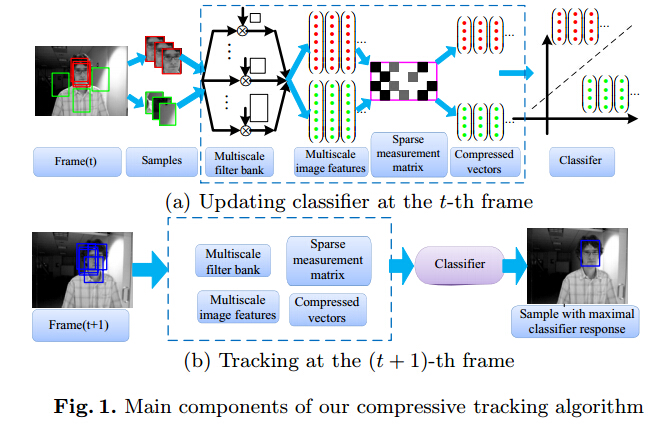
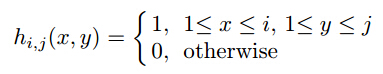
**一、其它算法介绍**

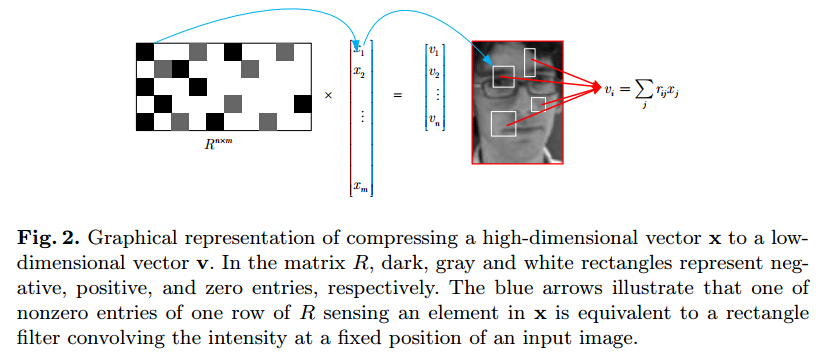
**Real-time compressive tracking**

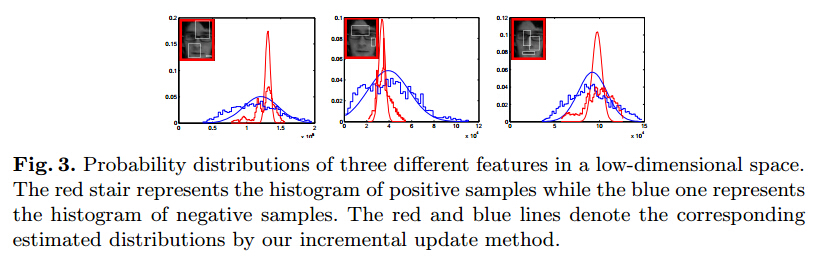
下面这部分将详细的讲述跟踪算法。我们假定第一帧中的跟踪窗口已经确定。在接下来的每一帧中，从当前目标位置附近采得正样本，从距离目标中心较远处采得负样本用于更新[分类器](https://so.csdn.net/so/search?q=%E5%88%86%E7%B1%BB%E5%99%A8&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/gguuookkaaii1234/article/details/_blank)。为了预测下一帧中目标的位置，我们从目标中心位置附近采得样本，再找的具有分类器得分最大的一个作为跟踪结果。

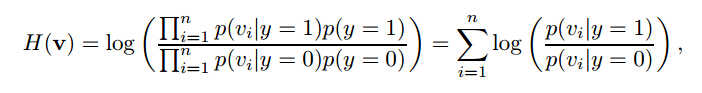
)

1)有效降维  
对每一个样本z，将其与一系列矩形滤波器卷积表示。  


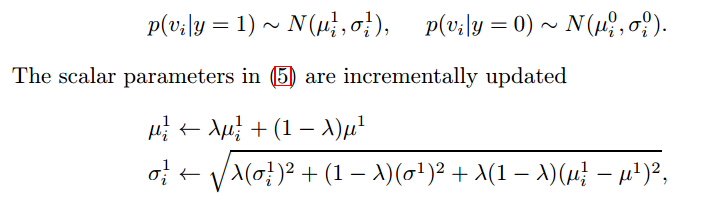
其中i和j分别是矩形滤波器的宽和高。然后将每一个滤波后的图像表示成列向量，再将他们联结成一个非常高维的多尺度图像特征向量。维度通常在十的六次方到十的十次方。我们采用一个非常稀疏的随机矩阵将x映射到低维空间向量v上。随机矩阵R只需离线计算一次，在整个跟踪过程保持不变，因此计算负荷比较小。在图2中，我们仅仅需要存储矩阵R中的非零项。



