

1 实验目的

使用Matlab完成图像坐标变换和仿射变换，加深对课堂知识的理解。

2 实验一 坐标变换

求出给定的九张图片中所有圆形物体对应的在世界坐标系之间的距离。

2.1 实验数据

(1) 已知九张图片中每个圆形物体的中心距相机原点的水平距离（Zc，单位cm）。

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：

$$I = \begin{bmatrix} 650.1821 & 0 & 315.8990 \\ 0 & 650.5969 & 240.3104 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

(1)

2.2 实验过程

(1) 求解外参矩阵

由相机外参矩阵的定义可知

$$M_2 = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 2.9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2)

(2) 读入图片及Zc和相机内参

```
Img = cell(9);
for i = 1:9
    img = imread(['./affine/image/', num2str(i), '.jpg']);
    Img{i} = img;
end

I = [650.1821 0 315.8990; 0 650.5969 240.3104; 0 0 1.0000];
R = [0 1 0; 0 0 1; 1 0 0];
T = [0;-2.9;0];
```

```
Zc = {[37.5 64];
      [84 39.5];
      [84 67];
      [35.5 27];
      [27 21];
      [27 26];
      [38 46 37];
      [33.5 46 37];
      [33.5 46 37 29];
};
```

(3) 彩色图片转换成灰度图，并识别图中圆形物体，并识别中心位置

这一步我使用 `imfindcircles()` 接口函数，并且手动调整参数，识别出了每张图片中的圆形物体。由于我是手动调整参数，所以我已经试出了每张图片中圆形物体的半径范围和相对背景明暗，我将这些信息存放在数组中，后续循环识别时使用。

识别出圆形物体后使用 `viscircles()` 接口在图片上画出圆的轮廓。

% 存放暗色圆识别时的敏感度

```
SenDark = [0.95, 0.95, 0.95, 0.95, 0.95, 0.95, 0.95, 0.97, 0.97];
```

% 存放暗色圆半径范围的左值

```
RadRangeL = [30, 30, 30, 30, 30, 30, 30, 60, 60];
```

% 存放暗色圆半径范围的右值

```
RadRangeR = [90, 90, 35, 35, 35, 35, 200, 100, 200];
```

```
figure;
```

```
for i = 1:9
```

```
    img = Img{i};
```

% 转换为灰度图

```
    gray_img = rgb2gray(img);
```

% 识别亮色圆

```
    [bright_circle, bright_radii] = imfindcircles(gray_img, [30 200],
'ObjectPolarity','bright','Sensitivity', 0.94);
```

% 识别暗色圆

```
    [dark_circle, dark_radii] = imfindcircles(gray_img, [RadRangeL(i) RadRangeR(i)],
'ObjectPolarity','dark','Sensitivity', SenDark(i));
```

% 将圆合并

```
    circle = [dark_circle;bright_circle];
```

```
    radii = [dark_radii;bright_radii];
```

% 画成子图

```
    subplot(3,3,i),imshow(gray_img);
```

```
    viscircles(circle, radii, 'EdgeColor', 'r');
```

```
end
```

输出结果如下：

由输出结果可见，使用 `imfindcircles()` 接口识别出圆形物体效果很好，可以进行接下来的实验。

(4) 转齐次坐标并求取小球中心在真实世界中的坐标

该步骤的原理公式如下：

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = I * \left(R * \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \right) \quad (3)$$

