3.1 运动目标跟踪

视觉跟踪(目标跟踪):指实时估计视频中目标的运动状态，从而实现对目标的位置和运动趋势的判定，亦称为目标跟踪。

该技术在智能人机交互、机器人控制、交通安全监控、医疗成像、军事指导、天文探测和电视制作等领域有着广泛的应用前景。基于目标跟踪等视频监控的智能化技术必将给计算机视觉在公共安全领域中的应用提供了广阔的前景，是物联网应用的必然要求。

3.1 运动目标跟踪

- 运动跟踪难点：
  - 跟踪的目标复杂多变：在跟踪过程目标可能发生缩放旋转等表观形变和自遮挡、目标之间及目标与背景之间的遮挡问题。
  - 目标跟踪的应用环境复杂多变：如光照变化、颜色或运动状态相近物体的干扰和场景的动态变化等。这就要求目标跟踪过程要实时考虑周围场景的变化，以实现长期有效的目标跟踪。

3. 运动目标跟踪

- 目标视觉跟踪的目的是在图像序列中进行目标定位，可以有一定程度的像素分类不准确，只需结果足可区分目标和背景。

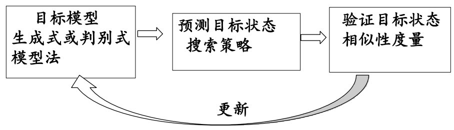


图1 目标跟踪的原理

3.1 运动目标跟踪

- 目标建模：

选定跟踪目标后，目标建模是指为目标基于某些显著特征建立鲁棒、与背景区分度高的数学模型。目标模型对目标的准确跟踪有着至关重要的影响。由于目标本身特征的多样性、跟踪过程中不稳定性（如目标可能发生位移、旋转、尺度变化和形变等），目标建模一直是目标跟踪研究的热点和难点。和背景的判别。

- 目标建模过程：

可分解为两步：第一是采用各种不同的视觉特征构建鲁棒的目标特征描述(表示)。第二是基于特征描述借助于统计学习等方法,为目标建立高可靠的数学模型。

- 目标建模作用：

特征提取后的目标数学建模作用在于：在后续视频帧用此目标模型，评估搜索得到的候选目标区域的正确性以最终估计出目标的状态。

3.1 运动目标跟踪

- 目标模型：

➢生成式模型法将目标跟踪视为匹配问题，是先学习一个表示目标的表观模型，然后在目标可能运动的范围内搜索与该生成模型最相似的区域作为跟踪的结果。

➢判别式模型。将跟踪问题视为目标/背景两类的分类问题。该方法利用机器学习中贝叶斯分类器、SVM、adaboost和决策树等方法，在线学习正负样本在特征空间的分类模型，从而实现目标和背景的判别。

3.1 运动目标跟踪

- 目标状态预测：

在视频首帧确立目标模型后，就可以在下一帧图像中采用特征匹配(生成式的目标模型)或分类学习方法(判别式的目标模型)进行目标状态的估计。

- ➢ 为了减少目标搜索的区域以提高跟踪算法的实时性，我们常常对目标运动轨迹进行预测，搜索预测可能的候选目标区域。
- ➢ 目前主要有确定性和随机性两种搜索策略：确定性搜索策略将跟踪问题看成函数最优化问题，例如经典的基于梯度下降的meanshift算法。随机性搜索策略将跟踪问题看成状态最优的估计问题，代表性经典方法有粒子滤波算法。

3.1 运动目标跟踪

- 目标跟踪验证策略：

对预测得到的候选目标区域，利用目标模型并基于一定相似性度量准则判定当前帧的目标位置，即对预测的目标状态进行验证的过程。

验证实施的核心是相似性度量准则，即度量候选目标区域特征和目标特征模型间的相似程度。传统相似性度量方法有基于巴氏距离等直方图相似性度量和基于KL距离等高斯分布度量方法。

3.2 运动目标跟踪

- 目标模型更新：

在目标跟踪过程中可能出现目标外观变化、遮挡和光照变化等问题，故目标模型的自适应更新是很重要的。

目标模型更新方法与目标模型建立方法密切相关。模型更新的方法大致可分为替代型更新和学习型更新：

- ➢ 替代型模型更新方法即用跟踪的结果更新目标模板。替代型模型跟踪方法难点在于判断是否进行模型更新。
- ➢ 学习型模型更新是通过学习的方法建立并更新目标模型。

3.4运动目标跟踪-基于生成式目标跟踪

基于生成式模型的目标跟踪方法：

- 基于稀疏光流的目标跟踪（已讲）
- Meanshift
- Camshift
- 粒子滤波跟踪框架
- 基于运动历史图的目标跟踪
- 团块的目标跟踪

3.4.1运动目标跟踪-基于生成式目标跟踪

Mean shift：均值漂移

均值漂移(*Mean Shift*)是Fukunaga等人提出的一种非参数概率密度梯度估计算法。该算法被广泛应用到诸多相关域，如模式分类、图像分割以及目标跟踪等方面。

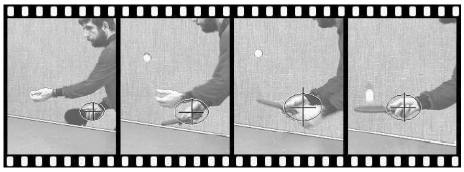
在跟踪领域，*Mean Shift*的跟踪算法是一种以目标区域像素值的概率分布为特征的跟踪算法。

Meanshift目标跟踪方法

- 基于生成式的目标跟踪方式-Meanshift：

目标特征提取
基于彩色空间量化
⇨
目标模型
核函数直方图建目标模型，估计目标的概率密度
⇩
预测目标状态
确定性搜索策略
⇦
验证目标状态
概率密度相似性比较
更新分类器

Mean-Shift物体跟踪

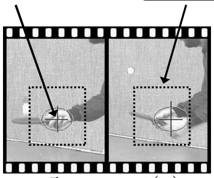


… ⟶ …

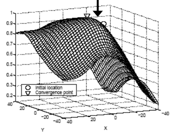
Mean-Shift物体跟踪

目标跟踪算法

从当前帧的模板位置开始 ⟶ 在下一帧搜索模板的邻近区域 ⟶ 通过相似性函数找到一个最佳的候选区域.