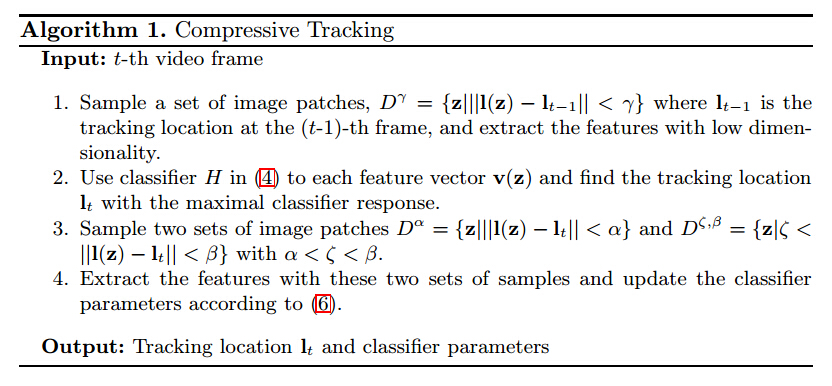
2)分析低维压缩特征  
低维特征v中的每个分量是不同尺度下空间分布矩形特征的线性组合。由于测量矩阵的系数可能是正的或负的，压缩特征计算相对像素值差的方式非常类似于Haar-like特征。  


3)分类器的构造与更新  
对于每个样本z，它的低维表示是v。我们假定v中所有元素是独立分布的，并且用朴素贝叶斯分类器建模，  


其中y是表示样本标签的二元变量，假定p(y=1)=p(y=0)。  
Diaconis and Freedman指出高维随机向量的随机映射几乎都是高斯分布的。因此，分类器H中的条件分布被假定为具有四个参数的高斯分布，

