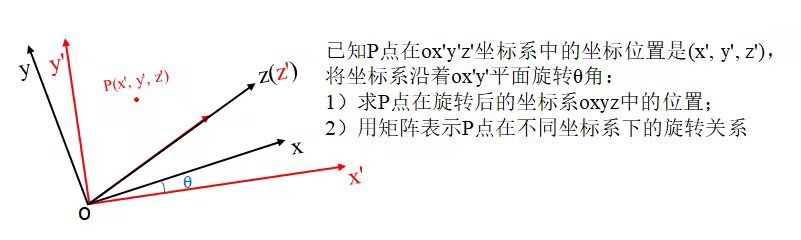
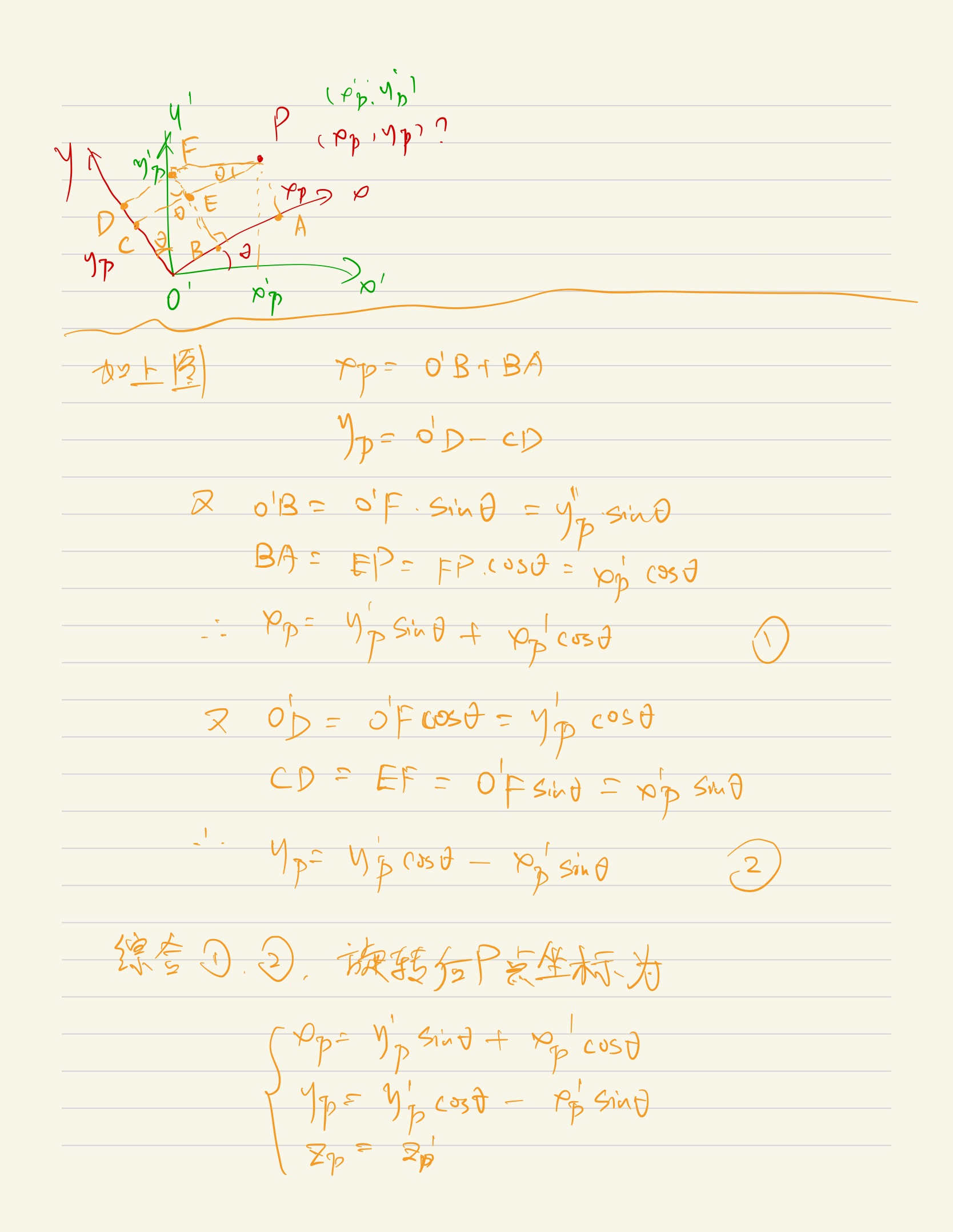
题目一:

**做题时长：10分钟（请写上时长）**





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**

