第五章

# 目标跟踪

李静

# 目标检测与目标跟踪

目标检测和目标跟踪有什么区别?

目标检测是在图像和视频中扫描和搜寻目标，即在一个场景中对目标进行定位和识别。

目标跟踪主要针对视频流。根据视频第一帧图像所带有的目标信息，对后续视频帧中的目标进行预测和定位。

可以充分利用帧间信息，目标周围的环境信息，及根据周边环境推测得到的三维信息等，更加高效的确定目标所在的位置。

主要解决目标存在变化、发生遮挡或者运动出视野等情况时的跟踪问题。

# 目标跟踪——常规分类

- 生成式模型：反映同类别相似度。此类方法首先建立目标模型或者提取目标特征, 在后续帧中进行相似特征搜索，逐步迭代实现目标定位。缺点：在光照变化, 运动模糊, 分辨率低, 目标旋转形变等情况下，目标跟踪准确性不高。

常用算法：光流法、粒子滤波、Meanshift 算法 、Camshift算法、 Kernel Correlation Filter(KCF)、SRDCF、stable、EBT、SRDCFdecon。

- 判别式模型（track by detection）：反映不同类别差异。通过对比目标模型和背景信息的差异, 将目标模型提取出来, 从而得到当前帧中的目标位置。

常用算法：TLD、MIL，OAB，struck，MEEM，支持向量机。

# 目标跟踪——按时间分类

- 早期跟踪模型：大部分是生成式模型。

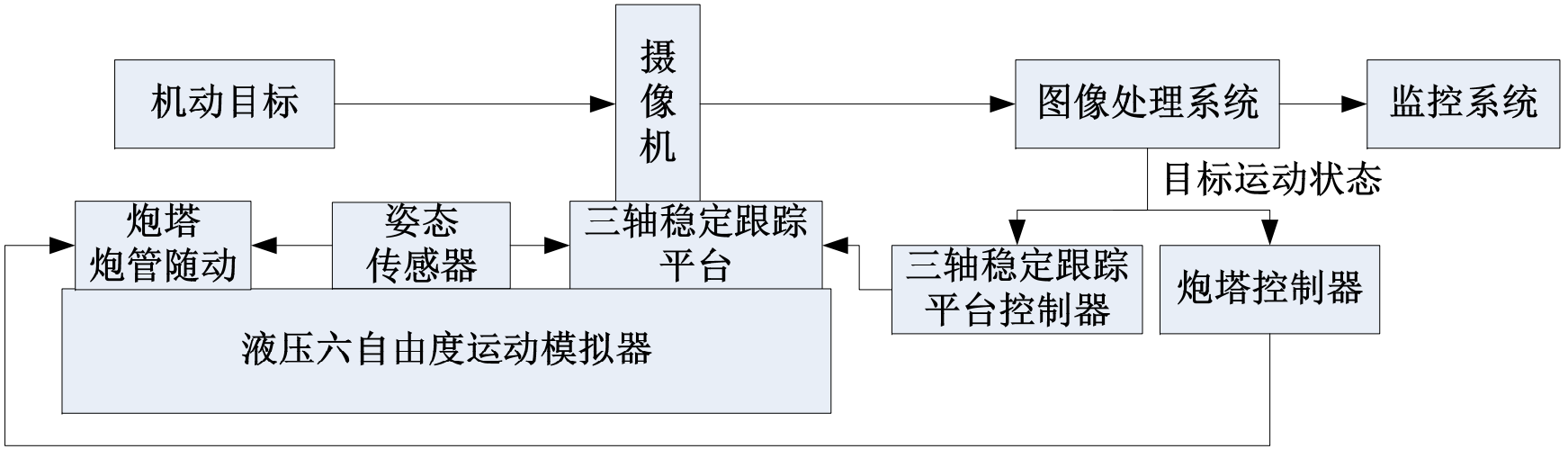
  （1）基于目标模型建模：基于区域匹配、基于特征点跟踪、基于主动轮廓跟踪、光流法等。

