第3章

人脸跟踪

人脸跟踪是指给定人脸目标在第一帧的位置，自动确定其在随后帧中的位置。人脸跟踪是视频中通用目标跟踪的一种特殊形式，视频序列中的目标跟踪研究还处在初级阶段，存在很多问题有待解决。在实际应用中，既要保证目标定位的准确性，又要保证目标跟踪的实时性，难度很大。在算法设计中，视频序列的目标跟踪主要有以下几个难点：

1. 目标外观变化

被跟踪的感兴趣目标（这里指人脸）的外观随着时间不断发生变化，如摄像机相对于目标的视角产生变化、环境的光照等外界条件发生变化等，这些都会导致目标的外观发生变化，另外目标本身由于非刚性产生的变化也会导致外观发生明显变化。

2. 遮挡问题

当视频序列中有多个跟踪目标时，由于摄像机的位置和两个或两个以上的目标在同一直线上，造成得到的序列中目标被其他目标遮挡的情况；或者目标被周围环境中的非目标障碍物挡住。这时就不能从视频中得到该目标的信息，如何有效地处理这种情况也是能够现实应用的目标跟踪系统所应该关注的焦点之一。

3. 目标和背景相似

当目标和背景比较相似的时候，此时目标相对于背景的鉴别性不够。计算机要通过捕捉到前景和背景之间的差别来实现对目标的跟踪就会遇到困难。比如，在包含多个人脸的视频序列中，算法跟踪某个人脸，这时如何准确地锁定感兴趣的人脸也是一个很大的问题，也是实际目标跟踪过程中所需要关注的。

一个典型的视频跟踪器包含两个组成部分：目标表示和定位（target representation and location）、滤波和数据关联（filtering and data association）。这两个部分如何组合依赖于特定的应用环境，例如复杂场景中的人脸跟踪算法主要依赖于目标的表示和定位，而在对航拍视频中的目标跟踪时，由于目标的运动有比较强的规律性，滤波和数据关联等高层处理方法就需要起主导作用。

人脸跟踪是通用目标跟踪的一种，现有的目标跟踪方法可以分为确定性跟踪算法和随机跟踪算法两类。本章分别介绍这两种方法以及它们在人脸跟踪上的应用，本章各节将安排如下：

3.1 节介绍确定性跟踪算法，包括目标表示、目标定位、算法实现和相关扩展。

3.2 节介绍随机跟踪算法，包括基于动力学系统模型的方法和基于统计模式识别的方法。

3.1 确定性跟踪算法

确定性跟踪算法以核跟踪为代表。核跟踪(kernel-based object tracking)由 Comaniciu (2003)提出，又称为 Mean Shift 跟踪，具有简单、易用、速度快等特点，自从其提出后就成为目标跟踪的一个重要研究方向。核跟踪的重心在于目标的表示和定位。目标用直方图表示，在计算直方图时，用一个各向同性的核函数对目标进行加权，使得直方图是一个关于目标位置的平滑函数。目标模型和候选模型的相似度利用 Bhattacharyya 系数来表示。目标的跟踪通过最大化该相似度来实现。由于直方图表示是一个关于目标位置的平滑函数，所以该相似度函数也是一个关于目标位置的平滑函数，目标的跟踪可以通过梯度下降的方法在平滑的相似度曲面上搜索其极值点实现，避免了耗时的穷尽搜索过程。核跟踪除自身可以实现对目标的有效跟踪外，还可以与 Kalman 滤波、粒子滤波等滤波算法进行有效地结合，扩大其应用范围。

3.1.1 目标表示

首先选定一个特征空间，作为基准的目标模型(target model)，用该特征空间中的概率密度函数 $\boldsymbol{q}$ 来表示，例如模板模型可以用颜色的概率密度函数来表示。不失一般性，假定目标模型的中心位置为原点。在后续帧中，候选目标(target candidate)的中心位置为 $\boldsymbol{y}$，用特征空间中的概率密度函数 $\boldsymbol{p}(\boldsymbol{y})$ 表示。这两个概率密度函数可以从图像数据中估计得到。为了降低运算的复杂度，这两个概率密度函数通常用包含 M 个区间的直方图近似表示，于是

$$\hat{\boldsymbol{q}} = \{\hat{q}_u\}_{u=1,\cdots,M} \qquad \sum_{u=1}^{M} \hat{q}_u = 1 \tag{3.1.1}$$

$$\hat{\boldsymbol{p}} = \{\hat{p}_u\}_{u=1,\cdots,M} \qquad \sum_{u=1}^{M} \hat{p}_u = 1 \tag{3.1.2}$$

$\hat{\boldsymbol{p}}$和$\hat{\boldsymbol{q}}$的相似度函数定义为

$$\hat{\rho}(\boldsymbol{y}) \triangleq \rho[\hat{\boldsymbol{p}}(\boldsymbol{y}), \hat{\boldsymbol{q}}] \tag{3.1.3}$$

函数 $\hat{\rho}(\boldsymbol{y})$ 表示了候选目标模型和实际目标模型的相似度，使得该函数最大的候选目标位置表明了被跟踪目标的真实位置。在计算直方图时，如果仅仅采用像素值信息，空间信息就会完全丢失，那么图像上相邻位置上的相似度就会有比较大的差异。梯度下降的方法不能用来搜索这样的函数的极值点，只能采用比较耗时的穷尽搜索法。为了便于利用梯度下降法进行快速搜索，在计算直方图时，可采用一个各向同性的核函数对目标上不同的像素位置进行加权。由携带了空间信息的核函数的权重来定义特征表示，$\hat{\rho}(\boldsymbol{y})$便成了空间位置 $\boldsymbol{y}$ 的平滑函数。

目标用图像上的一个椭圆区域表示。为了降低目标不同维数的影响，目标首先被归一化为单位圆。这可以通过将目标的行数和列数分别用长度 h_x 和宽度 h_y 进行缩放来实现。