其中R是世界坐标系到相机坐标系的旋转变换矩阵，T是世界坐标系原点到相机坐标系原点的位移向量，这里实际上是将外参 M_2 拆成了R和T两部分， $\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$ 是要求的世界坐标系下的坐标。

```
for i = 1:9
    disp(['第',num2str(i),'张图片: ']);

    img = Img{i};
    % 转换为灰度图
    gray_img = rgb2gray(img);
    % 识别亮色圆
    [bright_circle, bright_radai] = imfindcircles(gray_img, [30 200],
    'ObjectPolarity','bright','Sensitivity', 0.94);
    % 识别暗色圆
    [dark_circle, dark_radai] = imfindcircles(gray_img, [RadRangeL(i) RadRangeR(i)],
    'ObjectPolarity','dark','Sensitivity', SenDark(i));
    % 将圆合并
    circle = [dark_circle;bright_circle];
    radai = [dark_radai;bright_radai];

    disp("识别到的所有圆形物体的像素坐标");
    disp(circle)

    % 存放圆形物体的真实坐标
    coordinate = [];
    for n = 1:length(circle)
        % 计算圆形物体的相机坐标
        carmera = [circle(n, 1); circle(n, 2); 1]*Zc{i}(n);
        carmera = carmera - I*T;
        world = inv(I*R) * carmera;
        % 将圆的信息加入到数组中
        coordinate = [coordinate, world];
    end

    % 计算圆形物体在世界坐标系下的真实距离
    disp("圆形物体两两之间求距离");
    for p = 1:(size(coordinate, 2) - 1)
        for q = (p+1):size(coordinate, 2)
            disp(norm(coordinate(:,p)- coordinate(:,q)));
        end
    end
end
end
```

输出结果为:

第1张图片： 识别到的所有圆形物体的像素坐标		第7张图片： 识别到的所有圆形物体的像素坐标	
275.6738 224.2138 414.2729 231.3589		175.9810 248.4925 511.2541 230.3340 370.7043 228.7527	
圆形物体两两之间求距离		圆形物体两两之间求距离	
29.0918		23.4381	
		11.3972	
		13.9837	
第2张图片： 识别到的所有圆形物体的像素坐标		第8张图片： 识别到的所有圆形物体的像素坐标	
242.0206 231.3112 100.4994 220.4103		59.1625 250.7434 166.9292 247.0930 377.1216 226.8002	
圆形物体两两之间求距离		圆形物体两两之间求距离	
44.6407		12.7860	
		17.1245	
		16.7098	
第3张图片： 识别到的所有圆形物体的像素坐标		第9张图片： 识别到的所有圆形物体的像素坐标	
240.3739 230.5498 399.4435 234.0230		53.9350 251.0143 157.9260 246.0000 356.6929 228.9427 523.0144 217.4086	
圆形物体两两之间求距离		圆形物体两两之间求距离	
25.0341		12.7145	
		16.2457	
		23.2297	
		16.2571	
		26.6040	
		10.5820	

这便是题目要求的所有距离。

2.3 实验一结果分析

因为事先并不知道 Z_c 与图中各圆的对应关系，我根据识别出的圆形物体的半径，假定橙子的大小都相同，乒乓球的大小都相同，结合透视学中近大远小的公理，手动修正了输入的 Z_c ，得到了最后的输出结果，但我并不能保证输出结果的正确性，这只是我在已知信息下得到的最优答案。

3 实验二 仿射变换

3.1 相关背景知识

仿射变换是线性变换和平移变换的叠加，仿射变换包括缩放(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。

仿射变换的公式如下：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{00} & R_{01} & T_x \\ R_{10} & R_{11} & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

具体的变换公式如下：

$$\text{平移变换: } \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\text{旋转变换: } \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\text{缩放变换: } \begin{bmatrix} x'' \\ y' \\ w'' \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$\text{错切变换: } \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a_{12} & 0 \\ a_{21} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3.2 实验过程

定义仿射矩阵后，需要使用 `projective2d()` 函数让3x3矩阵转换成对应输入图片的尺寸，之后使用 `imwarp` 函数对图片进行变换。

% 读取图片

```
img = imread('./affine/ScissorSeven.jpg');
```

% 显示原始图片

```
figure
subplot(2,3,1), imshow(img);
title('Original Image');
```

% 缩放效果及对应矩阵

```
scaling_factor = 0.2;
scaling_matrix = [scaling_factor 0 0; 0 scaling_factor 0; 0 0 1];
T = projective2d(scaling_matrix);
scaled_img = imwarp(img, T);
subplot(2,3,2), imshow(scaled_img);
title('Scaled Image');
```

% 平移效果及对应矩阵

```
translation_matrix = [1 0 0; 0 1 0; 50 100 1];
T = projective2d(translation_matrix);
%translated_img = imwarp(img, T);
translated_img = imtranslate(img, [50, 100]);
subplot(2,3,3), imshow(translated_img);
title('Translated Image');
```