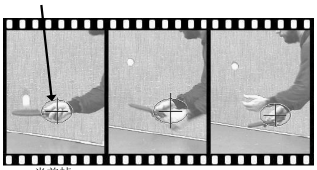
$\vec{q}$ $\vec{p}(y)$

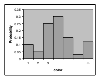


$f[\vec{p}(y),\vec{q}]$

Mean-Shift 物体跟踪目标表示

在当前帧选取一个合适的模板 ⟶ 选择一个特征空间(例如彩色空间的量化) ⟶ 在选择的特征空间中表示选取的模板



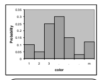


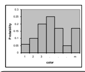
… 当前帧 ⟶ …

Mean-Shift 物体跟踪

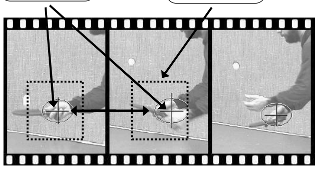
相似性函数:

$f(y)=f[\vec{q},\vec{p}(y)]$





从当前帧模板的位置开始 ⟶ 在下一帧模板附近的区域进行搜索 ⟶ 通过算法找到最佳的候选区域

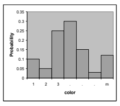


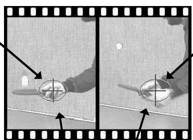
在后面视频的帧，重复上面的过程

模板 候选区域
当前帧
… ⟶ …

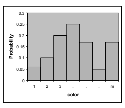
Mean-Shift物体跟踪

模板区域
(质心为取模板时的点)





候选区域
(质心为y)



$\vec{q}=\{q_u\}_{u=1..m}$ $\sum_{u=1}^{m} q_u = 1$?

$\vec{p}(y)=\{p_u(y)\}_{u=1..m}$ $\sum_{u=1}^{m} p_u = 1$?

相似性函数: $f(y)=f[\vec{q},\vec{p}(y)]$

Mean-Shift物体跟踪

相似度函数的确定

模板区域: $\vec{q}=(q_1,\ldots,q_m)$

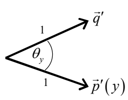
候选区域: $\vec{p}(y)=(p_1(y),\ldots,p_m(y))$

相似性函数: $f(y)=f[\vec{p}(y),\vec{q}]=?$

Bhattacharyya 系数

$\vec{q}'=(\sqrt{q_1},\ldots,\sqrt{q_m})$

$\vec{p}'(y)=(\sqrt{p_1(y)},\ldots,\sqrt{p_m(y)})$



$$f(y)=\cos\theta_y=\frac{p'(y)^T q'}{\|p'(y)\|\cdot\|q'\|}=\sum_{u=1}^{m}\sqrt{p_u(y)q_u}$$

核函数直方图建模-无参密度估计

PDF in feature space
- Color space
- Scale space
- Actually any feature space you can conceive
- …

Data

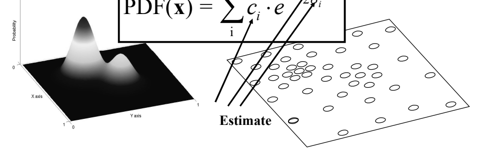
parametric Density Estimation ⇨ p(x｜θ) 极大似然估计法 贝叶斯估计

Non-parametric Density Estimation ⇨ p(x)未知 直方图法 Kernel KNN法

Parametric Density Estimation 参数密度估计

Assumption : The data points are sampled from an underlying PDF

$$\mathrm{PDF}(\mathbf{x}) = \sum_i c_i \cdot e^{-\frac{(\mathbf{x}-\mu_i)^2}{2\sigma_i^2}}$$



极大似然估计

- 估计类条件概率的常用策略：先假定其具有某种确定的概率分布形式，再基于训练样本对概率分布参数估计。
- 记关于类别 c 的类条件概率为 $P(\mathbf{x} \mid c)$，
  - 假设 $P(\mathbf{x} \mid c)$ 具有确定的形式被参数 θ_c 唯一确定，我们的任务就是利用训练集 D 估计参数 θ_c

极大似然估计Maximum Likelihood estimation (MLE)

- 令 D_c 表示训练集中第 c 类样本的组合的集合，假设这些样本是独立的，则参数 θ_c 对于数据集 D_c 的似然是

$$P(D_c \mid \boldsymbol{\theta}_c) = \prod_{\mathbf{x} \in D_c} P(\mathbf{x} \mid \boldsymbol{\theta}_c)$$

  - 对 θ_c 进行极大似然估计，寻找能最大化似然 $P(D_c \mid \theta_c)$ 的参数值 $\hat{\theta}_c$。直观上看，极大似然估计是试图在 $\hat{\theta}_c$ 所有可能的取值中，找到一个使数据出现的“可能性”最大值。

极大似然估计

- 令 D_c 表示训练集中第 c 类样本的组合的集合，假设这些样本是独立的，则参数 θ_c 对于数据集 D_c 的似然是

$$P(D_c \mid \boldsymbol{\theta}_c) = \prod_{\mathbf{x} \in D_c} P(\mathbf{x} \mid \boldsymbol{\theta}_c)$$

  - 对 θ_c 进行极大似然估计，寻找能最大化似 $P(D_c \mid \theta_c)$ 的参数值 $\hat{\theta}_c$。直观上看，极大似然估计是试图在 θ_c 所有可能的取值中，找到一个使数据出现的“可能性”最大值。
- 式 (7.9) 的连乘操作易造成下溢，通常使用对数似然(log-likelihood)

$$LL(\boldsymbol{\theta}_c) = \log P(D_c \mid \boldsymbol{\theta}_c) = \sum_{\mathbf{x} \in D_c} \log P(\mathbf{x} \mid \boldsymbol{\theta}_c)$$

- 此时参数 θ_c 的极大似然估计 $\hat{\theta}_c$ 为 $\hat{\boldsymbol{\theta}}_c = \underset{\boldsymbol{\theta}_c}{\arg\max}\, LL(\boldsymbol{\theta}_c)$

极大似然估计

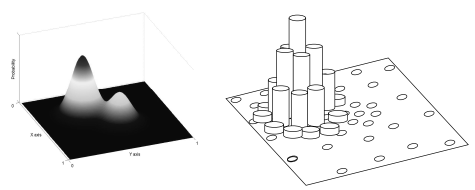
- 例如，在连续属性情形下，假设概率密度函数 $p(\mathbf{x} \mid c) \sim N(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$，则参数 $\boldsymbol{\mu}_c$ 和 $\boldsymbol{\sigma}_c^2$ 的极大似然估计为

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{|D_c|} \sum_{\mathbf{x} \in D_c} \mathbf{x} \tag{7.12}$$

$$\hat{\boldsymbol{\sigma}}_c^2 = \frac{1}{|D_c|} \sum_{\mathbf{x} \in D_c} (\mathbf{x} - \hat{\boldsymbol{\mu}}_c)(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)^{\mathrm{T}} \tag{7.13}$$