假定$\{\boldsymbol{x}_i^*\}$，$i=1,2,\cdots,N$ 是目标模型所在区域归一化后每个像素点的位置，且该区域的中心位于原点。一个各向同性、凸的和单调递减的核剖面(profile)函数 $k(x)$ 为离中心较

远的像素值分配较低的权重。因为离中心较远的区域的像素通常容易被遮挡或者受到背景中物体的干扰，所以采用这样的权值策略可以提高概率密度估计的鲁棒性。

函数 $b:\mathbf{R}^2\rightarrow\{1,2,\cdots,M\}$ 将像素 $\boldsymbol{x}_i^*$ 映射到量化后的特征空间中对应的区间 $b(\boldsymbol{x}_i^*)$。目标模型在特征空间中的每一点 $u=1,2,\cdots,M$ 的概率可以计算为

$$\hat{q}_u = C\sum_{i=1}^{N} k(\|\boldsymbol{x}_i^*\|^2)\delta[b(\boldsymbol{x}_i^*)-u] \tag{3.1.4}$$

其中 δ 是 Kronecker delta 函数。归一化常数 C 可以由约束条件 $\sum_{u=1}^{M}\hat{q}_u = 1$ 得到，即

$$C = \frac{1}{\sum_{i=1}^{N} k(\|\boldsymbol{x}_i^*\|^2)} \tag{3.1.5}$$

设 $\{\boldsymbol{x}_i\}_{i=1,2,\cdots,N_h}$ 是中心位置位于 $\boldsymbol{y}$ 的候选目标归一化后的像素的位置，采用带宽为 h 的核剖面函数 $k(x)$，候选目标在特征空间中每一点 $u=1,2,\cdots,M$ 的概率密度可以计算为

$$\hat{p}_u(\boldsymbol{y}) = C_h\sum_{i=1}^{N_h} k\left(\left\|\frac{\boldsymbol{y}-\boldsymbol{x}_i}{h}\right\|^2\right)\delta[b(\boldsymbol{x}_i)-u] \tag{3.1.6}$$

其归一化常数

$$C_h = \frac{1}{\sum_{i=1}^{N_h} k\left(\left\|\frac{\boldsymbol{y}-\boldsymbol{x}_i}{h}\right\|^2\right)} \tag{3.1.7}$$

需要指出的是由于所有像素点 $\boldsymbol{x}_i$ 关于其中心 $\boldsymbol{y}$ 是对称的，所以 C_h 是一个不依赖于 $\boldsymbol{y}$ 的常数，给定不同的带宽 h，C_h 可以预先计算得到。带宽 h 表征了目标的大小，即有多少像素点位于目标区域中。

相似度函数表达式(3.1.3)通过式(3.1.4)和式(3.1.6)继承了核剖面函数 $k(x)$ 的相关性质。一个可微的核剖面函数 $k(x)$ 可以产生一个可微的相似度函数，通过高效的梯度优化方法可以较快地找到相似度函数的极大值。核函数的连续性等价于在图像中各个点上进行插值。基于核函数平滑的直方图目标表示对相似度量 ρ 没有限制，任何基于直方图距离的度量方法都可以直接采用。最广泛采用的是 Bhattacharyya 系数，即

$$\hat{\rho}(\boldsymbol{y}) \triangleq \rho[\hat{\boldsymbol{p}}(\boldsymbol{y}),\hat{\boldsymbol{q}}] = \sum_{u=1}^{M}\sqrt{\hat{p}_u(\boldsymbol{y})\,\hat{q}_u} \tag{3.1.8}$$

Bhattacharyya 系数具有比较直观的几何解释，它等于两个 m 维的单位向量 $\left(\sqrt{\hat{p}_1},\cdots,\sqrt{\hat{p}_M}\right)^{\mathrm{T}}$ 和 $\left(\sqrt{\hat{q}_1},\cdots,\sqrt{\hat{q}_M}\right)^{\mathrm{T}}$ 间夹角的余弦值。同时还可以将式(3.1.8)解释为这两个向量归一化的相关值。Bhattacharyya 系数采用离散化的概率密度表示，对目标尺度的变化具有一定的鲁棒性。同时直方图的表示可以表征任意形状的概率密度函数，适用性比较广。

3.1.2 目标定位

为了跟踪目标，需要在当前帧最大化式(3.1.8)。目标定位的过程需要以目标在上一帧的位置作为起始点，然后在附近搜索。假设目标在上一帧的位置为 $\hat{\boldsymbol{y}}_0$，首先计算目标在当前帧 $\hat{\boldsymbol{y}}_0$ 处的直方图表示 $\{\hat{p}_u(\hat{\boldsymbol{y}}_0)\}_{u=1,\cdots,M}$。在 $\hat{p}_u(\hat{\boldsymbol{y}}_0)$ 附近，利用 Taylor 展开，式(3.1.8)可以近似为

$$\rho[\hat{\boldsymbol{p}}(\boldsymbol{y}),\hat{\boldsymbol{q}}] \approx \frac{1}{2}\sum_{u=1}^{M}\sqrt{\hat{p}_u(\boldsymbol{y}_0)\,\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{M}\hat{p}_u(\boldsymbol{y})\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\boldsymbol{y}_0)}} \tag{3.1.9}$$

