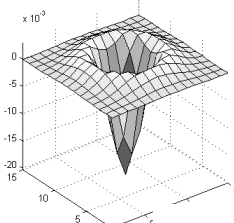
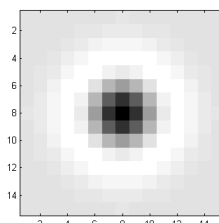
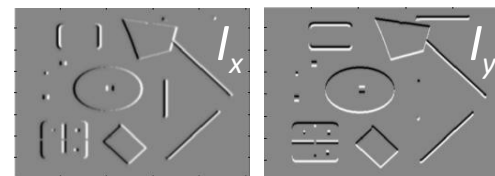
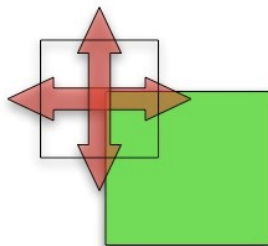
Pomen značilnih točk

- Zanesljivost zaznave v sliki
 - Na istem mestu, ponovljivost
 - Podobna okolica
- Računska učinkovitost iskanja ujemanja
 - Hiter izračun opisa
 - Hitra primerjava



Detektorji značilnih točk

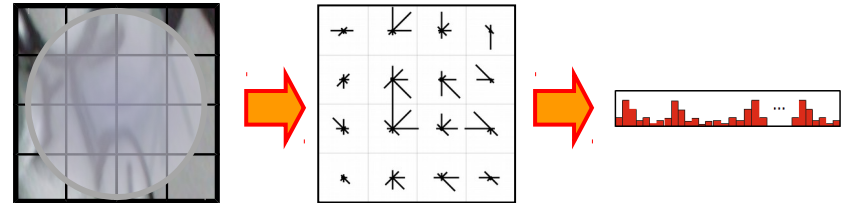
- Harris
 - Avtokorelacija
 - Prvi odvodi slike
- Hessian
 - Drugi odvodi slike
 - Invarianten na skalo
- LoG, DoG
 - detekcija “pack”
 - Piramida
- FAST
 - strojno učenje
- MSER
 - stabilne regije



