

Research on Video Target Tracking Based on Particle Filter and Mean-Shift

Siya Wang¹, Ziliang Feng²

¹College of Computer Science, Sichuan University, Chengdu Sichuan

²The National Key Discipline Laboratory of Visual Synthesis Graphics Technology, Sichuan University, Chengdu Sichuan

Email: wendywang0620@163.com

Received: Aug. 23rd, 2017; accepted: Sep. 5th, 2017; published: Sep. 11th, 2017

Abstract

Target tracking has been already a research direction in the field of pattern recognition and computer vision. Especially, video target tracking has become a focus of study. Target tracking plays an important role in the intelligent monitoring, traffic monitoring, traffic statistics, and so on, due to the variability of shooting environment, the fluid of target motion state, the shading of targets and the interference of likeness, etc. These factors make video target tracking more complicated. Video target tracking can be defined as select single or multiple target from the video sequences filming with cameras, and can give target location accurately and timely, and then get target motion trajectory and its motor habit. Mainly in the framework of particle filter algorithm, this paper analyzes the factors which affect the effect of particle filter tracking, and focuses on the study of the target feature extraction, the similarity measurement method between models, controlling the number of particles, in order to improve the robustness, accuracy and instantaneity of the algorithm. Particle filter video target tracking algorithm based on Mean Shift is proposed. The algorithm uses an iterative process of Mean Shift algorithm after the particle initialization and resampling to reduce the amount of calculation. The algorithm updates the target model after estimating location of the target, in order to adapt to changes of the target. The experimental results show that the effect of the particle filter tracking based on Mean Shift is better, and it needs shorter time.

Keywords

Particle Filter, Target Tracking, Mean Shift

基于Mean Shift的粒子滤波视频目标跟踪算法

王思雅¹, 冯子亮²

¹四川大学计算机学院, 四川 成都

²视觉合成图形图像技术国家重点学科实验室, 四川 成都

Email: wendywang0620@163.com

收稿日期: 2017年8月23日; 录用日期: 2017年9月5日; 发布日期: 2017年9月11日

摘要

目标跟踪一直是计算机视觉领域的重要研究方向, 由于拍摄环境的多变性、目标运动状态的不固定性、目标间的相互遮挡以及相似物的干扰等因素使视频目标跟踪变得更加复杂。本文以粒子滤波算法为框架研究目标跟踪算法, 重点对目标特征提取、模型间相似度的度量方法、控制跟踪过程中粒子数量进行研究, 以提高算法的鲁棒性、准确性和实时性。提出了一种基于Mean Shift的粒子滤波视频目标跟踪算法。该算法为了减少光照的干扰, 采用HSV颜色空间的核函数加权颜色直方图来描述颜色特征, 在粒子初始化和重采样后采用一次Mean Shift算法的迭代过程将一些作用弱的粒子不参与计算, 以减少计算量。完成一次目标估计后更新目标模型, 以适应目标在运动过程的变化。对比实验结果表明, 本算法具有较好的实时性。

关键词

粒子滤波, 目标跟踪, 均值漂移

Copyright © 2017 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在目标跟踪过程中粒子滤波[1] [2] [3] [4] [5]的方法里有些粒子对目标位置的确定可能起的作用很少或者根本不起作用, 这些粒子涉及粒子权重的大量计算, 这样无疑增加了计算量; 如果粒子能够尽量在目标的真实位置附近, 估计出的目标位置就会更准确。均值漂移(Mean Shift) [3]-[12]算法是一种不断向目标真实位置漂移的跟踪算法, 将均值漂移算法和粒子滤波算法相结合可以更快地得到跟踪结果[4] [5] [6]。

Mean Shift 算法是一种基于统计迭代的无参数密度估计方法, 它的高效性使其可适用于实时性要求高的场合; 但其算法要求视频相邻帧间要有目标重叠部分, 如果目标被遮挡或运动较快, 算法就不能向目标中心漂移, 而是跟踪到错误区域, 造成跟踪失败。

Mean Shift 算法实质是寻找目标的最优值并传播到下一帧中, 如果最优值不是准确的目标位置, 这种错误将逐帧积累, 最后甚至会导致跟踪失败。粒子滤波在跟踪过程中不仅保留最优值, 而且也保留部分次优值, 只是权值不同的传播到下一帧中。即使目标丢失, 目标的多种可能性也会传递下去, 当目标重新出现时, 权值发生变化, 粒子滤波依然能跟踪, 所以粒子滤波对目标遮挡具有较好的鲁棒性。