该式在$\{\hat{p}_u(\hat{\boldsymbol{y}})\}_{u=1,\cdots,M}$和$\{\hat{p}_u(\hat{\boldsymbol{y}}_0)\}_{u=1,\cdots,M}$相差不大时成立，这个假设一般在连续的帧间是成立的。通过去掉违反条件的区间值，对每一个区间 $u=1,2,\cdots,M$，条件$\hat{p}_u(\hat{\boldsymbol{y}}_0)>0$ 也是肯定可以满足的。代入式(3.1.6)，式(3.1.9)可以简化为

$$\rho[\hat{\boldsymbol{p}}(\boldsymbol{y}),\hat{\boldsymbol{q}}] \approx \frac{1}{2}\sum_{u=1}^{M}\sqrt{\hat{p}_u(\boldsymbol{y}_0)\,\hat{q}_u} + \frac{C_h}{2}\sum_{u=1}^{N_h}\omega_i k\left(\left\|\frac{\boldsymbol{y}-\boldsymbol{x}_i}{h}\right\|^2\right) \tag{3.1.10}$$

其中

$$\omega_i = \sum_{u=1}^{M}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\boldsymbol{y}_0)}}\delta[b(\boldsymbol{x}_i)-u] \tag{3.1.11}$$

可以通过最大化式(3.1.10)的右边直接最大化式(3.1.8)。该式等价于最大化以 $\boldsymbol{y}$ 为中心，利用核剖面函数 $k(x)$估计的概率密度函数。该概率密度函数的极值点可以通过 Mean Shift 算法[Comaniciu 2002]迭代得到，即

$$\hat{\boldsymbol{y}}_1 = \frac{\sum_{i=1}^{N_h}\boldsymbol{x}_i\omega_i g\left(\left\|\frac{\hat{\boldsymbol{y}}_0-\boldsymbol{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{N_h}\omega_i g\left(\left\|\frac{\hat{\boldsymbol{y}}_0-\boldsymbol{x}_i}{h}\right\|^2\right)} \tag{3.1.12}$$

其中

$$g(x) = -k'(x) \tag{3.1.13}$$

Bhattacharyya 系数 $\rho[\hat{\boldsymbol{p}}(\boldsymbol{y}),\hat{\boldsymbol{q}}]$最大化的算法流程包括如下步骤：

(1) 输入：目标模型$\{\hat{q}_u\}_{u=1,\cdots,M}$和它在上一帧的位置$\hat{\boldsymbol{y}}_0$。

(2) 初始化目标在当前帧的位置为$\hat{\boldsymbol{y}}_0$，计算此处的直方图$\{\hat{p}_u(\hat{\boldsymbol{y}}_0)\}_{u=1,\cdots,M}$，并计算$\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_0),\hat{\boldsymbol{q}}]=\sum_{u=1}^{M}\sqrt{\hat{p}_u(\boldsymbol{y}_0)\,\hat{q}_u}$。

(3) 根据式(3.1.11)计算权值$\{\omega_i\}_{i=1,\cdots,N_h}$。

(4) 根据式(3.1.12)估计目标在下一处的位置$\hat{\boldsymbol{y}}_1$。

(5) 计算$\{\hat{p}_u(\hat{\boldsymbol{y}}_1)\}_{u=1,\cdots,M}$和$\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_1),\hat{\boldsymbol{q}}]=\sum_{u=1}^{M}\sqrt{\hat{p}_u(\boldsymbol{y}_1)\hat{q}_u}$。

(6) 当 $\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_1),\hat{\boldsymbol{q}}]<\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_0),\hat{\boldsymbol{q}}]$，执行 $\hat{\boldsymbol{y}}_1\leftarrow\frac{1}{2}(\hat{\boldsymbol{y}}_0+\hat{\boldsymbol{y}}_1)$，计算 $\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_1),\hat{\boldsymbol{q}}]$。

(7) 如果 $\|\hat{\boldsymbol{y}}_1-\hat{\boldsymbol{y}}_0\|<\varepsilon$ 则停止，否则$\hat{\boldsymbol{y}}_0\leftarrow\hat{\boldsymbol{y}}_1$，转步骤(2)。

3.1.3 跟踪算法实现

上面算法步骤(7)中的停止迭代参数 ε 限制了位置向量$\hat{\boldsymbol{y}}_0$ 和$\hat{\boldsymbol{y}}_1$ 都对应原始图像坐标的同一个像素值。当其取值小于一个像素时，可以达到亚像素的精度。当应用系统有实时性限制时，可以限制 Mean Shift 迭代的次数为 $N_{\max}$，一般取 20 即可。在实际中，一般平均的迭代次数更小，大约为 4。

在实际实现时，跟踪算法可以比上述算法流程更加简单。上述算法中的步骤(6)是为了

防止 Mean Shift 迭代过程中可能产生的数值稳定性的问题，该问题来源于对 Bhattacharyya 系数的线性近似。然而，对很多目标长时间的跟踪实验表明，在新的位置处的 Bhattacharyya 系数只有很少的情况会比原来的位置更小。因此，在实际应用中步骤(6)可以不必采用，那么相应地步骤(2)到步骤(5)中的 Bhattacharyya 系数也不必计算。于是实际实现时只需要计算步骤(3)中的权值和步骤(4)中的新的迭代位置，然后在步骤(7)中根据位移的步长判断算法是否终止。

当核剖面函数为 Epanechnikov 函数时，有

$$k(x)=\begin{cases}\dfrac{1}{2}C_d^{-1}(d+2)(1-x) & \text{如果 } x\leqslant 1\\ 0 & \text{其他}\end{cases} \tag{3.1.14}$$

这种情况下，该核剖面函数对应的 $g(x)$ 是常数，式(3.1.12)可以简化为

$$\hat{\boldsymbol{y}}_1=\frac{\sum_{i=1}^{N_h}\boldsymbol{x}_i\omega_i}{\sum_{i=1}^{N_h}\omega_i} \tag{3.1.15}$$

Bhattacharyya 系数的最大化可以解释为匹配滤波的过程。实际上，式(3.1.8)表示单位向量$\sqrt{\hat{\boldsymbol{q}}}$和$\sqrt{\hat{\boldsymbol{p}}(\boldsymbol{y})}$的相关系数，分别代表了目标模型和候选目标。Mean Shift 算法等价于最大化这个标量化的相关系数。