- 也就是说，通过极大似然法得到的正态分布均值就是样本均值，方差就是 $(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)(\mathbf{x} - \hat{\boldsymbol{\mu}}_c)^{\mathrm{T}}$ 的均值，这显然是一个符合直觉的结果。
- 需注意的是，这种参数化的方法虽能使类条件概率估计变得相对简单，但估计结果的准确性严重依赖于所假设的概率分布形式是否符合潜在的真实数据分布。

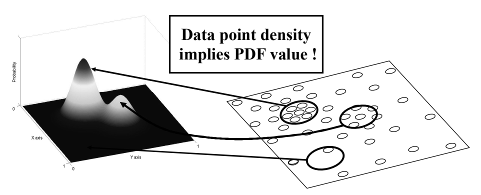
Non-Parametric Density Estimation



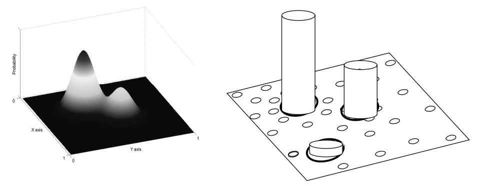
无参密度估计

Finding modes in a set of data samples, manifesting an underlying probability density function (PDF) in $\mathbf{R}^N$

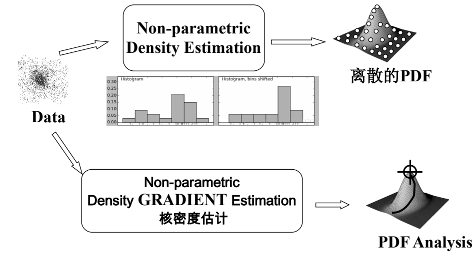
Assumption : The data points are sampled from an underlying PDF



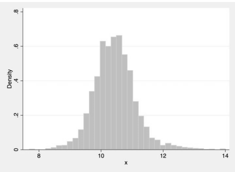
Non-Parametric Density Estimation 非参数密度估计



核函数直方图建模-无参密度估计



无参数估计-直方图



估计X=x处的密度函数值，选一个x附近的小区间，数一下在这个区间里面的点的个数，除以总个数，应该是一个比较好的估计。

统计落在[x-h,x+h]区间的点的个数。f(x)的估计写成：

$$\hat{f}_h(x) = \frac{1}{2h}\frac{\#x_i \in [x-h, x+h]}{N} = \frac{1}{2Nh}\sum_{i=1}^{N} 1(x-h \le x_i \le x+h) = \frac{1}{Nh}\sum_{i=1}^{N}\frac{1}{2}\cdot 1(\frac{|x-x_i|}{h} \le 1)$$

无参数估计-直方图

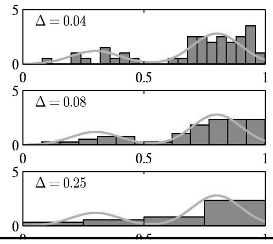
统计落在[x-h,x+h]区间的点的个数。f(x)的估计写成：

$$\hat{f}_h(x) = \frac{1}{2h}\frac{\#x_i \in [x-h, x+h]}{N} = \frac{1}{2Nh}\sum_{i=1}^{N} 1(x-h \le x_i \le x+h) = \frac{1}{Nh}\sum_{i=1}^{N}\frac{1}{2}\cdot 1(\frac{|x-x_i|}{h} \le 1)$$

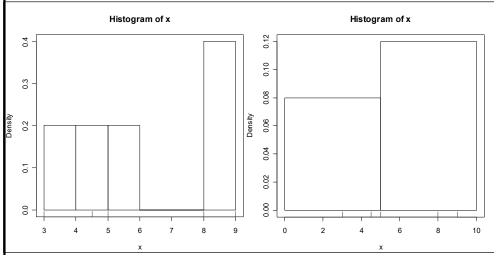
h怎么选?

$$p_i = \frac{n_i}{N\,\Delta_i}$$

$h = c \cdot N^{-1/5}$



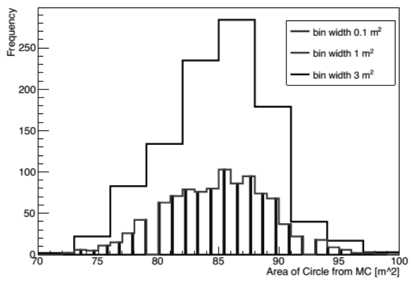
无参数估计-直方图 - Bin 宽度



hist(x,breaks=5,right=F,prob=T) hist(x,breaks=2,right=F,prob=T)

Area=1

无参数估计-直方图 - Bin 宽度



无参数估计-直方图 - Bin 宽度

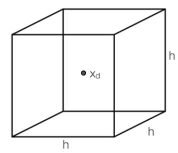
$$\hat{f}_h(x) = \frac{1}{2h}\frac{\#x_i \in [x-h, x+h]}{N} = \frac{1}{Nh}\sum_{i=1}^{N}\frac{1}{2}\cdot 1(\frac{|x-x_i|}{h} \le 1)$$

令 $K_0(t) = \frac{1}{2}\cdot 1(t<1)$

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{N} K_0(\frac{x-x_i}{h})$$

如果用标准正态分布的密度函数作为K，估计就变成了：

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{N}\phi(\frac{x-x_i}{h})$$

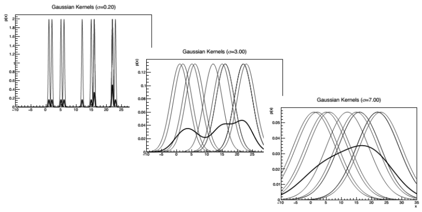


$$P_{KDE}(\vec{x}) = \frac{1}{Nh^D}\sum_{n=1}^{N} K(\frac{\vec{x}-\vec{x}_n}{h})$$

$$\begin{aligned} P_{KDE}(x=6) &= \frac{1}{Nh^D}\sum_{n=1}^{N} K(\frac{\vec{x}-\vec{x}_n}{h}) \\ &= \frac{1}{12\times 3^1}\left[K\left(\frac{1-6}{3}\right)+K\left(\frac{2-6}{3}\right)+K\left(\frac{5-6}{3}\right)+K\left(\frac{6-6}{3}\right)+\dots K\left(\frac{23-6}{3}\right)\right] \\ &= \frac{1}{12\times 3^1}\left[0+0+1+1+0+0+0+0+0+0+0+0\right] \end{aligned}$$

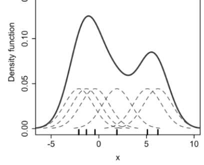
非参数核密度估计- Kernel Density Estimation

$$P_{KDE}(\vec{x}) = \frac{1}{Nh^D}\sum_{n=1}^{N} K(\frac{\vec{x}-\vec{x}_n}{h}) \qquad K(\vec{x},\sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^D} e^{-\frac{||\vec{x}-\vec{x}_n||^2}{2\sigma^2}}$$