Mean Shift 算法求解 Mean Shift 向量的过程, 是逐步向目标密度增大的方向移动的过程, 也就是逐步向目标真实的位置移动的过程。将 Mean Shift 算法的优点应用到粒子滤波算法中, 可以提高粒子滤波跟踪算法的效率。本文验证了将 Mean Shift 算法与粒子滤波相结合的方法, 在实时性和鲁棒性上有了提升。

2. Mean Shift 算法

在 d 维空间有 n 个样本点, 在 x 点的 Mean Shift 向量可表示为

$$M_h(x) = \frac{\sum_{x_i \in s_h} (x_i - x)}{k} \tag{1}$$

式中 s_h 表示半径为 h 的区域, k 表示有 k 个样本点落在 s_h 区域中, $(x_i - x)$ 表示 x_i 相对于 x 的偏移量。 n 个样本点的位置各不相同, 有的样本点恰好在 s_h 区域中, 不在区域中的样本点就没有作用。在 s_h 区域中样本点相对于 x 的偏移量的方向各不相同, $M_h(x)$ 就是将各个偏移向量相加后求平均。如图 1 所示 $M_h(x)$ 就会指向偏移向量最密集的区域, 即样本最多、概率密度最大的方向。

Mean Shift 向量, 只要落在 s_h 区域中的样本点, 它们对 $M_h(x)$ 的贡献都是平等的。离基准点 x 的距离应该对 Mean Shift 向量的求解有影响所以引入核函数 $K(x)$ 。Mean Shift 算法常选择积分均方误差下的最优核函数 Epanechnikov 核函数。它的定义如下

$$K_E(x) = \begin{cases} \frac{1}{2} C_d^{-1} (d+2) (1 - \|x\|^2) & \|x\| < 1 \\ 0 & \|x\| \geq 1 \end{cases} \tag{2}$$

其中 C_d 为 d 维空间的单位球体体积。

故而 Mean Shift 向量可表示为

$$M_h(x) = \frac{\sum_{x_i \in s_h} (x_i - x) K_E(x_i - x)}{\sum_{x_i \in s_h} K_E(x_i - x)} \tag{3}$$

3. 粒子滤波

粒子滤波的基本思想是, 对于待求的变量, 假设已知先验概率, 则可用足够多的粒子来逼近真实的后验概率密度。粒子滤波与维数无关, 在解决高维问题上有很强的优势。

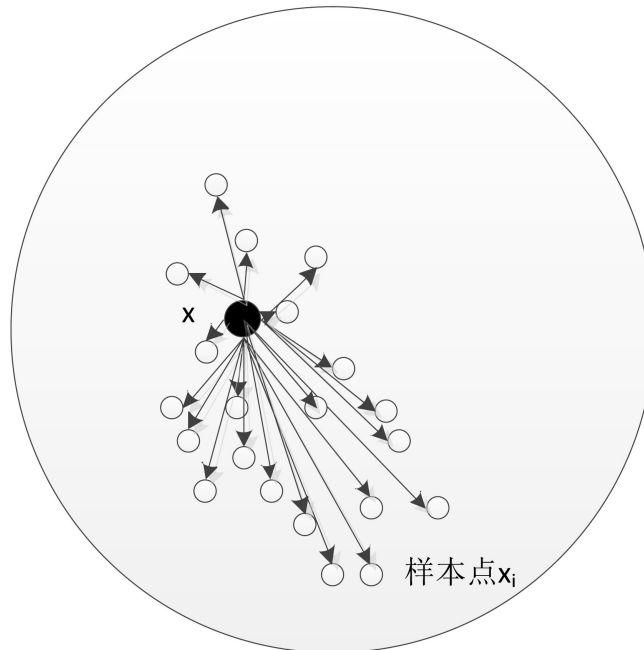


Figure 1. The diagram of mean shift
图 1. Mean shift 示意图

3.1. 标准粒子滤波

在 k 时刻用多个样本点, 即粒子集 $\{X_{0k}^{(i)}, \omega_{0k}^{(i)}\} (i=1, \dots, N)$ 来描述状态的后验概率密度 $p(X_{0k}|Z_{1k})$, $X_{0k}^{(i)}$ 表示在 0 到 k 时刻随机抽取 N 个粒子, 0 到 k 时刻粒子的权重为 $\omega_{0k}^{(i)}$, 且 $\sum_{i=1}^N \omega_{0k}^{(i)} = 1$, k 时刻目标的后验概率密度可近似表示为

$$p(X_{0k}|Z_{1k}) \approx \sum_{i=1}^N \omega_k^{(i)} \delta(X_{0k} - X_{0k}^{(i)}) \quad (4)$$