Opisniki značilnih območij

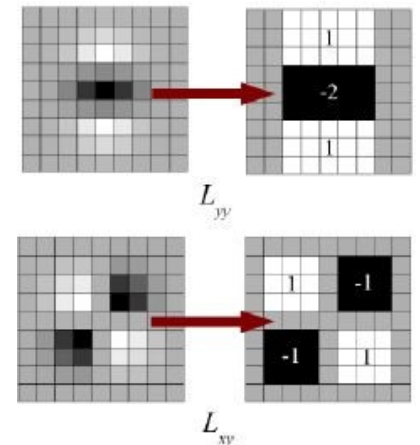
- SIFT

- Histogrami gradientov



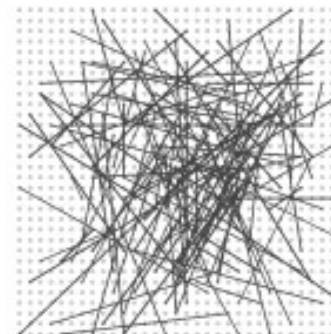
- SURF

- Pospešeni SIFT
- Binarni filtri, integralna slika



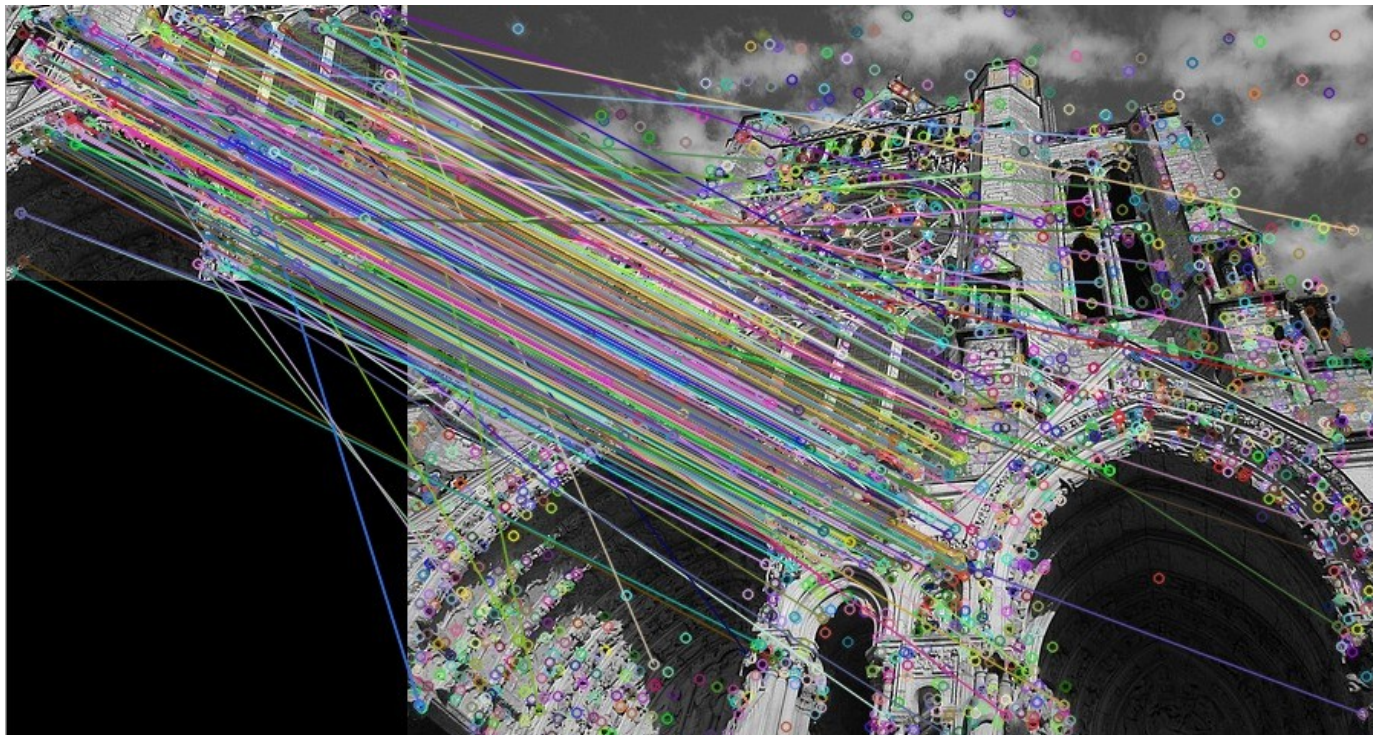
- BRIEF

- Naučen vzorec primerjav
- Binarni vektor



Iskanje ujemanj

- Primerjava opisnikov
 - Vsak z vsakim
 - Pametno preverjanje
- Simetrično ujemanje

