

计算机视觉——目标跟踪

2022年春季
桑农 王岳环



目标跟踪任务

Image 1

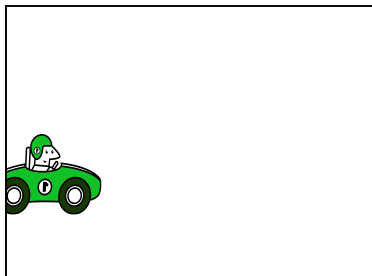


Image 2

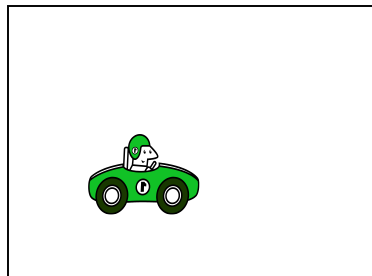


Image 3

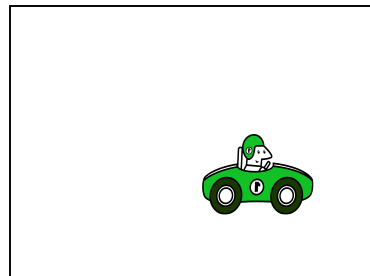
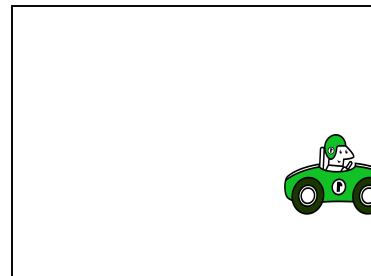
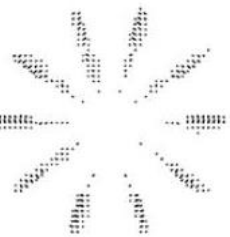


Image 4

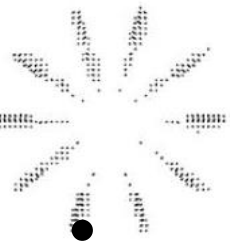


- 可以确定目标的位置吗？
- 可以确定目标的速度吗？
- 可以预测目标的位置吗？

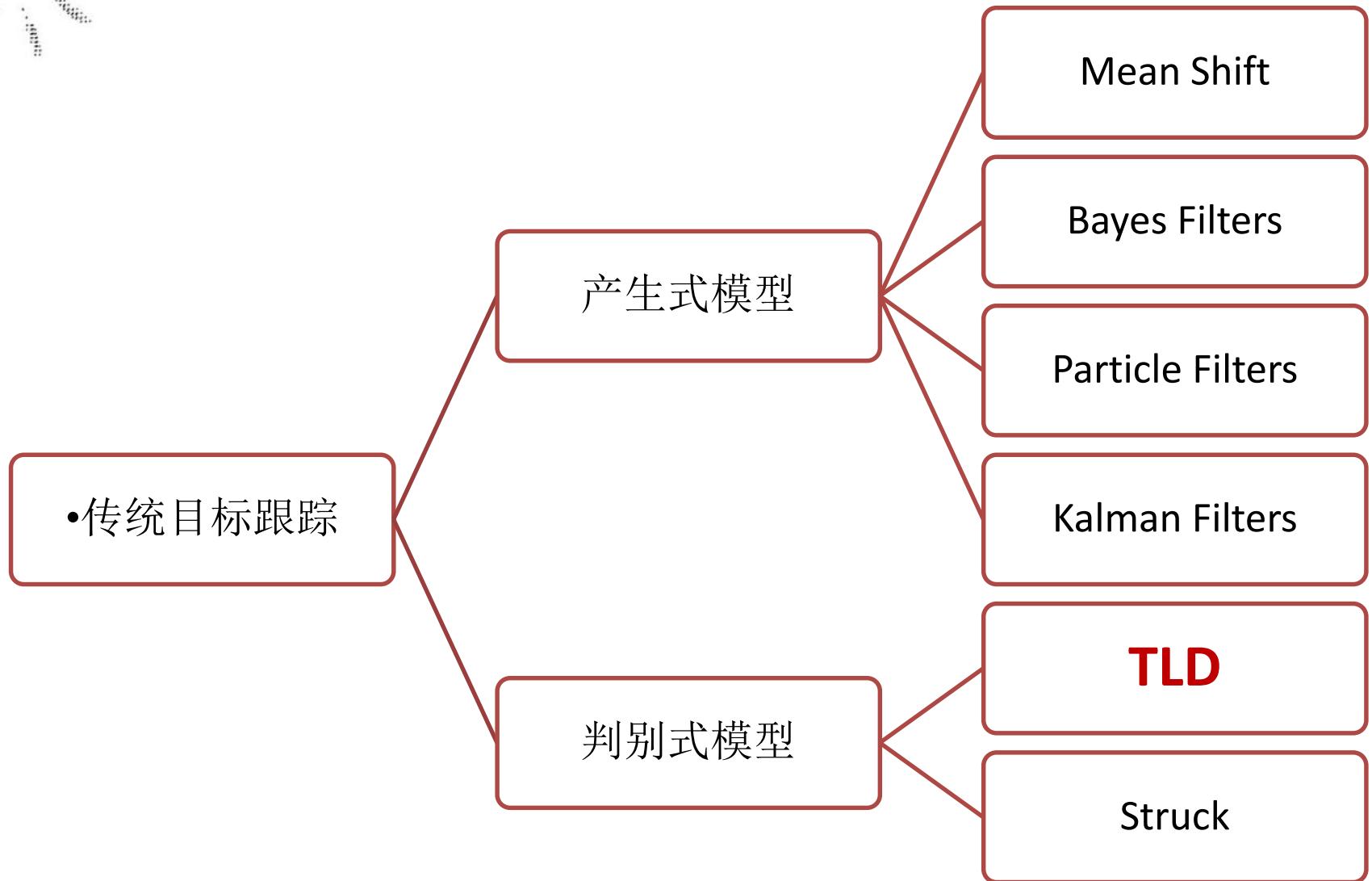


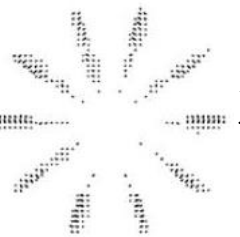
目标跟踪问题

- 给定图像序列
- 找到运动目标的位置
- 摄像头可以是禁止或者运动
- 假定：每一幅图像中都能找到目标
- 问题：在序列图像上进行跟踪
- 目标跟踪是计算机视觉的基本问题之一



目标跟踪方法





基于检测的目标跟踪方法

- 检测 VS 跟踪



t=1



t=2

...



t=20



t=21

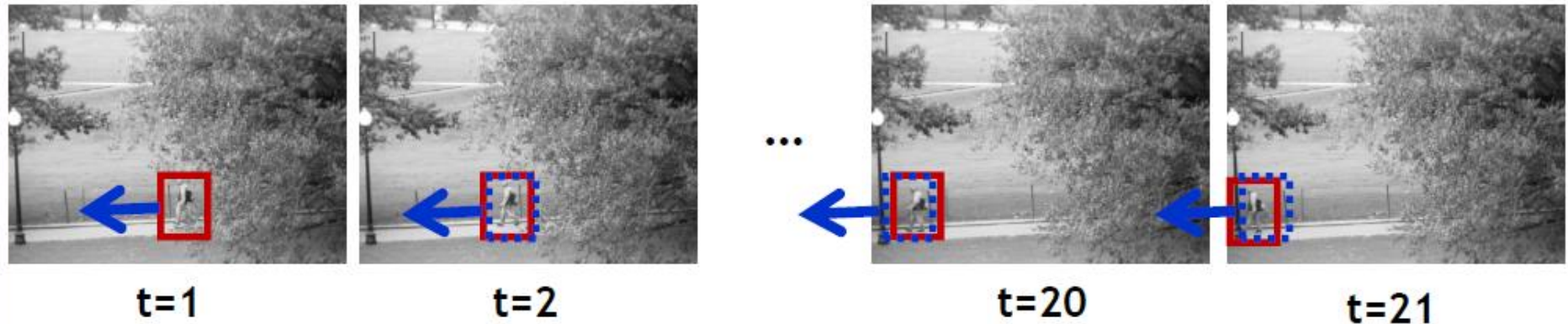
Detection vs. Tracking



- **Detection**

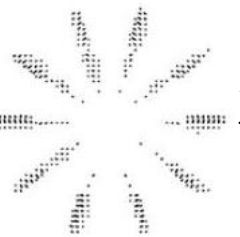
- We detect the object independently in each frame and can record its position over time, e.g., based on blob's centroid or detection window coordinates.

Detection vs. Tracking



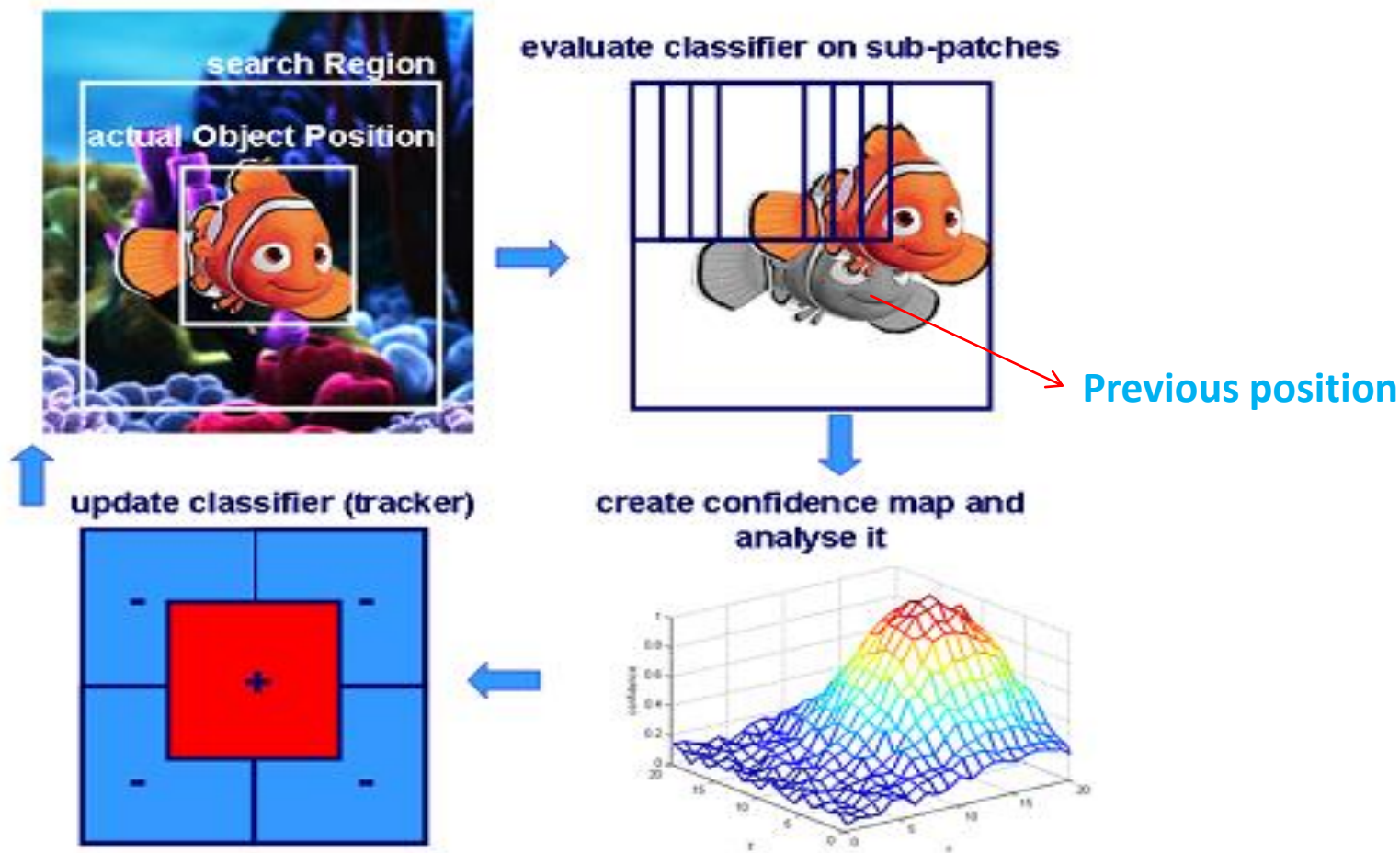
- Tracking with *dynamics*:

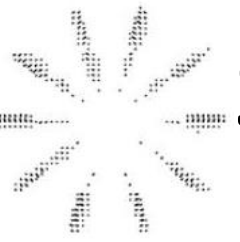
- We use image measurements to estimate the object position, but also incorporate the position predicted by dynamics, i.e., our expectation of the object's motion pattern.