上述算法并没有考虑被跟踪目标的尺度变化问题，由于在实际的跟踪过程中目标的尺度一般是变化的，所以式(3.1.6)中的带宽 h 需要自适应地进行调整。假设目标在上一帧的带宽为 h_{prev}，当前帧最佳的带宽 h_{opt} 可以通过运行上述跟踪算法 3 次得到，每次的带宽分别为 $h=h_{\text{prev}}$、$h=h_{\text{prev}}+\Delta h$、$h=h_{\text{prev}}-\Delta h$，典型的取值为 $\Delta h=0.1h_{\text{prev}}$。选择使得 Bhattacharyya 系数最大的带宽作为当前帧的最佳带宽 h_{opt}。为了降低对带宽变化的敏感性，当选帧的最终带宽一般取为

$$h_{\text{new}}=\gamma h_{\text{opt}}+(1-\gamma)h_{\text{prev}} \tag{3.1.16}$$

其中 γ 的默认值为 0.1。

假定 N 为每帧的平均迭代次数，那么在步骤(2)中需要计算加权的直方图$\{\hat{p}_u(\hat{\boldsymbol{y}}_0)\}_{u=1,\cdots,M}$，在步骤(3)中需要计算对每一个像素点的权值。那么在一个单一尺度上的计算量为

$$C_O=N(c_H+N_hc_S)\approx NN_hc_S \tag{3.1.17}$$

其中 c_H 是计算直方图的计算量，c_S 是每个像素点一次加法、一次开方和一次除法的计算量。上述近似表达式假定了直方图的区间个数 M 和像素点的个数 N_h 处于同一个数量级。

比较该算法与不采用梯度优化方法的算法的复杂度。假定不采用梯度优化方法时搜索的范围局限于目标的大小。全局搜索的第一步是计算 N_h 个直方图，假定每个直方图计算窗口的大小是 N_h 的平方根，为了提高计算效率，可以将该窗口平移 N_h 次(水平方向 $\sqrt{N_h}$ 次，垂直方向 $\sqrt{N_h}$ 次)。在每次平移时需要计算 $2\sqrt{N_h}$ 次加法运算，因此总的计算量为 $c_H+2N_h\sqrt{N_h}c_A$，其中 c_A 表示每次加法运算的计算量。全局搜索的第二步是计算 N_h 个 Bhattacharyya 系数，这同样可以通过计算一次系数然后逐渐平移窗口得到，总的计算量为 $Mc_S+2N_h\sqrt{N_h}c_S$，于是没有经过优化的搜索方法的总计算量为

$$C_{NO} = c_H + 2N_h\sqrt{N_h}c_A + (M + 2N_h\sqrt{N_h})c_S \approx 2N_h\sqrt{N_h}c_S \tag{3.1.18}$$

式(3.1.18)和式(3.1.17)的比值为 $2\sqrt{N_h}/N$，在一个典型的跟踪场景中，目标的大小为 50×50 像素(即 $\sqrt{N_h}=50$)，Mean Shift 算法的平均迭代次数为 $N=4.19$ 次，那么 Mean Shift 快速跟踪相对于全局优化的加速比为 $2\times 50/4.19\approx 24$ 倍。

将 Mean Shift 算法直接应用到人脸跟踪上的一个例子如图 3.1.1 所示。人脸的模型为一矩形区域，直方图在 RGB 空间中表示，区间个数为 16×16×16。为了快速地适应尺度的变化，最佳的尺度大小通过最大化颜色的相关系数来实现。由图 3.1.1 可见，当人在第 45 帧转头和第 95 帧的颜色变化时，该算法均能稳定地跟踪到人脸。

图 3.1.1 Mean Shift 人脸跟踪示例

3.1.4 多核跟踪

原始的核跟踪算法将目标视为整体，然后利用直方图进行表示，丢掉了空间信息，在非刚体目标跟踪中获得了较好的效果。但这种做法有两个主要缺点：一是降低了目标模型的鉴别性；二是在目标被部分遮挡时不能对目标进行准确地定位。为了解决这两个问题，下面介绍一种基于直方图表示和 Bhattacharyya 系数相似度度量的多核跟踪算法[贾 2009]。

多核跟踪将目标分块，针对每块提取出用核函数加权的直方图。目标块的大小可以变化以获得不同尺度的信息，块与块之间可以部分重叠以获得不同位置的信息。假设目标的中心位置为原点，分为 V 个矩形块，第 v 个矩形块的中心位置为 $\boldsymbol{l}^v$，尺度为 h^v，对应的核函数为 k^v，那么该块对应的核函数加权的直方图 $\hat{\boldsymbol{q}}^v=[\hat{q}_1^v,\cdots,\hat{q}_u^v,\cdots,\hat{q}_M^v]$ 可以计算如下：

$$\hat{q}_u^v = Q^v\sum_{i=1}^{n^v}k^v\left(\left\|\frac{\boldsymbol{y}_i-\boldsymbol{l}^v}{h^v}\right\|^2\right)\delta[b(\boldsymbol{y}_i)-u] \tag{3.1.19}$$

$$Q^v = \left[\sum_{i=1}^{n^v}k^v\left(\left\|\frac{\boldsymbol{y}_i-\boldsymbol{l}^v}{h^v}\right\|^2\right)\right]^{-1} \tag{3.1.20}$$