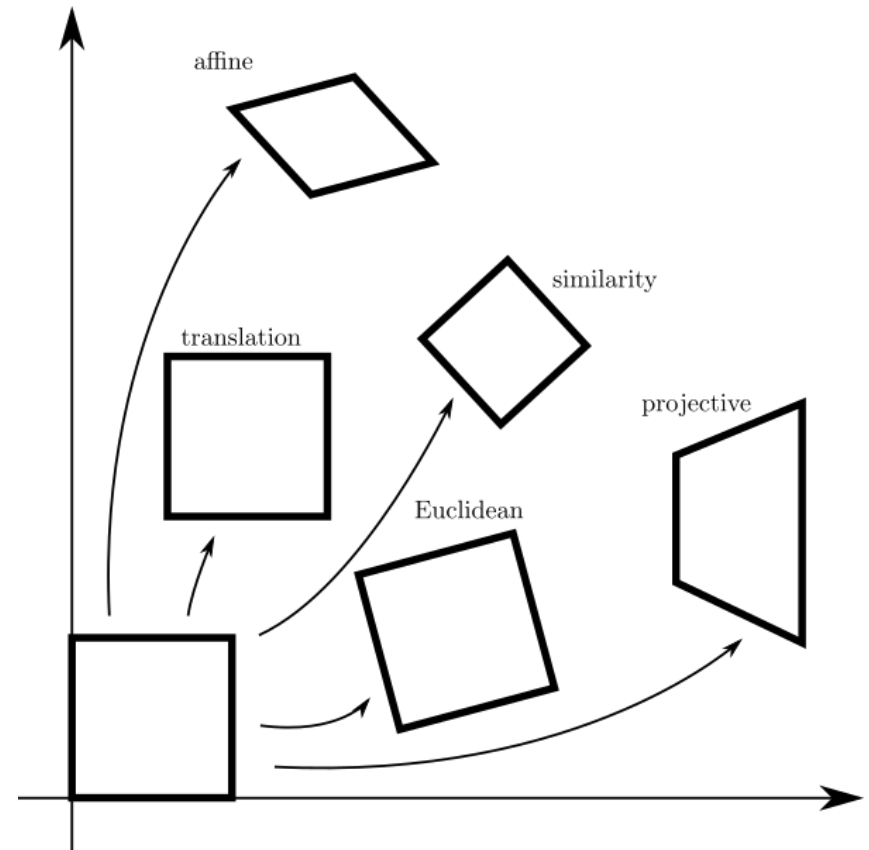


Modeliranje transformacije



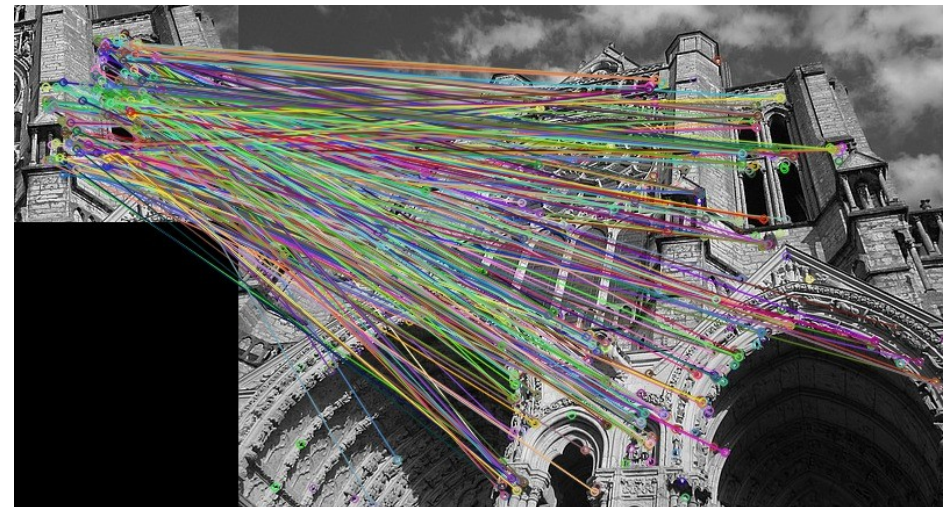
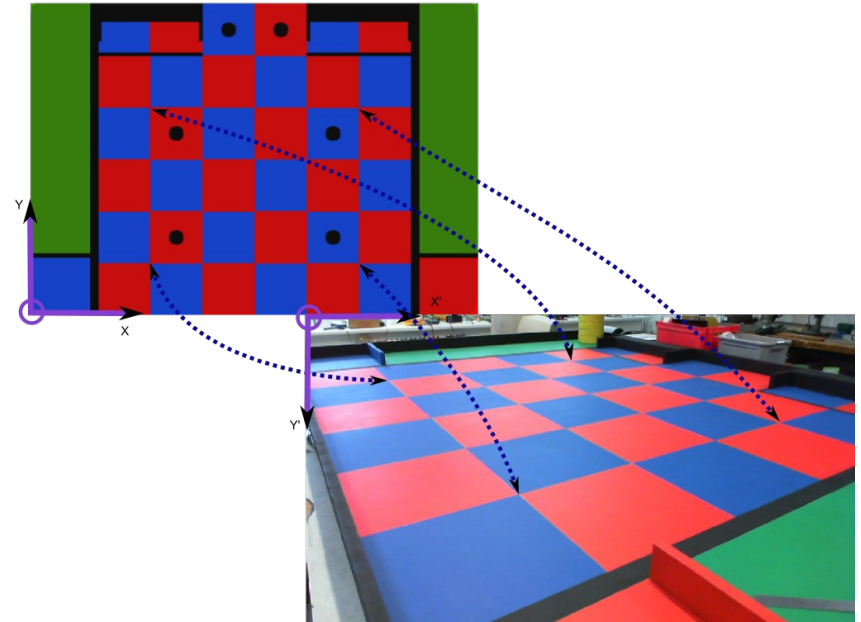
Linearne preslikave