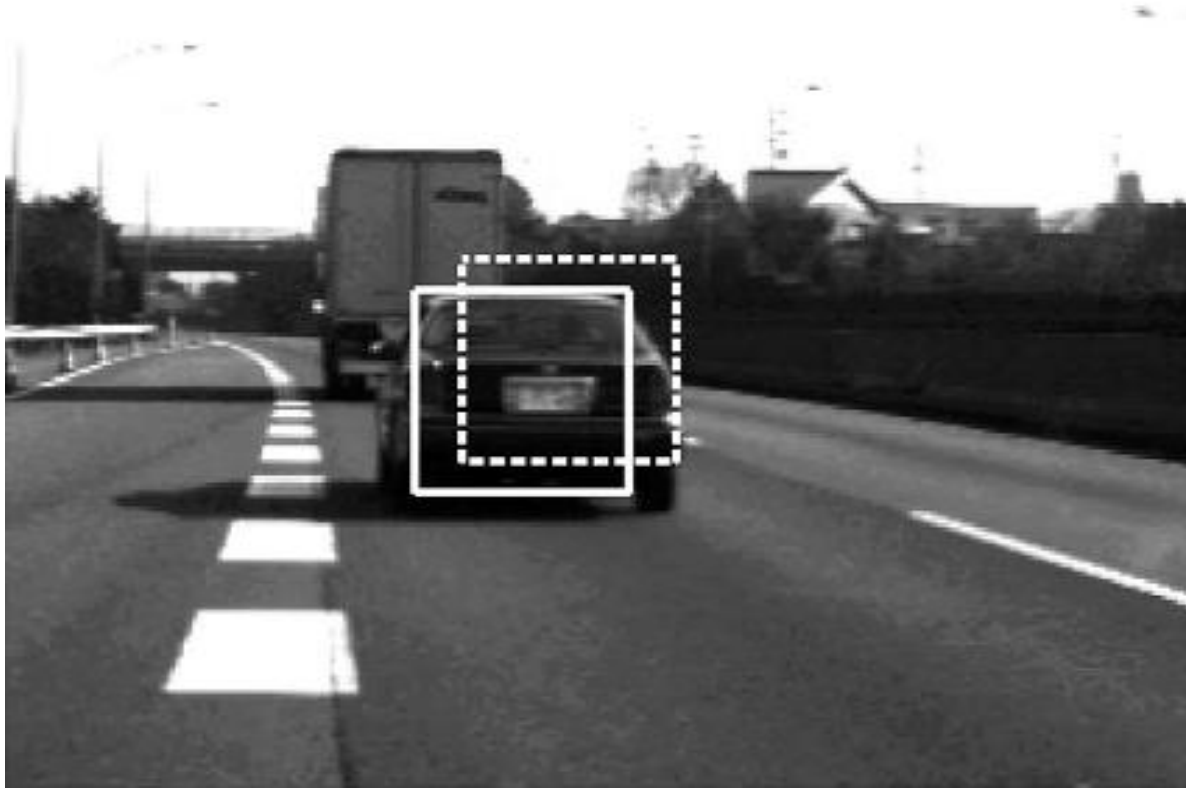
基于检测/分类的目标跟踪框架

Different
sampling
methods

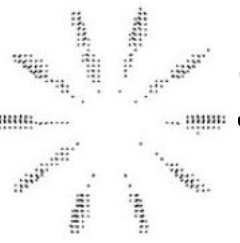




Support Vector Tracking



1: Avidan S., [Support Vector Tracking](#). IEEE Trans. On Pattern Analysis and Machine Intelligence, 2004.



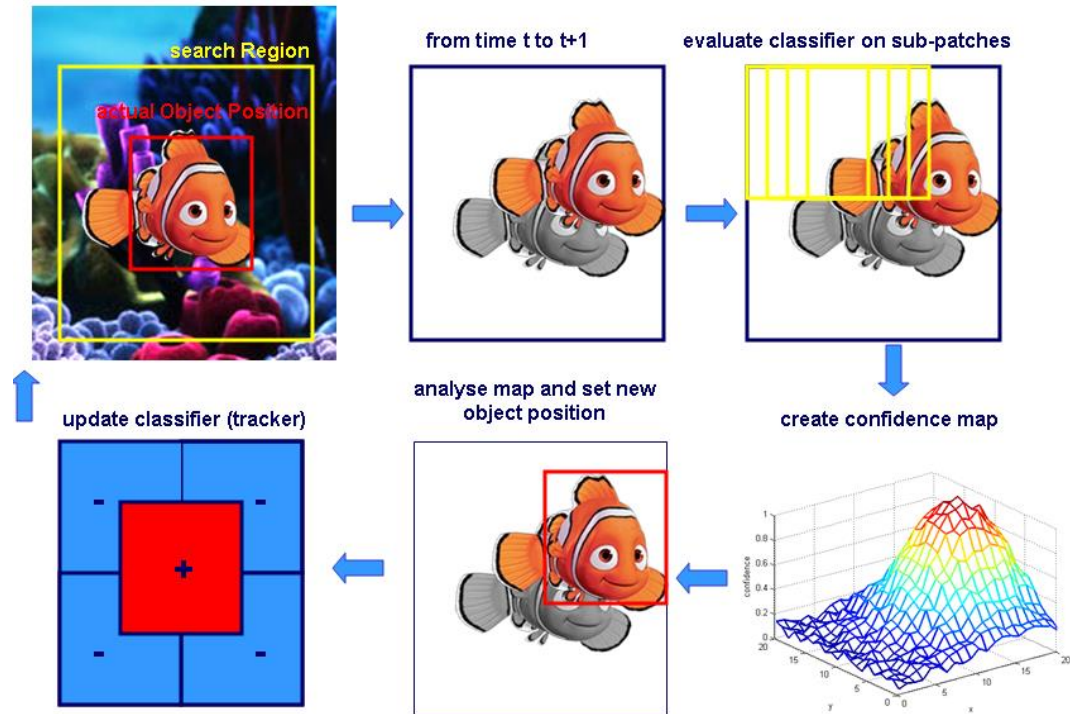
Support Vector Tracking

- 主要跟踪汽车的尾部
- 方法：
 - 求取边缘图像（主要基于这种特殊应用，汽车尾部图像水平垂直边缘比较强）获取候选image patches
 - 用离线训练的SVM分类器来进一步验证该候选image patch是否是真正的目标
- 特点
 - 必须有关于目标的先验知识，适应性不强

On-line boosting

Grabner, H., Bischof, H.: **On-line boosting and vision**. In: Proc. CVPR.
Volume 1. (2006) 260-267

Grabner, H., Grabner, M., Bischof, H. **Real-Time Tracking via On-line Boosting**.
BMCV, 2006



