深 圳 大 学 实 验 报 告

课程名称：机器人学导论

实验项目名称：Sparse Stereo and Position-Based Visual Servoing

学院：电子与信息工程学院

专业：电子信息工程

指导教师：郑琪

报告人：陈闻天 学号：2023280259

班级：04

实验时间：2024 年 10 月 29 日

实验报告提交时间：2024 年 11 月 4 日

教务处制

Aim of Experiment:

1. Learn the 3D Triangulation and Sparse Stereo
2. Learn the Position-Based Visual Servoing and use it to control the motion of robot

Experiment Content:

1. Use 3D Triangulation to finish Sparse Stereo
2. Simulate the Position-Based Visual Servoing with RVC toolbox

Experiment Process：

(1) 14.3.1 3D Triangulation

a) Estimate the fundamental matrix

b) Display the epipolar lines

c) Examine the camera focal length

d) Determine the essential matrix

e) decompose camera intrinsics matrix to determine the camera motion

f) Scale the translation unit vector

g) Compute the 3D world point locations

h) Extract last column

i) Superimpose the distance to each point on the image of the courtyard

j) Display the generated structure

(2) 15.1 Position-Based Visual Servoing

a) Define a camera with default intrinsic parameters

b) Set the world point

c) Project the world point onto image-plane

d) Estimate the pose of goal with respective to camera

e) Estimate new pose of camera after the motion of camera

f) Create a PBVS object

g) Implement the PBVS control algorithm

h) Plot the result

Data Logging and Processing:

(1) 14.3.1 3D Triangulation

a) The aim of experiment: to illustrate sparse stereo with 3D Triangulation.

b) Code and comment:

14.3.1 3D Triangulation

```
rng(0) % set random seed for reproducibility of results
% estimate the fundamental matrix
[F,in] = estimateFundamentalMatrix(matchedPts1,matchedPts2);
% display the epipolarlines
epiLines = epipolarLine(F',matchedPts2(in));
pts = lineToBorderPoints(epiLines,size(im1));
imshow(im1), hold on
line(pts(1:40,[1 3])',pts(1:40,[2 4])',color="y");
% a structure of text strings
md = imfinfo("walls-l.jpg");
%  the focal length
f = md.DigitalCamera.FocalLength
%  model camera
md.Model
%  construct the cameraIntrinsics object
flenInPix = (f/1000)/1.5e-6;
imSize = size(im1);
principalPoint = imSize/2+0.5;
camIntrinsics = cameraIntrinsics(flenInPix,principalPoint,imSize)
% The essential matrix
rng(0) % set random seed for reproducibility of results
[E,in] = estimateEssentialMatrix(matchedPts1,matchedPts2, ...
    camIntrinsics,MaxDistance=0.18,Confidence=95);
% determine the camera motion
inlierPts1 = matchedPts1(in);
inlierPts2 = matchedPts2(in);
pose = estrelpose(E,camIntrinsics,inlierPts1,inlierPts2);
pose.R
pose.Translation
% scale the translation unit vector
t = pose.Translation*0.3;
% set up camera projection matrices
tform1 = rigidtform3d;   % null tranforms by default
camMatrix1 = cameraProjection(camIntrinsics,tform1);
cameraPose = rigidtform3d(pose.R,t);
tform2 = pose2extr(cameraPose);
camMatrix2 = cameraProjection(camIntrinsics,tform2);
% the 3D world point locations
P = triangulate(inlierPts1,inlierPts2,camMatrix1,camMatrix2);
% extract their last column
z = P(1:100,3);
% superimpose the distance
circles = [inlierPts1.Location(1:100,:) repmat(15,[100 1])];
imAnnotated = insertObjectAnnotation(im1,"circle", ...
    circles,z,FontSize=50,LineWidth=4);
imshow(imAnnotated)
%The generated structure
pc = pointCloud(P);
pcshow(pcdenoise(pc),VerticalAxis="Y",VerticalAxisDir="Down");
```

(2) 15.1 Position-Based Visual Servoing

a) The aim of experiment: estimate the relationships between the relevant poses for a PBVS system

b) Code and comment:

15.1 Position-Based Visual Servoing

```
T_C = se3();  % identity pose
% define a camera
cam = CentralCamera("default",pose=T_C);
% the world points
P = mkgrid(2,0.5);
% the image-plane projections of the world points
p = cam.plot(P,objpose=se3(eye(3),[0 0 3]));
p'  % transpose for display
% estimate G
Te_C_G = cam.estpose(P,p);
printtform(Te_C_G)
T_Cd_G = se3(eye(3),[0 0 1]);
T_delta = Te_C_G*T_Cd_G.inv();
% a small move
lambda=0.05;
T_inc = interp(se3,T_delta,lambda);
printtform(T_inc)
% The new value of camera pose
cam.T = cam.T*T_inc;
% the initial pose of the camera in world coordinates
T_C0 = se3(rotmz(0.6),[1 1 -3]);
% the desired pose of the goal with respect to the camera
T_Cd_G = se3(eye(3),[0 0 1]);
% create an instance of the PBVS class
pbvs = PBVS(cam,P=P,pose0=T_C0,posef=T_Cd_G, ...
  axis=[-1 2 -1 2 -3 0.5])
% implements the PBVS control algorithm
pbvs.run(100);
% plot the simulation results
pbvs.plot_p();
pbvs.plot_vel();
pbvs.plot_camera();
```

Experimental Results and Analysis:

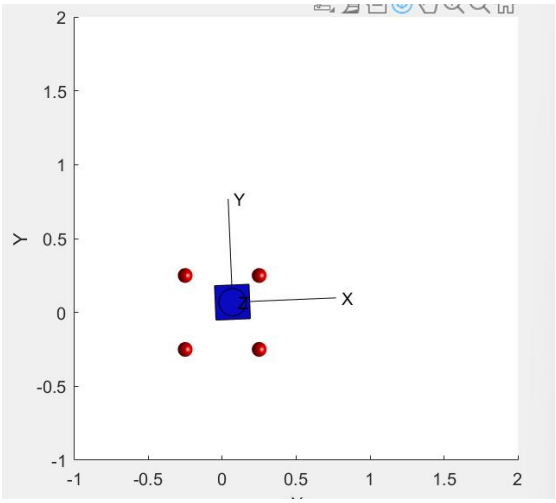
(1) 14.3.1 3D Triangulation

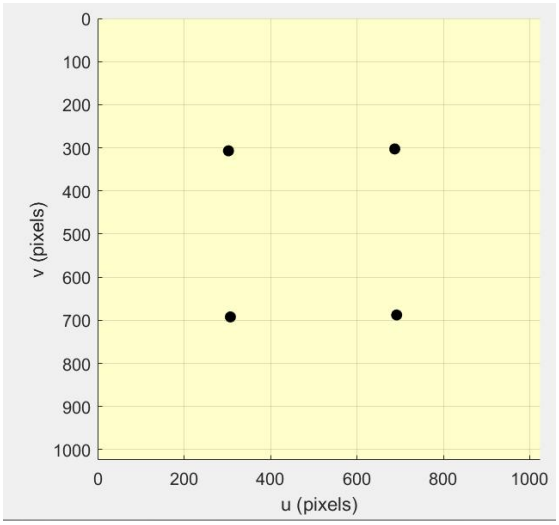
a) The outcome of experiment:

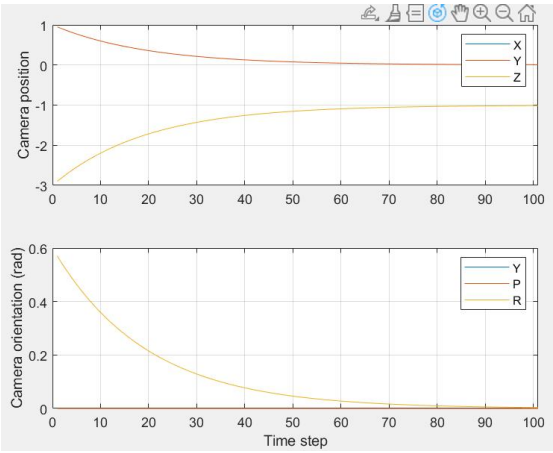
b) The analysis of experiment: successfully used information from two overlapping images to infer the 3-dimensional position of points in the world.

(2) 15.1 Position-Based Visual Servoing

a) The outcome of experiment:







b) The analysis of experiment: the camera's translation and orientation have converged smoothly on the desired values , which followed a curved path in the image

指导教师批阅意见：

成绩评定：

指导教师签字：

年　　月　　日

备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。