其中权值通过重要性采样得到, 从重要密度函数 $q(X_{0k}|Z_{1k})$ 抽取样本粒子, 则重要性权重表示为

$$\omega_k^{(i)} = \frac{p(X_{0k}^{(i)}|Z_{1k})}{q(X_{0k}^{(i)}|Z_{1k})} \quad (5)$$

若重要密度函数可以表示成递推的形式

$$q(X_{0k}|Z_{1k}) = q(X_k|X_{0k-1}, Z_{1k})q(X_{0k-1}|Z_{1k-1}) \quad (6)$$

那么由 $q(X_k|X_{0k-1}, Z_{1k})$ 得到的粒子集 $\{X_k^{(i)}\} (i=1, \dots, N)$ 和由 $q(X_{0k-1}|Z_{1k-1})$ 得到的粒子集 $\{X_{0k-1}^{(i)}\} (i=1, \dots, N)$ 能获得新的粒子集 $\{X_{0k}^{(i)}\} (i=1, \dots, N)$ 。

由于后验概率密度可表示成

$$\begin{aligned} p(X_{0k}|Z_{1k}) &= \frac{p(Z_k|X_{0k}, Z_{1k-1})p(X_{0k}|Z_{1k-1})}{p(Z_k|Z_{1k-1})} \\ &= \frac{p(Z_k|X_{0k}, Z_{1k-1})p(X_k|X_{0k-1}, Z_{1k-1})}{p(Z_k|Z_{1k-1})} p(X_{0k-1}|Z_{1k-1}) \\ &= \frac{p(Z_k|X_k)p(X_k|X_{k-1})}{p(Z_k|Z_{1k-1})} p(X_{0k-1}|Z_{1k-1}) \\ &\propto p(Z_k|X_k)p(X_k|X_{k-1})p(X_{0k-1}|Z_{1k-1}) \end{aligned} \quad (7)$$

将公式(6)和(7)带入(5)中, 得到更新的重要性权重

$$\begin{aligned} \omega_k^{(i)} &\propto \frac{p(Z_k|X_k^{(i)})p(X_k^{(i)}|X_{k-1}^{(i)})p(X_{0k-1}^{(i)}|Z_{1k-1})}{q(X_k^{(i)}|X_{0k-1}^{(i)}, Z_{1k})q(X_{0k-1}^{(i)}|Z_{1k-1})} \\ &= \omega_{k-1}^{(i)} \frac{p(Z_k|X_k^{(i)})p(X_k^{(i)}|X_{k-1}^{(i)})}{q(X_k^{(i)}|X_{0k-1}^{(i)}, Z_{1k})} \end{aligned} \quad (8)$$

如果重要密度函数只和它的上一个时间状态 X_{k-1} 还有现在时刻的观测量 Z_k 有关, 则 $q(X_k|X_{0k-1}, Z_{1k}) = q(X_k|X_{k-1}, Z_{1k})$ 。在计算过程中只需保存现在时刻的粒子集, 而不需要关心它前面时刻的粒子集, 这样大量的节约了计算时间。修改后的重要性权重为

$$\omega_k^{(i)} \propto \omega_{k-1}^{(i)} \frac{p(Z_k|X_k^{(i)})p(X_k^{(i)}|X_{k-1}^{(i)})}{q(X_k^{(i)}|X_{k-1}^{(i)}, Z_{1k})} \quad (9)$$

标准粒子滤波重要性密度函数选择最易于实现的先验概率密度

$$q\left(X_k^{(i)} \mid X_{k-1}^{(i)}, Z_{1:k}\right) = p\left(X_k^{(i)} \mid X_{k-1}^{(i)}\right) \quad (10)$$

将(9)式带入(10)式中得到化简的重要性权重

$$\omega_k^{(i)} \propto \omega_{k-1}^{(i)} p\left(Z_k \mid X_k^{(i)}\right) \quad (11)$$

