

# Osnove robotike

## LV4 Robotski vid

Denis Lazor

**Zadatak:** Koristeći robotski vid i rubove predmeta kao značajke treba postaviti alat robotskog manipulatora iznad tog istog predmeta.

**Manipulator:** Motoman MH50-35

**Model:** Nacrt i bokocrt u AutoCad-u skalirani s 0.1 uz poštivanje vidljivih dimenzija i odokativnog određivanja ostalih. Na modelu se nalaze i označeni koordinatni sustavi zajedno sa svim njihovim osima.

## Dijelovi MATLAB skripti:

### → my\_robot\_script\_vision.m

```
% Setting parameters explained at beginning of script
zT0 = -14.5;  -- Postavljanje visine plohe na vrijedost jednakoj dnu robota
xmin = 50;    -- Domet robotske ruke // Nije isti u svim smjerovima zbog grade robota
xmax = 155.1; -- pa je uzet maksimalan domet za smjer s najmanjim dometom
method = 'edges'; -- Metoda prepoznavanja rubova

my_rvsim;
```

### → my\_createscene.m

```
my_robot_script_CreateRobot; -- Pozivanje vlastite skripte za izradu modela robota

plotbox = [-500 600 -500 600 -500 600]; -- Dimenzije 3D prozora za simulaciju
```

### → my\_rvtraj.m

```
% Initial robot configuration -- Postavljanje robota u početni položaj
T60(:, :, 1) = [1 0 0 -19.3474;
                0 1 0 0 ;
                0 0 1 196 ;
                0 0 0 1 ];

qgr = [8 8 8 8 8 8 100]'; -- Postavljanje ograničenja brzine i ubrzanja robota
ddqgr = [16 16 16 16 16 16 10000]'; -- --||--
```

### → my\_rvsim.m

```
% Box parameters -- Promjena dimenzija predmeta
a = 20;
b = 11;
c = 10;
```

➔ my\_rvboxpose.m

```
function TA0 = my_rvboxpose(P,f,TC0,zC,c)

uc = 320;
vc = 240;
fi_ = []; % Angles array
ro_ = []; % Distances array
p = []; % Edges intersections coordinates

for i=1:4
    fi_ = [fi_ P(1,i)];
    ro_ = [ro_ P(2,i)];
end

% Solving coordinates of lines(edges) intersections
for i = 1:2
    for j = 3:4

        syms uk vk 'real';
        eqn1 = (uk-uc)*cos(fi_(i)) + (vk - vc)*sin(fi_(i)) == ro_(i);
        eqn2 = (uk-uc)*cos(fi_(j)) + (vk - vc)*sin(fi_(j)) == ro_(j);
        [uk, vk] = solve([eqn1, eqn2]);
        p = [p [double(uk) double(vk)]];

    end
end

% Finding coordinates of object center of mass
u = (p(1,1)+p(1,2)+p(1,3)+p(1,4))/4;
v = (p(2,1)+p(2,2)+p(2,3)+p(2,4))/4;

% Translations
x = (u - uc)*((zC-c)/f);
y = (v - vc)*((zC-c)/f);
z = zC - c/2;

% Translation vector
tAC = [x y z]';

% Distances between detected edges
% --> Bigger distance between edges -> shorter edges and vice versa
d1 = abs(double(ro_(1)) - double(ro_(2)));
d2 = abs(double(ro_(3)) - double(ro_(4)));

% Finding the angle of object
% --> x axis in direction of longer edge
if d1 > d2
    alpha = fi_(1);
else
    alpha = fi_(3);
end

% Object rotation matrix
RAC = [cos(alpha) -sin(alpha) 0;
        sin(alpha)  cos(alpha) 0;
        0           0        1];
```

```
% Translation matrix between object and camera
TAC = [RAC tAC;
       0 0 0 1];

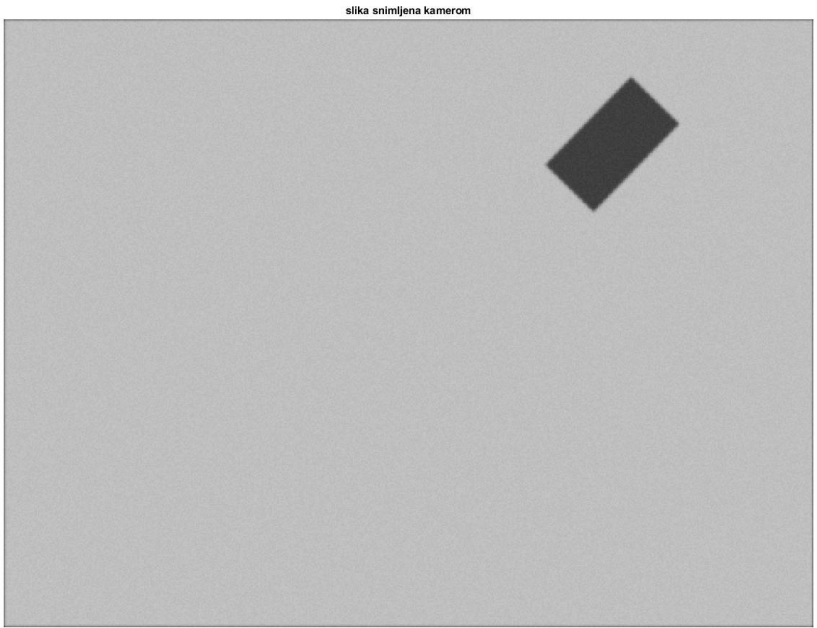
% Translation matrix between object and manipulator base coordinate system
TA0 = TC0*TAC;
```