- Translacija (1)
- Evklidska preslikava (2)
 - Rotacija
- Podobnost (2)
 - Skaliranje
- Afina preslikava (3)
 - Striženje
- Projekcija / Homografija (4)
 - Točke med ravninama
 - Ne ohranja vzporednosti premic



Preslikava iz podatkov

- Množica parov točk
 - Znana geometrija
 - Značilne točke
- Sistem linearnih enačb
 - Število neznank
 - Predeterminiran sistem
- Pokritost slike
 - Interpolacija
 - Ekstrapolacija

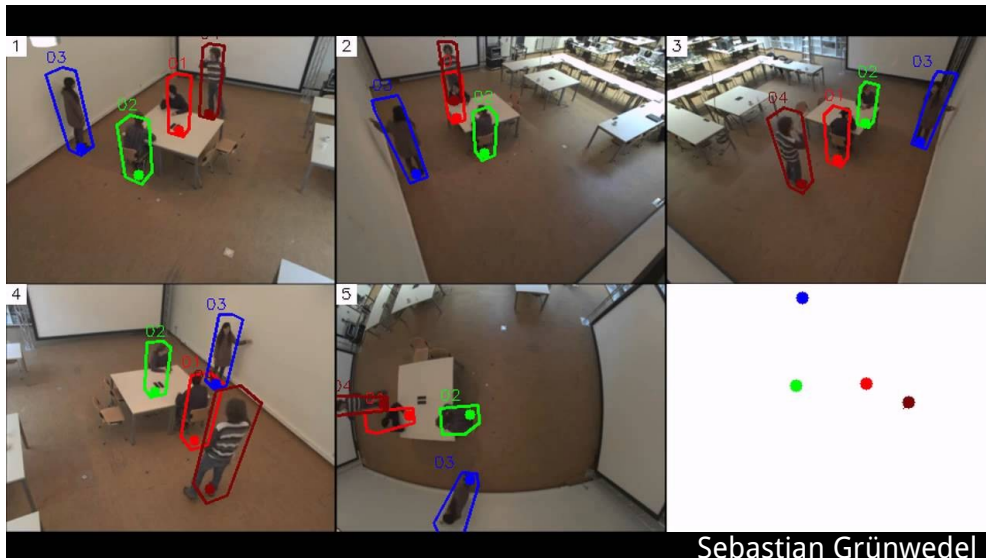


Robustno določanje parametrov

- Minimiziranje kvadratne napake
 - Vsi podatki (pari ujemanj)
 - Ni robustno
- Iskanje konsenza (RANSAC)
 - Naključna izbira minimalne podmnožice
 - Izračun modela
 - Štetje podpore med ostalimi pari
 - Izbor najboljšega modela po več ponovitvah

Uporaba preslikav

- Obogatena resničnost
- Več-kamerni sistemi
- 3D rekonstrukcija



Carlos Hernández Esteban

Še nekaj nasvetov

- Kalibracija
 - Več poizkusov
 - Pri spletnih kamerah so parametri lahko “splošni”
- Iskanje ujemanj
 - Združevanje večih tipov značilnih točk
 - Ocenite kako hiter algoritem rabite
- Iskanje preslikave
 - Čimveč primerov
 - Pokritost slike

Vaje 4: kamera in homografija

- Kalibracija kamere
- Preslikava med kamerami
- Detekcija značilnih točk
- Iskanje ujemanj
- Homografija

Kalibracija kamere (calibrate.cpp)

Lokacija vzorca:

- findChessboardCorners + cornerSubPix
- findCirclesGrid

Kalibracija kamere:

- calibrateCamera

Usage: calibration [params]

```
-c, --camera (value:0)
    Camera ID
--columns, -x (value:5)
    Number of points in width
-h, --help (value:true)
    Print this help
-o, --output (value:camera.yaml)
    Size of rectangle or circle in millimeters
-p, --pattern (value:chessboard)
    Pattern type (chessboard,circles)
--rows, -y (value:8)
    Number of points in height
--size (value:50)
    Size of rectangle or circle in millimeters
```

Plan B: uporabite generični model za spletne kamere.