  （2）基于搜索的方法：为了减少搜索范围，一种方式通过预测减少搜索范围，如：Kalman滤波、粒子滤波等。另一种为内核算法，运用最速下降法的原理, 向梯度下降方向对目标模板逐步迭代, 直到迭代到最优位置。如：Meanshift、Camshift。

- 相关滤波模型：属于判别式模型，相关滤波之前，所有的跟踪都是在时域上进行，涉及复杂的矩阵求逆计算，速度慢。相关滤波在频域上进行，利用循环矩阵可以在频域对角化的性质，减少了运算量。常用算法：MOSSE、CSK、KCF、 BACF、SAMF。

- 深度学习模型：基于相关滤波，使用深度学习对特征提取、特征搜索等方向进行改进。

# 一、目标跟踪平台


机动目标
摄像机
图像处理系统
监控系统
目标运动状态
炮塔
炮管随动
姿态
传感器
三轴稳定跟踪
平台
三轴稳定跟踪
平台控制器
炮塔控制器
液压六自由度运动模拟器

# 一、目标跟踪平台

# 二、基于Camshift的目标跟踪

CamShift算法的全称是"Continuously Adaptive Mean-Shift"，即：连续自适应的MeanShift算法。

基本思想：对视频序列的所有图像帧都作MeanShift运算，并将上一帧的结果（即搜索窗口的中心位置和窗口大小）作为下一帧MeanShift算法的搜索窗口的初始值，如此迭代下去。

特点：

- meanShift是针对单幅图像寻找最优迭代结果，而camShift则是针对视频序列来处理，并对该序列中的每一帧图片都调用meanShift来寻找最优迭代结果。
- 由于camShift针对一个视频序列进行处理，从而可保证不断调整窗口的大小，因此当目标的大小发生变化的时候，该算法就可以自适应地调整目标区域继续跟踪。

# 二、基于Camshift的目标跟踪

## 1. Meanshift

给定d维空间$R^d$的n个样本点 ，i=1, …, n, 在空间中任选一点x，那么Mean Shift向量的基本形式定义为:

$$M_h = \frac{1}{K}\sum_{x_i \in S_k}(x_i - x)$$

$S_k$是一个半径为h的高维球区域, 满足以下关系的y点的集合,

$$S_h(x) = \left\{ y : (y - x_i)^T (y - x_i) < h^2 \right\}$$

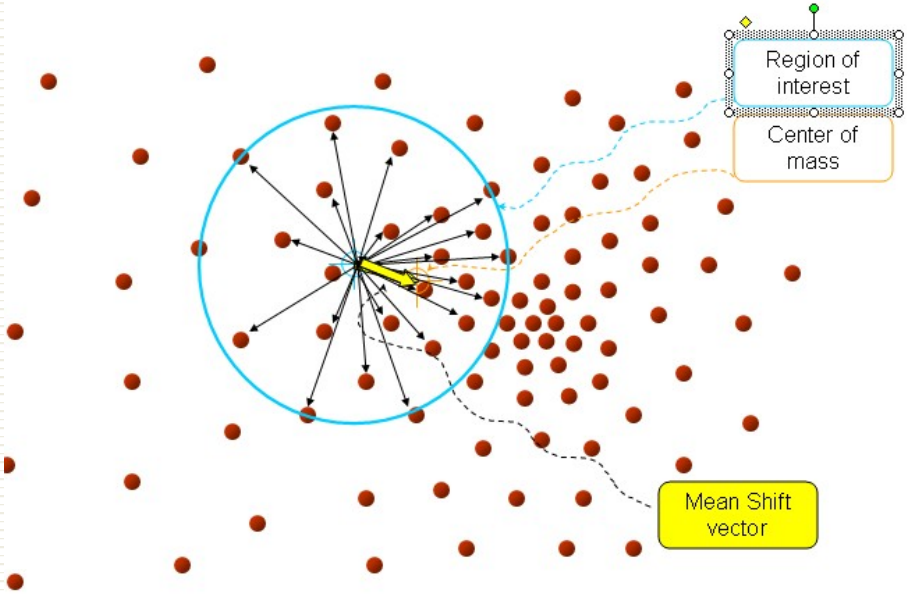
k表示在这n个样本点$x_i$中, 有k个点落入$S_k$区域中。

# 二、基于Camshift的目标跟踪

## 步骤一：

在d维空间中，任选一个点，然后以该点为圆心，h为半径做一个高维球，因为有d维，d可能大于2，所以是高维球。

落在这个球内的所有点和圆心都会产生一个向量，向量是以圆心为起点落在球内的点为终点。然后把这些向量都相加取平均，结果就是Meanshift向量。



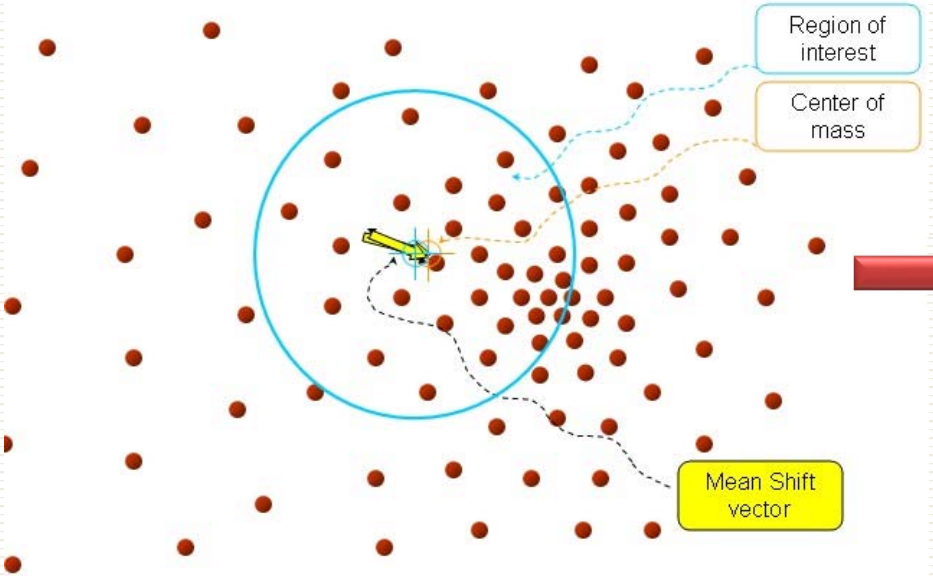

$$M_h = \frac{1}{K}\sum_{x_i \in S_k}\left(x_i - x\right)$$

黄色箭头即$M_h$

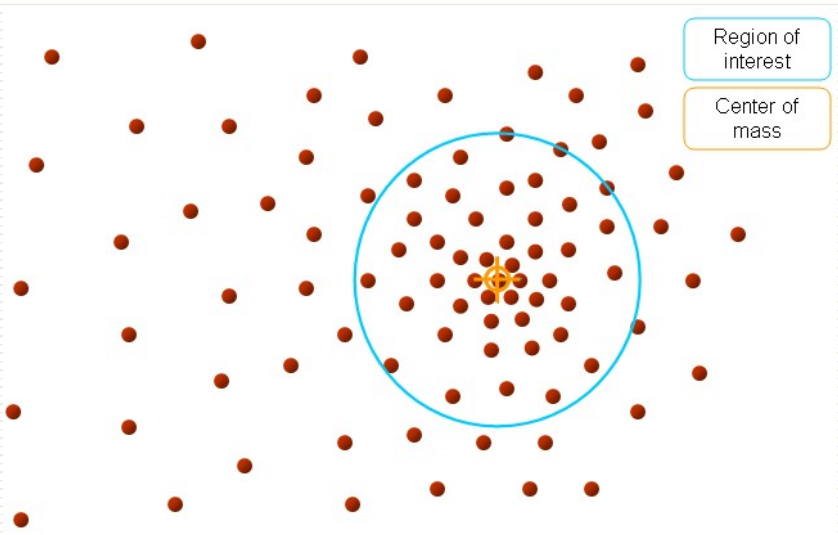
# 二、基于Camshift的目标跟踪

## 步骤二：

再以meanshift向量的终点为圆心，再做一个高维的球。重复步骤一，就可得到一个meanshift向量。如此重复下去，meanshift算法可以收敛到**概率密度最大**的地方，也就是最稠密的地方。