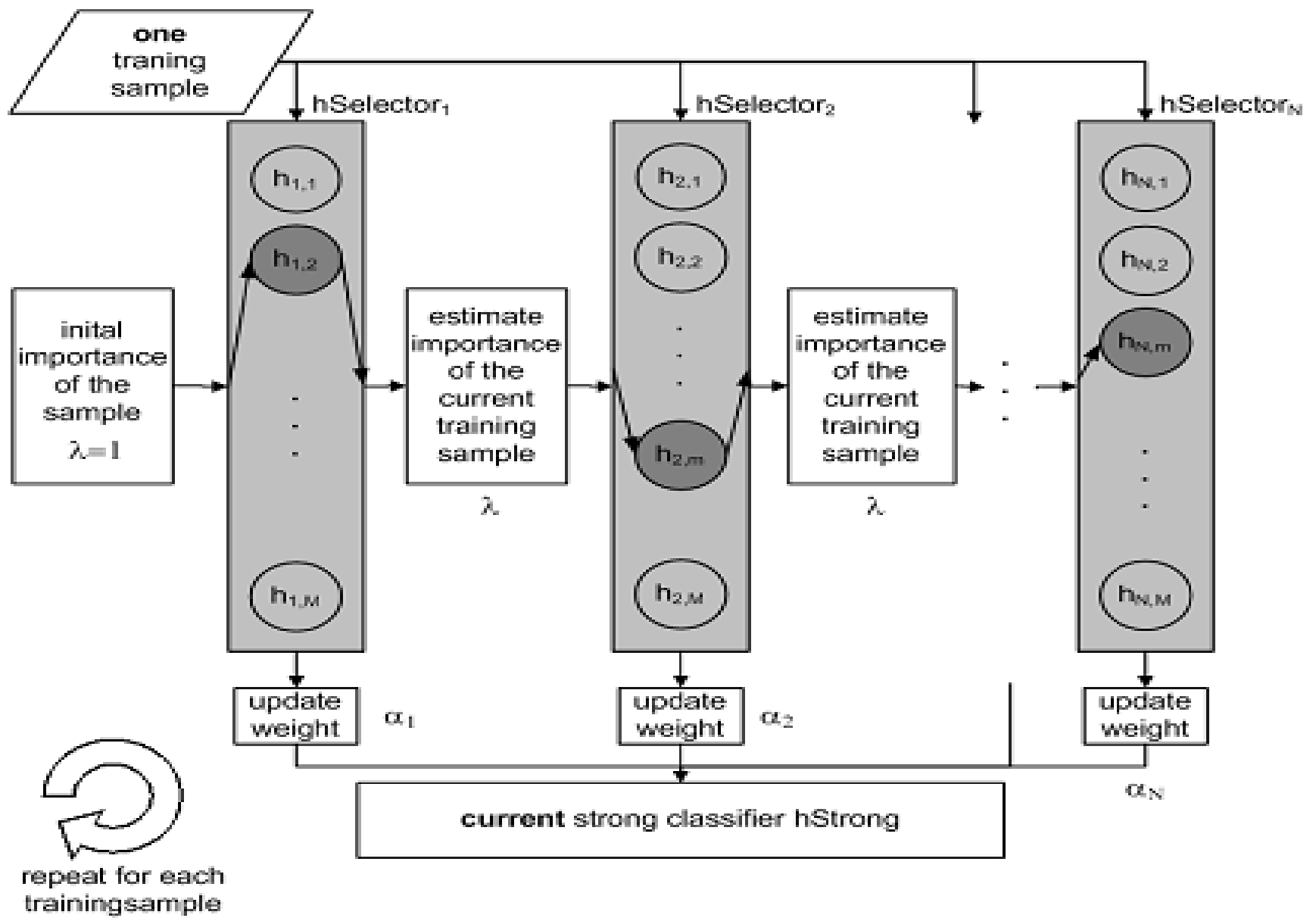
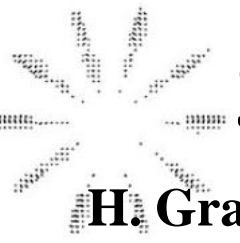
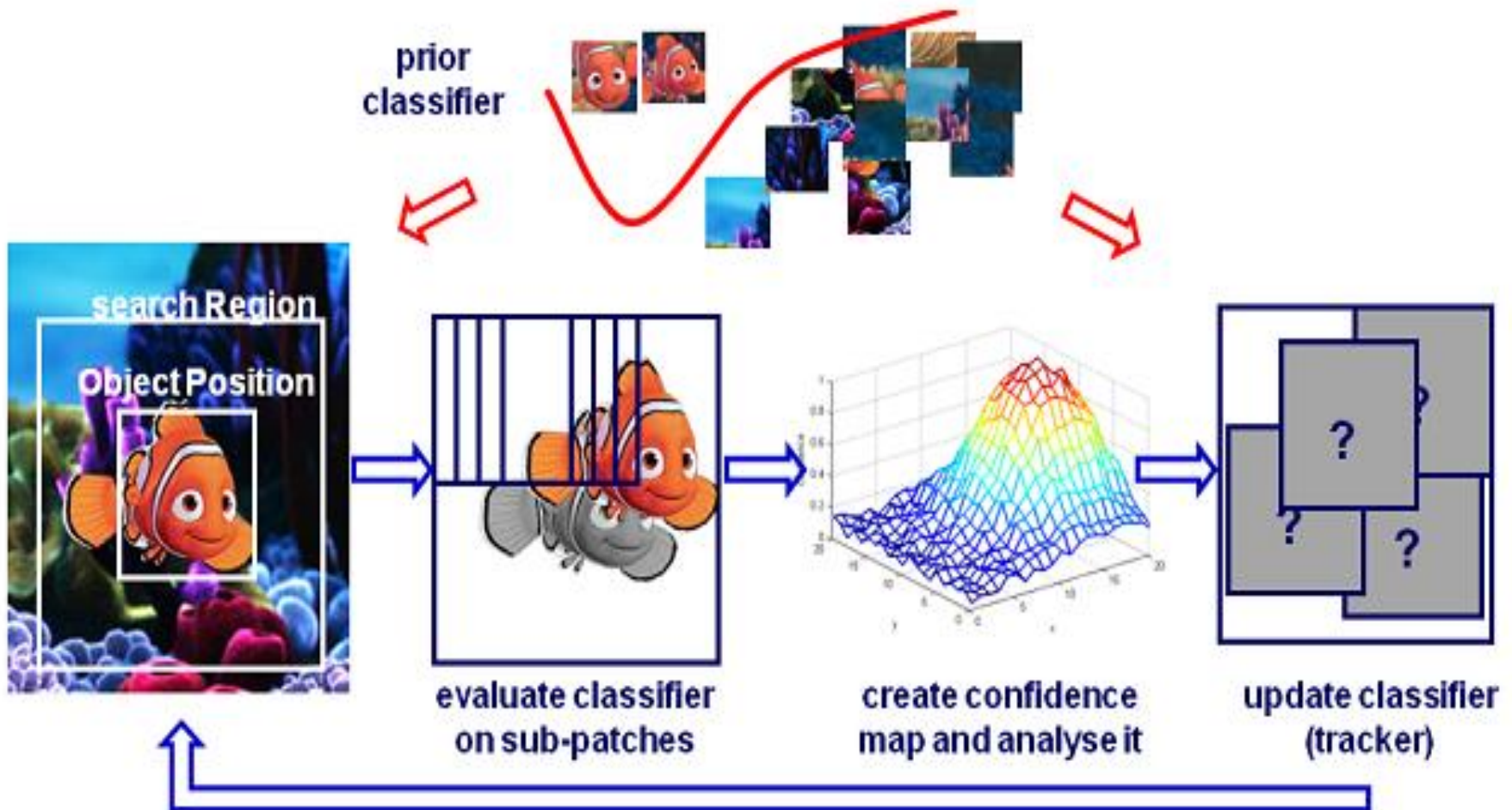


Figure 1. Novel on-line boosting for feature selection.

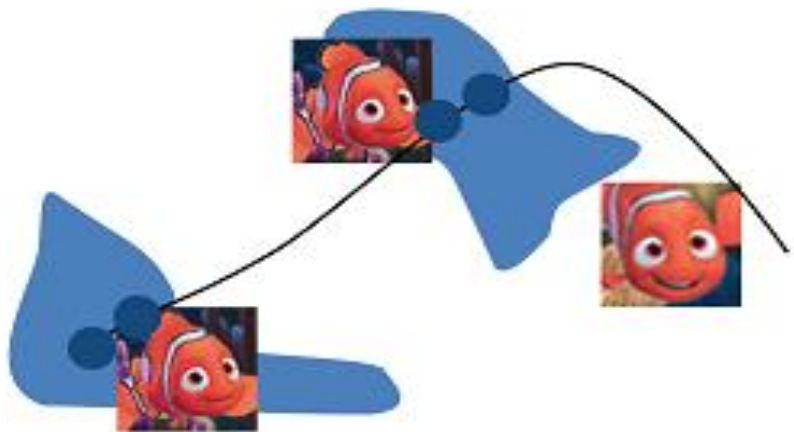


Semi-supervised on-line boosting

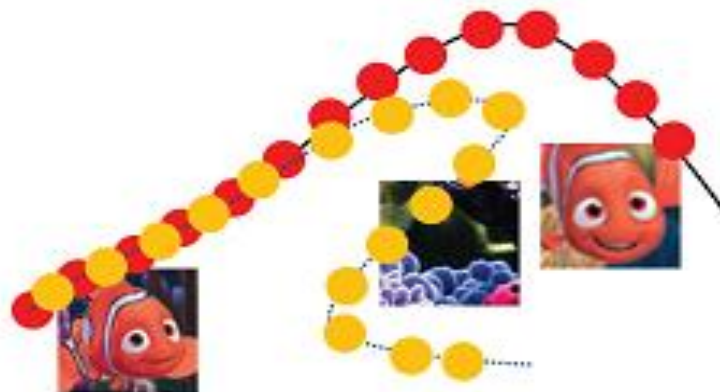
H. Grabner, C. Leistner, and H. Bischof. **Semi-supervised on-line boosting for robust tracking.** In Proc. ECCV, 2008.