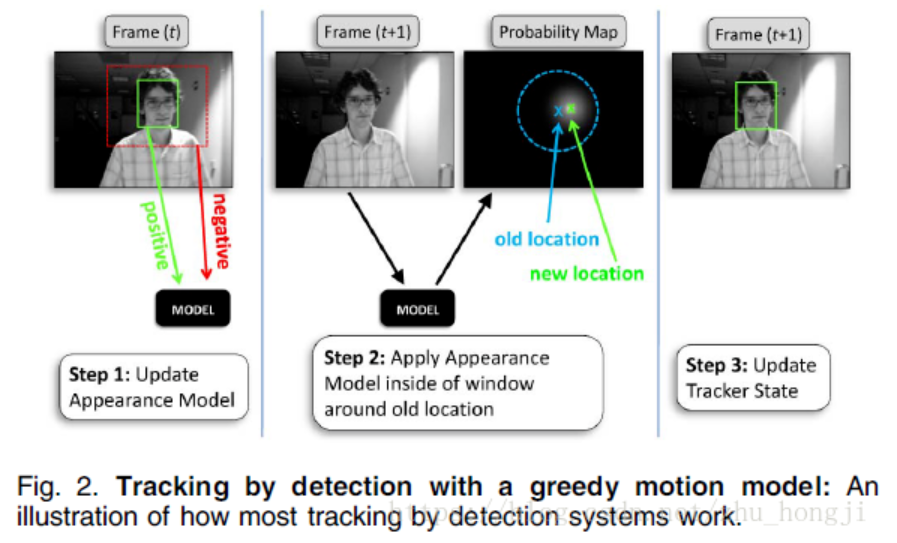


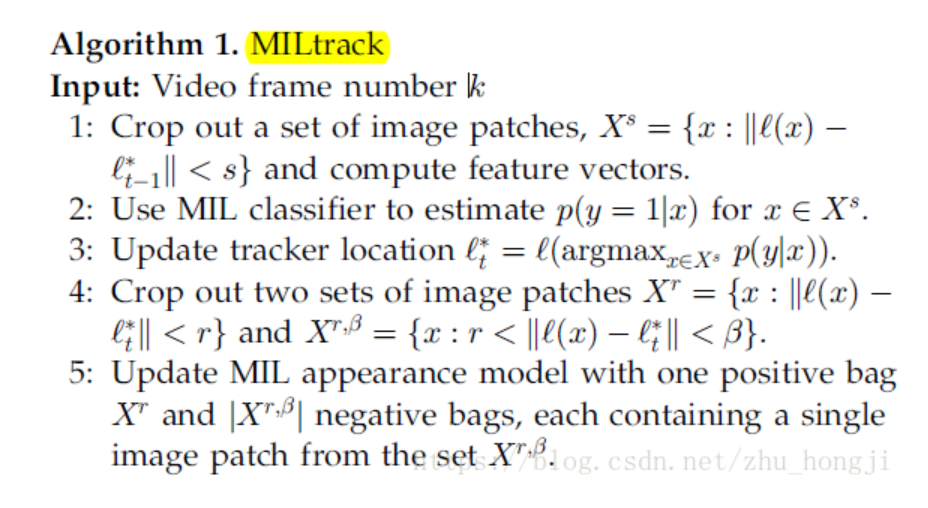
4)下面是算法主要步骤的总结：  
输入：视频第t帧序列  
1.在上一帧目标中心周围选取样本，并提取低维特征。  
2.通过分类器H找出具有最大分类得分的特征向量，其对应的目标位置作为跟踪结果。  
3.采集正负样本集。  
4.提取样本特征并更新分类器。  
输出：跟踪结果和分类器参数



**Robust\_Object\_Tracking\_with\_Online\_Multiple\_Instance\_Learning**

MIL跟踪算法





MIL跟踪算法：

1、由前一帧目标位置得到当前帧目标可能位置的集合，并计算相应的特征向量（使用Haar-like特征）表示图像块x的位置，包括坐标(x,y)和尺度； 表示第（t-1）帧（前一帧）的目标位置； s为参数，度量图像块与前一帧目标位置的最大距离。

2、使用MIL分类器计算集合中所有图像块的概率（图像块x中包含目标的概率）

3、更新跟踪器目标位置，选出概率最大的图像块作为当前帧的目标位置

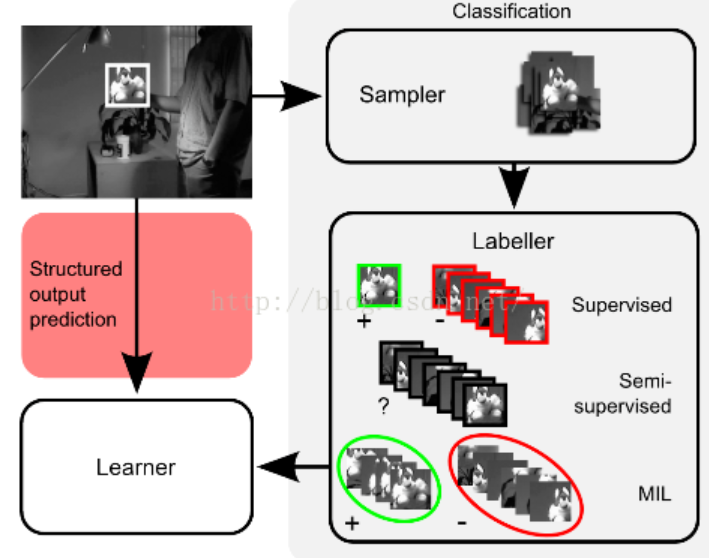
4、得到目标位置以后，扩展样本： 扩展为正样本，放在一个包里面，并标签为正；

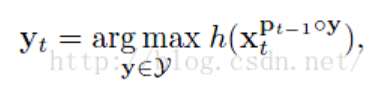
， 扩展负样本，每个负样本单独放在一个包里面，并标签为负；

5、用扩展后的正包和负包对MIL外观模型 进行更新；

同时论文还考虑了引入尺度变换参数，好处是结果可以更加的准确；坏处是增加了参数的空间维度，因此可根据需求决定是否采用。

**Struck\_Structured\_Output\_Tracking\_with\_Kernels**

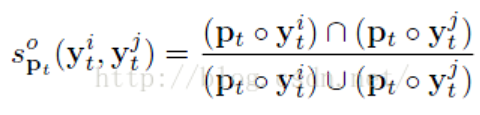


..................1

在这个框图中，作者要监督的学习label，半监督的学习样本，利用MIL来搞多种 特征。

第一，要得到好的正负样本，因为之前对样本贴标签的时候没有考虑transformation，第二是所有的样本看做了等权重啦训练SVM，然而有些负样本其实不是那么的负。很可能会对model产生不好的影响，比如drift。第三个问题是可能的噪声label。下面介绍怎么解决这个问题：