将重要性权重归一化

$$\omega_k^{(i)} = \omega_k^{(i)} / \sum_{i=1}^N \omega_k^{(i)} \quad (12)$$

在递推的过程中, 粒子的权重发生了变化, 可能它的权重已经变得很小, 但还在重复的进行后验概率的计算, 这样增加了无用的计算量, 也使后验概率不准确, 这就是粒子的退化现象。

为了减少粒子的退化现象, 提出了重采样的思想。重采样的思想实际就是将权重小的粒子剔除, 复制权重大的粒子的过程。重复的重采样会使粒子的多样性消失, 所以设定描述退化程度的尺度值 N_{eff} , 尺度值小于某个设定的阈值时才重采样, 重采样后粒子的权重相等。

尺度值定义为

$$N_{eff} = \frac{1}{\sum_{i=1}^N \left(\omega_k^{(i)}\right)^2} \quad (13)$$

标准粒子滤波算法可以用图 2 的流程图表示。

3.2. 标准粒子滤波应用于跟踪

粒子滤波目标跟踪的过程大致分为以下几个方面。粒子滤波跟踪的流程图如下图 3。

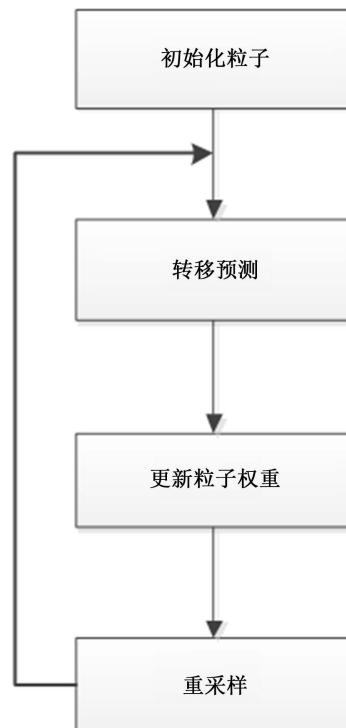


Figure 2. The flow chart of the algorithm of particle filter
图 2. 标准粒子滤波算法流程图

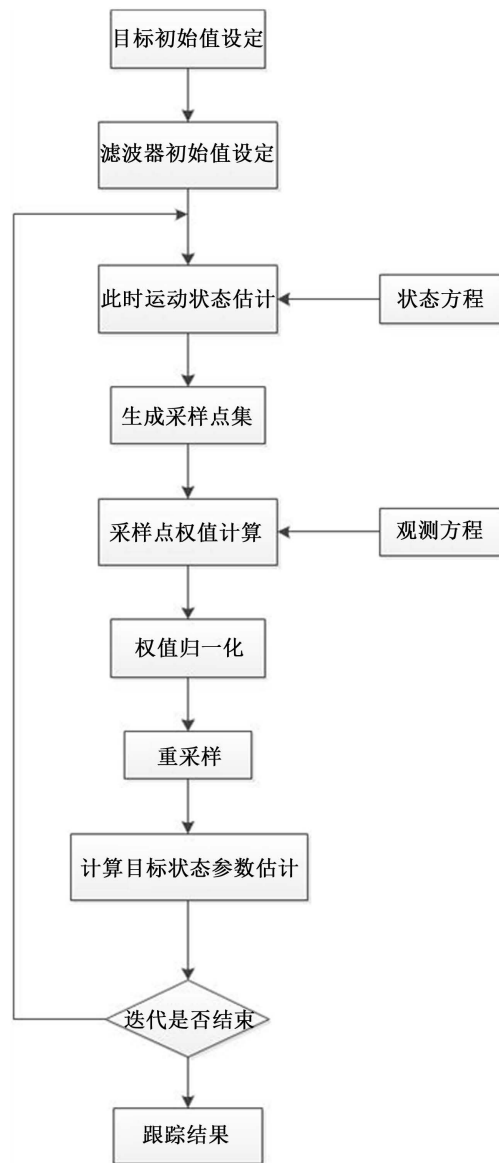


Figure 3. The flow chart of the tracking algorithm of particle filter
图 3. 粒子滤波跟踪算法流程图