黄色箭头即$M_h$（meanshift向量）




最终的结果

# 二、基于Camshift的目标跟踪

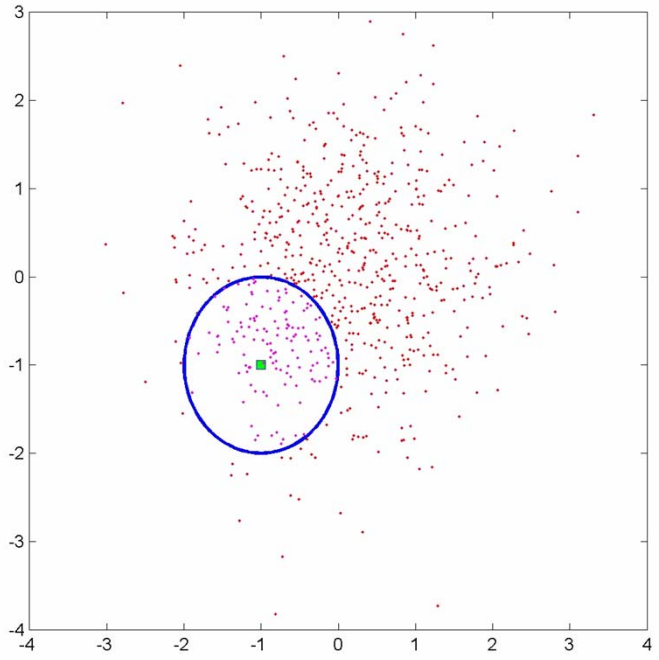
**总结**：假设有一堆同类的数据，目标是去寻找一个数据密度最大的部分。简单的来看，meanshift就是提供了一个方法来求解上面的问题。