漂移 vs. 适应性



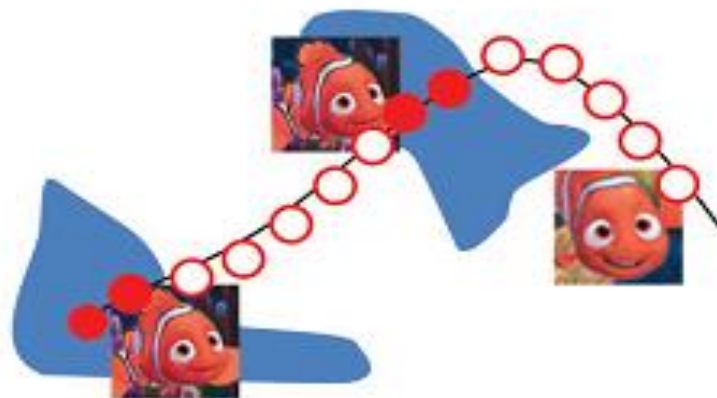
(a) Detection



(b) Supervised tracking



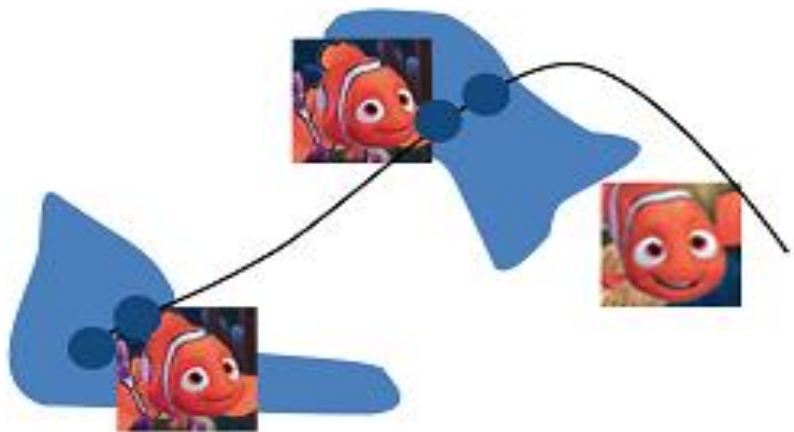
(c) Semi-supervised tracking



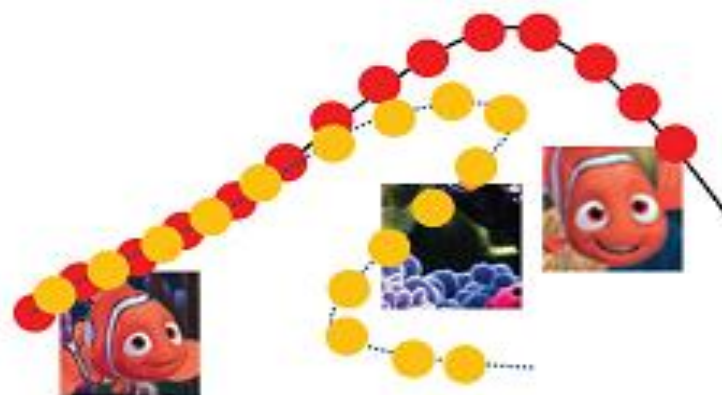
(d) Our Approach

Detection: 固定的检测器，避免漂移的能力，但是没有自适应能力

漂移 vs. 适应性



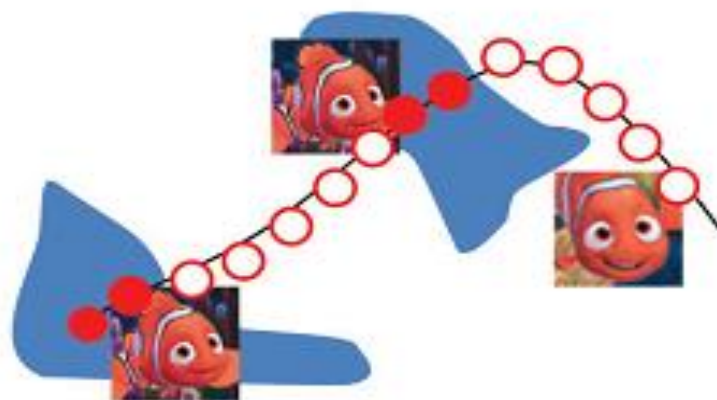
(a) Detection



(b) Supervised tracking



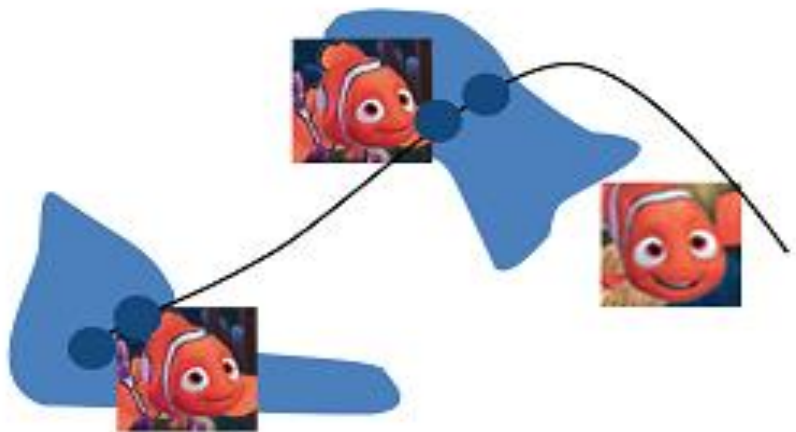
(c) Semi-supervised tracking



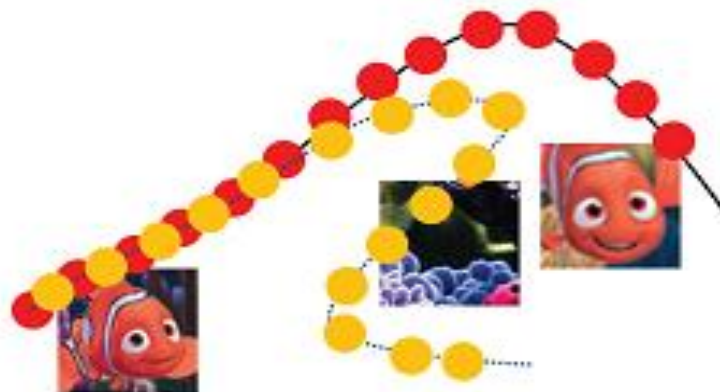
(d) Our Approach

Supervise tracking: 适应目标变化

漂移 vs. 适应性



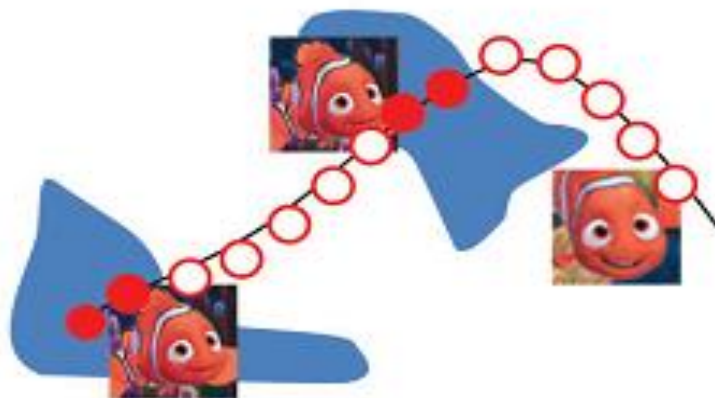
(a) Detection



(b) Supervised tracking



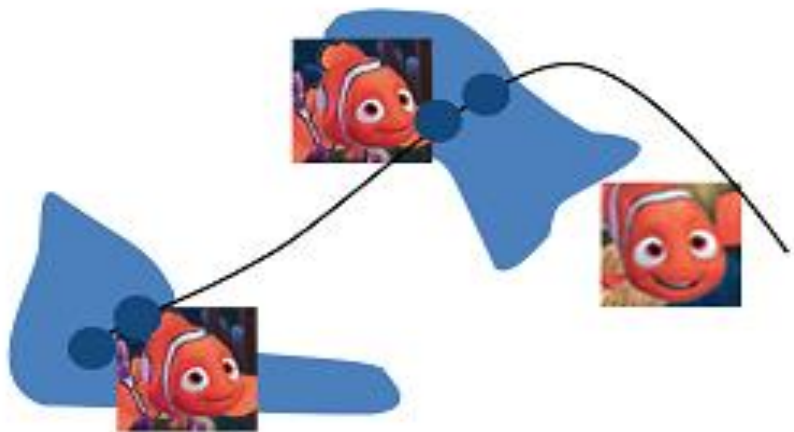
(c) Semi-supervised tracking



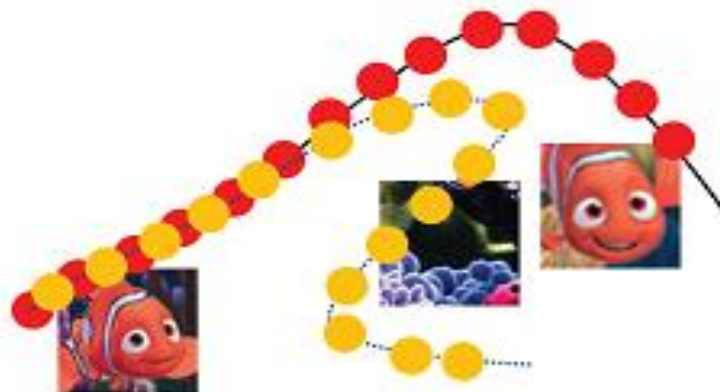
(d) Our Approach

Semi-supervised tracking: 漂移和自适应直接平衡

漂移 vs. 适应性



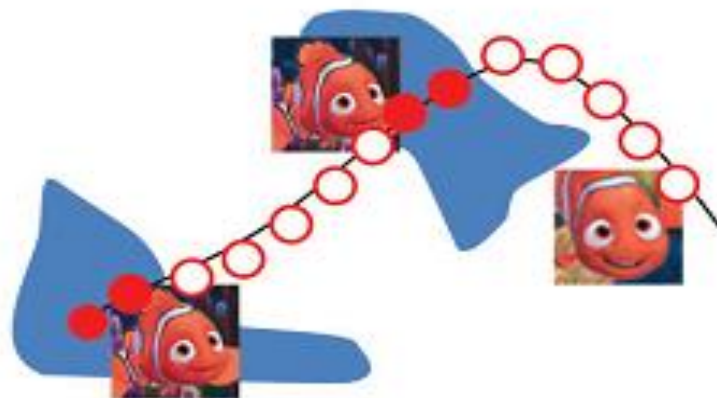
(a) Detection



(b) Supervised tracking

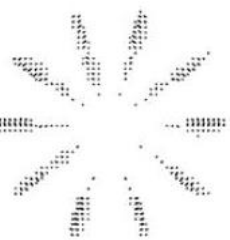


(c) Semi-supervised tracking



(d) Detector, recognizer and tracker

在半监督基础上又加了对特定目标的先验

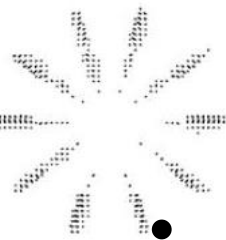


TLD 目标跟踪方法

TLD: Tracking-Learning-Detection

作者相关文章:

- Z. Kalal, K. Mikolajczyk, and J. Matas, "[Tracking-Learning-Detection](#)," Pattern Analysis and Machine Intelligence 2011.
- Z. Kalal, K. Mikolajczyk, and J. Matas, "[Face-TLD: Tracking-Learning-Detection Applied to Faces](#)," International Conference on Image Processing, 2010.
- Z. Kalal, K. Mikolajczyk, and J. Matas, "[Forward-Backward Error: Automatic Detection of Tracking Failures](#)," International Conference on Pattern Recognition, 2010, pp. 23-26.
- Z. Kalal, J. Matas, and K. Mikolajczyk, "[P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints](#)," Conference on Computer Vision and Pattern Recognition, 2010.



TLD 目标跟踪方法

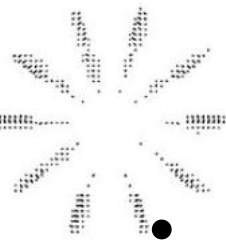
- 目标

长期, 实时 目标跟踪 (未知类别)

- 挑战

- 表观变化
- 部分或全部遮挡
- 目标可能短期消失
- 相似物体
- 实时性能
-

模型更新:
逐帧更新
选择更新



TLD 目标跟踪方法

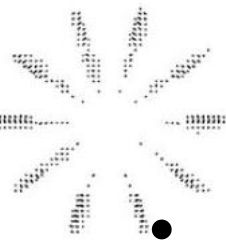
- 目标

长期, 实时 目标跟踪 (未知类别)

- 挑战

- 表观变化
- 部分或全部遮挡
- 目标可能短期消失
- 相似物体
- 实时性能
-

通过检测器初始化跟踪器



TLD 目标跟踪方法

- 目标

长期, 实时 目标跟踪 (未知类别)

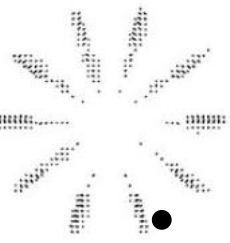
- 挑战

- 表观变化
- 部分或全部遮挡
- 目标可能短期消失
- 相似物体
- 实时性能
-



重识别

A light orange rectangular callout box with a blue border and a blue pointer tail pointing towards the '目标可能短期消失' (target may disappear for a short time) item in the list of challenges.



TLD 目标跟踪方法

- 目标

长期, 实时 目标跟踪 (未知类别)

- 挑战

- 表观变化
- 部分或全部遮挡
- 目标可能短期消失
- 相似物体
- 实时性能
-

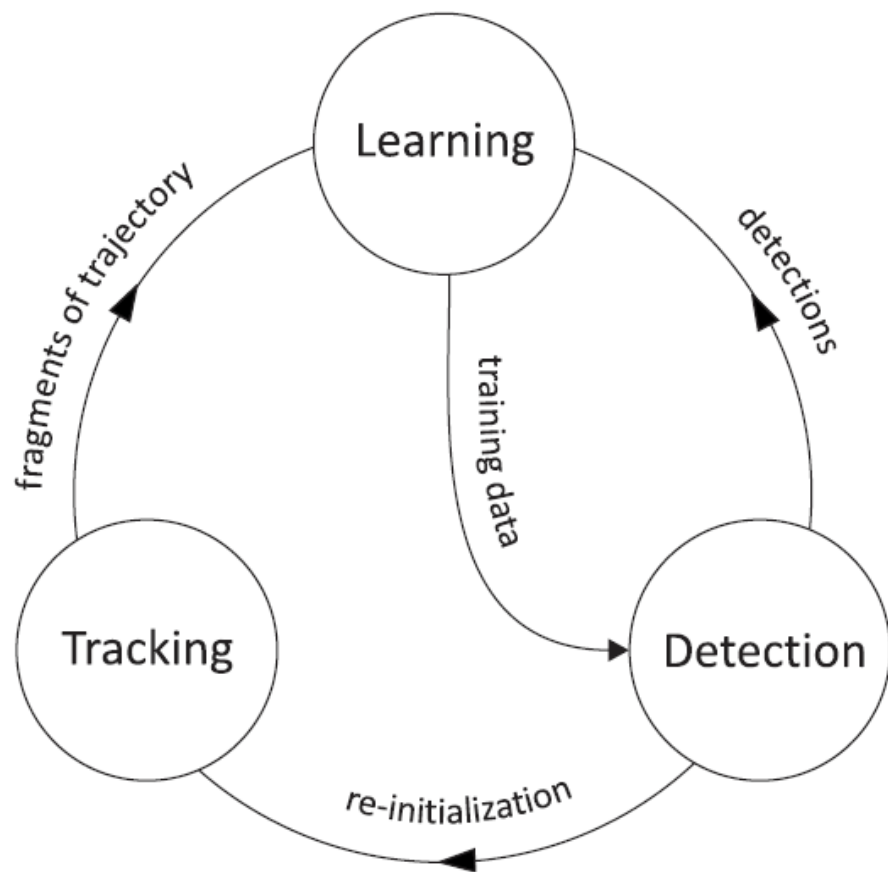
- 方法

- TLD: a Framework that integrates adaptive Tracking with online Learning of object-specific Detector