Rektifikacija (rectify.cpp)

Izračun optimalne matrike

`getOptimalNewCameraMatrix(intrinsics, distortion, size, outliers)`

- `size` - velikost končne slike
- `outliers` - število črnih elementov (0 - noben, 1 - max)

Inicializacija preslikovalne tabele

`initUndistortRectifyMap(intrinsics, distortion, R,
camera_matrix, size, type, in_map, out_map)`

- `R` - se uporablja v stereo vidu (drugače je `Mat()`)
- `camera_matrix` - matrika kamere
- `size` - velikost slike
- `type` - tip preslikovalnih tabel

Uporaba preslikovalne tabele

`remap(in, out, in_map, out_map, interpolation)`

Položaj kamere (pose.cpp)

Iskanje položaja objekta s pari 2D in 3D točk

`solvePnP(src, dst, intrinsics, distortion, rotation, translation)`

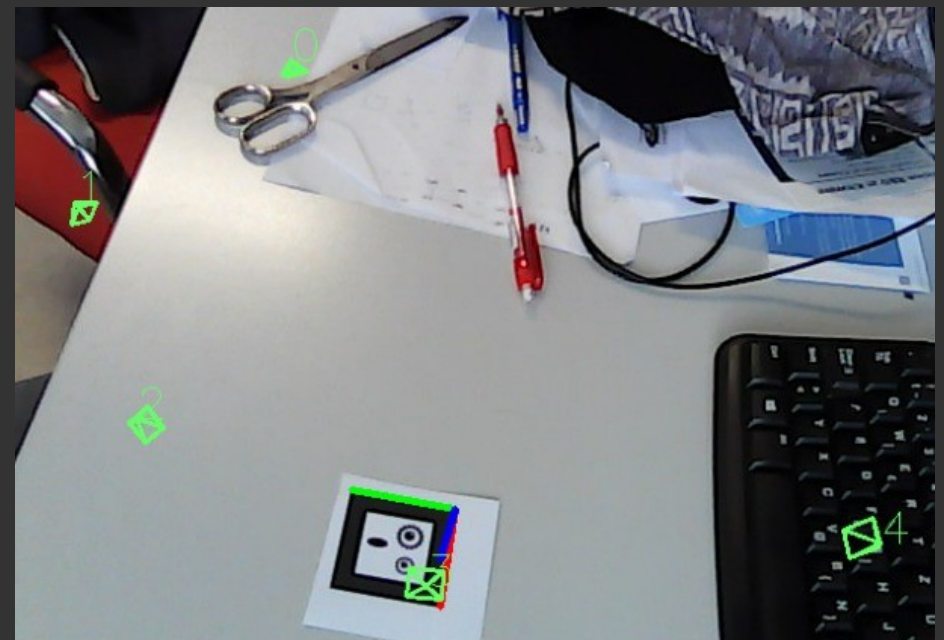
- `src` - izhodiščne točke (npr. oznaka)
- `dst` - končne točke (npr. v prostoru kamere)
- `intrinsics` - intrinzični parametri
- `distortion` - popačenje leče
- `rotation` - ocenjena rotacija (kvarternion)
- `translation` - ocenjena translacija

`projectPoints(in, rotation, translation, intrinsics, distortion, out)`

`Rodrigues(in, out)`

`perspectiveTransform(in, out, T)`

- 2D in 3D
- Homogene koordinate



Značilne točke (keypoints.cpp)