1. 初始化随机选择一个数据点，以它为中心，并且设置一个半径r，这个作为每一次选择的数据区域；

2. 以中心为圆心，做一个半径为r的高维球（二维上它应该是一个圆）。这样可以确定落在球中的所有点；

3. 以圆心为起点，以每一个球内的点为终点，生成一个向量，将所有向量进行矢量相加，就可以得到一个和向量；

4. 以和向量的终点为圆心，以r为半径重复上面的操作，meanshift可以收敛到密度最大的部分。

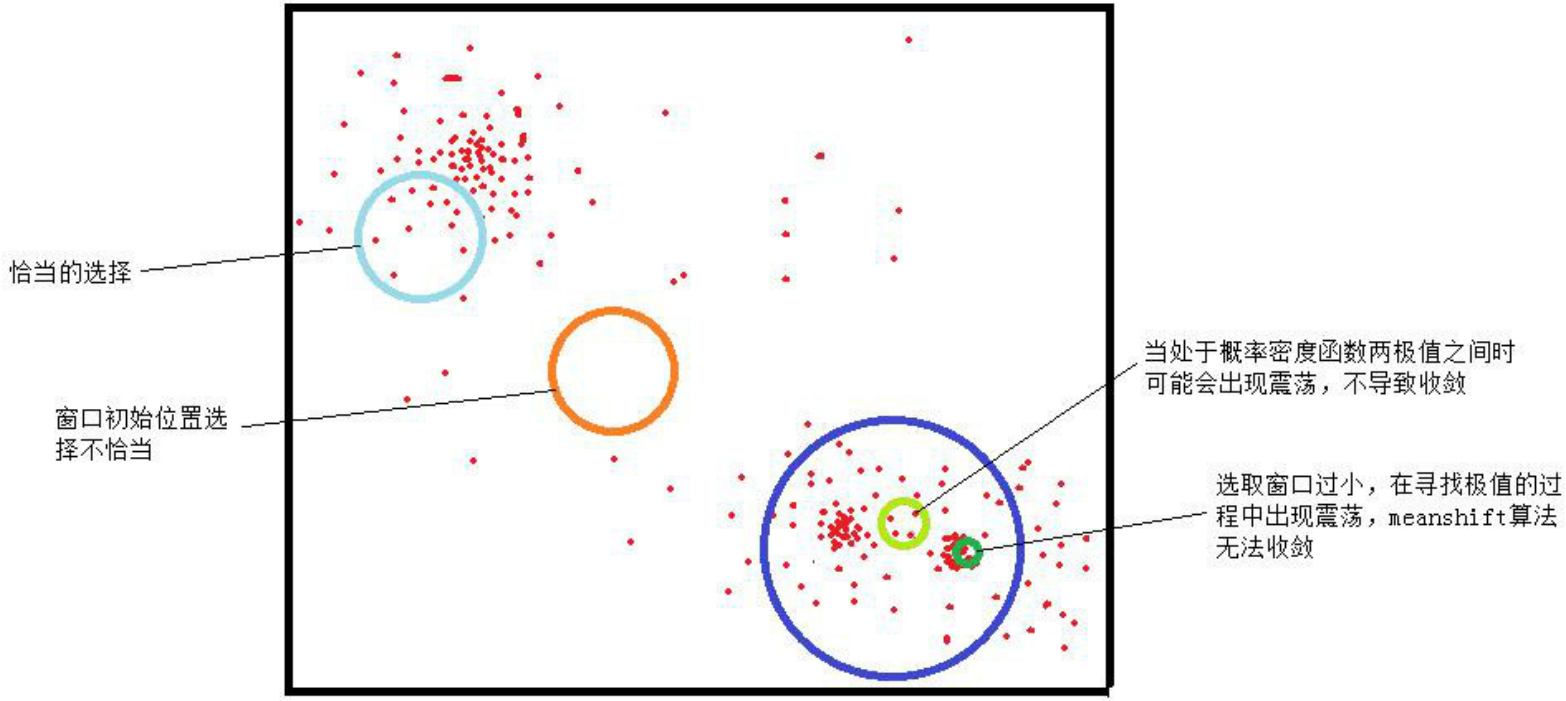



二维空间均值迁移过程

# 二、基于Camshift的目标跟踪

Meanshift算法的适用范围及优缺点：

- meanshift方法适合概率密度函数有极值且在某一局部区域内唯一，即选择的特征数据点能够较为明显的判定目标，即显著特征点。
- meanshift算法受初始值的影响很大，算法收敛的速度和程度在很大程度上和选取的窗口有关，窗口选取的是否恰当很大程度上决定于目标（特征数据点的分布情况）。



# 二、基于Camshift的目标跟踪

## 2. Camshift

**原理**：利用目标的颜色直方图模型将图像转换为颜色概率分布图，初始化一个搜索窗的大小和位置，并根据上一帧得到的结果自适应调整搜索窗口的位置和大小，从而定位出当前图像中目标的中心位置。分为三个部分：

（1）色彩投影图（反向投影）

- RGB颜色空间对光照亮度变化较为敏感，为了减少此变化对跟踪效果的影响，首先将图像从RGB空间转换到HSV空间。
- 对其中的H分量作直方图，在直方图中代表了不同H分量值出现的概率或者像素个数，就是说可以查找出H分量大小为h的概率或者像素个数，即得到了颜色概率查找表。
- 将图像中每个像素的值用其颜色出现的概率替换，就得到了颜色概率分布图。这个过程称为反向投影，颜色概率分布图是一个灰度图像。