% 旋转效果及对应矩阵

```
rotation_angle = 30; % 顺时针旋转30度
rotation_matrix = [cosd(rotation_angle) -sind(rotation_angle) 0; sind(rotation_angle)
cosd(rotation_angle) 0; 0 0 1];
T = projective2d(rotation_matrix);
```

```
rotated_img = imwarp(img, T);
subplot(2,3,4), imshow(rotated_img);
title('Rotated Image');
```

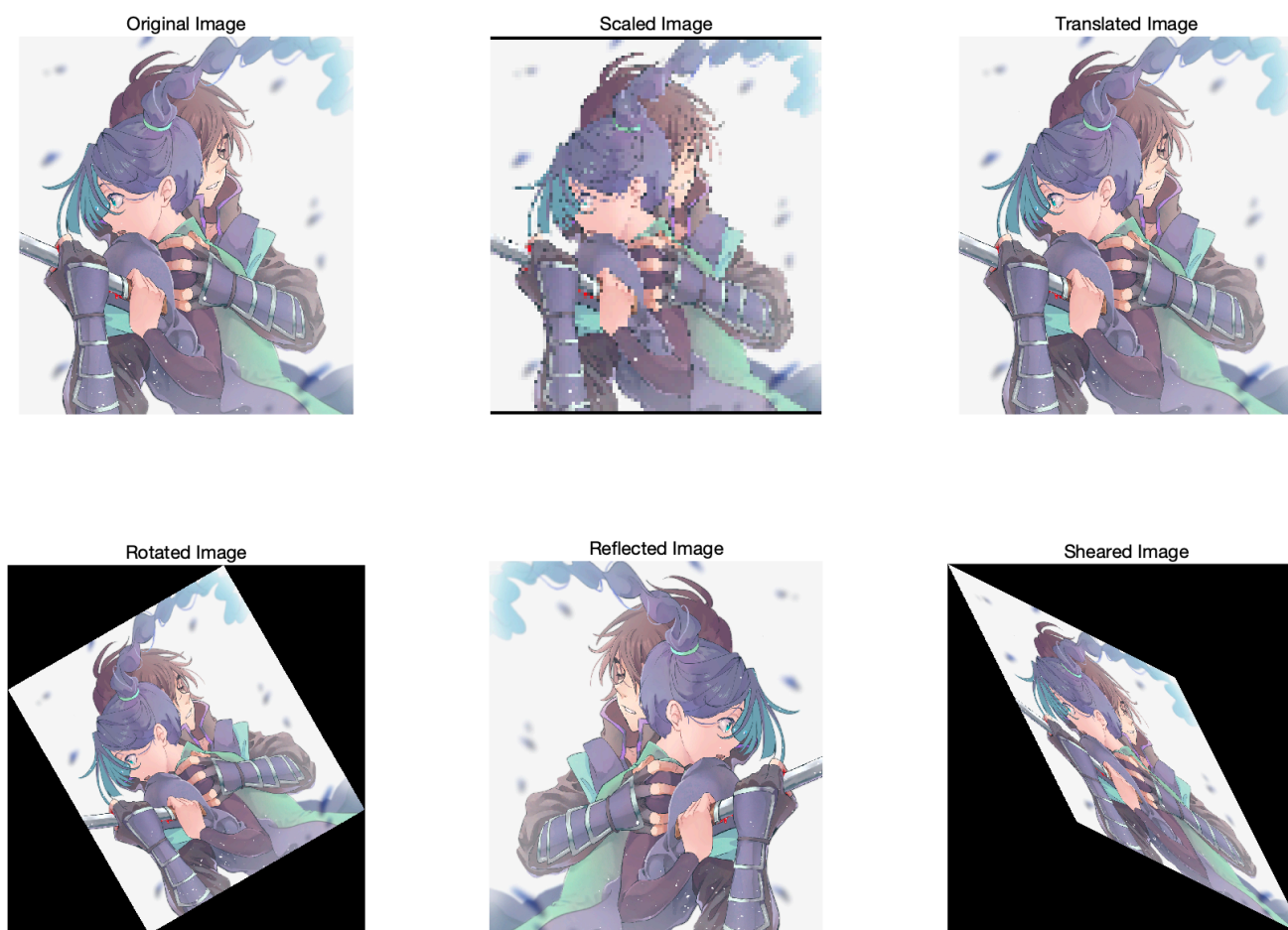
% 反射效果及对应矩阵

```
reflection_matrix = [-1 0 0; 0 1 0; 0 0 1];
T = projective2d(reflection_matrix);
reflected_img = imwarp(img, T);
subplot(2,3,5), imshow(reflected_img);
title('Reflected Image');
```