系统状态转移模型的描述

运动模型是目标跟踪的基础, 运动模型的建立应该符合实际, 在描述运动中位置和速度是最重要的指标。在视频跟踪中, 相邻帧间目标的位置移动一般不会很大, 所以选择二阶常速模型来建立运动模型。在描述目标的运动状态时目标的轮廓大小也是考虑的因素, 用矩形或椭圆表示目标轮廓大小。最终目标状态可表示为

$$X = [x, x_v, y, y_v, h_x, h_y]^T \quad (14)$$

其中 (x, y) 表示目标的位置, 即目标的轮廓中心; (x_v, y_v) 表示目标在 x 方向和 y 方向上的速度; $h = (h_x, h_y)$ 表示目标轮廓的宽度和高度。目标的位置采用匀速模型, 目标的轮廓大小在运动过程中波动性比较大, 所以采用随机扰动模型。因此目标随时间的运动状态可以表示为

$$X_k = \begin{bmatrix} 1 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} X_{k-1} + \begin{bmatrix} \frac{\Delta T^2}{2} & 0 & 0 & 0 \\ \Delta T & 0 & 0 & 0 \\ 0 & \frac{\Delta T^2}{2} & 0 & 0 \\ 0 & \Delta T & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} W_k \quad (15)$$

式中 ΔT 为采样周期; W_k 为高斯白噪声。

目标初始值设定

在视频第一帧中选定目标, 对区域内的特征建立目标模型。假设以直方图作为描述

$$\{p^{(u)}\}_{u=1,2,\dots,m}, \sum_{u=1}^m p^{(u)} = 1 \quad (16)$$

粒子的初始化

目标的初始状态就是初始化粒子的状态 $\{X_0^{(i)}\} (i=1, \dots, N)$, 每个粒子的权重 $\omega_0^{(i)}$ 。 $X_0^{(i)}$ 表示第 i 个粒子的状态, 共有 N 个粒子, 每个粒子的权重 $\omega_0^{(i)} = 1/N$ 。

转移预测

经过状态转移方程, 粒子的位置发生了变化。

观测模型的描述

以每一个粒子为中心建立候选模型

$$q(z) = \{q^{(u)}\}_{u=1,2,\dots,m}, \sum_{u=1}^m q^{(u)} = 1 \quad (17)$$

粒子权重的确定

每一个候选模型与目标模型相似度的计算, 进而确定观测模型似然函数 $p(Z_k | X_k^{(i)})$, 计算出每个粒子的权重, 最后归一化粒子权重, 确定目标的状态估计。

重采样

为了防止粒子的退化, 采用重采样的方法把权重小的粒子剔除, 权重大的粒子保留并复制。

没有跟踪到最后一帧则循环进行转移预测、粒子权重的确定、重采样的工作, 直到最后一帧。

4. Mean Shift 粒子聚类的粒子滤波流程

粒子滤波算法适用范围之广使粒子滤波在各个领域受到欢迎, 但是粒子滤波算法由于需要不断的更新粒子使计算量成倍增加, 算法的实时性不是很好, 粒子滤波目标跟踪算法同样也面临实时性的问题。

Mean Shift 向量总是向着密度增大的方向漂移, 将这一点应用在粒子滤波中可以起到很好的作用。为了更准确的逼近目标, 粒子滤波算法采用大量的粒子, 大量粒子的计算使粒子滤波的实时性降低。将 Mean Shift 算法中的聚类特性应用在粒子滤波中, 对粒子进行聚类, 使粒子更接近目标的真实位置, 也减少了粒子数目, 降低了计算量, 提高了跟踪效率。本文采用在粒子更新权重后对粒子聚类。

目标跟踪中运动模型和观测模型的建立都非常关键, 由于 Mean Shift 算法中用到概率密度和核函数, 所以在建立模型时, 采用颜色的核密度估计。

4.1. 观测模型及定位

在建立模型时这里采用颜色特征, 但是 RGB 颜色空间对光照敏感, 不同的光照强度, 我们肉眼看到

的颜色可能不同。还有另一种颜色空间 HSV (Hue, Saturation, Value)更接近于人们对颜色的感知, 且对光照不敏感。HSV 颜色空间输出的数据都是 double 类型, 则每个颜色变量的取值范围是, HSV 颜色空间如图 4 所示, H 表示色调, 用角度来表示, 一般以红色轴为, 顺着逆时针方向依次增大, 红、绿、蓝三原色依次相差。 V 表示某一像素的灰度值, V 越大越明亮。 S 表示饱和度, 它到 V 轴的距离, S 越大颜色越鲜艳。

