第二次实验手册——坐标变换与仿射变换

实验一 坐标变换

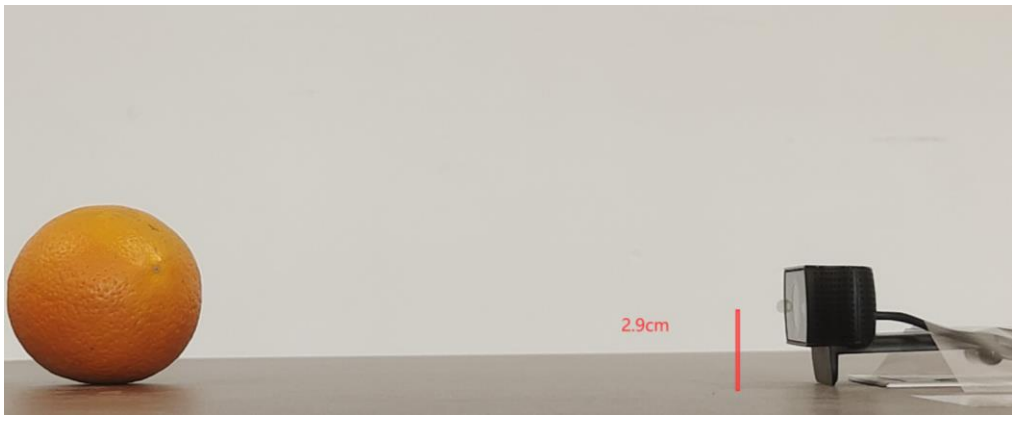
一、实验内容：

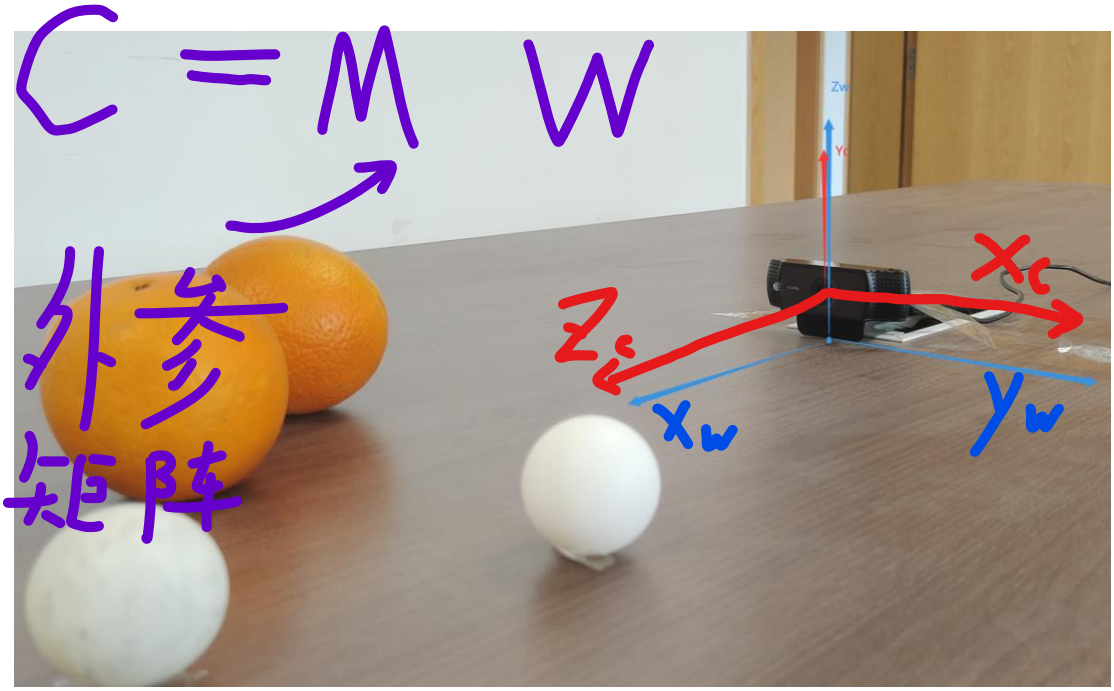
给定一张图片，求图片中所有圆形物体对应的在世界坐标系之间的距离。

二、相关细节：

1、拍摄场景：

世界坐标系 Zw 轴在相机光心正下方，并垂直于桌面，Xw 轴和 Yw 轴与相机坐标系 Zc、Xc 同向，具体可自行定义，相机光心距离桌面垂直 2.9cm。





2、已知数据

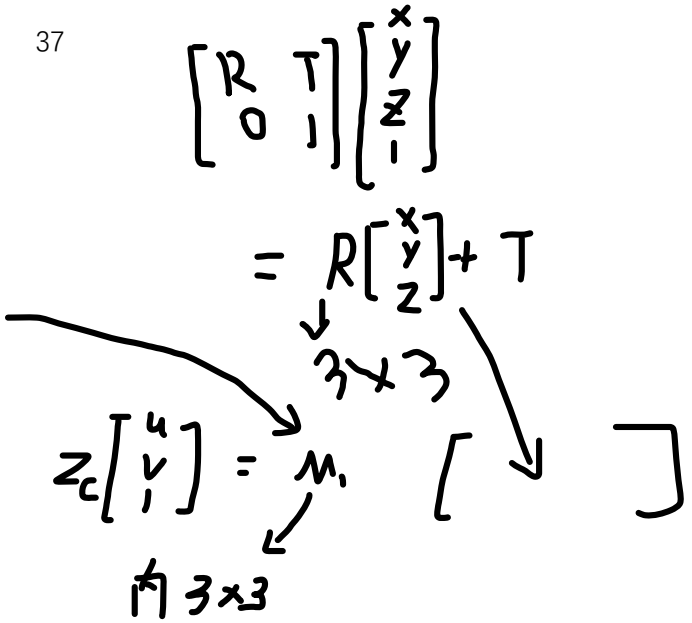
1）Image 文件夹中 9 张图片依次对应的 Zc(原点距小球中心的水平距离，单位 cm)如下：

橙子：　37.5　64
橙子：　39.5　84
橙子：　67　84
乒乓球：27　35.5
乒乓球：21　27
乒乓球：26　37
乒乓球：38　　橙子：37　46
乒乓球：33.5　46　橙子：37
乒乓球：33.5　46　橙子：29　37

2）相机内参如下：

```
I =

  650.1821         0   315.8990
         0  650.5969   240.3104
         0         0     1.0000
```



三、实验步骤

1、 求解外参矩阵

2、 读入图片及 Zc 和相机内参

3、 彩色图片转灰度图，判断目标小球相对背景的是明是暗，判断目标小球的大致半径范围

4、 识别图片小球的中心位置

（1）

```
[circle,radii] = imfindcircles(img,[40 200],'ObjectPolarity','dark','Sensitivity',0.97)
```

imfindcircles()：寻找图像上圆的函数

返回值：在图像上找到的圆的中心和半径[centers, radii]

参数：

1）img 是导入的图像

2）[40 200]是圆的半径范围

3）ObjectPolarity 是设置找到图像内相对于背景色更亮或者更暗的圆，bright 或 dark

4）Sensitivity 是敏感因子，默认设置为 0.85，其范围在[0,1]之间，随着灵敏度系数的增加，imfindcircles 会检测到更多的圆形对象，包括弱圆形和部分遮盖的圆形。 较高的灵敏度值也会增加错误检测的风险。

（2）

```
viscircles(circle,radii,'Color','b');
```

viscircles()：画图像上圆的轮廓，需要输入通过上步得到的两个参数 centers, radii，另外可以设置轮廓的颜色和线条样式，例如：'Color','b'，'LineStyle'等。

