深 圳 大 学 实 验 报 告

课程名称：机器人学导论

实验项目名称：**Image Formation**

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 **22** 日

实验报告提交时间：**2024** 年 **10** 月 **28** 日

教务处制

Aim of Experiment:

1. Master the perspective camera model.
2. Learn to calibration of camera and calibrate camera.
3. Learn to decompose the camera matrix

Experiment Content:

1. Model a perspective camera
2. Model a perspective camera at discrete image plane
3. Use camera matrix to describe the features of camera
4. Project some points and lines
5. Calibrate camera with 3D target
6. Decompose the camera calibration matrix
7. Estimate pose with camera calibration matrix
8. Use camera calibration tool to calibrate camera

Experiment Process：

(1) 13.1.2 Modeling a Perspective Camera

a) Set a world point

b) Project this point at image-plane

c) Move camera and project again

(2) 13.1.3 Discrete Image Plane

a) Set image-plane coordinate

b) Project point at discrete image-plane

(3) 13.1.4 Camera Matrix

a) Display camera intrinsic parameter matrix

b) Display camera matrix

c) Display the field of view

(4) 13.1.5 Projecting Points

a) Create some grid points

b) Check the coordinates of the points

c) Project points at image-plane

d) Visualize camera view

e) Project line at image-plane

f) Create cube object and project it at image-plane

(5) 13.1.6 Lens Distortion

a) Set additional intrinsic parameters

(6) 13.2.1 Calibrating with a 3D Target

a) Create markers

b) Create unknown pose

c) Set random seed

d) Create a camera object

e) Estimate the camera matrix

(7) 13.2.2 Decomposing the Camera Calibration Matrix

a) Create null space

b) Convert the null space to cartesian form

c) Decompose the estimated matrix

d) Plot the calibration markers

e) Visualize the result

(8) 13.2.3 Pose Estimation with a Calibrated Camera

a) create a calibrated camera with known parameters

b) project point

c) compare the pose

(9) 13.2.4 Camera Calibration Tools

a) Launch calibration tool

b) Choose figures

c) Check the parameters of camera

Data Logging and Processing:

(1) 13.1.2 Modeling a Perspective Camera

a) The aim of experiment: model a central-perspective camera and change the pose of camera

b) Code and comment:

13.1.2 Modeling a Perspective Camera

```
%create a central perspective camera with 0.15 focal
cam = CentralCamera(focal=0.015);
% the world point
P = [0.3 0.4 3.0];
% projection
cam.project(P)
% move the camera
cam.project(P,pose=se3(eye(3),[-0.5 0 0]))
```

(2) 13.1.3 Discrete Image Plane

a) The aim of experiment: calculate the point on image-plane modeled by pixels

b) Code and comment:

13.1.3 Discrete Image Plane

```
%creata a image-plane
cam = CentralCamera(focal=0.015,pixel=10e-6, ...
      resolution=[1280 1024],center=[640 512],name="mycamera")
% nonhomogeneous image-plane coordinates
cam.project(P)
```

(3) 13.1.4 Camera Matrix

a) The aim of experiment: display the camera matrix and the field of view

b) Code and comment:

13.1.4 Camera Matrix

```
% intrinsic parameter matrix
cam.K
% camera matrix
cam.C
% th field of view
fov = cam.fov();
rad2deg(fov)
```

(4) 13.1.5 Projecting Points

a) The aim of experiment: project multiple points or lines

b) Code and comment:

13.1.5 Projecting Points

```
% create 3x3 grid of points
P = mkgrid(3,0.2,pose=se3(eye(3),[0 0 1.0]));
% display grid point X Y Z
P(1:4,:)
% The image-plane coordinates of these grid points
cam.project(P)

cam.plot(P)
% The camera pose
Tcam = se3(eul2rotm([0 0.9 0]),[-1 0 0.5]);
% oblique view of the plane
cam.plot(P,pose=Tcam)
% the vanishing point
cam.project([1 0 0 0],pose=Tcam)
% return the image-plane coordinates
p = cam.plot(P,pose=Tcam);
% oblique viewing case
p(1:4,:)
% The vertices of a cube
cube = mkcube(0.2,pose=se3(eye(3),[0 0 1]));

cam.plot(cube);
% create an edge
[X,Y,Z] = mkcube(0.2,"edge",pose=se3(eye(3),[0 0 1]));
% display
cam.mesh(X, Y, Z)

Tcam = se3(eul2rotm([0 0.8 0]),[-1 0 0.4]);
cam.mesh(X,Y,Z,pose=Tcam);
% a cube tumbling in space
theta = [0:500]/100*2*pi;
[X,Y,Z] = mkcube(0.2,[],"edge");
for th=theta
  T_cube = se3(eul2rotm(th*[1.3 1.2 1.1]),[0 0 1.5]);
  cam.mesh(X,Y,Z,objpose=T_cube); drawnow
end
```