其中 n^v 表示第 v 个矩形块中像素的个数，Q^v 表示第 v 个矩形块直方图的归一化系数。假设候选目标和被跟踪目标的大小相同，则中心位置位于 $\boldsymbol{y}$ 的候选目标模型的第 v 个矩形块的直方图 $\hat{\boldsymbol{p}}(\boldsymbol{y})^v=[\hat{p}(\boldsymbol{y})_1^v,\cdots,\hat{p}(\boldsymbol{y})_u^v,\cdots,\hat{p}(\boldsymbol{y})_M^v]$ 可以计算如下：

$$\hat{p}(\boldsymbol{y})_u^v = P^v\sum_{i=1}^{n^v}k^v\left(\left\|\frac{\boldsymbol{y}_i-\boldsymbol{l}^v-\boldsymbol{y}}{h^v}\right\|^2\right)\delta[b(\boldsymbol{y}_i)-u] \tag{3.1.21}$$

$$P^v = \left[\sum_{i=1}^{n^v} k^v \left(\left\| \frac{\boldsymbol{y}_i - \boldsymbol{l}^v - \boldsymbol{y}}{h^v} \right\|^2 \right) \right]^{-1} \tag{3.1.22}$$

其中 P^v 表示第 v 个矩形块的直方图的归一化系数，由于像素点 $\boldsymbol{y}_i$ 关于其中心 $\boldsymbol{l}^v + \boldsymbol{y}$ 对称，所以其值并不依赖于坐标 $\boldsymbol{y}$，可以在跟踪过程开始前预先计算得到。

将各个块的直方图组合在一起，目标模型表示为 $\hat{\boldsymbol{q}} = [\hat{\boldsymbol{q}}^1, \cdots, \hat{\boldsymbol{q}}^v, \cdots, \hat{\boldsymbol{q}}^V]$，则中心位置位于 $\boldsymbol{y}$ 的候选目标模型的直方图可以表示为 $\hat{\boldsymbol{p}}(\boldsymbol{y}) = [\hat{\boldsymbol{p}}(\boldsymbol{y})^1, \cdots, \hat{\boldsymbol{p}}(\boldsymbol{y})^v, \cdots, \hat{\boldsymbol{p}}(\boldsymbol{y})^V]$，目标模型和候选目标模型的相似度定义为各个块对应的 Bhattacharyya 系数的平均值，即

$$\hat{\rho}(\boldsymbol{y}) \equiv \frac{1}{V} \sum_{v=1}^{V} \sum_{u=1}^{M} \sqrt{\hat{p}_u^v(\boldsymbol{y})\, \hat{q}_u^v} \tag{3.1.23}$$

目标的跟踪通过最大化式(3.1.23)来实现。在目标的初始位置 $\hat{\boldsymbol{y}}_0$，利用 Taylor 展开，式(3.1.23)可以近似为

$$\begin{aligned}
\hat{\rho}(\boldsymbol{y}) &\approx \frac{1}{2V} \sum_{v=1}^{V} \sum_{u=1}^{M} \left(\sqrt{\hat{p}_u^v(\hat{\boldsymbol{y}}_0)\, \hat{q}_u^v} + \hat{p}_u^v(\boldsymbol{y}) \sqrt{\frac{\hat{q}_u^v}{\hat{p}_u^v(\hat{\boldsymbol{y}}_0)}} \right) \\
&= \frac{1}{2V} \sum_{v=1}^{V} \sum_{u=1}^{M} \left[P^v \sum_{i=1}^{n^v} k^v \left(\left\| \frac{\boldsymbol{y}_i - \boldsymbol{y} - \boldsymbol{l}^v}{h^v} \right\|^2 \right) \delta[b(\boldsymbol{y}_i) - u] \sqrt{\frac{\hat{q}_u^v}{\hat{p}_u^v(\hat{\boldsymbol{y}}_0)}} \right] + C \\
&= \frac{1}{2V} \sum_{v=1}^{V} P^v \sum_{i=1}^{n^v} k^v \left(\left\| \frac{\boldsymbol{y}_i - \boldsymbol{y} - \boldsymbol{l}^v}{h^v} \right\|^2 \right) \sum_{u=1}^{M} \left(\delta[b(\boldsymbol{y}_i) - u] \sqrt{\frac{\hat{q}_u^v}{\hat{p}_u^v(\hat{\boldsymbol{y}}_0)}} \right) + C \\
&= \frac{1}{2V} \sum_{v=1}^{V} P^v \sum_{i=1}^{n^v} k^v \left(\left\| \frac{\boldsymbol{y}_i - \boldsymbol{y} - \boldsymbol{l}^v}{h^v} \right\|^2 \right) \omega_i^v + C
\end{aligned} \tag{3.1.24}$$

其中

$$C = \frac{1}{2V} \sum_{v=1}^{V} \sum_{u=1}^{M} \left(\sqrt{\hat{p}_u^v(\hat{\boldsymbol{y}}_0)\, \hat{q}_u^v} \right) \tag{3.1.25}$$

$$\omega_i^v = \sum_{u=1}^{M} \left(\delta[b(\boldsymbol{y}_i) - u] \sqrt{\frac{\hat{q}_u^v}{\hat{p}_u^v(\hat{\boldsymbol{y}}_0)}} \right) \tag{3.1.26}$$

$\hat{\rho}(\boldsymbol{y})$的梯度为

$$\nabla \hat{\rho}(\boldsymbol{y}) = \frac{1}{V} \sum_{v=1}^{V} \frac{P^v}{(h^v)^2} \sum_{i=1}^{n^v} g^v \left(\left\| \frac{\boldsymbol{y}_i - \boldsymbol{y} - \boldsymbol{l}^v}{h^v} \right\|^2 \right) (\boldsymbol{y}_i - \boldsymbol{y} - \boldsymbol{l}^v) \omega_i^v \tag{3.1.27}$$