5、转齐次坐标并求取小球中心在真实世界中的坐标

inv(A)：矩阵求逆

6、距离计算

norm(A)：求向量范数

实验二 仿射变换

仿射变换（Affine Transformation）其实是另外两种简单变换的叠加：一个是线性变换，一个是平移变换，本次实验主要让大家初步了解仿射变换在图像处理中的效果

仿射变换变化包括缩放(Scale)、平移(transform)、旋转(rotate)、反射(reflection)、错切(shear mapping)，原来的直线仿射变换后还是直线，原来的平行线经过仿射变换之后还是平行线，这就是仿射。

仿射变换的公式

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{00} & R_{01} & T_x \\ R_{10} & R_{11} & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

其中 r 为线性变换相关部分，t 为平移变换相关

任务一：根据下图公式，自行选择一张图片，分别做出缩放，平移，旋转，反射，错切的效果，并写出对应效果的矩阵

基本坐标变换——变换矩阵总结

$$\begin{matrix} A & B & C \\ D & E & F \\ G & H & I \end{matrix}$$

A、E 控制缩放
C、F控制平移
B、D控制错切

平移变换 translation	旋转变换 ratotion(逆时)	缩放变换 scaling =尺度变换	错切变换 shearing =倾斜、剪切、偏移
$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\alpha & sin\alpha & 0 \\ -sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a_{12} & 0 \\ a_{21} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

实验步骤

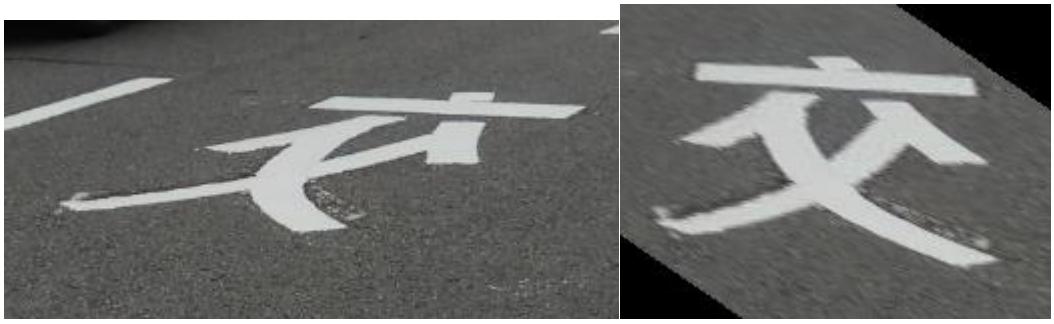
1）自定义一个仿射矩阵

2）利用 maketform()函数将自定义的矩阵转换为仿射变换用的矩阵 T

3）利用 imwrap()函数将原图转换为经过矩阵 T 仿射后的图片

任务二：构建一个合适的仿射矩阵，将任务图矫正，写出构建的过程矩阵

示例：



任务图：

（说明这次变换要用到什么效果，并且写出对应矩阵）

本次实验相关函数：

T = Projective2d(array)

T = Maketform(array)

以上两个函数作用相同，返回一个与输入矩阵相同的用于仿射变换的数据结构

Img2 = imwrap(img1,T)

输入原图 img1 与仿射变换结构 T，输出变换后的图片

```
H_e=projective2d([1 2.2 0;
                  0 1.5 0;
                  0  0  1]');
newimg=imwarp(img,H_e);
figure,imshow(newimg);
```

T = Maketform('affine',input,output)

该函数输出 input->output 的仿射矩阵

其中 input 与 output 皆为有三个点坐标数据的数组（点一一对应，即 I1->O1）

例如：input = [1 1 ; 2 3 ; 4 5]，其中的点为图片的像素坐标

(*选做)任务三:

利用仿射变换做出如下效果，并写出过程矩阵

若效果不佳，请说明原因

相关知识:

图像融合:

F α B C

Image Composite (合成)：$C = \alpha F + (1-\alpha)B$

%Alpha 通道

[I,map,Alpha] = imread('b1.png');

注意，合成时 F 与 B 的图像 size 要一致才可以融合