→ Kod skripte `my_rvboxpose.m` predstavlja isprogramirani predložak za slučaj prepoznavanja rubova koji je je objasnjen na online terminu lv-a. Generalni princip je detekcija rubova nekom od metoda za detekciju što smo dobili implementirano pomoću informacija o udaljenosti od ishodišta i kutevima tih linija u koordinatnom sustavu dolazimo do informacijama o njihovim sjecištima, odnosu duljina, centra mase pa konačno dobivamo poziciju predmeta u stvarnom prostoru u odnosu na kamera pa isto tako i njegovu poziciju u odnosu na bazni koordinatni sustav robotskog manipulatora nakon čega možemo inverznom kinematikom postaviti vrh alata iznad željenog objekta.

- Cijeli kodovi se nalazi u skriptama s prefixom 'my\_'

REZULTATI:

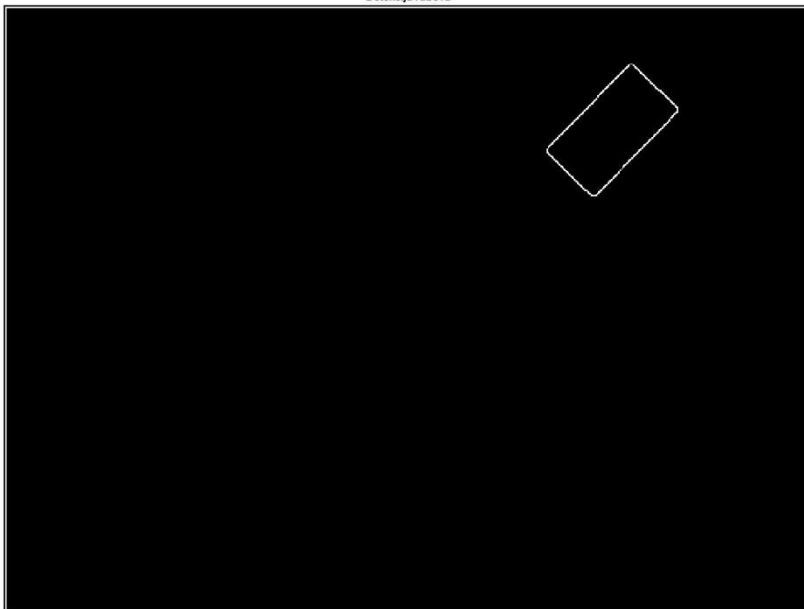
➔ Slike dobivene detekcijom

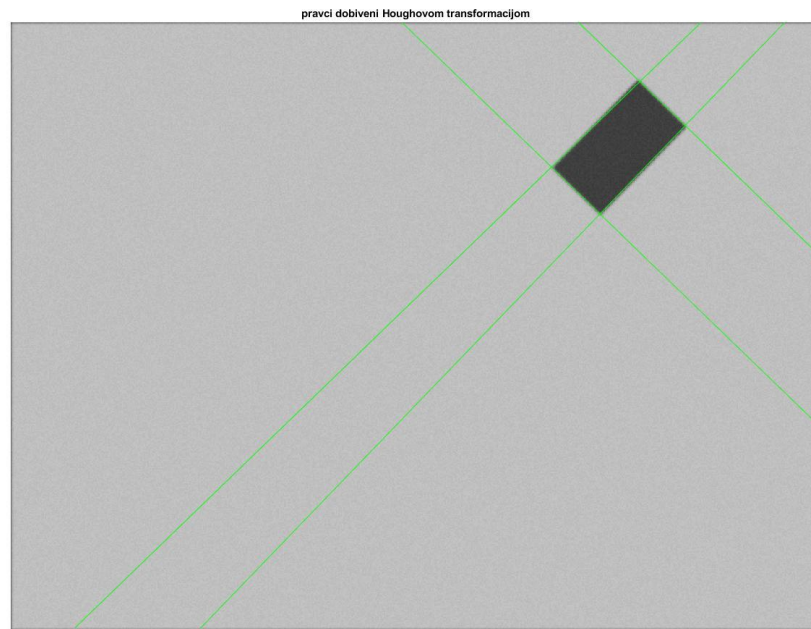


Houghova ravnina

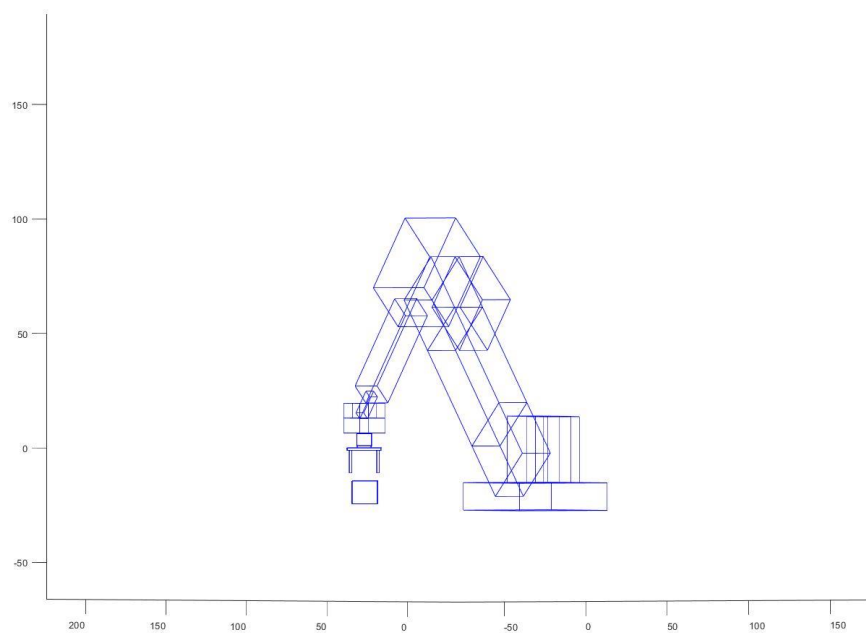


Detekcija rubova





→ Rezultat detekcije i inverzne kinematike



➔ Dodatni testovi za nasumične položaje predmeta