为最大化 $\hat{\rho}(\boldsymbol{y})$，取在 $\hat{\boldsymbol{y}}_0$ 处的位移向量为

$$\Delta \hat{\boldsymbol{y}}_0 = \frac{\sum_{v=1}^{V} \frac{P^v}{(h^v)^2} \sum_{i=1}^{n^v} g^v \left(\left\| \frac{\boldsymbol{y}_i - \hat{\boldsymbol{y}}_0 - \boldsymbol{l}^v}{h^v} \right\|^2 \right) (\boldsymbol{y}_i - \hat{\boldsymbol{y}}_0 - \boldsymbol{l}^v) \omega_i^v}{\sum_{v=1}^{V} \frac{P^v}{(h^v)^2} \sum_{i=1}^{n^v} g^v \left(\left\| \frac{\boldsymbol{y}_i - \hat{\boldsymbol{y}}_0 - \boldsymbol{l}^v}{h^v} \right\|^2 \right) \omega_i^v} \tag{3.1.28}$$

即新的目标位置为

$$\hat{\boldsymbol{y}}_1 = \frac{\sum_{v=1}^{V} \frac{P^v}{(h^v)^2} \sum_{i=1}^{n^v} g^v \left(\left\| \frac{\boldsymbol{y}_i - \hat{\boldsymbol{y}}_0 - \boldsymbol{l}^v}{h^v} \right\|^2 \right) (\boldsymbol{y}_i - \boldsymbol{l}^v) \omega_i^v}{\sum_{v=1}^{V} \frac{P^v}{(h^v)^2} \sum_{i=1}^{n^v} g^v \left(\left\| \frac{\boldsymbol{y}_i - \hat{\boldsymbol{y}}_0 - \boldsymbol{l}^v}{h^v} \right\|^2 \right) \omega_i^v} \tag{3.1.29}$$

当核剖面函数为式(3.1.14)表示的 Epanechnikov 函数时，式(3.1.29)可以简化为

$$\hat{\boldsymbol{y}}_1 = \frac{\sum_{v=1}^{V} \frac{P^v}{(h^v)^2} \sum_{i=1}^{n^v} (\boldsymbol{y}_i - \boldsymbol{l}^v)\omega_i^v}{\sum_{v=1}^{V} \frac{P^v}{(h^v)^2} \sum_{i=1}^{n^v} \omega_i^v} \tag{3.1.30}$$

综上可知，多核跟踪算法流程主要包括如下步骤：

(1) 输入：目标模型$\{\hat{q}_u^v\}_{u=1,\cdots,M}^{v=1,\cdots,V}$和它在上一帧的位置$\hat{\boldsymbol{y}}_0$。

(2) 初始化目标在当前帧的位置为$\hat{\boldsymbol{y}}_0$，计算此处的直方图$\{\hat{p}_u^v(\hat{\boldsymbol{y}}_0)\}_{u=1,\cdots,M}^{v=1,\cdots,V}$，并计算$\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_0),\hat{\boldsymbol{q}}] = \sum_{v=1}^{V}\sum_{u=1}^{M}\sqrt{\hat{p}_u^v(\boldsymbol{y}_0)\,\hat{q}_u^v}$。

(3) 根据式(3.1.26)计算权值$\{\omega_i^v\}$。

(4) 根据式(3.1.29)估计目标在下一个的位置$\hat{\boldsymbol{y}}_1$。

(5) 计算$\{\hat{p}_u^v(\hat{\boldsymbol{y}}_1)\}_{u=1,\cdots,M}^{v=1,\cdots,V}$和$\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_1),\hat{\boldsymbol{q}}] = \sum_{v=1}^{V}\sum_{u=1}^{M}\sqrt{\hat{p}_u^v(\hat{\boldsymbol{y}}_1)\,\hat{q}_u^v}$。

(6) 当$\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_1),\hat{\boldsymbol{q}}] < \rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_0),\hat{\boldsymbol{q}}]$时，执行$\hat{\boldsymbol{y}}_1 \leftarrow \frac{1}{2}(\hat{\boldsymbol{y}}_0 + \hat{\boldsymbol{y}}_1)$，计算$\rho[\hat{\boldsymbol{p}}(\hat{\boldsymbol{y}}_1),\hat{\boldsymbol{q}}]$。

(7) 如果$\|\hat{\boldsymbol{y}}_1 - \hat{\boldsymbol{y}}_0\| < \varepsilon$则停止，否则$\hat{\boldsymbol{y}}_0 \leftarrow \hat{\boldsymbol{y}}_1$，转步骤(2)。

3.2 随机跟踪算法

按照算法模型分类，随机跟踪算法主要包括两类，一类是基于动力学系统模型的算法，如基于卡尔曼滤波(Kalman filtering)的跟踪算法，基于粒子滤波的跟踪算法[Isard 1998]，其随机性体现在状态转移方程和观测方程中存在的随机噪声；另一类则是基于统计模式识别的算法，如基于支持向量机的跟踪算法[Avidan 2004]，基于 Adaboost 的跟踪算法[Avidan 2007]，其随机性则体现在利用统计学的知识对跟踪目标建立模型。本节将重点介绍基于粒子滤波的随机跟踪算法。

3.2.1 基于动力学系统模型的方法

基于 Kalman 滤波器的目标跟踪算法假设状态向量在空间的分布符合高斯分布，同时假设系统的状态转移方程和观测方程中的噪声也符合高斯分布。算法基于最小均方误差准则，对下一状态进行估计，即在视频序列中对下一帧中的目标状态进行估计。目标状态一般是指目标的位置，有时还包括其他信息，如目标运动的速度、加速度等。这依赖于具体建模过程中的状态变量的选取。但是卡尔曼方法的最优性是基于单模高斯密度的，当系统状态变量不符合高斯概率密度分布时，卡尔曼滤波产生的就是有偏的估计，其最小均方差也不再得到保证。再则，卡尔曼滤波的状态转移方程和观测方程都是线性的，这也给实际的应用增加了比较大的难度，因为实际中的系统不一定是线性的。卡尔曼滤波由于过强的假设，不能代表其他可能的状态向量分布的情况和其他非线性系统的情况，因此该方法有很大的局限性。

