VJEŽBA 3: HOUGHOVA TRANSFORMACIJA

Opis vježbe:

Potrebno je pomoću, prethodno kalibrirane, web kamere uslikati objekt kvadratnog oblika koji je postavljen na milimetarskom papiru na stolu. Primjenom Houghove transformacije (HT) treba odrediti parametre ρ i θ najdominantnijeg pravca, koji odgovara jednom od rubova objekta na slici. Pod najdominantnijim pravcem podrazumijeva se pravac kojem pripada najveći broj 'glasova' u akumulacijskoj ravnini. Implementacija HT u biblioteci OpenCV vraća popis detektiranih pravaca koji su razvrstani prema broju 'glasova' počevši od najdominantnijeg. Primjenom odgovarajuće transformacije, odrediti ρ' i θ' tog pravca u koordinatnom sustavu milimetarskog papira. Provjeriti koliko je odstupanje dobivenog pravca od stvarnog (odgovarajućeg) ruba objekta.

Priprema:

a) Downloadirati VS2008 projekt odnosno zip datoteku CameraCalibration.zip te raspakirati sadržaj na disk.
b) Pokrenuti projekt te uz pomoć *kalibracijskog panela* kalibrirati kameru. Kalibracijski panel možete napraviti pomoću slike Chessboard.pdf.
c) Pročitati u dokumentaciji OpenCV biblioteke detalje vezane za metode `cvInitUndistortMap`, `cvFindExtrinsicCameraParams2`, `cvHoughLines2` te `cvRodrigues2`.
d) Analizirati Listing 3.1

```
//Display undistorted images

CvMat *intrinsic = (CvMat*)cvLoad("Intrinsics.xml");
CvMat *distortion = (CvMat*)cvLoad("Distortion.xml");
// Build the undistort map that we will use for all
// subsequent frames.
//
IplImage* mapx = cvCreateImage( cvGetSize(image), IPL_DEPTH_32F, 1 );
IplImage* mapy = cvCreateImage( cvGetSize(image), IPL_DEPTH_32F, 1 );
cvInitUndistortMap(intrinsic,distortion,mapx,mapy);

// Just run the camera to the screen, now showing the raw and
// the undistorted image.
//
cvNamedWindow( "Undistort");
while(image)
{
       IplImage *t = cvCloneImage(image);
       cvShowImage( “Calibration”, image ); // Show raw image
       cvRemap(t,image,mapx,mapy ); // Undistort image
       cvReleaseImage(&t);
       cvShowImage("Undistort", image); // Show corrected image
       //Handle pause/unpause and ESC
       int c = cvWaitKey(15);
       if(c == 'p')
       {
              c = 0;
              while(c != 'p' && c != 27)
              {
                     c = cvWaitKey(250);
              }
       }
       if(c == 27)
              break;
       image = cvQueryFrame( capture );
}
```

Listing 3.1.
(Izvor: Learning OpenCV Computer Vision with the OpenCV Library, Gary Bradski & Adrian Kaehler str. 401)

Rad na vježbi:

U aplikaciji napravljenoj u prvoj vježbi treba:

a) Napisati funkciju tako da korisnik pomoću web kamere uslika objekt koji se nalazi na milimetarskom papiru na stolu. Omogućiti u programu da se mišem može označiti (klikom) četiri ugla milimetarskog papira na slici. Pomoću metode `cvFindExtrinsicCameraParams2` odrediti rotacijsku matricu i translacijski vektor uz pomoć prethodne učitane intrinsične matrice (Intrinsics.xml) te koeficijente distorzije (Distortion.xml). Primjenom izraza u prilogu treba odrediti koliko se pravac, dobiven na slici pomoću Houghove transformacije, podudara s odgovarajućim rubom objekta na stolu.