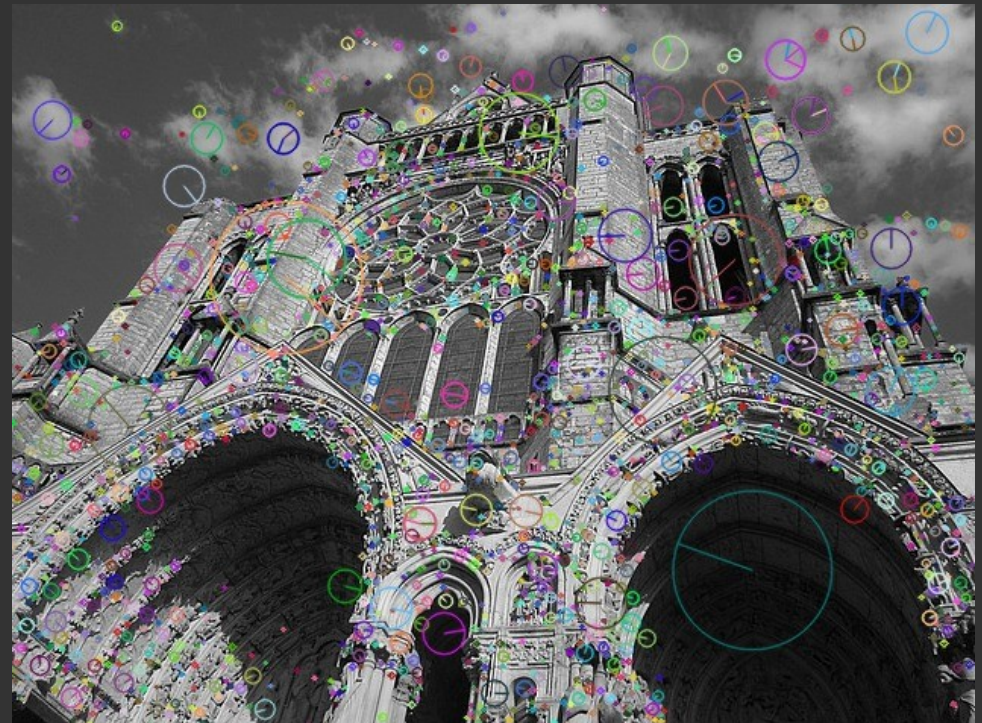
```
#include <opencv2/features2d.hpp> // < FAST, ORB, AKAZE  
#include <opencv2/xfeatures2d.hpp> // < SIFT, SURF
```

```
using namespace cv::xfeatures2d;
```

```
std::vector<KeyPoint> keypoints;
```

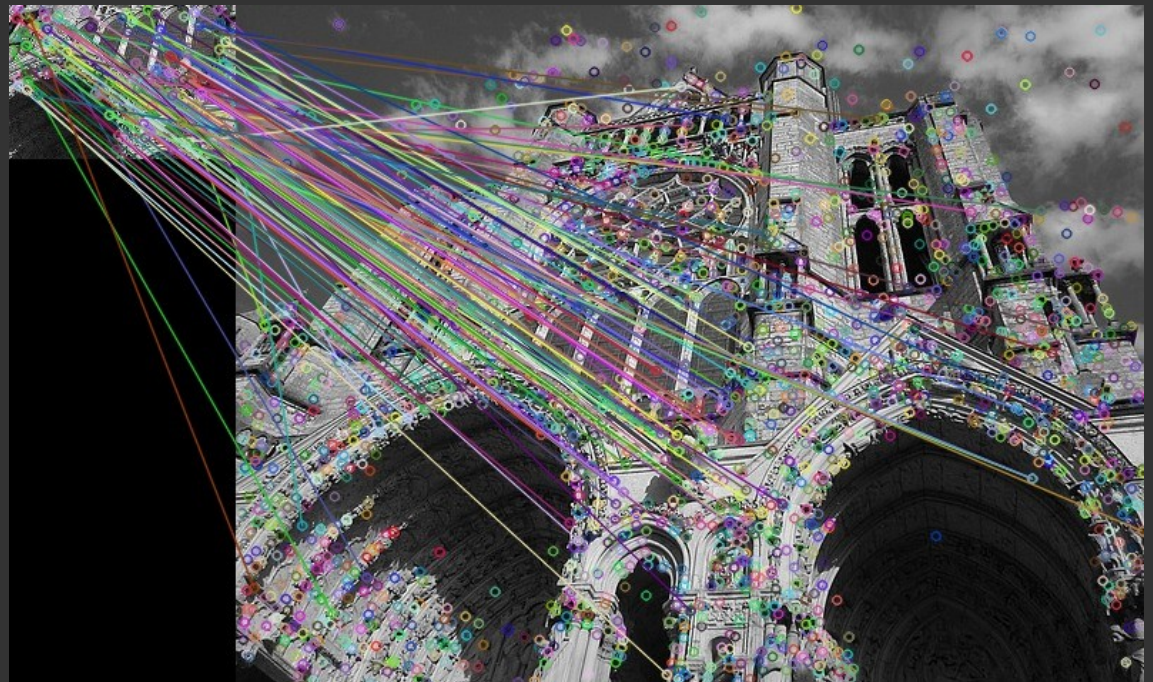
```
Ptr<Feature2D> detector = SIFT::create();  
detector->detect(img, keypoints);
```

```
drawKeypoints(...);
```