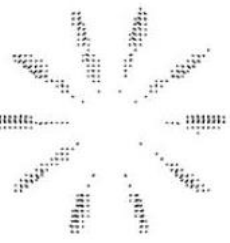


TLD目标跟踪方法

- TLD 框架

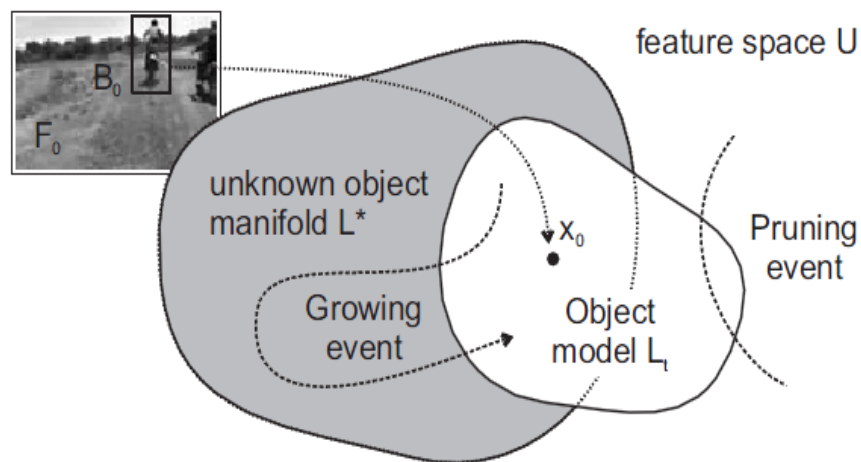


- **跟踪器**是观察帧与帧之间的目标运动
- **检测器**将每帧图像当成独立的，然后去定位。检测器可以定位之前所有出现的目标
- **学习器**评估检测器的误差，并更新检测器防止后续错误



TLD 目标跟踪方法

- TLD 框架



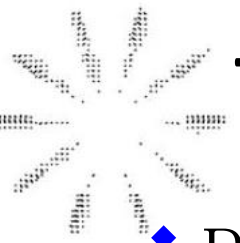
- **Growing events** 通过在线模型扩展空间
- **Pruning events** 去除错误样本



TLD目标跟踪方法

❖ P-N learning

- **目的：**提高检测器性能。采用半监督学习方法进行在线检测器更新，每一帧评估当前检测器，针对错误情况进行更新避免将来出现类似错误
- **两类检测器错误：**假阴 && 假阳
- P-N learning主要思想是利用两类“专家（expert）”解决这两类错误
 - **P-expert**：识别假阴
 - **N-expert**：识别假阳



TLD目标跟踪方法

❖ P-N learning

- P-N learning**主要思想是利用两类“专家（expert）”解决这两类错误**
 - **P-expert**: 识别假阴
 - **N-expert**: 识别假阳
- **专家的特点**
 - 采用独立的信息
 - 可以反驳对方
 - 可能发生错误



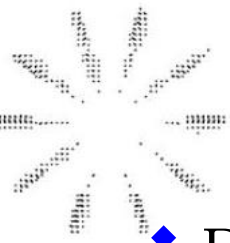
TLD 目标跟踪方法

❖ P-N learning

■ 数据

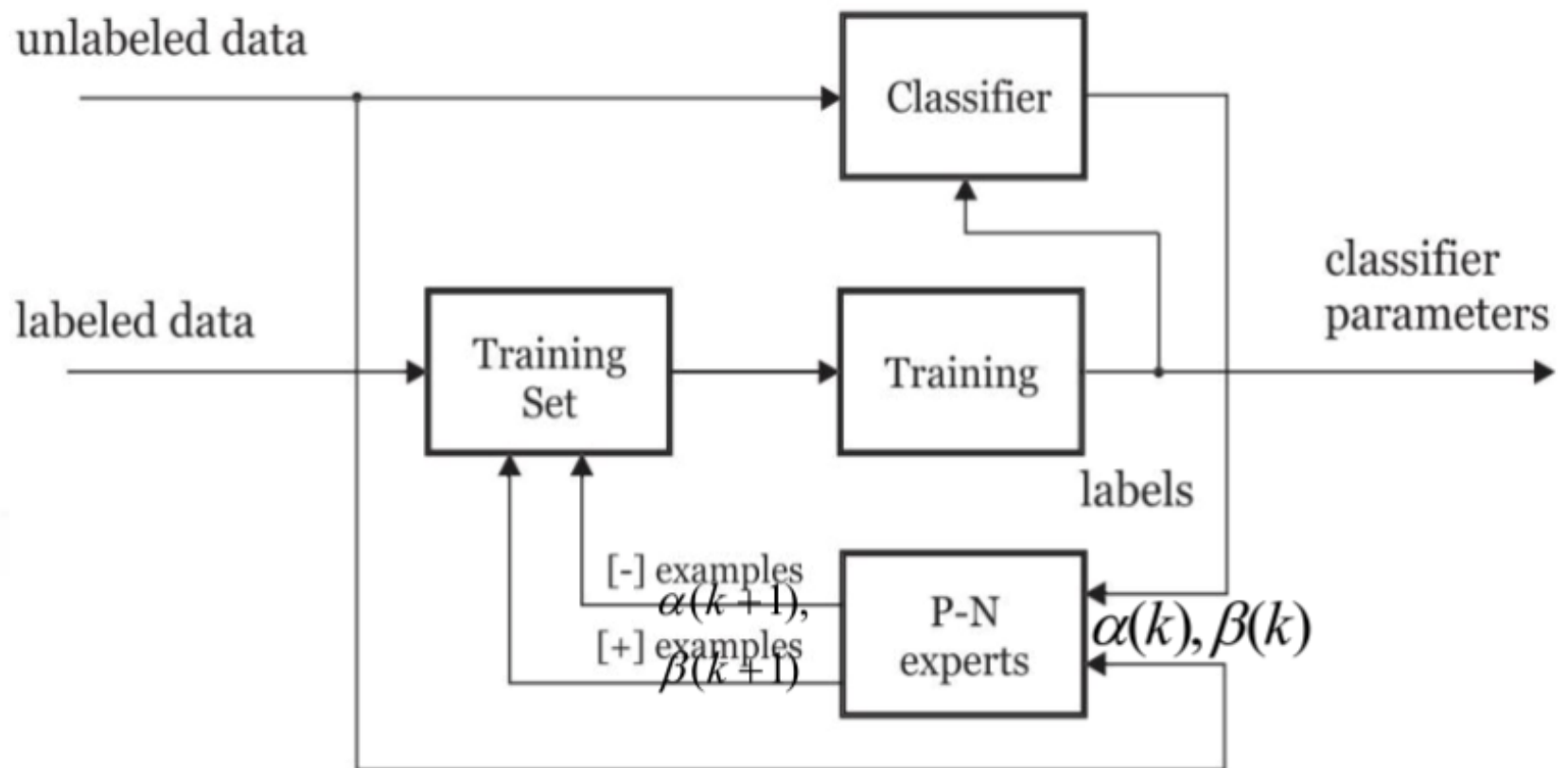
- 标记数据集: $L_l = (\mathcal{X}, \mathcal{Y})$
- 未标记数据集: $X_u = (X)$

- P-N学习的任务是根据标记数据集学习一个分类器 $f: \mathcal{X} \rightarrow \mathcal{Y}$ ，并利用未标记数据集提升其性能
- 分类器 $f: \mathcal{X} \rightarrow \mathcal{Y}$ 为一个来自由参数 θ 参数化的函数族 F 的函数，训练过程和参数 θ 的估计有关



TLD 目标跟踪方法

❖ P-N learning



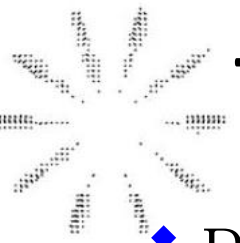


TLD目标跟踪方法

❖ P-N learning

■ Procedures of P-N Learning

- 学习过程的初始化：将标记数据集 L 嵌入训练集
- 训练集用于监督学习，训练一个分类器，即估计初始参数 Θ^0
- 在第 k 次迭代中，利用之前训练好的分类器对未标记样本进行分类 $y_u^k = f(x_u | \Theta^{k-1})$, for all $x_u \in X_u$.
- P-N experts对分类进行性能评估，找到错误分类的样本，并加入训练集
- 重新训练分类器，即估计参数 Θ^k
- 迭代，直到收敛或者达到迭代次数



TLD 目标跟踪方法

❖ P-N learning

■ P-N Learning 的关键要素

- 估计分类器的错误：将两种错误分别处理

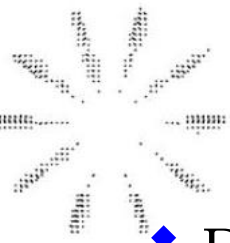


TLD目标跟踪方法

❖ P-N learning

■ P-N Learning的关键要素

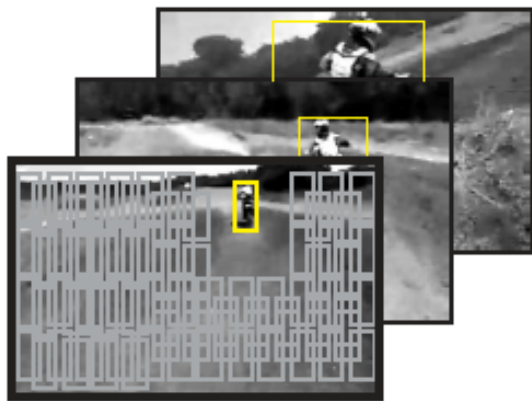
- 估计分类器的错误：将两种错误分别处理
- 无监督的数据集根据当前分类器分为两部分，每部分都用不同的expert处理
 - P-expert 分析错分的负样本，赋予正的标签并加入训练集。在第 k 次迭代，P experts输出 $n+(k)$ 个正样本
 - N-expert 分析错分的正样本，赋予负的标签并加入训练集。在第 k 次迭代，N experts输出 $n-(k)$ 个负样本