Neke od metoda i struktura OpenCV-a korisne za rad na vježbi:

Metode:

```
cvLoad; cvCreateMat; cvCreateImage; cvRemap; cvInitUndistortMap; cvHoughLines2;
cvRodrigues2; cvFindExtrinsicCameraParams2
```

Linkovi:

Intrinsics.xml	- http://moodle.etfos.hr/mod/resource/view.php?id=10680
Distortion.xml	- http://moodle.etfos.hr/mod/resource/view.php?id=10681
CameraCalibration.zip	- http://moodle.etfos.hr/mod/resource/view.php?id=17548
Chessboard.pdf	- http://moodle.etfos.hr/mod/resource/view.php?id=17549

Prilog

Za pravac opisan parametrima ρ i θ na slici (2D), dobiven web kamerom, treba odrediti odgovarajuće parametre pravca ρ' i θ' u 3D prostoru. Neka je (u,v) koordinata točke koja leži na pravcu (ρ, θ) na slici, a (x,y,z) odgovarajuća točka u prostoru koja se također nalazi na pravcu (ρ', θ') u prostoru. Tada vrijedi sljedeće:

$$\mathbf{m} = s \cdot \mathbf{P} \cdot (\mathbf{R} \cdot \mathbf{p} + \mathbf{t}) \tag{1}$$

gdje je :

- $\mathbf{m} = [u \quad v \quad 1]^T$ – koordinate točke na slici;
- $\mathbf{p} = [x \quad y \quad z]^T$ – koordinate točke u koordinatnom sustavu milimetarskog papira S_0;
- $\mathbf{P} = \begin{bmatrix} f_x & 0 & u_c \\ 0 & f_y & v_c \\ 0 & 0 & 1 \end{bmatrix}$ – projekcijska matrica, gdje su f_x , f_y, u_c i v_c intrinsični parametri kamere;
- $\mathbf{R}$ – rotacijska matrica koja opisuje orijentaciju k. s. S_0 u odnosu na k. s. kamere;
- $\mathbf{t}$ – translacijski vektor koji opisuje poziciju k. s. S_0 u odnosu na k. s. kamere.

Iz (1) dobije se:

$$\mathbf{m} = s \cdot (\mathbf{A} \cdot \mathbf{p} + \mathbf{b}) \tag{2}$$

gdje je:

$\mathbf{A} = \mathbf{P} \cdot \mathbf{R} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3]^T$;

$\mathbf{b} = \mathbf{P} \cdot \mathbf{t} = [b_1 \quad b_2 \quad b_3]^T$.

Iz (2) vrijedi sljedeće:

$$u = s(\mathbf{a}_1 \cdot \mathbf{p} + b_1),$$

$$v = s(\mathbf{a}_2 \cdot \mathbf{p} + b_2),$$

$$1 = s(\mathbf{a}_3 \cdot \mathbf{p} + b_3) \Rightarrow s = \frac{1}{\mathbf{a}_3 \cdot \mathbf{p} + b_3}$$

Jednadžba pravca na slici opisana parametrima ρ i θ glasi:

$$u \cdot cos\theta + v \cdot sin\theta = \rho \tag{3}$$

odnosno:

$$(\mathbf{a}_1 \cdot \mathbf{p} + b_1) \cdot cos\theta + (\mathbf{a}_2 \cdot \mathbf{p} + b_2) \cdot sin\theta = \rho \cdot (\mathbf{a}_3 \cdot \mathbf{p} + b_3) \tag{4}$$

odnosno:

$$\lambda_x \cdot x + \lambda_y \cdot y = \lambda_\rho \tag{5}$$

gdje je:

$$\lambda_x = a_{11}cos\theta + a_{21}sin\theta - \rho a_{31}$$
$$\lambda_y = a_{12}cos\theta + a_{22}sin\theta - \rho a_{32}$$
$$\lambda_\rho = (a_{33}z + b_3)\rho - (a_{13}z + b_1)cos\theta - (a_{23}z + b_2)sin\theta$$

Pošto je z = 0, vrijedi

$$\lambda_\rho = b_3\rho - b_1cos\theta - b_2sin\theta.$$

Izraz (5) predstavlja jednadžbu pravca u xy-ravnini k. s. S_0. Isti se pravac može opisati i jednadžbom

$$x \cdot cos\theta' + y \cdot sin\theta' = \rho',$$

gdje je:

$$\theta' = atan2(\lambda_y, \lambda_x),$$

$$\rho' = \frac{\lambda_\rho}{\sqrt{{\lambda_x}^2 + {\lambda_y}^2}}.$$