Ujemanja (matching.cpp)

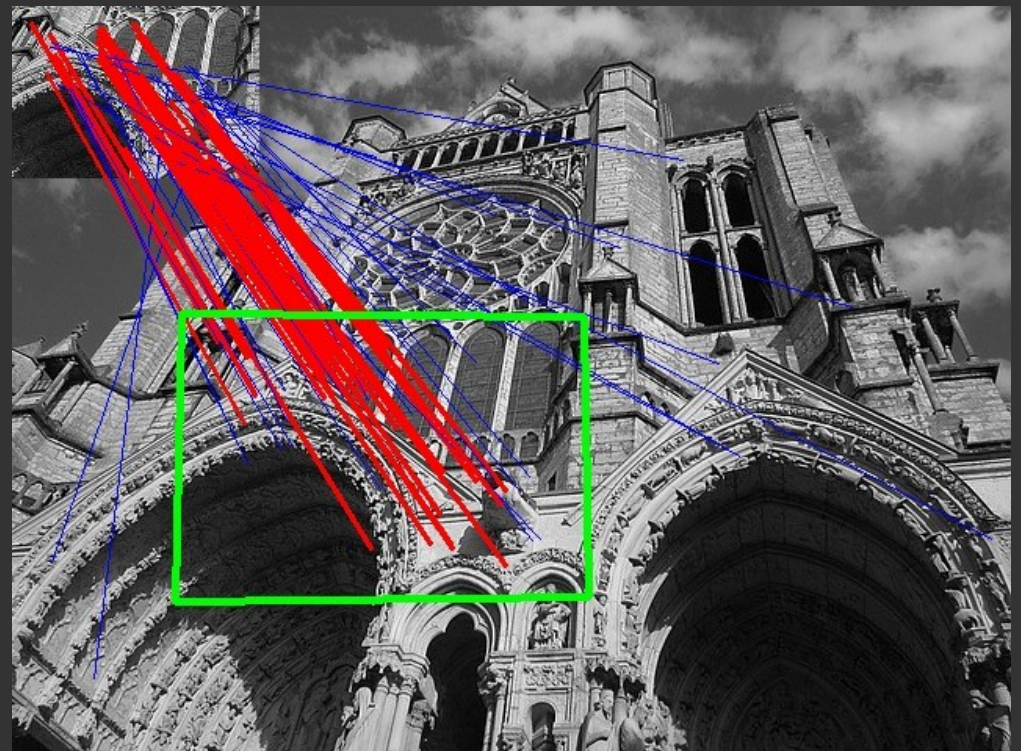
```
Ptr<Feature2D> descriptor = SIFT::create();  
Ptr<DescriptorMatcher> matcher = new BFMatcher();  
vector<DMatch> matches;  
  
descriptor->compute(img, keypoints, descriptors);  
descriptor->detectAndCompute(img, Mat(), keypoints, descriptors);  
  
matcher->match(descriptors1, descriptors2, matches);
```



Prileganje modela (homography.cpp)

```
H = findHomography(src, dst, method, error, inliers)
```

- src
- dst
- method - metoda za iskanje modela
 - None - uporabi vse točke
 - RANSAC - Random Sample Consensus algorithm
 - LMEDS - Least Median algorithm
 - RHO - PROSAC algorithm
- error - threshold
- inliers - uporabljene točke



Naloga

- Kalibrirajte svojo kamero
- Zajemite sliko predloge
- V predlogi poiščite značilne točke in jih najдите v testni sliki, uporabite več opisnikov
 - Za vsak opisnik poiščite ujemanja
 - Izberite določeno število najboljših parov
- Izračunajte homografijo
 - Združite pare iz večih opisnikov
- Preslikajte v sliko pravokotnik v velikosti predloge

Reference

- Snov se bolj podrobno obravnava pri predmetih
 - Umetno zaznavanje
 - Razvoj inteligentnih sistemov
- http://docs.opencv.org/3.1.0/da/d9b/group_features2d.html
- http://docs.opencv.org/3.1.0/d9/d0c/group_calib3d.html