# 二、基于Camshift的目标跟踪

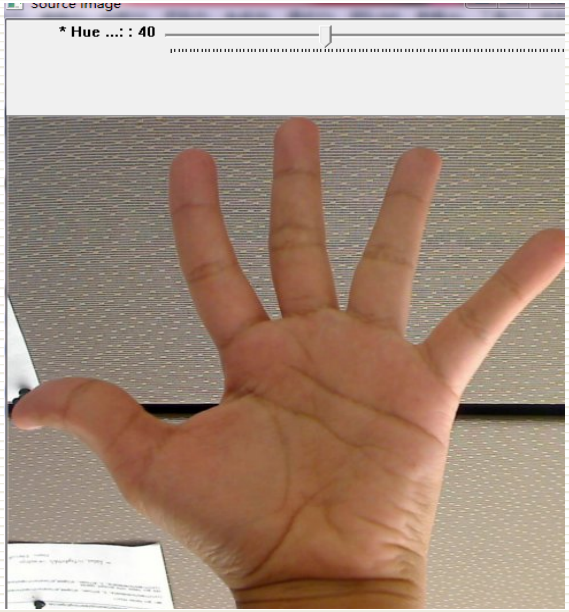
所谓反向投影就是首先计算某一特征的直方图模型，然后使用模型去寻找测试图像中存在的该特征。

利用H分量直方图解释**反向投影原理**：

- 获取测试图像中每个像素的H分量数据$h_{i,j}$，并找到$h_{i,j}$在H分量直方图中的bin的位置；
- 查询H分量直方图中对应bin的数值；
- 将该数值存储在新的图像BackProjection中，也可以先归一化H分量直方图数值到0-255范围，这样可以直接显示BackProjection图像（单通道图像）；
- 通过对测试图像每个像素采取以上步骤，可以得到最终的BackProjection图像。

# 二、基于Camshift的目标跟踪

*BackProjection*中储存的数值代表了测试图像中该像素属于皮肤区域的概率。亮起的区域是皮肤区域的概率更大，而更暗的区域则表示更低的概率。




原图

直方图

反向投影图

# 二、基于Camshift的目标跟踪

## （2）meanshift

meanshift算法是一种密度函数梯度估计的非参数方法，通过迭代寻优找到概率分布的极值来定位目标。算法过程为：

① 在颜色概率分布图中选取搜索窗W

② 计算零阶距： $M_{00}=\sum_x\sum_y I(x,y)$

计算一阶距： $M_{10}=\sum_x\sum_y xI(x,y)$ $M_{01}=\sum_x\sum_y yI(x,y)$

计算搜索窗的质心：$x_c=M_{10}/M_{00}$ $y_c=M_{01}/M_{00}$

③ 调整搜索窗大小

宽度为：$s=\sqrt{M_{00}/256}$ 长度为1.2s；

④ 移动搜索窗的中心到质心，如果移动距离大于预设的固定阈值，则重复②③④，直到搜索窗的中心与质心间的移动距离小于预设的固定阈值，或者循环运算的次数达到某一最大值，停止计算。

# 二、基于Camshift的目标跟踪

## （3）camshift

将meanshift算法扩展到连续图像序列，就是camshift算法。它将视频的所有帧做meanshift运算，并将上一帧的结果，即搜索窗的大小和中心，作为下一帧meanshift算法搜索窗的初始值。如此迭代下去，就可以实现对目标的跟踪。算法过程为：

① 初始化搜索窗

② 计算搜索窗的颜色概率分布（反向投影）

③ 运行meanshift算法，获得搜索窗新的大小和位置。

④ 在下一帧视频图像中用③中的值重新初始化搜索窗的大小和位置，再跳转到②继续进行。

优点：camshift能有效解决目标变形和部分遮挡的问题，对系统资源要求不高，时间复杂度低，在简单背景下能够取得良好的跟踪效果。

缺点：当背景较为复杂，或者有许多与目标颜色相似像素干扰的情况下，会导致跟踪失败。因为它单纯的考虑颜色直方图，忽略了目标的空间分布特性，所以这种情况下需加入对跟踪目标的预测算法。

# 实验课——编写程序

## 1、编写Camshift目标跟踪方法