Tutorial question and answer 1 (updated 2021.2.19) A discussion is added at the end of this document.

A small fly is at position [1.5,1.2,4]T meters in the world coordinate system. A camera has focal length F=3mm, (ox,oy)= (320, 240). A pixel is a square, the width sx=sy=3um.

1. A camera is the world coordinate system (the camera and world coordinates are the same). Calculate the image pixel coordinates (u1,v1) of the fly.
2. The camera is moved to a new position by a rotation of Rcam2, the rotation is performed by rotating around the x-axis first by 5 degrees, the y-axis by 8 degrees, finally z-axis by 7 degrees.

Translation Tcam2 is [0.3,0.4,0.5] T meters. Calculate again the image pixel coordinates (u2,v2) of the fly.

Answer1a:

Since the camera and world coordinates are the same, so then, ,T

Note for the lecture notes

refer to slide 53, select the top 3 rows

Apply to here

Hence

, note: 1 is added to the last row of *pw* to make it a homogenous coordinate

s is an arbitrary scaling factor

=

Hence s=4

Answer (a):

u1==2780/4=695

v1==2160/4=540

Answer1b:

Rotating around the x-axis first by 5 degrees, then around y-axis by 8 degrees, finally around the z-axis by 7 degrees.

=, since for rotation matrix R=R-1=RT and

=

Given

=

=

=

It is shown earlier that Rc2=Rcx\*Rcy\*Rcz

Given Tc2=[0.3,0.4,0.5]T

Hence

answer (b):

u2 = 540.6764

v2 = 509.3849

%MATLAB CODE

clear

clc

format short

%%%%%%%% part (a)%%%%%%%%%%%%%%%%%%%%555

mint=[1000 0 320

 0 1000 240

 0 0 1]

mext=[1 0 0 0

 0 1 0 0

 0 0 1 0]

pw=[1.5, 1.2, 4,1]'

img=mint\*mext\*pw

u=img(1)/img(3)

v=img(2)/img(3)

%%%%%%%%%%%%%%%% part (b) %%%%%%%%%%%%%%%%%%%

Tc2=[0.3,0.4,0.5]'

an\_x=5\*pi/180;

an\_y=8\*pi/180;

an\_z=7\*pi/180;

Rcz=[cos(an\_z) sin(an\_z) 0

 -sin(an\_z) cos(an\_z) 0

 0 0 1]

Rcy=[cos(an\_y) 0 -sin(an\_y)

 0 1 0

 sin(an\_y) 0 cos(an\_y) ]

Rcx=[1 0 0

 0 cos(an\_x) sin(an\_x)

 0 -sin(an\_x) cos(an\_x)] % bug fixed, changes: red marked

Rc2=Rcx\*Rcy\*Rcz

mext2=[Rc2 , Rc2\*(-Tc2)]

img2=mint\*mext2\*pw

u2=img2(1)/img2(3)

v2=img2(2)/img2(3) %bug fixed

Discussion:

Question: When I read the tutorial samples about camera model provided on course web page, I have met some confusion.

In this tutorial and the code shown above, the point P in world coordinate is in meter, and it’s directly applied to the final calculation.

But in “Matlab Demo and Exercise.pptx”, P in world coordinate is also in meter, but it’s firstly divided by the pixel length (in page3, “ X=[0.2,0.3,3.0]'/pwm”)

I’m really confused about that. Could you please give me a more detailed explanation?

Answer:

When calculating the homogeneous coordinates  of the pixels, you use the formula [su,sv,s]', where

[su,sv,s]'=M\_intrinsic\*M\_extrinsic \*P=       project\_matrix\*P

size 3x1      3x3            3x4                   4x1        4x4              4x1

and P=[X,Y,Z]'

su depends on X

sv depends on Y

s depends on Z

In the end, u=su/s=some\_terms \* X/Z

and, v=su/s=some\_terms \* Y/Z

The essential idea is 'u' depends on the ratio of X/Z, so no matter you use pixel or meters , the result will be the same.

But in practice, I understand that some rounding error map occurs, because pwm (pixel width in meters) is very small.

Conclusion: If you want to find the 2D projected pixel location of a 3D object point [X,Y,Z] , presenting [X,Y,Z] in  pixel or meter in the calculation will generate the same result.