非参数核密度估计- Kernel Density Estimation

- **核密度估计的方法生成的密度估计是一个连续的光滑的曲线。**
- **具体的方法是在对应的数据点上放置一个kernel函数，然后将所有数据点上的kernel叠加在一起，就可以构成一个光滑的密度函数。**



图中红色的虚线表示的是在对应6个数据点上放置的kernel函数，而蓝色的线代表的就是将所有的kernel函数叠加在一起构成的密度函数。

非参数核密度估计- Kernel Density Estimation

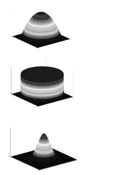
归一化

A function of some finite number of data points $x_1 \dots x_n$

Data

Examples:不同核

- **Epanechnikov Kernel** $K_E(\mathbf{x}) = \begin{cases} c\left(1-\|\mathbf{x}\|^2\right) & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$
- **Uniform Kernel** $K_U(\mathbf{x}) = \begin{cases} c & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$
- **Normal Kernel** $K_N(\mathbf{x}) = c\cdot\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$



Mean shift向量

- ✓如果样本点x_i从目标概率密度函数f(x)中采样得到，那么特征空间中样本点最密集的区域，即被认为的目标位置(这个区域有更多来自f(x)采样点）。
- ✓寻找数据集合中密度最大数据的分布位置可以对标准密度梯度进行估计。

Kernel Density Estimation Gradient

$$\nabla P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla K(\mathbf{x}-\mathbf{x}_i)$$

Give up estimating the PDF !
Estimate ONLY the gradient

核函数:

$$K(\mathbf{x}-\mathbf{x}_i) = ck\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

Size of window

可得到：

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n} \nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \bullet \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

$\mathbf{g}(\mathbf{x}) = -k'(\mathbf{x})$

Computing The Mean Shift / Kernel Density Estimation Gradient

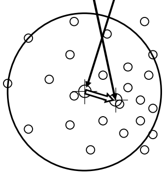
$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n} \nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \bullet \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

$\mathbf{g}(\mathbf{x}) = -k'(\mathbf{x})$

计算均值漂移-Computing The Mean Shift

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n} \nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right] \bullet \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

Yet another Kernel density estimation !



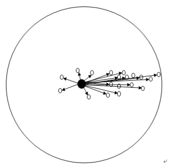
Simple Mean Shift procedure:
• Compute mean shift vector

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)} - \mathbf{x}\right]$$

•Translate the Kernel window by m(x)

计算均值漂移-Computing The Mean Shift

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)} - \mathbf{x}\right]$$

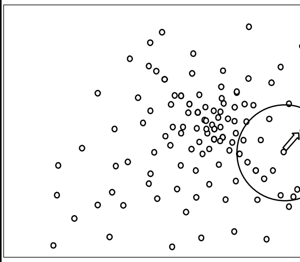


为便于理解，假设g (x)=1

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} x_i}{N} - x\right] = \bar{x} - x$$

- ➢ 中间的实心黑点表示x点,也是核函数g (x)的中心。
- ➢ 周围的空心白点是样本点$\mathbf{x}_i$。
- ➢ (xi - x)样本点xi相对于点x的偏移向量,箭头表示
- ➢ mean shift向量就是对落入区S_h中的n个样本点相对于点x的偏移向量求和然后再平均。

Mean shift：均值漂移

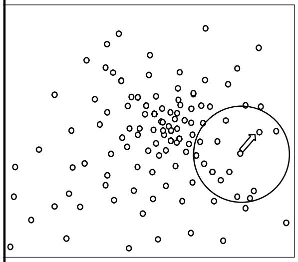


1、初始化目标
2、以R为半径画高维球，计算落在高维球内点到中心点距离（高斯加权）
3、更新目标的位置
4、重复2,3步骤，直到停止迭代

停止迭代条件：
1、前后两次中心点距离小于某阈值
2、迭代次数超过某阈值

在当前帧中通过反复迭代搜索特征空间中样本点最密集的区域，即被认为的目标位置，从而达到跟踪的目的。

Mean shift：均值漂移

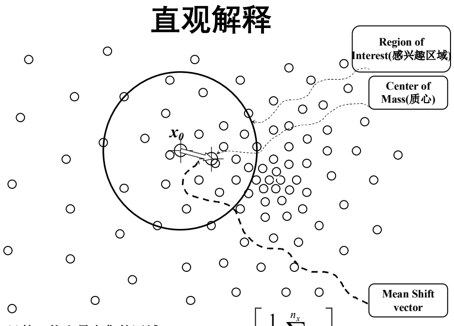


Meanshift (1)首先计算当前点的偏移均值，(2)移动该点到其偏移均值；(3)然后以此为新的起始点，继续移动，直到满足一定的条件结束。

初始化目标(给定初始化*x*,核函数G(X)，允许误差ε, meanshift 计算三步：
(1) 计算$m_h(x)$
(2) 把$m_h(x)$赋给x
(3) 如果$\| m_h(x) - x \| < \varepsilon$,则循环结束；否则执行步骤(1)

在当前帧中通过反复迭代搜索特征空间中样本点最密集的区域，即被认为的目标位置，从而达到跟踪的目的。

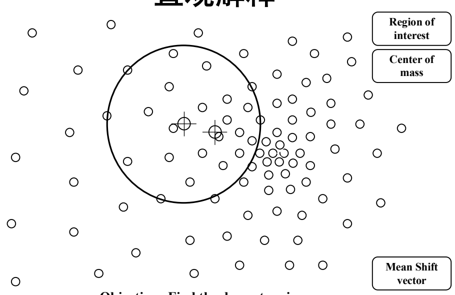
直观解释



目的：找出最密集的区域

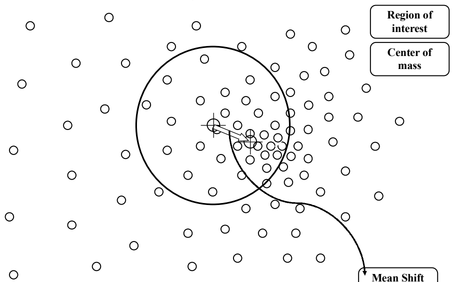
$$M_h(\mathbf{y}) = \left[\frac{1}{n_x}\sum_{i=1}^{n_x} \mathbf{x}_i\right] - x_0$$