粒子滤波器是一种蒙特卡洛(Monte Carlo)的方法，通过非参数化的模型来实现贝叶斯

递推(Bayesian inference)。它没有线性系统的假设,因而适用于任何能用状态空间模型表示的非线性系统,这是卡尔曼滤波方法无法实现的。粒子滤波利用粒子集合(粒子即概率论中的样本)来近似表示状态向量在状态空间的概率密度,因此不受高斯分布的约束,甚至不要求状态向量的概率密度分布有显式表达。这一点大大扩展了粒子滤波的应用领域。

粒子滤波算法实现上非常简便。由于粒子滤波允许非线性的观测方程,可以采用很多种观测方式,而不再是一个简单的线性变换。这种观测方程的非线性性质使得观测量非常灵活,可以加入很多观测值。在目标跟踪中,观测值可以为颜色直方图、检测器的输出值等。因此,粒子滤波作为一个跟踪算法的框架,可以灵活地选择观测值,这比确定性跟踪算法中的核跟踪算法(Mean Shift)[Comaniciu 2003]要灵活得多。

在实际应用中,需对系统的状态空间进行建模。假设状态空间的观测值在离散时刻可以得到,即可在时间轴上进行固定间隔采样,并在采样时刻上得到目标的观测值。同时,可用微分方程对状态空间进行建模,此时考虑状态向量和观测向量两个主要的向量。其中状态向量包括了所关心系统的所有相关描述信息,比如跟踪系统中目标的位置、外观、速度、加速度等;而观测向量则是指在当前时刻得到的信息,比如粒子的颜色特征、纹理特征、检测器输出的响应等。如何选择这两个向量,是系统建模的第一步。

为了描述整个系统,必须对状态向量随着时间的演化进行建模,同时也要对观测向量进行建模。一般认为状态向量和观测向量都可以用概率的形式描述,因此状态向量的概率密度随着新的观测向量的到来而不断更新,这个过程可以用贝叶斯更新来描述。总体的框架就是用两个概率分布函数分别描述状态向量和观测向量。借助状态空间方程可用前一时刻的状态向量分布对当前时刻的状态向量分布进行预测。在预测的过程中,会出现很多不必要的扰动,对这个扰动,可用随机噪声对其进行建模。在预测之后用当前时刻的观测向量通过贝叶斯方法对状态向量进行更新,这样通过不断地输入观测向量,就可以得到状态向量在时域上的演化。如果状态向量包括跟踪过程中目标的运动位置,那么这个贝叶斯更新过程就能实现该目标的跟踪。

下面基于 CONDENSATION 算法[Isard 1998]以及粒子滤波的文献[Arulampalam 2002]来详细介绍粒子滤波器的各个组成部分。

1. 状态密度的离散时间传播

设概率密度传播过程在离散时间 t 发生。t 时刻被建模的目标的状态向量标记为 $\boldsymbol{x}_t$,故该状态向量对应的历史状态表示为 $\boldsymbol{x}_t=(x_1,x_2,\cdots,x_t)$。同时,在 t 时刻的观测向量(即该时刻的图片特征的集合)被标记为 $\boldsymbol{z}_t$,而观测向量的历史状态标记为 $\boldsymbol{z}_t=(z_1,z_2,\cdots,z_t)$。

2. 随机动态

假设该动态传播过程是一个马尔科夫过程,即

$$P(\boldsymbol{x}_{t+1}\mid\boldsymbol{x}_t)=P(\boldsymbol{x}_{t+1}\mid x_t) \tag{3.2.1}$$

当前时刻新状态出现的概率只与其前一个状态有关,与再之前的状态无关。而对于实际的应用问题,状态向量 $\boldsymbol{x}$ 一般是多维的,因此该条件概率密度函数比较复杂,需要根据实际情况来确定。

3. 观测向量

假设观测向量 $\boldsymbol{z}_t$ 是相互独立的,且与该动态的传播过程也相互独立,则可以用下式表达:

$$p(\boldsymbol{z}_t, x_{t+1} \mid \boldsymbol{x}_t) = p(x_{t+1} \mid x_t) \prod_{i=1}^{t} p(z_i \mid x_i) \tag{3.2.2}$$

且

$$p(\boldsymbol{z}_t \mid \boldsymbol{x}_t) = \prod_{i=1}^{t} p(z_i \mid x_i) \tag{3.2.3}$$

该观测过程通过在每个时间 t 定义指定条件密度 $p(z_t|x_t)$来实现，通常将其设为与时间无关的函数 $p(z|x)$。

4. 传播过程

给定一个马尔科夫链以及独立的观测，条件密度 $p(z_t|x_t)$在时域的传播准则如下：

$$p(x_{t+1} \mid \boldsymbol{z}_{t+1}) = k_{t+1} p(z_{t+1} \mid x_{t+1}) p(x_{t+1} \mid \boldsymbol{z}_t) \tag{3.2.4}$$

其中

$$p(x_{t+1} \mid \boldsymbol{z}_{t+1}) = \int_{x_t} p(x_{t+1} \mid x_t) p(x_t \mid \boldsymbol{z}_t) \tag{3.2.5}$$

且 k_{t+1}是归一化常数。该传播准则可以简单地理解成贝叶斯准则，是通过先验概率和观测值推导后验概率的一种方式。

5. 用粒子集合表示任意概率密度

在实际的应用过程中，涉及的概率密度函数可能是非高斯的，甚至不能显式表示。当概率密度函数不能用显式表示时，就迫切需要一种能够有效地表示任何形状的概率密度函数的方法。下面将详细介绍如何用粒子集合来表示任意的概率密度函数。

对于一个概率密度函数，如果能够按照这种概率分布进行抽样，得到 N 个粒子(即样本)，那么就可以用这 N 个粒子来近似地表示该概率密度函数。直观上，当 N 越大的时候，该概率密度函数的表示方式应该越准确。