拍摄设备输出的一般都是 RGB 模式的图像, RGB 色彩空间对光照敏感, 为了使跟踪算法能适应光照的变化, 将 RGB 颜色空间转化到 HSV 颜色空间。RGB 颜色空间可以很容易的转化到 HSV 颜色空间, 即从笛卡尔坐标系映射到柱坐标系。首先将 RGB 的取值范围都变成区间, 即 double 数据类型。如果 RGB 的数据类型是 unit 8, 则三个颜色分量的每个值都除以 256; 如果 RGB 的数据类型是 unit 16, 则三个颜色分量的每个值都除以 65,535。然后按照下面的公式

$$V = \max(\max(R, G), B) \quad (18)$$

$$S = \begin{cases} 0 & V = 0 \\ 1 - \frac{\min(R, G, B)}{V} & V \neq 0 \end{cases} \quad (19)$$

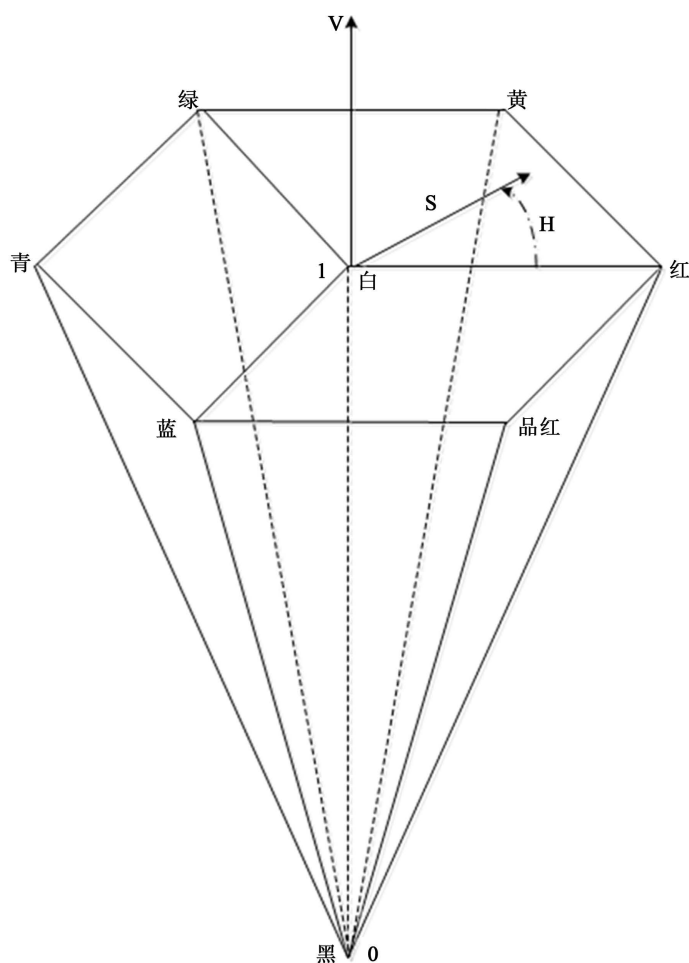


Figure 4. HSV color spaces
图 4. HSV 颜色空间

$$H = \begin{cases} 0^\circ & V = \min(R, G, B) \\ 60^\circ \times \left(\frac{G - B}{V - \min(R, G, B)} \right) & V = R, G \geq B \\ 60^\circ \times \left(2 + \frac{B - R}{V - \min(R, G, B)} \right) & V = G \\ 60^\circ \times \left(4 + \frac{R - G}{V - \min(R, G, B)} \right) & V = B \\ 60^\circ \times \left(6 + \frac{G - B}{V - \min(R, G, B)} \right) & V = R, G < B \end{cases} \quad (20)$$

在光照发生变化时 HSV 颜色空间直方图变化不明显。如图 5(a)和图 5(b)是两张同一景物但光照有所差别的照片，我们可以看到光照发生变化时，图 5(a)、图 5(b)的 GRB 和 HSV 颜色空间的颜色直方图分别如图 6、图 7 所示。可以看出图 6 和图 7 的 RGB 颜色直方图差别很大，但 HSV 颜色直方图却差别不大。