直观解释



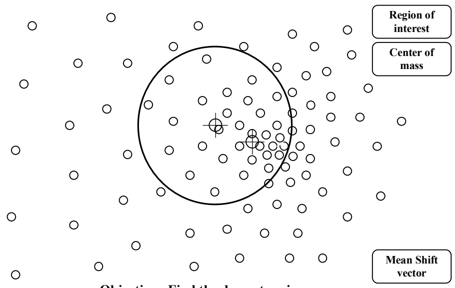
Objective : Find the densest region
Distribution of identical billiard balls

直观解释



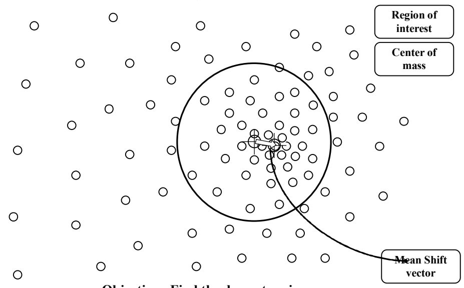
Objective : Find the densest region
Distribution of identical billiard balls

直观解释



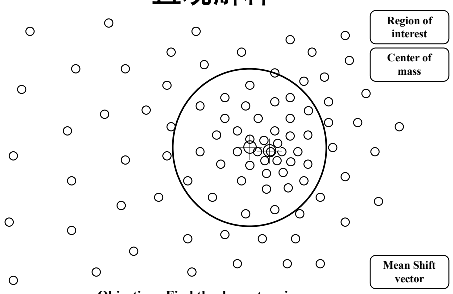
Objective : Find the densest region
Distribution of identical billiard balls

直观解释



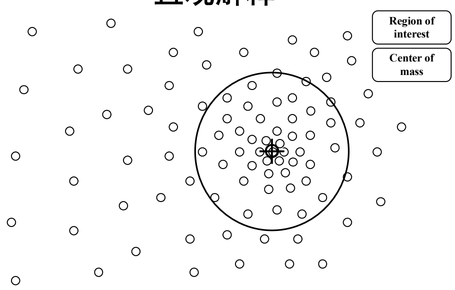
Objective : Find the densest region
Distribution of identical billiard balls

直观解释



Objective : Find the densest region
Distribution of identical billiard balls

直观解释

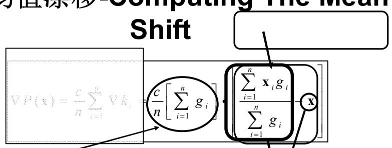


Objective : Find the densest region
Distribution of identical billiard balls

Mean shift向量

- ✓如果样本点x_i从一个概率密度函数f(x)中采样得到，那么特征空间中样本点最密集的区域，即被认为的目标位置(这个区域有更多来自f(x)采样点）。
- ✓如何从当前位置寻找到快速收敛到概率密度最大的方向？沿着概率密度函数的梯度方向。
  - ➢因为概率密度梯度方向是概率密度增加最大的方向，所以沿着概率密度函数的梯度方向可以最快周到样本集最密集的区域。

计算均值漂移-Computing The Mean Shift

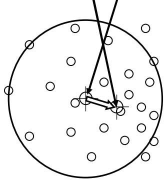


Yet another Kernel density estimation !

Simple Mean Shift procedure:

- Compute mean shift vector

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h} \right)}{\sum_{i=1}^{n} g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h} \right)} - \mathbf{x} \right]$$

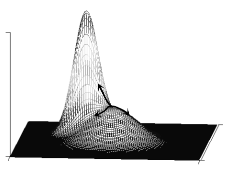


- Translate the Kernel window by m(x)

均值漂移步骤

1.计算meanshift向量

平均的偏移向量会指向样本分布最多的区域，也就是概率密度函数的梯度方向

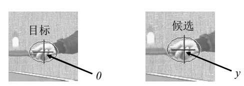


2. 根据向量移动核窗

最终核函数的中心点收敛到数据空间中密度最大的点。Meanshift向量总是指向密度增加最大的方向。(点移动到密度函数的局部极大值点处)

Mean-Shift物体跟踪

$\{x_i\}_{i=1..n}$ 目标像素位置



$k(x)$ 可微,各向同性，凸面，递减的核函数
次要像素受到背景干扰的影响

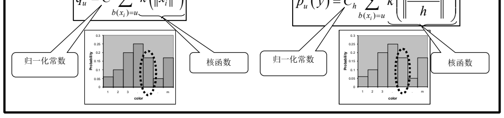
$b(x)$ 判断像素x是否属于$(1..m)$中的一个特征值

特征值u在模板区域中的概率

$$q_u = C \sum_{b(x_i)=u} k\left(\|x_i\|^2\right)$$

特征值u在候选区域中的概率

$$p_u(y) = C_h \sum_{b(x_i)=u} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)$$



Mean-Shift物体跟踪

$$f(y)=\sum_{u=1}^{m}\sqrt{p_u(y)q_u}$$

模板位置: y_0
候选区域位置: y

$$f(y)\approx\underbrace{\frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u(y_0)q_u}}_{\text{与y独立}}+\underbrace{\frac{1}{2}\sum_{u=1}^{m}\boxed{p_u(y)}\boxed{\sqrt{\frac{q_u}{p_u(y_0)}}}}$$

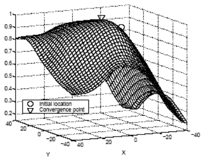
$$p_u(y)=C_h\sum_{b(x_i)=u}\boxed{k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}$$

$$\frac{C_h}{2}\sum_{i=1}^{n}\boxed{w_i}\boxed{k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}$$

Mean-Shift物体跟踪

相似度的最大化

模型 $$\boxed{\frac{C_h}{2}\sum_{i=1}^{n}w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}$$ = 寻求最大值



3.4运动目标跟踪-基于生成式目标跟踪

Meanshift算法：
均值漂移(Mean Shift)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：
计算一阶距：
计算搜索窗的质心：
(3).调整搜索窗大小：
(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2)3)4)，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。

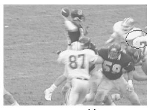
Mean-Shift物体跟踪

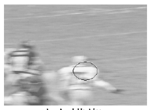
特征空间: 16×16×16 量化 RGB
目标: 在第一帧手动选择
平均迭代次数: 4

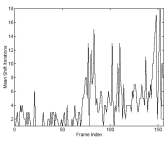
Mean-Shift物体跟踪

结果


局部隐藏


干扰


动态模糊



3.4.2运动目标跟踪-基于生成式目标跟踪

基于生成式模型的目标跟踪方法： *cpp-example-camshiftdemo.exe*

camShift（Continuously Adaptive Mean-SHIFT）算法即：连续自适应的MeanShift算法。

- ✓ 其基本思想是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。
- ✓ 简单点说meanShift是针对单张图片寻找最优迭代结果，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。
- ✓ 正是由于camShift针对一个视频序列进行处理，从而保证其可以不断调整窗口的大小，如此一来，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。