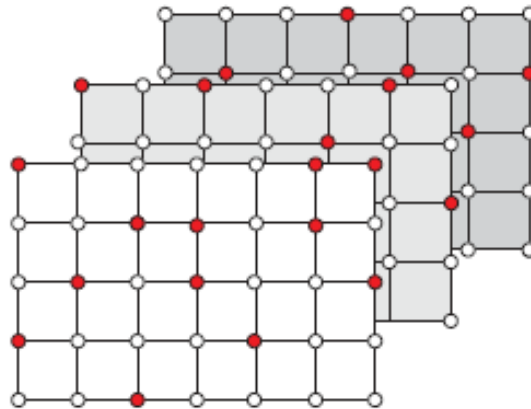
TLD 目标跟踪方法

❖ P-N learning

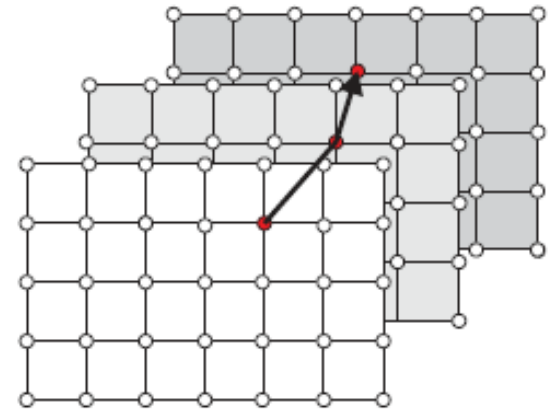
■ Design of experts



a) scanning grid

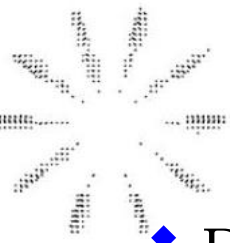


b) unacceptable labeling



c) acceptable labeling

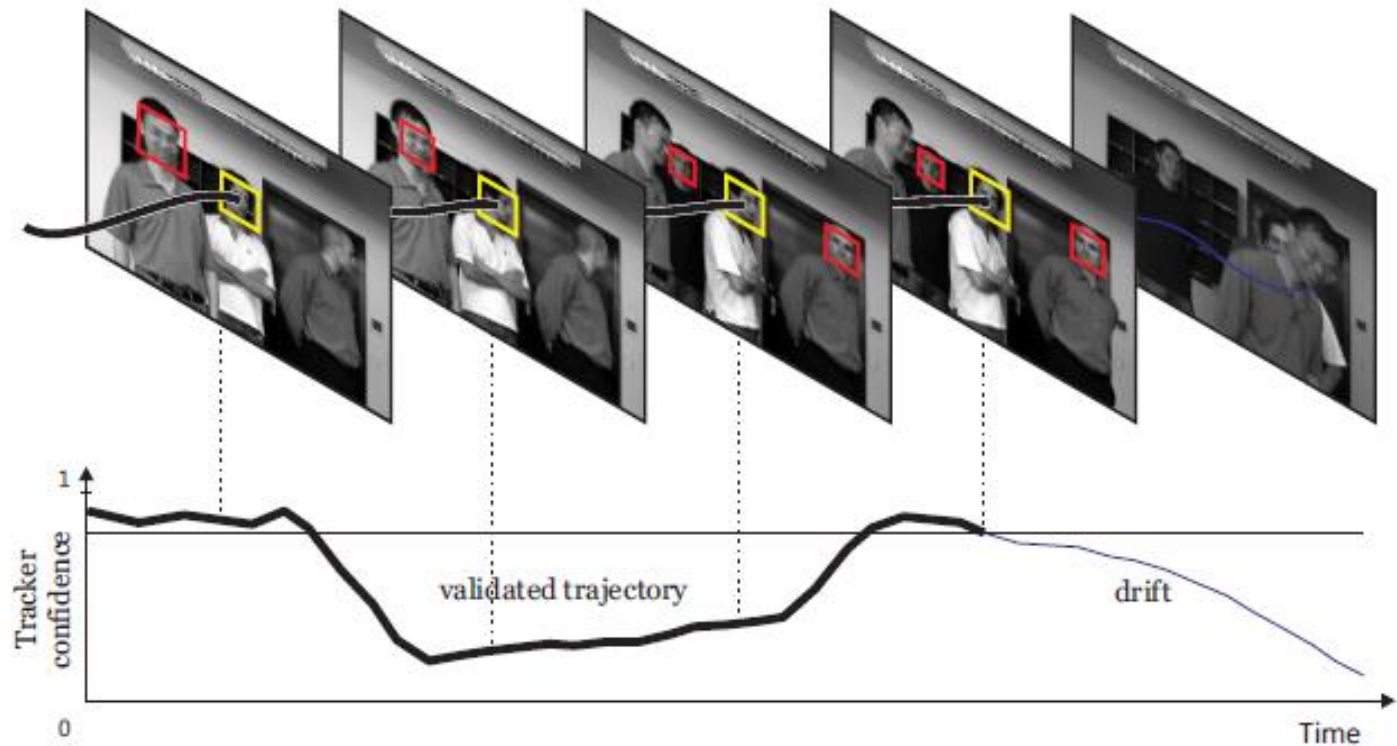
- 每个图像块的标签是**独立的**
- 这种唯一连续的性质称为**结构性**
- P-N 学习的关键就是找到这种结构性的数据，来判别检测器所产生的错误标签



TLD 目标跟踪方法

❖ P-N learning

■ Design of experts





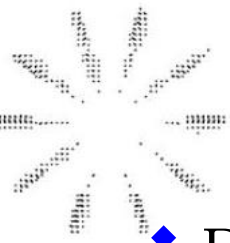
TLD目标跟踪方法

❖ P-N learning

■ Design of experts

P-experts寻找视频序列中的**时域**上的结构性特征，并且假设目标是沿着轨迹线移动的。P-experts记录目标在上一帧中的位置，并根据帧与帧之间的跟踪算法来预测目标在当前帧中的位置。如果**检测模块**将目标在**当前帧中的位置**标记为**负标签**，那么P-experts就产生一个**正样本**。

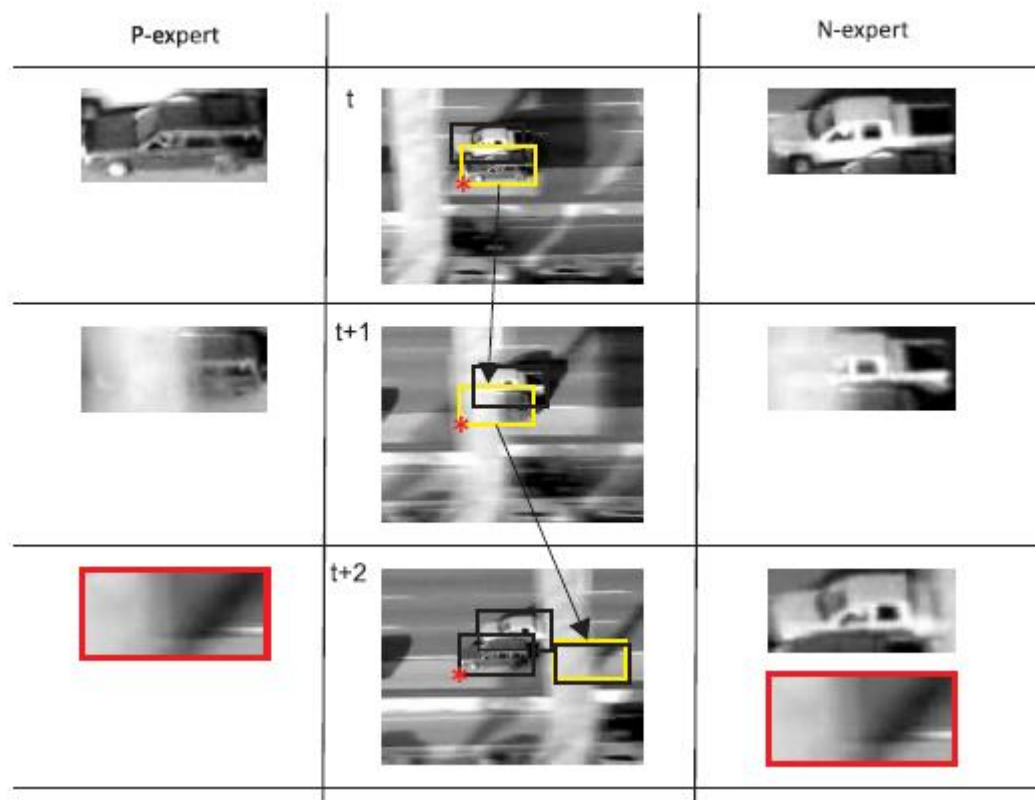
N-experts寻找视频序列中的**空域**上的结构性特征，并且假设目标在一个视频帧中只可能出现在一个位置。N-experts对**检测模块**在当前帧中的所有输出结果以及**跟踪模块**的输出结果进行分析，并找到**置信度最高**的结果。同**置信度最高**区域之间没有重叠的图相片被认定为**负样本**。另外，具有最大可能性的那个区域，被用于**重新初始化**跟踪模块。



TLD 目标跟踪方法

❖ P-N learning

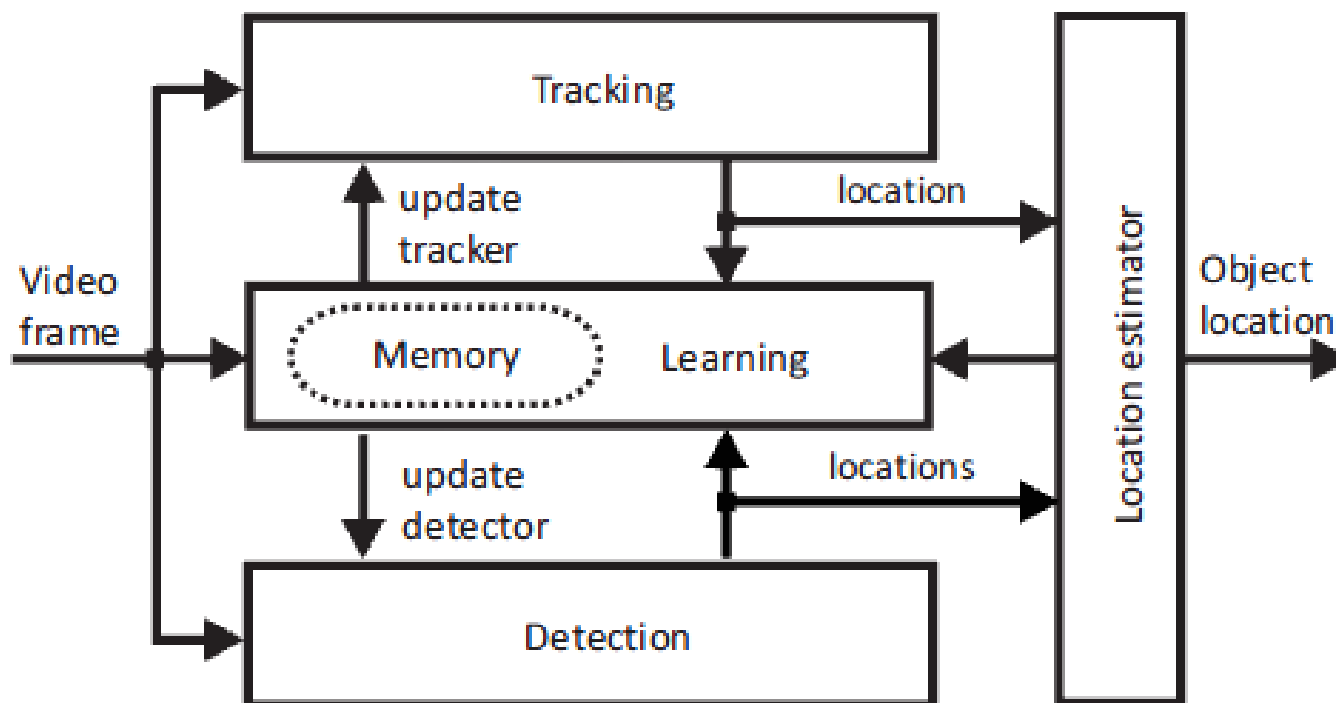
■ Design of experts





TLD 目标跟踪方法

❖ TLD实现





TLD实现

1.一些基本定义

(1)在任意时刻，被跟踪目标都可以用其**状态属性**来表示。该状态属性可以是一个**方框**或一个指示目标是否可见的**标记**。

方框由它的**位置**和**尺度**来决定，有**固定**的长宽比。两个**方框**的**空间相似度**由**重叠度**来量化，定义为两个方框**交集与并集之比**。