Figure 5. The pictures of same scene in different light
图 5. 同一场景光照不同的照片

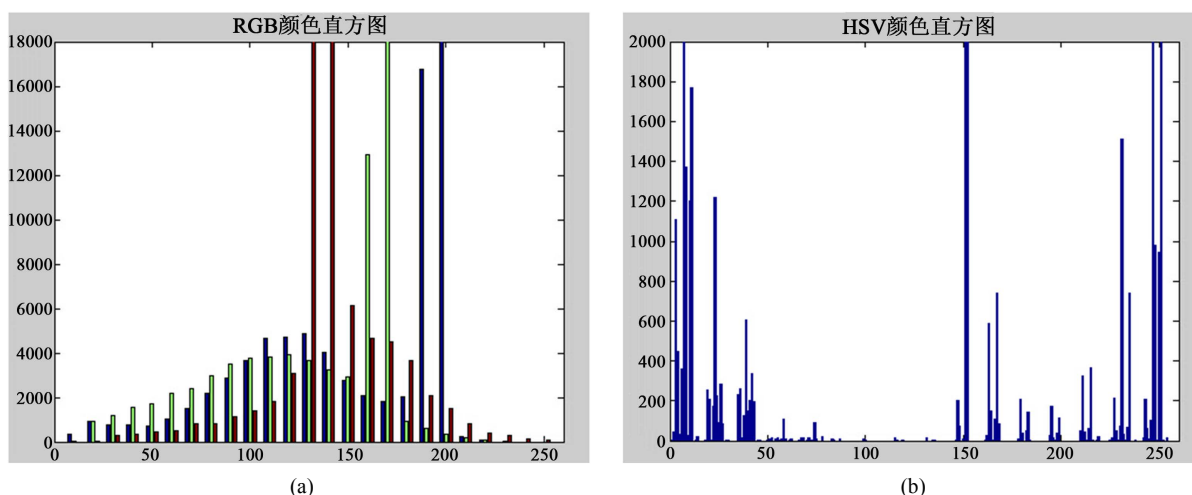


Figure 6. Color histograms in different color spaces of Figure 5(a)
图 6. 图 5(a)在不同颜色空间的颜色直方图

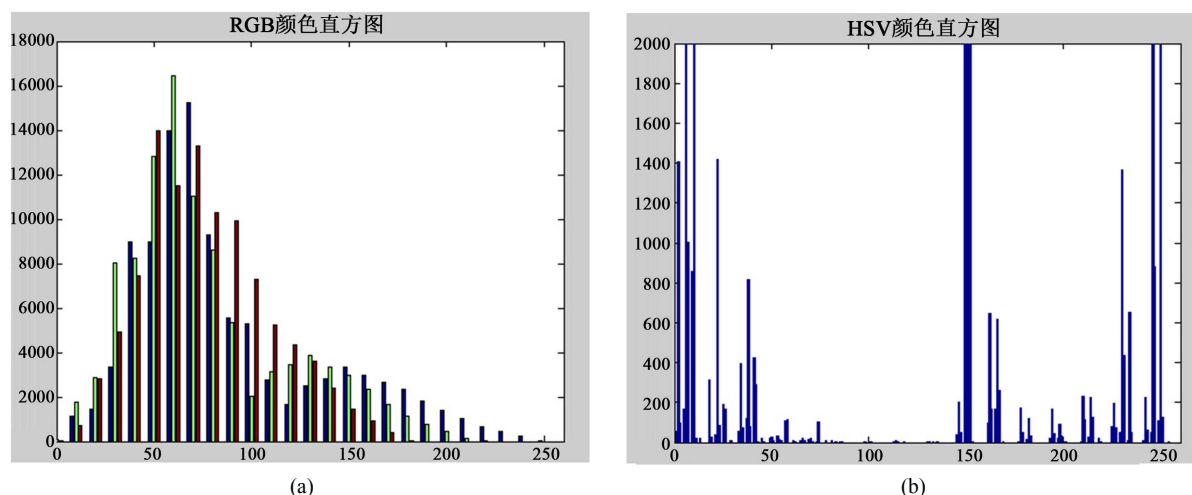


Figure 7. Color histograms in different color spaces of Figure 5(b)

图 7. 图 5(b)在不同颜色空间的颜色直方图

为了能更好的适应光照变化, 将颜色信息转换到 HSV 颜色空间。这一章中采用 HSV 颜色空间的核函数加权颜色直方图, 它考虑了光照影响, 也考虑了像素位置对建立模型的影响。