令 $x_{0:k}=(x_0, x_1, \cdots, x_k)$是状态变量；$D_k=(y_1, y_2, \cdots, y_k)$是到当前时刻为止的所有观测值(为了与上面部分有所区别，这里用 D_k 表示观测值，这里的 D_k 可以理解成前后文的观测值 $\boldsymbol{z}_k$)。同时初始概率分布已知为 $p(x_0)$。

系统模型表示为(w 为权重)

$$x_k = f_k(x_{k-1}, w_k) \quad \leftrightarrow \quad p(x_k \mid x_{k-1}) \tag{3.2.6}$$

观测模型表示为(v 为权重)

$$y_k = h_k(x_k, v_k) \quad \leftrightarrow \quad p(y_k \mid x_k) \tag{3.2.7}$$

用采样得到的粒子来表示概率密度函数如下：

$$\hat{p}(x_{0:k} \mid D_k) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_{0:k}^i}(dx_{0:k}) \tag{3.2.8}$$

其中 $x_{0:k}^i$是从后验概率 $p(x_{0:k}|D_k)$中抽样得到的样本。这样的近似表示相当直观。如果能够从某个概率密度函数中得到足够多的样本粒子，则就可以通过这些粒子来近似表示对应的概率密度。

但是从有些概率密度中高效快速地采样是非常困难的，因此需要将该问题转化。从要表示的概率密度 $p(x_{0:k}|D_k)$中采样是困难的，但总能找到一个容易采样的概率密度函数 $q(x_{0:k}|D_k)$，使得从 $q(x_{0:k}|D_k)$中采样是容易的。基于从 $q(x_{0:k}|D_k)$中采样得到的粒子，希望能够表示出 $p(x_{0:k}|D_k)$的分布性质。虽然没有显式地写出 $p(x_{0:k}|D_k)$的表达式，但

是能够求出服从 $p(x_{0:k}|D_k)$分布的变量的任意函数值，这样前面的目的也就可以达到了。

假设 $E[g(x)]$是希望得到的值。例如，当 $g(x)=x$ 时，那么就是在求期望

$$E[g(x_{0:k})]=\int g(x_{0:k})\frac{p(x_{0:k}\mid D_k)}{q(x_{0:k}\mid D_k)}q(x_{0:k}\mid D_k)\mathrm{d}x_{0:k}$$
$$=\int g(x_{0:k})\frac{p(D_k\mid x_{0:k})p(x_{0:k})}{p(D_k)q(x_{0:k}\mid D_k)}q(x_{0:k}\mid D_k)\mathrm{d}x_{0:k}$$
$$=\int g(x_{0:k})\frac{w_k(x_{0:k})}{p(D_k)}q(x_{0:k}\mid D_k)\mathrm{d}x_{0:k} \tag{3.2.9}$$

$$P(D_k)=\int p(D_k,x_{0:k})\mathrm{d}x_{0:k}=\int\frac{p(D_k\mid x_{0:k})p(x_{0:k})q(x_{0:k}\mid D_k)}{q(x_{0:k}\mid D_k)}\mathrm{d}x_{0:k}$$
$$=\int w_k q(x_{0:k}\mid D_k)\mathrm{d}x_{0:k} \tag{3.2.10}$$

由式(3.2.9)可以知道，$E[g(x)]$可以用从 $q(x_{0:k}|D_k)$中采样得到的粒子来表示，每个粒子必须带权重 $w_k(x_{0:k})/(D_k)$。对权重的迭代式子为

$$w_k=\frac{p(D_k\mid x_{0:k})p(x_{0:k})}{q(x_k\mid x_{0:k-1},D_k)q(x_{0:k-1}\mid D_{k-1})}$$
$$=w_{k-1}\frac{p(D_k\mid x_{0:k})}{p(D_{k-1}\mid x_{0:k-1})}\frac{p(x_{0:k})}{p(x_{0:k-1})}\frac{1}{q(x_k\mid x_{0:k-1},D_k)} \tag{3.2.11}$$
$$=w_{k-1}\frac{p(y_k\mid x_k)p(x_k\mid x_{k-1})}{q(x_k\mid x_{0:k-1},D_k)}$$

因此可以得到

$$E(g(x_{0:k}))=\frac{\int(g(x_{0:k})w_k(x_{0:k}))q(x_{0:k}\mid D_k)\mathrm{d}x_{0:k}}{\int w_k(x_{0:k})q(x_{0:k}\mid D_k)\mathrm{d}x_{0:k}} \tag{3.2.12}$$

$$\overline{E(g(x_{0:k}))}=\frac{\frac{1}{N}\sum_{i=1}^{N}g(x_{0:k}^i)w_k(x_{0:k}^i)}{\frac{1}{N}\sum_{i=1}^{N}w_k(x_{0:k}^i)}=\sum_{i=1}^{N}g(x_{0:k}^i)\,\tilde{w}_k(x_{0:k}^i) \tag{3.2.13}$$

因此可以通过不断的迭代实现求解任何时刻所关心的状态向量的任何函数，而不需要显式地表达该函数。

6. 重采样

由于粒子滤波产生了很多粒子，而在中间会出现退化问题，这意味着会花费大量的时间去计算那些权重非常低的粒子，这是不希望看到的。为此可通过重采样的方式来实现去掉那些权重太低的粒子，而增加权重高的粒子的作用。以 $x_{0:k}^i$表示采样前的粒子，$\tilde{w}_k(x_{0:k}^i)$为其对应的权重，经过重采样的粒子为 $x_{0:k}^j$，采样后的粒子权重为 $1/N$，则重采样过程可表示为

$$[x_{0:k}^i,\tilde{w}_k(x_{0:k}^i)]\rightarrow(x_{0:k}^j,1/N) \tag{3.2.14}$$

常用的重采样步骤如下：

(1) 初始化：$c_1=0$。

(2) FOR $i=2:N$

$-c_1=c_{i-1}+\tilde{w}_k^i$

END FOR