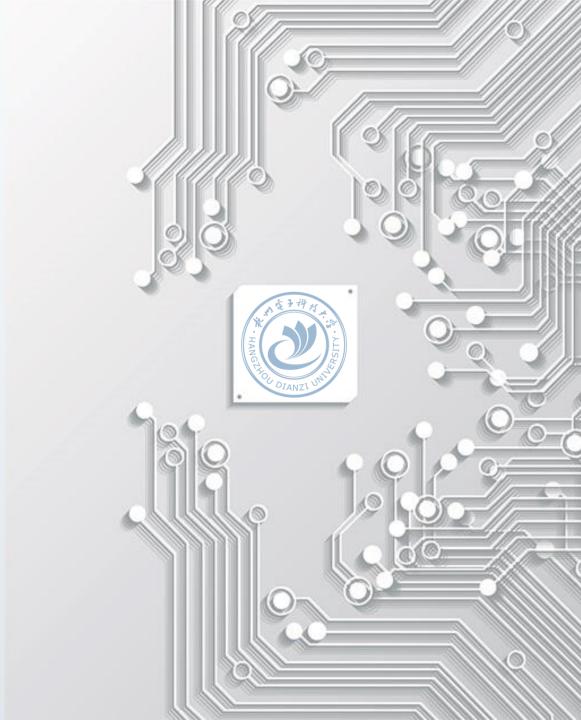
# 数字图像处理

第十三周课堂练习

李竹 杭州电子科技大学 电子信息学院



# 讨论1

- 1. MeanShift和particle filter提高匹配精度和降低运算时间的思路?
- 2. 如果并非视频文件,如工业上的零件定位任务,有什么可以减少运算时间的办法吗?





# 讨论2

1. 传统的SAD等相似度计算方法是否具有尺度、光照、旋 转鲁棒性? 如何设计具有鲁棒性的相似度计算方法?

$$D(i,j) = \sum_{s=1}^{M} \sum_{t=1}^{N} |S(i+s-1,j+t-1) - T(s,t)|.$$

利用第九周缩写的基于HOG的图片相似度计算的函数,完成一个基于HOG的目标物体追踪程序。以下图中的右图为模板,实现在右图中的定位功能。





#### OpenCV模板匹配功能调用练习。

```
OpenCV封装好的选取roi函数,
返回一个Rect2d 数据

if (cnt == 0) {
    Rect2d r;
    r = selectROI(frame, true);
    tempMat = frame(r);
    tempMat.copyTo(refMat);
    destroyAllWindows();
}
```

#### 模板匹配主函数

```
int match_method = 0;
matchTemplate(frame,refMat,resultMat,match_method);
```

```
TM_SQDIFF = 0,
TM_SQDIFF_NORMED = 1,
TM_CCORR = 2,
TM_CCORR_NORMED = 3,
TM_CCOEFF = 4,

TM_CCOEFF = 5
```

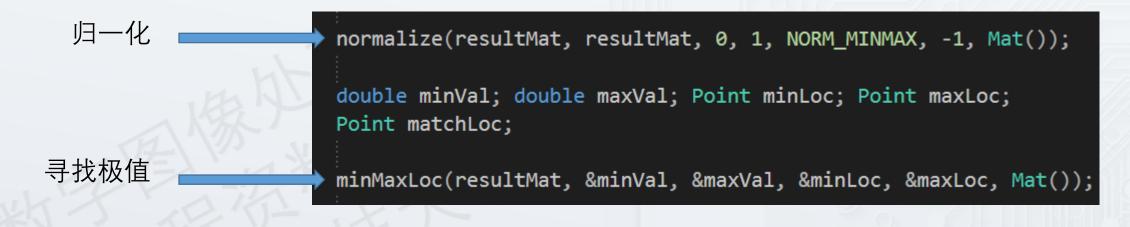
0.输入图像: 8位图像或32位浮点图像

1.模板图像: size必须小于输入图像,并且与输入图像为同一类型的图像

2.输出结果: 假设输入图像尺寸为W×H, 模板为w×h, 则该输出图像的尺

寸为(W-w+1)×(H-h+1)

3.相似度算法



- 0.输入图像: 单通道图像
- 1.返回最小值
- 2.返回最大值
- 3.最小值的位置
- 4.最大值的位置
- 5.mask