3.4运动目标跟踪-基于生成式目标跟踪

基于生成式模型的目标跟踪方法： *cpp-example-camshiftdemo.exe*

camShift（Continuously Adaptive Mean-SHIFT）算法可以分为三个部分：

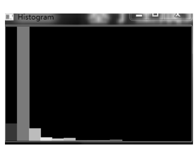
一、计算色彩投影图：（目标模型）

（1）减少光照变化对目标跟踪的影响，RGB颜色空间转换到HSV颜色空间；
（2）计算**被跟踪目标区域**的H分量的颜色直方图。
（3）**每个像素的值**用其颜色出现的概率进行替换，得到与原图等大的颜色概率分布图（**反向投影图就是概率分布图，投影图中某个像素点的值就是这个点符合目标的概率分布**）；

3.4运动目标跟踪-基于生成式目标跟踪

计算Back Projection的步骤：反向投影

1. 计算被跟踪目标的色彩直方图。在各种色彩空间中，只有HSI空间(或与HSI类似的色彩空间)中的H分量可以表示颜色信息。所以在具体的计算过程中，首先将其他的色彩空间的值转化到HSI空间，然后会其中的H分量做1D直方图计算。



3.4运动目标跟踪-基于生成式目标跟踪

计算Back Projection的步骤：反向投影

2. 根据获得的色彩直方图将原始图像转化成色彩概率分布图像，这个过程就被称作“Back Projection”。

- ✓ 反向投影图中每个像素的值用其颜色出现的概率进行替换。
- ✓ 投影图中某个像素点的值就是这个点符合目标的概率分布；点越亮，就说明这个点属于物体的概率越大。

3.4运动目标跟踪-基于生成式目标跟踪

计算Back Projection的步骤：反向投影

2. 根据获得的色彩直方图将原始图像转化成色彩概率分布图像，这个过程就被称作“Back Projection”。

- ✓ 反向投影图中每个像素的值用其颜色出现的概率进行替换。
- ✓ 投影图中某个像素点的值就是这个点符合目标的概率分布；点越亮，就说明这个点属于物体的概率越大。

3.4运动目标跟踪-基于生成式目标跟踪

基于生成式模型的目标跟踪方法：

camShift（Continuously Adaptive Mean-SHIFT）算法可以分为三个部分：

一、计算色彩投影图：（目标模型）

二、meanShift寻优（目标位置预测）

meanShift算法是一种非参数概率密度估计方法，它通过不断迭代计算得到最优搜索窗口的位置和大小

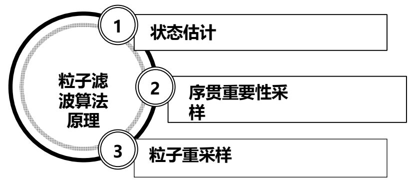
三、camShift跟踪算法（目标验证及更新）

camShift其实就是在视频序列的每一帧当中都运用meanShift，并将上一帧的meanShift结果作为下一帧的初始值，如此不断循环迭代，就可以实现目标的跟踪了

3.4.3 运动目标跟踪-基于生成式目标跟踪

研究方法和过程

粒子滤波跟踪算法

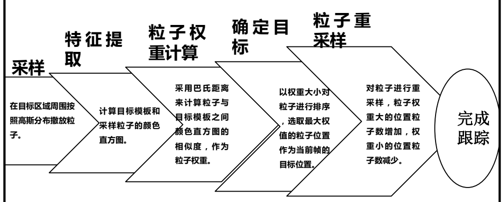


66

3.4.3运动目标跟踪-基于生成式目标跟踪

研究方法和过程

粒子滤波跟踪算法



研究方法和过程

粒子滤波算法原理

最优贝叶斯估计

动态时变系统可以描述为：

$x_k=f_k(x_{k-1},v_{k-1})$，$z_k=h_k(x_k,n_k)$

若已知状态的初始概率密度函数为$p(x_0|z_0)=p(x_0)$，则状态预测方程为：

状态更新方程为：

68

研究方法和过程

粒子滤波算法原理

序贯重要性采样

（1）从提议分布$q(x_k|x_{k-1},z_k)$中随机抽取n个有限样本；

（2）逐点计算对应的过程动态分布$p(x_k|x_{k-1})$和观测似然分布$p(z_k|x_k)$；

（3）按下式计算对应样本的重要性系数，即样本权值；

（4）权值归一化；

（5）利用下式对系统真实后验概率$p(x_k|z_{1:k})$进行估计。

69

研究方法和过程

粒子滤波算法原理

粒子重采样

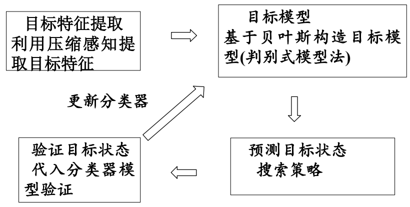
通过对粒子和相应权值表示的概率密度函数重新采样，增加权值较大的粒子数，在采样总数保持不变的情况下，权值较大的粒子被多次复制，从而实现重采样过程。

70

判别式目标跟踪方法

3.1 运动目标跟踪

- 基于判别式的目标跟踪方式-朴素贝叶斯分类器：



3.1 运动目标跟踪

- 目标模型：
  - ➢生成式模型法将目标跟踪视为匹配问题，是先学习一个表示目标的表观模型，然后在目标可能运动的范围内搜索与该生成模型最相似的区域作为跟踪的结果。
  - ➢判别式模型。将跟踪问题视为目标/背景两类的分类问题。该方法利用机器学习中贝叶斯分类器、SVM、adaboost和决策树等方法，在线学习正负样本在特征空间的分类模型，从而实现目标和背景的判别。

3.1 运动目标跟踪

生成式模型法：生成式算法通过描述目标的表观特征，处理当前帧与下一帧的关系，即在图像中搜索与模型最匹配的区域作为跟踪结果。该模型提取目标特征构建表现模型不论采用全局特征还是局部特征，其本质是在目标表示的高维空间中，找到与目标模型最相邻的候选目标作为当前估计，以适应目标在局部的各种尺度和旋转的变化。

✓ 缺点是过于关注目标本身，忽略背景信息，容易产生漂移现象。例如：LK光流法、meanshift 算法等。

3.1 运动目标跟踪

基于判别式模型的方法：

判别式模型将跟踪问题看做分类或回归问题，目的是寻找一个判别函数，将目标从背景中分离出来，从而实现对目标的跟踪。分类判别式模型：判别式算法区分目标和背景，表现比生成式算法更为鲁棒，目前也是更为流行的研究方向。