(2) **目标的外观**用**图片** p 来表示，每一个图片都是从跟踪框内部采样得到的，并被**重采样**到一个**归一化**的分辨率（15*15像素）。两个图相框 p_i, p_j 的相似度：

$$S(p_i, p_j) = 0.5(NCC(p_i, p_j) + 1)$$

NCC是一个**归一化相关性系数**。



TLD目标跟踪方法

1.一些基本定义

(3) **目标模型**是一个代表迄今为止观测到的**被检测目标**及其**周围背景**的数据结构，它是一系列**正样本**和**负样本**的集合

$$M = \{p_1^+, p_2^+, \dots, p_m^+, p_1^-, p_2^-, \dots, p_m^-\}$$

任意给定一个**图像片P**和**目标模型M**，我们定义几个量化指标：

① 正样本最近邻相似度

$$S^+(p, M) = \max_{p_i^+ \in M} S(p, p_i^+)$$

② 负样本最近邻相似度

$$S^-(p, M) = \max_{p_i^- \in M} S(p, p_i^-)$$

③ 前50%正样本的正最近邻相似度

$$S_{50\%}^+(p, M) = \max_{p_i^+ \in M \wedge i < \frac{m}{2}} S(p, p_i^+)$$

④ 相对相似度

$$S^r = \frac{S^+}{S^+ + S^-}$$

⑤ 保守相似度

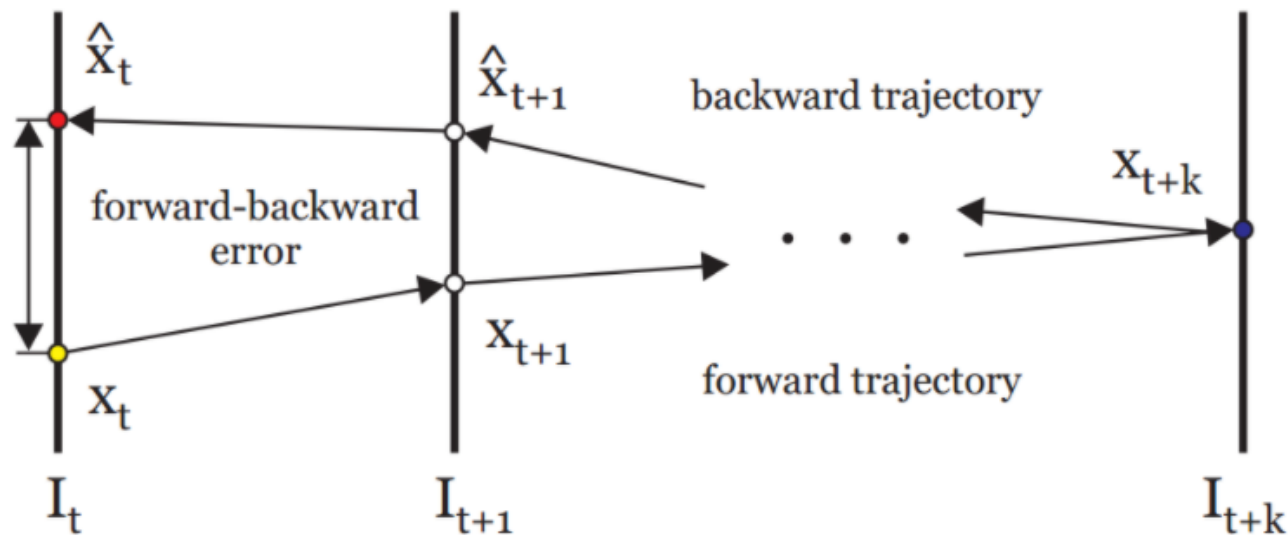
$$S^c = \frac{S_{50\%}^+}{S_{50\%}^+ + S^-}$$



TLD实现

2.跟踪模块--Tracker

带有**失败检测** (Failure detection) 的**中值流** (Median-Flow) 跟踪算法，定义**FB误差** (forward-backward error)：



- 从时间 t 的初始位置 $x(t)$ 开始追踪产生时间 $t+p$ 的位置 $x(t+p)$ ；
- 再从位置 $x(t+p)$ 反向追踪产生时间 t 的预测位置 $\hat{x}(t)$ ；
- 初始位置 $x(t)$ 和预测位置 $\hat{x}(t)$ 之间的欧氏距离就作为追踪器在 t 时间的**FB误差**。

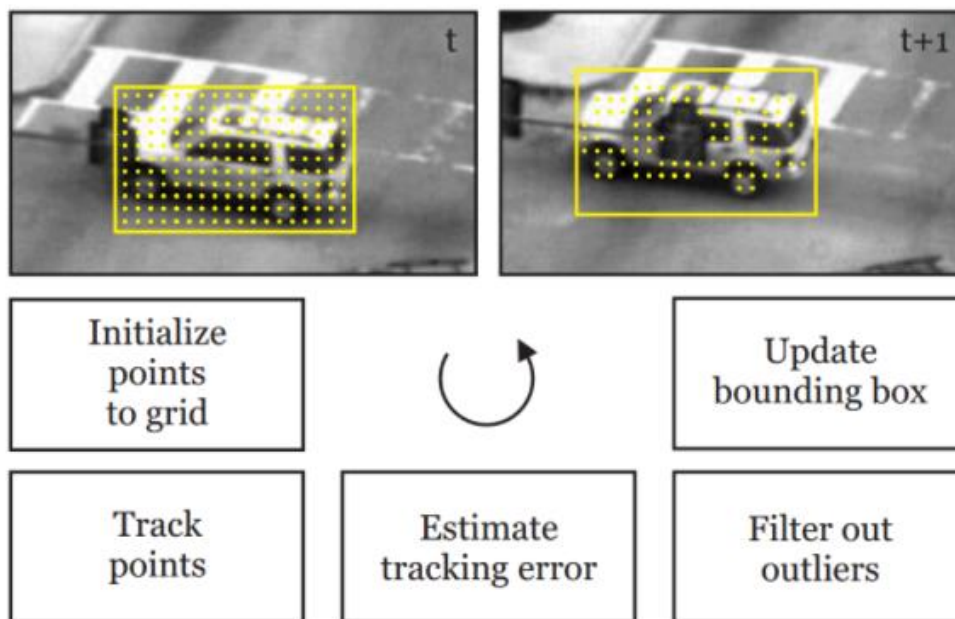


TLD实现

2.跟踪模块--Tracker

■ 中值流跟踪器

- ① 在第 t 帧的目标框中均匀撒点并采样作为特征点；
- ② 使用金字塔LK光流法跟踪特征点在第 $t+1$ 帧的对应位置。
- ③ 计算跟踪点的FB误差，筛选出FB误差最小的一半点作为最佳追踪点。
- ④ 根据这些点的坐标变化变化计算第 $t+1$ 帧包围框的位置和尺度（平移和尺度的变化取中值）



中值流跟踪器原理



TLD实现

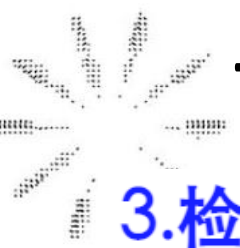
2.跟踪模块--Tracker

■ 跟踪失败检测

跟踪模块的跟踪失败检测算法：

中值流跟踪算法的前提假设是目标是可见的，所以当目标完全被遮挡或者消失于视野，则不可避免地出现跟踪失败。

让 d_i 表示其中某一个点的移动位移， d_m 表示位移中值，则残差可定义为 $|d_i - d_m|$ 。如果残差大于10个像素，那么就认为跟踪失败。这个策略能够很稳定地就确定出由剧烈移动或者遮挡所造成的跟踪失败。

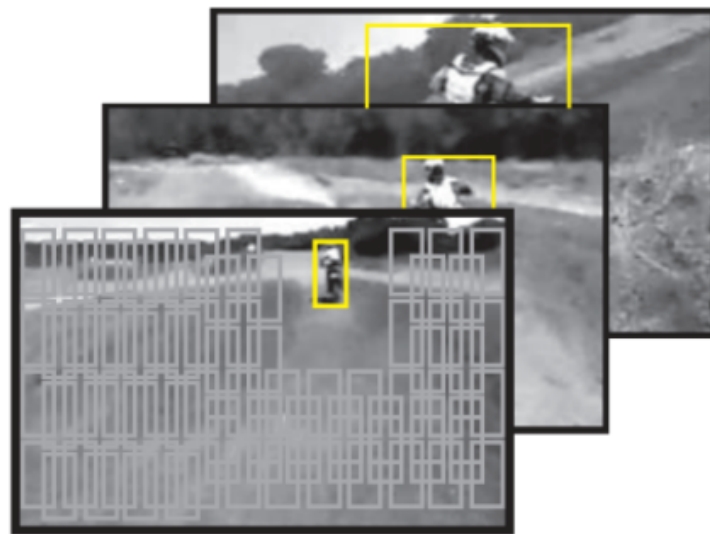


TLD实现

3.检测模块--Detection

检测器通过一个**扫描窗口**来扫描输入图像，然后判断出每一个图像块是否含有目标。在**初始跟踪框**的基础上，扫描窗口的参数设置为：

- **缩放比例**的步长系数为1.2，产生20种尺度系数
- **水平步长**是宽度的10%，**垂直步长**是高度的10%
- 最小的扫描窗口大小为20个像素。



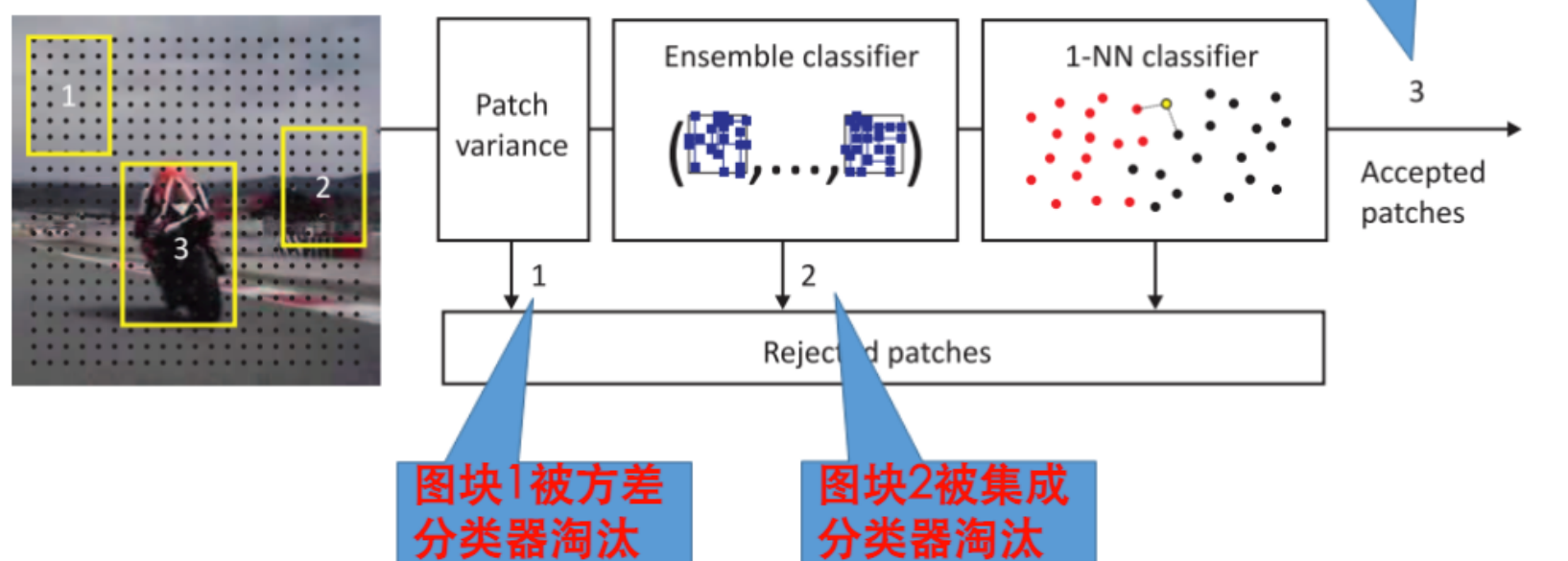
如此，对于QVGA 图片（240*320）来说会产生约**5万个**扫描方框，具体数字取决于初始框的长宽比。这是一个非常巨大的数字，每个图像块的处理必须非常高效。

TLD实现

3.检测模块--Detection

检测模块使用**级联分类器** (Cascaded classifier) 来判断输入的图像块中是否还有目标，划分为**三个模块**：

- ① 方差分类器 (Patch Variance)
- ② 集成分类器 (Ensemble Classifier)
- ③ 最近邻分类器 (NN Classifier)