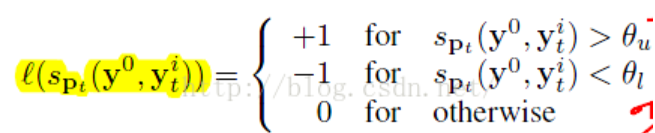
首先文中讲了一年传统的生成正负样本的方法：

..................2

在这里，Pt表示当前帧target的位置，yt表示估计出来的translation,于是对于两个样本(也就是两个bounding box)i,j，我们计算他们的重叠率：



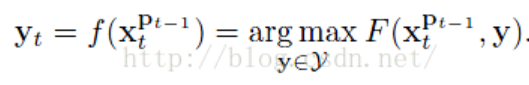
作者定义了一个label函数l，如下所示：

.................3

其中y0可以看做是对target位置的0偏移，也就是不偏移。于是和现有位置重叠率大于阈值\theta\_u的就看做是正样本而小于\theta\_l的看做是负样本，简单说就是去掉一些不好的正负样本，就是让正样本更加正，负样本更加负。

======================================

首先先祭出structured output SVM，作者说我们不去学习一个预测标签的分类器出来，我们学习一个能直接估计目标偏移量的函数出来，也就是说对于函数y不再是正负一，而是或者正或者负的偏移量。这个一般的SVM搞不定，就得靠structured output SVM，

.....................4

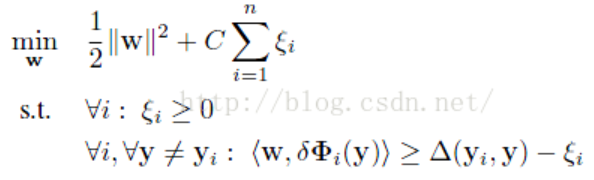
比较公式4和公式1，发现我们这个时候目标函数包含了样本的标签y. 什么意思？根据机器学习算法，我们知道，这个y是可以学习的，也就是说样本的标签是可以学习的，我们就可以随着tracking的过程，不断的update 样本标签，就出现了这样的情形：在前几帧某个样本还看做是正样本，但是在最近一帧中被看做负样本。

这个F的物理意义是衡量（x,y）的兼容性。特别的，F写成：



其中\pha(x,y)是一个联合核map，用来从一些样本集合:

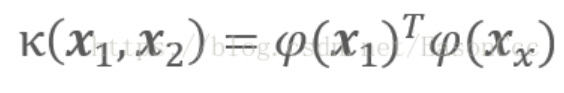
中学出一个最大边际出来，目标函数如下：

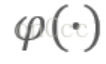


1. **与KCF比较**
2. 与上面算法相比，KCF用了核技巧，**核技**巧用一句话来概括的话，就是“更高维空间中内积的快速运算”。

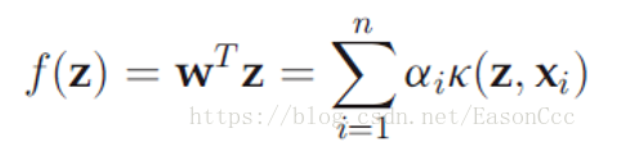
KCF用核技巧通过核函数隐式地将X空间映射到Z空间中，不必显示地计算，就可以实现快速运算。

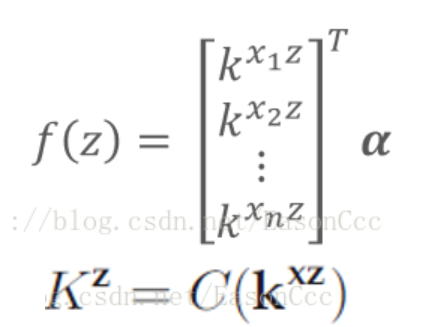
比如传统方法计算过程

核技巧的计算方法如式

上式中，为核函数，核函数的选取原则就是使得样本在升维后的Z空间中变得线性可分(在读可以达到目标的情况下，应该选方便于计算的)。

1. 与上面算法相比，KCF的检测更为快速

为了进一步提高处理速度，KCF对检测的过程也进行了简化加速。检测过程，也就是利用回归器与输入数据相关的过程，再次拿出公式，

可以将其写成如下的形式：

通过傅立叶变换求出两个单通道向量的核相关内积，给出当输入为多通道（彩色图，特征图）时，核相关计算的公式

：