基于判别式模型目标跟踪方法

- 在现实中通常难以直接获得。机器学习所要实现的是基于有限的训练样本尽可能准确地估计出后验概率 $P(c \mid \mathbf{x})$。

基于判别式模型目标跟踪方法

- 在现实中通常难以直接获得。机器学习所要实现的是基于有限的训练样本尽可能准确地估计出后验概率 $P(c \mid \mathbf{x})$。
- 主要有两种策略：
  - 判别式模型（discriminative models）
    - 给定$\mathbf{x}$，通过直接建模$P(c \mid \mathbf{x})$, 来预测 c
    - 决策树，BP神经网络，支持向量机
  - 生成式模型（generative models）
    - 先对联合概率分布 $P(\mathbf{x}, c)$建模，再由此获得 $P(c \mid \mathbf{x})$
    - 生成式模型考虑

$$P(c \mid \mathbf{x}) = \frac{P(\mathbf{x}, c)}{P(\mathbf{x})}$$

贝叶斯决策论

- 生成式模型

$$P(c \mid \mathbf{x}) = \frac{P(\mathbf{x}, c)}{P(\mathbf{x})}$$

贝叶斯决策论

- 生成式模型

$$P(c \mid \mathbf{x}) = \frac{P(\mathbf{x}, c)}{P(\mathbf{x})}$$

□基于贝叶斯定理，$P(c \mid \mathbf{x})$可写成

$$P(c \mid \mathbf{x}) = \frac{\boxed{P(c)}P(\mathbf{x} \mid c)}{\boxed{P(\mathbf{x})}}$$

先验概率
样本空间中各类样本所占的比例，可通过各类样本出现的频率估计（大数定理）

用于归一化的“证据”（evidence）因子，与类标记无关

贝叶斯决策论

- 生成式模型

$$P(c \mid \mathbf{x}) = \frac{P(\mathbf{x}, c)}{P(\mathbf{x})}$$

$P(c \mid \mathbf{x})$

□基于贝叶斯定理，　　可写成

类标记 c相对于样本 $\mathbf{x}$ 的“类条件概率”（class-conditional probability)，或称“似然”。

$$P(c \mid \mathbf{x}) = \frac{\boxed{P(c)}\boxed{P(\mathbf{x} \mid c)}}{\boxed{P(\mathbf{x})}}$$

先验概率
样本空间中各类样本所占的比例，可通过各类样本出现的频率估计（大数定理）

“证据”（evidence）因子，对给定样本x, p(x)与类标记无关

贝叶斯分类器

Training data:

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

(1)估计P(X,Y)，即估计P(X|Y)和P(Y)
(2)根据贝叶斯定理计算$P(Y \mid X^{new})$

如何计算P(X|Y)?

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y) = P(X_1|Y)P(X_2|Y)$$

主要障碍：所有属性上的联合概率，难以从有限训练样本估计获得

朴素贝叶斯分类器

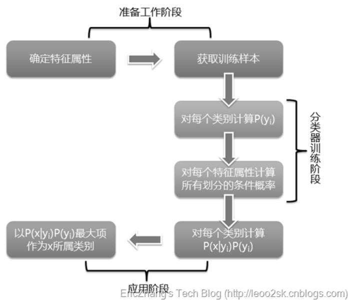
属性条件独立：在给定Y下，假设X1,..Xn属性相互独立

$$P(X_1 \dots X_n|Y) = \prod_i P(X_i|Y)$$

去分母

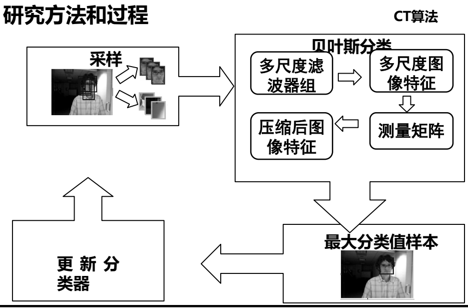
3.2 运动目标跟踪-基于判别式的目标跟踪法

朴素贝叶斯分类器



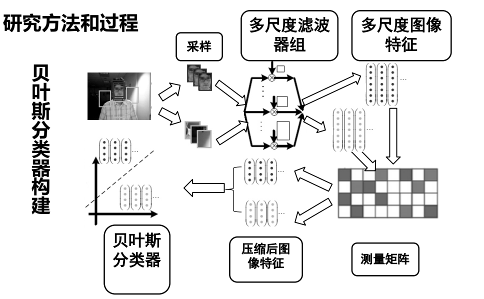
3.2 运动目标跟踪-基于朴素贝叶斯的目标跟踪

研究方法和过程



3.1 运动目标跟踪-基于朴素贝叶斯的目标跟踪

研究方法和过程



3.2 运动目标跟踪-基于判别式的目标跟踪法

基于朴素贝叶斯分类器的目标跟踪

跟踪采用算法为：通过第一帧手动框选的目标周围采集正负样本，训练朴素贝叶斯分类器，对当前帧图像中，以上一帧目标区域为中心，采用扫描窗口25像素进行检测采样，将采样得到的样本用朴素贝叶斯分类找到匹配度最高的那个样本，作为当前帧的目标。

演示　　总结

3.4运动目标跟踪-生成式模型的目标跟踪

基于团块的目标跟踪框架：

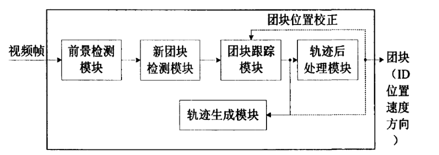
- 基于团块(blob)的跟踪算法基本原则是对候选像素进行图像分割，判断该像素是属于目标还是属于背景或者属于其他的区域。
- 该跟踪算法也可以称为基于图像分割的跟踪，因为它和图像分割所使用的基本方法是一致的，都是根据给定线索优化像素的选择、合并和分离，都把具有相同特征的像素点集合成一个区域。线索一般有目标的运动特征、纹理特征和图像深度信息等。

3.4 基于团块的运动物体跟踪

目标：希望把运动着的物体检测出来，对其编号并进行跟踪

block-团块跟踪\c-example-blobtrack_sample.bat

3.4 基于团块的运动物体跟踪



整个算法细分成几个模块实现(跟踪框架)：

（1）前景检测模块　（2）新团块检测模块

（3）团块跟踪模块　（4）轨迹生成模块

（5）轨迹后处理模块

3.4基于团块的运动物体跟踪

- **运动物体的前景检测模块**：判断每一个像素是前景还是背景，如可以利用混合高斯模型的背景差分法。输入数据为当前帧图像，输出为当前帧的前景掩码。OpenCV可以检测出完全静止的背景，实际运用中背景可能发生平移或旋转，为了满足背景变化也可以检测出运动物体，那就需要把摄像头的标定结合进来。
- **运动物体的团块特征检测模块**：使用前景检测的结果检测场景中运动物体的像素集合——团块，输入数据为当前帧的掩码和己有的团块，输出数据为新检测到的团块。为了满足在雷达系统上的应用，团块的图形特征和运动相关性分析都要考虑，而且要考虑团块之间以及团块与背景像素融合以后的处理。