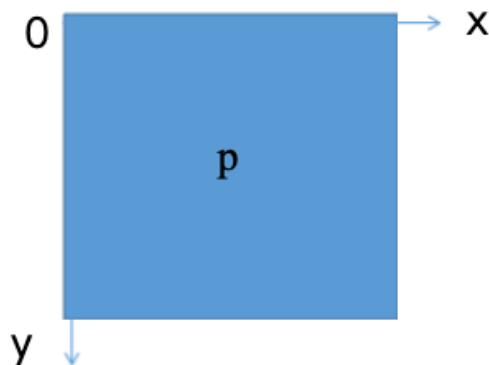
TLD实现

3.1 方差分类器

方差分类器淘汰所有**灰度值方差** (gray-value variance) 小于**初始跟踪框**内图像块的50%的图像块，**灰度值方差**可以表示为：

$$E(p^2) - E^2(p)$$

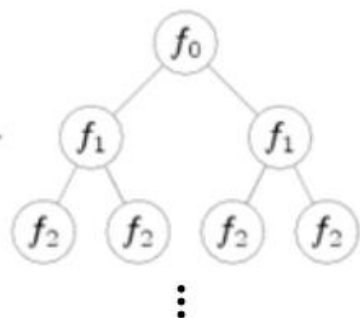
一般而言，这一步会淘汰**超过50%**的**非目标图像块**（比如，天空，街道等等）。



TLD实现

3.2 集成分类器

经过方差分类器之后未被去除的图像块进入集成分类器。集成分类器由 n ($n = 10$) 个独立的**基本分类器**组成，每个基本分类器对图像块进行 m ($m = 10$) 组**像素比较** (pixel comparisons)，并产生一个**二进制编码** x 。

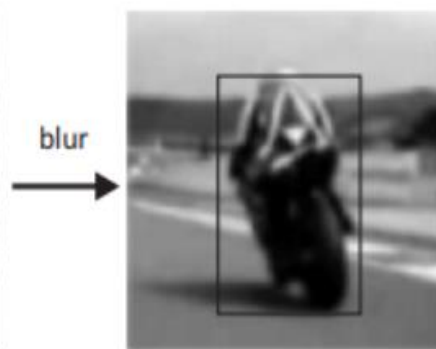


每个基本分类器为**树形结构** (随机森林分类器)，每层的节点的判断准则相同，即，在图像元中任意选取两点A和B，比较这两点的**亮度值**，若A的亮度大于B，则特征值为1，否则为0。

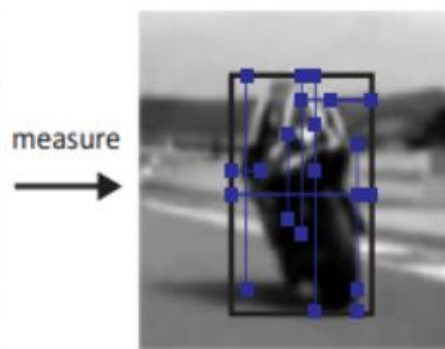
集成分类器由**10棵树**组成，每棵树有**13层**，图像块经过每棵树都会产生一个13位的二进制编码。



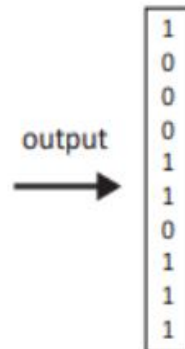
input image + bounding box



blurred image



pixel comparisons



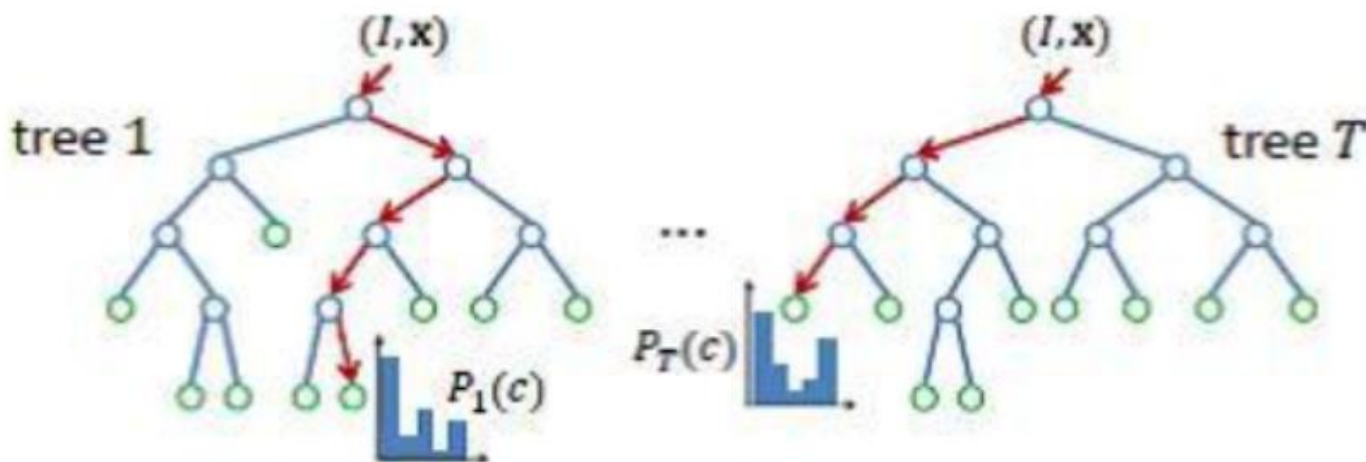
binary code



TLD实现

3.2 集成分类器

每棵树都对应一个关于二进制编码 x 的**后验概率分布** $P_i(y|x)$ ，其中 $y \in \{0, 1\}$ ， $y = 1$ 时表示正样本。



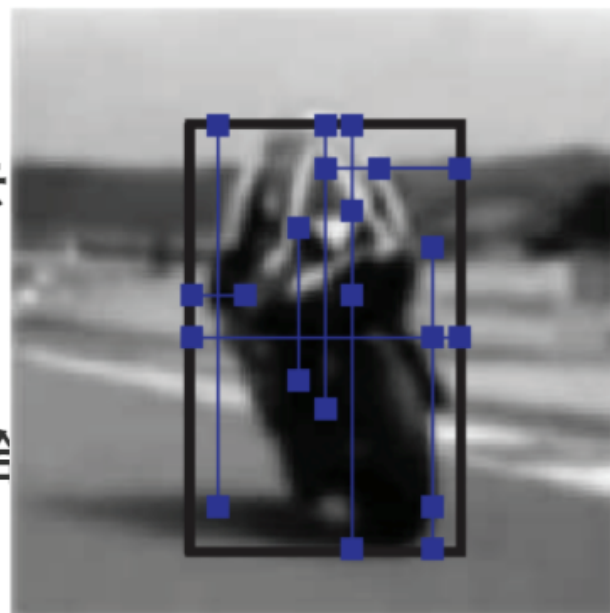
输入图像块经过所有基本分类器后，产生的10个后验概率值会被**平均**，集成分类器把平均后验概率大于0.5的图像块认为是目标。



TLD实现

■ 生成像素比较

- **对** 像素比较对的坐标离线、随机生成，并在运行过程中保持固定。
- 先用一个归一化 (15*15) 的patch去离散化像素空间，产生所有可能的垂直和水平的 pixel comparisons;
- 把这些pixel comparisons随机分配给10个分类器，每个分类器得到不同的pixel comparisons(特征集合，包含13组像素对)。
- 这样，保证了每个分类器基于一组不同的特征集合，而且所有的特征统一起来就可以覆盖整个patch了。





TLD实现

■ 后验概率分布

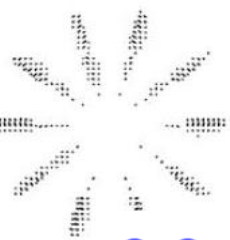
每个基本分类器 i 的后验概率分布 $P_i(y|x)$ 表示为：

$$P_i(y|x) = \frac{\#P}{\#P + \#N}$$

$\#P$ 和 $\#N$ 分别表示训练和更新过程中特征值 x 对应的正样本和负样本的数量。

■ 初始化及更新

在初始化阶段，所有后验概率的值均设为0。在运行时，集合分类器按如下更新：若已标注的样本被集合分类器错误分类，在更新相应的 $\#P$ 和 $\#N$ ，也就会更新 $P_i(y|x)$ 。



TLD实现

3.3 最近邻分类器

计算新样本的**相对相似度** S^r ，如大于阈值 θ_{NN} ，则认为为正样本。

$$S^r(p, M) > \theta_{NN}$$



TLD实现

4.综合模块--Integrator

- 跟踪器和检测器**独立同步**运行，各自输出预测到的目标框，而且检测器的输出结果很可能**不唯一**。
- 综合器先对检测器的结果进行**聚类**，然后根据检测器聚类结果和跟踪器跟踪结果的**置信度(confidence)**，对两者进行**加权**作为跟踪的最终输出结果,以达到最优的跟踪效果。
- 若跟踪器和检测器都没有得到目标框，则认为跟踪失败。



TLD实现

5.学习模块--Learning

学习模块的任务是在第一帧中**初始化**检测器，并且在运行过程中使用P-N expert 和 the N-expert**更新**检测器。

■ 初始化

- 在扫描网格中选择距离**初始目标框**最近的10个方框；
- 在每个**方框**的内部进行20次几何变换（ $\pm 1\%$ 范围的偏移， $\pm 1\%$ 范围的尺度变化， $\pm 10\%$ 范围的平面内旋转，并且在每个像素上增加方差为5的高斯噪声），那么10个box将产生**200个**仿射变换的方框，作为集合分类器的**正样本pX**；
- 目标框外的其他图像块都是负样本，将其中方差大于阈值的图像块都加入集合分类器的**负样本nX**；
- 如3.3小节所述，初始化后验概率。



TLD实现

5.学习模块--Learning

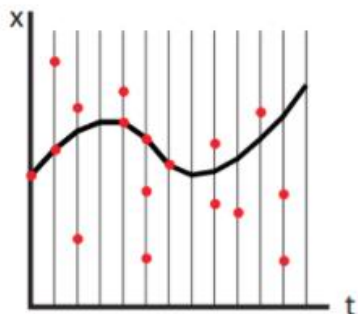
■ P-expert

P-expert 根据在线模型评估跟踪器结果的可靠性。如果当前位置可靠，则生成一组正样本去更新目标模型和集合分类器。生成正样本的方法与初始化时类似，但是只生成100个正样本。

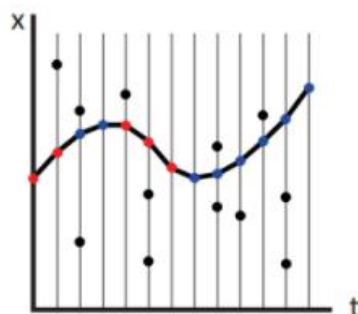
■ N-expert

N-expert 生成负训练样本。如果跟踪器结果可靠，那么距离目标位置较远（重叠度 < 0.2 ）的方框将被标注为负样本。

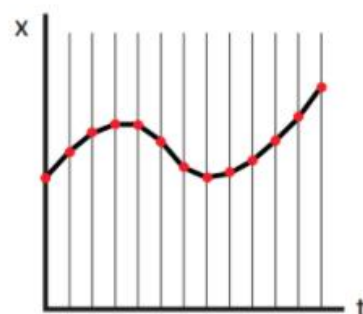
1) Track and detect



2) Identify new training examples



3) Update detector

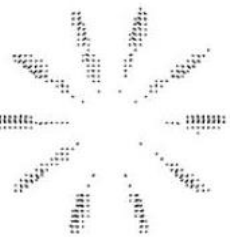


● positive examples
● negative examples



TLD 目标跟踪方法





TLD 目标跟踪方法

- 局限性
 - 不能很好解决发生平面外旋转问题
 - 不能很好跟踪铰链式的物体
 - 仅适合单目标跟踪
 -



The end !