% 错切效果及对应矩阵

```
shear_factor = 0.5;
shear_matrix = [1 shear_factor 0; shear_factor 1 0; 0 0 1];
T = projective2d(shear_matrix);
sheared_img = imwarp(img, T);
subplot(2,3,6), imshow(sheared_img);
title('Sheared Image');
```

输出结果如下：



结果分析：

从输出结果中不难发现，旋转、反射、错切操作都完美实现了，但是缩放和平移似乎没有达到预期的效果，这是因为在使用 `imwarp()` 接口时，图像所处坐标系和 `imwarp` 坐标系都发生了变化，所以难以看出缩放和平移。

但从输出结果可以看出，缩放的图片相较于原始图片明显更为模糊，这相当于图像在相同尺寸的情况下，包含的像素变少了，确实达到了缩放的效果。

3.3 仿射变换实战任务

题目中要求对如下任务图进行仿射变换，达到矫正的效果。

实现代码如下：

% 读取图片

```
img1 = imread('./affine/上善若水.jpg');
```

% 显示原始图片

```
figure  
subplot(1,2,1), imshow(img1);
```

% 仿射变换

```
affine_matrix = [cosd(-65) -sind(-65) 0; sind(-65) cosd(-65) 0; 0 0 1]*[1 -0.4 0; 0 1 0; 0 0 1]*  
[cosd(92) -sind(92) 0; sind(92) cosd(92) 0; 0 0 1];  
disp(affine_matrix);  
T = projective2d(affine_matrix);  
img1 = imwarp(img1, T);  
subplot(1,2, 2), imshow(img1);
```

仿射矩阵包含三部分，换句话说，总共进行了旋转、错切、旋转共三次仿射变换，将三个矩阵相乘就是最终的仿射矩阵，三次变换如下：

$$RotationMatrix1 = \begin{bmatrix} 0.4226 & 0.9063 & 0 \\ -0.9063 & 0.4226 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$ShearMatrix = \begin{bmatrix} 1 & -0.4 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$RotationMatrix2 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

最终达到的效果与原图的对比如下：



结果分析:

从输出结果来看，任务图中最主要的两个字“上”和“善”矫正的效果比较好，但另外两个字“若”和“水”由于透视角度的关系，需要用透视矫正进行进一步的调整

3.4 选做任务

使用仿射变换完成图像合成，并写出过程矩阵

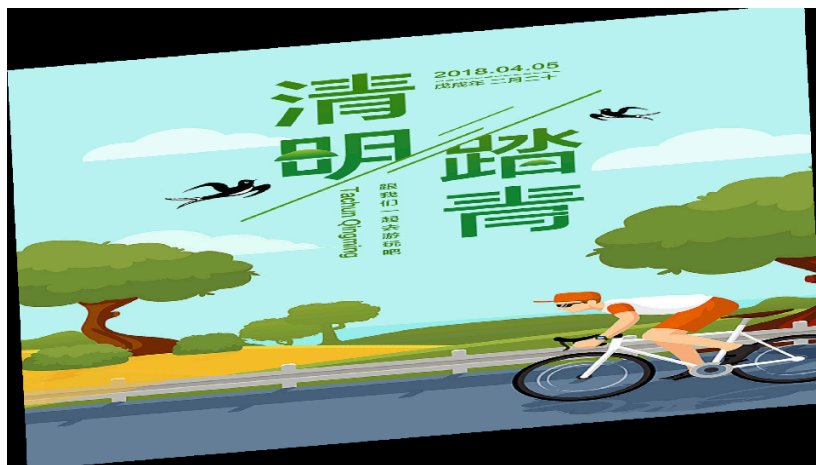


针对该任务，我的想法是先将海报进行仿射变换，使其变成适合电视屏幕的尺寸，之后在通过图像融合来让二者结合在一起。

我通过尝试，找到了最“适合”的仿射变换，代码如下：

```
board = imread("./affine/affine3_2.jpg")
shear_matrix = [6 -0.5 0; 0.1 2 0; 0 0 1];
T = projective2d(shear_matrix);
board = imwarp(board, T);
imwrite(board, "处理后.jpg");
figure;
imshow(board);
```

输出结果如下：



下一步，我将处理后海报的黑色背景去除，使其没有填充，并将其调整至与电视机图片相同的尺寸，下面是图像融合的代码：

```
img2 = imread('./affine/处理后.png');

R = double(img2(:,:,1));
G = double(img2(:,:,2));
B = double(img2(:,:,3));
```



```

%Alpha通道
[I,map,Alpha] = imread('./affine/处理后.png');

background = imread('./affine/电视2.png');

R1 = double(background(:,:,1));
G1 = double(background(:,:,2));
B1 = double(background(:,:,3));

%计算参数
a = double(Alpha)/255.00;

%三通道合成
img3(:,:,1) = uint8(R .* a + (1-a) .* R1);
img3(:,:,2) = uint8(G .* a + (1-a) .* G1);
img3(:,:,3) = uint8(B .* a + (1-a) .* B1);

figure
imshow(img3),title('合成的图片');
imwrite(img3,'combine.png')

```

输出结果分析：

从输出结果来看，虽然完成了图像融合的任务，但海报没有完全与电视机结合，这是因为电视机在图像中呈现的状态是由于透视而造成畸变的，所以需要对海报进行透视变换，一般的仿射变换的一条性质是只能将平行直线映射为平行直线，所以无法使图片变成“梯形”，所以最终融合的效果不是很好。

3.5 选做任务的扩展

我在网上查阅资料^[1]后，找到了一种比仿射变换进行图像融合更优的方法，就是将图像变成梯形来实现透视变换，这种方法的实现原理是创建一个转换对象，该对象采用图像的四个角点并将它们放置到目标梯形的相应角落。找到此变换后，使用此变换对象扭曲图像，确保指定坐标系，使原点相对于原始图像的左上角。一旦找到这个扭曲的图像，我们就会在相对于原始图像坐标系的图像帧上显示它。代码实现如下：

```

post = imread("./affine/affine3_2.jpg");
post = rgb2gray(post);
post1 = post;

% 获取海报大小
[m,n] = size(post);
% 使用零扩展和填充矩阵
post = padarray(post,[1866-m,2800-n],0,'post');

TV = imread("./affine/TV.jpg");
TV = rgb2gray(TV);

% 创建将原始图像坐标扭曲为梯形的透视变换
movingPoints = [1 1; size(post1,2) 1; 1 size(post1,1); size(post1,2) size(post1,1)];

fixedPoints = [0.4855    0.0823;2.2195    0.2682;0.5185    1.5433;2.1805    1.2822]*1000;

tform = fitgeotrans(movingPoints, fixedPoints, 'Projective');

```

% 创建一个参考坐标系，其中范围是图像的大小

```
RA = imref2d([size(post,1) size(post,2)], [1 size(post,2)], [1 size(post,1)]);
```

% 变换图像

```
[out,r] = imwarp(post, tform, 'OutputView', RA);
```

% 显示图像并将轴关闭

```
res = out+TV;
```

```
figure;
```

```
imshow(res);
```

```
axis off;
```

输出结果如下：



其中的拉伸参数是通过对电视机的四个边角上的像素采样得到的。这种变换和融合方法相较于之前介绍的方法完美复现题目的要求。

4 引用

[1] 选做任务参考链接：<https://qa.1r1g.com/sf/ask/2271343721/#>