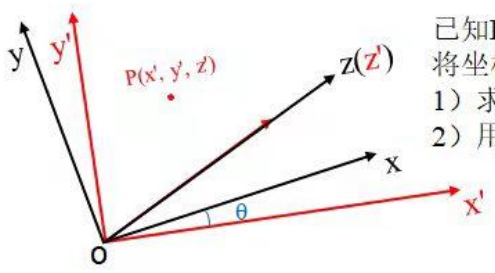
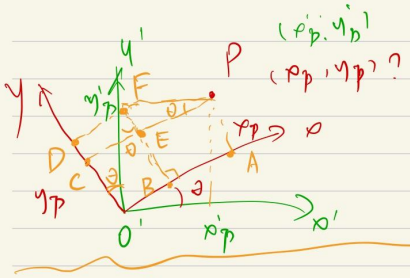
题目一:

做题时长：10 分钟（请写上时长）



已知P点在ox'y'z'坐标系中的坐标位置是(x', y', z')，
将坐标系沿着ox'y'平面旋转θ角：
1）求P点在旋转后的坐标系oxyz中的位置；
2）用矩阵表示P点在不同坐标系下的旋转关系



如上图 $x_p = O'B + BA$

$y_p = O'D - CD$

又 $O'B = O'F \cdot \sin\theta = y'_p \sin\theta$

$BA = EP = FP \cdot \cos\theta = x'_p \cos\theta$

$\therefore x_p = y'_p \sin\theta + x'_p \cos\theta$ ①

又 $O'D = O'F \cos\theta = y'_p \cos\theta$

$CD = EF = O'F \sin\theta = x'_p \sin\theta$

$\therefore y_p = y'_p \cos\theta - x'_p \sin\theta$ ②

综合①②，旋转后P点坐标为

$$\begin{cases} x_p = y'_p \sin\theta + x'_p \cos\theta \\ y_p = y'_p \cos\theta - x'_p \sin\theta \\ z_p = z'_p \end{cases}$$

题目二：用你熟悉的语言，实现上述两张图的特征匹配与图像拼接。

做题时长：45 分钟（请写上时长）

- 主要思想：

1. 分别输出两张图的特征点
2. 利用 KNN 的思想，对图 2 每个特征点 x，分别计算其与图 1 所有特征点的距离（相似度），然后根据结果进行排序，选出 K 个最相似的图 1 的特征点，将图 2 那个特征点 x 归类为这个 K 个图 1 特征点最多的类别标记；

- 代码如下：

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline

pic1 = cv2.imread('pic1.png')
pic2 = cv2.imread('pic2.png')
pic1_gray = cv2.cvtColor(pic1, cv2.IMREAD_GRAYSCALE)
pic2_gray = cv2.cvtColor(pic2, cv2.IMREAD_GRAYSCALE)
# test
# plt.imshow(pic1)
# plt.imshow(pic2)
```

```
orb = cv2.ORB_create()
pic1_key, pic1_desc = orb.detectAndCompute(pic1_gray, None)
pic2_key, pic2_desc = orb.detectAndCompute(pic2_gray, None)

pic1_keypoint = cv2.drawKeypoints(image=pic1, outImage=pic1, keypoints=pic1_key, flags=4, color=(0,0,0))
# plt.imshow(pic1_keypoint)

pic2_keypoint = cv2.drawKeypoints(image=pic2, outImage=pic2, keypoints=pic2_key, flags=4, color=(0,0,0))
# plt.imshow(pic2_keypoint)

bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
matches = bf.knnMatch(pic1_desc, pic2_desc, k=1)
match_result = cv2.drawMatchesKnn(pic1, pic1_key, pic2, pic2_key, matches[:30], pic2, flags=2)
plt.figure(figsize=(16, 9))
plt.imshow(match_result)
```

- **图像拼接结果**