(5) 13.1.6 Lens Distortion

a) The aim of experiment: set additional intrinsic parameters

b) Code and comment:

13.1.6 Lens Distortion

```
% Not runnable unless you define k1, k2, k3, p1, and p2
%set these parameter
k1=1.0;
k2=1.1;
k3=1.0;
p1=0.5;
p2=0.5;
cam = CentralCamera(focal=0.015,pixel=10e-6, ...
resolution=[1280 1024],center=[512 512], ...
distortion=[k1 k2 k3 p1 p2]);
```

(6) 13.2.1 Calibrating with a 3D Target

a) The aim of experiment: use 3D calibration target to estimate camera matrix

b) Code and comment:

13.2.1 Calibrating with a 3D Target

```
%The coordinates of the marker
P = mkcube(0.2);

T_unknown = se3(eul2rotm([0.3 0.2 0.1]),[0.1 0.2 1.5]);

rng(0) % set random seed for reproducibility of results
% creative a perspective camera whose parameters are unknown
cam = CentralCamera(focal=0.015,pixel=10e-6, ...
  resolution=[1280 1024],noise=0.05);
% calibration target points
p = cam.project(P,objpose=T_unknown);
% estimate the camera matrix
C = camcald(P,p)
```

(7) 13.2.2 Decomposing the Camera Calibration Matrix

a) The aim of experiment: decompose the camera parameters from the estimated camera matrix

b) Code and comment:

13.2.2 Decomposing the Camera Calibration Matrix

```
wo = null(C)'  % transpose for display
% convert to Cartesian form
h2e(wo)
% true value
T_unknown.inv.trvec
% decompose the previously estimated camera matrix
est = decomposeCam(C)

est.f/est.rho(1)
% true focal
cam.f/cam.rho(2)
% relative pose between the true and estimated camera pose
printtform(T_unknown*est.T)
% plot the calibration markers as small red sphere
hold on; plotsphere(P,0.03,"r")
plottform(eye(4,4),frame="T",color="b",length=0.3)
%estimated pose of the camera
est.plot_camera()
view([40 19])
```

(8) 13.2.3 Pose Estimation with a Calibrated Camera

a) The aim of experiment: estimate the unknown pose which transform from target to camera

b) Code and comment:

13.2.3 Pose Estimation with a Calibrated Camera

```
% create a calibrated camera with known parameters
cam = CentralCamera(focal=0.015,pixel=10e-6, ...
      resolution=[1280 1024],center=[640 512]);
% the coordinates of the marker
P = mkcube(0.2);

T_unknown = se3(eul2rotm([0.3 0.2 0.1]),[0.1 0.2 1.5]);
T_unknown.trvec
rad2deg(rotm2eul(T_unknown.rotm))
% image-plane coordinates of the object's points
p = cam.project(P,objpose=T_unknown);
% estimate the relative pose
T_est = cam.estpose(P,p);
T_est.trvec
rad2deg(rotm2eul(T_est.rotm))
```

(9) 13.2.4 Camera Calibration Tools

a) The aim of experiment: use camera calibration tool to complete camera calibration

b) Code and comment:

13.2.4 Camera Calibration Tools

```
% launch
cameraCalibrator

% Not runnable unless camera calibration is complete
% The calibration parameters
cameraParams
cameraParams.Intrinsics
cameraParams.Intrinsics.K
% app for stereo calibration
stereoCameraCalibrator
```

Experimental Results and Analysis:

(1) 13.1.2 Modeling a Perspective Camera

a) The outcome of experiment:

```
ans =

    0.0015    0.0020


ans =

    0.0040    0.0020
```

b) The analysis of experiment: the first answer is the point on the image plane, the second answer is the point on the image plane with camera moving

(2) 13.1.3 Discrete Image Plane

a) The outcome of experiment:

```
name: mycamera [central-perspective]
  focal length:   0.015
  pixel size:     (1e-05, 1e-05)
  principal pt:   (640, 512)
  number pixels:  1280 x 1024
  pose:           t = (0, 0, 0), RPY/zyx = (0, 0, 0) rad


ans =

   790   712
```

b) The analysis of experiment: from the result, the point on image-plane is (790,712)