3.4基于团块的运动物体跟踪

- **运动物体的团块跟踪模块**：包括团块列表和单团块跟踪，使用新团块检测的结果，输入数据为为当前帧、当前帧掩码和新团块，输出数据为当前帧的团块信息：ID号，位置，速度以及运动方向。可以利用连通分量跟踪、Meanshift和粒子滤波实现跟踪。
- **轨迹生成模块**：收集所有团块的位置，建立轨迹列表，进行相关性处理。
- **轨迹后处理模块**：进行运动物体标定时去除位置和运动趋势的抖动，对轨迹进行平滑处理。

3.4基于团块的运动跟踪实验结果图

- 对视频中运动车辆的检测，多数情况下能够检测出来，但有时新进入画面的车辆检测不到(主要受运动物体的前景检测模块性能影响)。标示出的运动方向和部分轨迹基本正确。

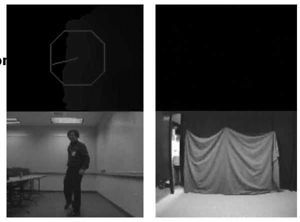
3.4.4 运动历史图-计算运动方向-行为分析

利用运动历史图(MHI-Motion History Images)可以求取出运动目标的大概运动方向(横线由圆心向外所指方向，实际上是用角度表示)，如下图所示：

演示c-example-motempl.exe

运动历史图

Overlay silhouettes, take gradient for normal op flow. Flood fill to segment motions.



3.4.4 运动历史图-(2)计算运动方向

- 运动历史图：基于的条件是运动目标相邻两帧之间在画面上存在交集，如海上雷达图像上的目标和交通路口的车辆都符合此条件。
- 此算法不需要外推和相关分析以及轨迹后处理就可以很清晰地显示出目标的轨迹、速度与方向，该算法基于帧差的运动目标检测出来的运动目标前景图像。具体的过程描述如下：

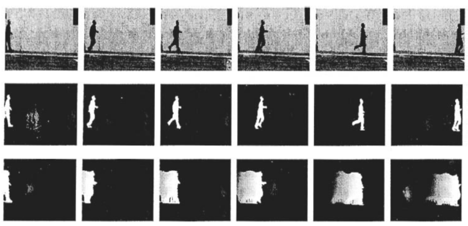
3.4.4 运动历史图-(2)计算运动方向

- 运动历史图**MHI**中每个像素的亮度都是这一点处运动历史的函数。

$$H_\tau(x,y,t)=\begin{cases}\tau & \text{if } D(x,y,t)=1\\ \max(0,H_\tau(x,y,t-1)-1) & \text{otherwise.}\end{cases}$$

其中τ为行为持续运动的时间。MHI是一个标量值图像，敏感于运动的方向，灰度的变化体现出行为运动的方向。最近运动的像素亮度值最大。MHI的生成很好地把运动的空间特征和时间信息都体现在图像中。它既在空间上很好地描述了运动发生的位置和区域，同时在时间上反映了运动是如何发生的。 图给出了人物奔跑形成MHI的全过程，从上至下依次为原始图像、二值化图像、MHI。

3.4.4 运动历史图-(2)计算运动方向



3.4.4 运动历史图-(2)计算运动方向

1. 存储检测出来的目标前景图像，过去的前景结果图帧灰度递减
2. 当前结果帧打上时间戳叠加存储到历史图像中；
3. 求运动历史图的梯度，求得运动梯度的方向图像。
4. 在运动历史图，分割成独立的运动区域。
5. 计算每个分割区域的运动方向：在选择的分割区域内依据方向图计算选择区域的整个运动方向，并且返回0到360之间的角度值。

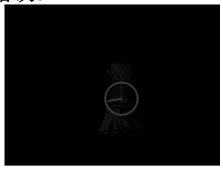
该算法简化了目标相关性的运算，可以在初始状态下对于目标运动趋势不了解的情况下实施对目标的稳定跟踪，同时具有非常高的实时性能。

演示c-example-motempl.exe

3.4.4 运动历史图-(2)计算运动方向

- 对视频中运动人体的检测，多数情况下检测的方向是正确的，在视频开始的前几帧和结束的几帧处检测的方向会出现错误。

- 对视频中运动人体的检测，多数情况下检测的方向是正确的，在视频开始的前几帧和结束的几帧处检测的方向会出现错误。



3.4.4 运动历史图-实验结果图

- 由于此方法对视频中运动人体的检测会出现将人体上身和下身分开检测的问题，检测出的运动人体的中心往往不是唯一的，因此，描绘出的运动轨迹并不准确，显得杂乱无章。

3.3 运动历史图-实验结果图

- 针对轨迹的问题，将此方法进行了改进：添加膨胀操作，此操作可以消除目标的不连续空洞。之后利用函数找到图像中所有轮廓，将轮廓中心保存在数组中，再用画线函数画出运动轨迹。

- 依据改进算法，能够将运动人体看做一个整体进行检测和跟踪，画出的轨迹比较准确。

3.4.4基于运动历史图的总结

运动历史图：

适用于简单的单运动目标检测和跟踪，针对单个低速的运动目标有良好的检测和跟踪效果，能够正确的检测运动方向，并描绘出清晰、平滑的运动轨迹。

不适合于运动速度太快的目标方向检测。

mhi - video.bat

本章小节

- 本节课介绍了几种常规的运动检测方法：
  - 运动检测
    - 差法
    - 帧间差分
    - 光流法
  - 目标跟踪及行为理解
    - 基于团块的运动目标跟踪
    - 运动历史图

本章小节

- 运动检测的难点：
  - 受诸多外界因素以及背景物体内在因素的影响，图像中的背景常常是动态变化的。
- 运动检测的两种总体思路：
  - 直接利用前景所特有的信息检测前景；
  - 先得到背景图象，然后将输入图象减去背景前景得到前景图象。
- 运动跟踪难点：
  - 跟踪的目标复杂多变，目标跟踪的应用环境复杂多变。

课堂练习题

运动目标跟踪主要步骤？

课堂练习题

生成式目标跟踪模型和判别式目标跟踪模型方法原理与区别？