(3) 13.1.4 Camera Matrix

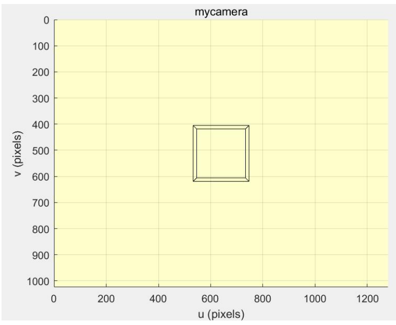
a) The outcome of experiment:

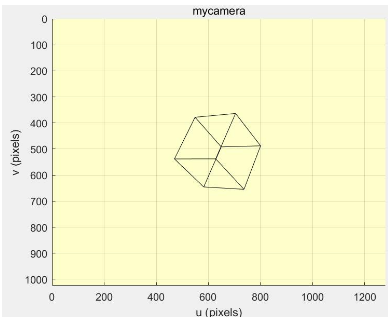
b) The analysis of experiment: the camera matrix is related the parameter we set before

```
ans =

   1.0e+03 *

    1.5000         0    0.6400
         0    1.5000    0.5120
         0         0    0.0010

ans =

   1.0e+03 *

    1.5000         0    0.6400         0
         0    1.5000    0.5120         0
         0         0    0.0010         0

ans =

   46.2127   37.6930
```

(4) 13.1.5 Projecting Points

a) The outcome of experiment:





b) The analysis of experiment: compared frontal view and oblique view, the position of these points on image-plane is different and some points from oblique view are not at the central of pixels

(5) 13.1.6 Lens Distortion

a) The outcome of experiment:

```
cam =

name: noname [central-perspective]
  focal length:   0.015
  distortion:     k=(1, 1.1, 1), p=(0.5, 0.5)
  pixel size:     (1e-05, 1e-05)
  principal pt:   (512, 512)
  number pixels:  1280 x 1024
  pose:           t = (0, 0, 0), RPY/zyx = (0, 0, 0) rad
```

b) The analysis of experiment: successfully to set additional intrinsic parameters.

(6) 13.2.1 Calibrating with a 3D Target

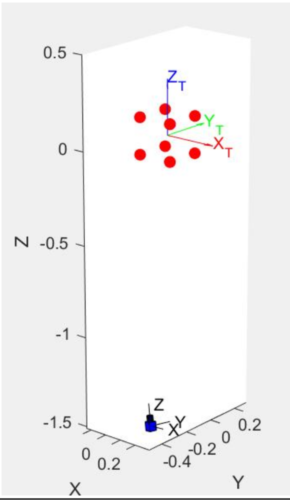
a) The outcome of experiment:

```
C =

  853.0895 -236.9378  634.2785  740.0438
  222.6439  986.6900  295.7327  712.0152
   -0.1304    0.0610    0.6495    1.0000
```

b) The analysis of experiment: successfully estimate camera matrix

(7) 13.2.2 Decomposing the Camera Calibration Matrix

a) The outcome of experiment:



b) The analysis of experiment: the outcome is close to the true value but is difficult to equal to the true value because the noisy

(8) 13.2.3 Pose Estimation with a Calibrated Camera

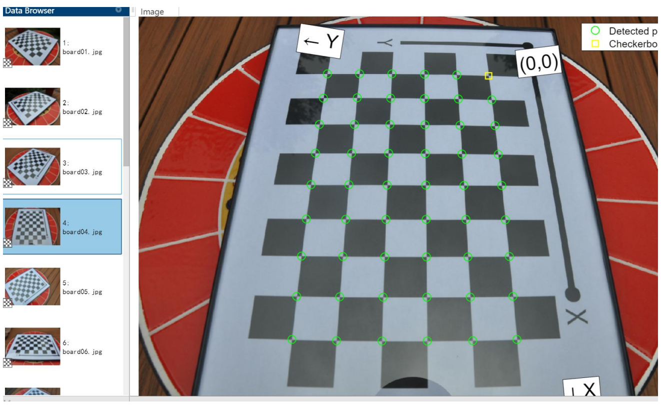
a) The outcome of experiment:

```
ans =

   17.1887   11.4592    5.7296


ans =

    0.1000    0.2000    1.5000


ans =

   17.1887   11.4592    5.7296
```

b) The analysis of experiment: compared the first answer and the third answer, the estimated pose is same as the required unknown pose

(9) 13.2.4 Camera Calibration Tools

a) The outcome of experiment:



b) The analysis of experiment: successfully seize the key points and use them to calibrate camera

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。