目标模型

在视频第一帧中选定需要跟踪的目标, 目标区域的中心位置为 $l = (x_0, y_0)$, 目标区域像素的位置为 $l_i = (x_i, y_i), i = 1, 2, \dots, n$, n 为目标区域内像素总数, 目标的颜色表示为

$$q^{(u)} = \frac{\sum_{i=1}^n k\left(\left\|\frac{l-l_i}{h}\right\|\right) \delta[b(l_i)-u]}{\sum_{i=1}^n k\left(\left\|\frac{l-l_i}{h}\right\|\right)} \quad (21)$$

式中 $k(\cdot)$ 是权值函数, 为了表示离目标远近属于目标区域的权值, 把其定义为

$$k(\|r\|) = \begin{cases} 1 - \|r\|^2 & \|r\| < 1 \\ 0 & \|r\| \geq 1 \end{cases} \quad (22)$$

h 表示目标区域的大小, $h = \sqrt{h_x^2 + h_y^2}$;

$\delta(x)$ 为狄拉克(Delta)函数, 其定义为 $\delta(x) = \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$;

$b(l_i)$ 表示位置为 l_i 的像素的颜色的索引号; u 表示颜色直方图中颜色索引号;

$u = 1, 2, \dots, m$, m 表示将 HSV 颜色空间划分为 m 个等级。

候选模型

与目标模型的建立基本相似, 但是每一个粒子都可以建立一个候选模型, 候选模型是以每个粒子的位置为中心为 t , 以 h 为半径找出和目标区域相同大小的区域, 区域内每个像素处的位置为 l'_i , 则第 i 个粒子对应的候选模型

$$p_i^{(u)} = \frac{\sum_{i=1}^n k\left(\left\|\frac{t-l'_i}{h}\right\|\right) \delta[b(l'_i)-u]}{\sum_{i=1}^n k\left(\left\|\frac{t-l'_i}{h}\right\|\right)} \quad (23)$$

更新粒子权重

目标模型和候选模型的相似度可用 Bhattacharyya 系数表示 $\rho_i = \sum_{u=1}^m \sqrt{q^{(u)} p_i^{(u)}}$

则目标模型 $q^{(u)}$ 和候选模型 $p_i^{(u)}$ 的相似距离表示为 $d_i = \sqrt{1 - \rho_i}$

得到目标模型和候选模型的相似度后, 进而可以得到颜色观测的似然函数

$$p(Z_k | X_k^{(i)}) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{d_i^2}{2\sigma^2}\right) \quad (24)$$

根据公式(14)下一时刻的粒子权重更新为

$$\omega_k^{(j)} = \omega_{k-1}^{(j)} p_c(Z_k | X_k^{(j)}) \quad (25)$$

对粒子权重归一化

$$\omega_k^{(j)} = \omega_k^{(j)} / \sum_{j=1}^N \omega_k^{(j)} \quad (26)$$

得到最终的粒子权重。

目标定位

经过上面的工作, 已经确定了运动模型和粒子的权重, 据此可以估计出目标的位置。目标的最优估计采用加权准则, 得到

$$\hat{X}_k = \sum_{i=1}^n \omega_k^{(i)} X_k^{(i)} \quad (27)$$

经过几次递归在最优目标估计中有些粒子的权重很小, 而有些粒子的权重很大, 权重很小的粒子几乎已经不起作用了, 但还要重复的计算, 增加了计算量, 所以需重采样, 再进行跟踪。

4.2. 改进的粒子滤波

对于粒子滤波来说, 粒子的数量至关重要。如果粒子数过多, 计算量增大, 如果计算的速度小于帧间隔那么就会丢帧。如果粒子过少, 粒子就不能很好的逼近目标, 那么粒子滤波的优势也将失去。在初始生成粒子还有重采样生成的粒子, 先采用 Mean Shift 算法中的聚类作用向目标区域移动。

在某帧中初始化粒子, 选定目标并建立目标模型, 在后续帧中目标的真实位置应该在前一帧选定目标的位置的附近, 但当目标移动过快或过慢, 可能真实的目标和候选位置有一定的差距。所以使用一次 Mean Shifts 聚类, 使搜索窗移动一次。如果相比于原来的候选区域, 中心位置向右移动, 则搜索窗口左边的粒子对目标的寻找起到的作用不大, 丢弃这些粒子; 如果相比于原来的候选区域, 中心位置向左移动, 则搜索窗口右边的粒子对目标的寻找起到的作用不大, 丢弃这些粒子。丢弃一些粒子之后再建立粒子的候选模型, 这样就降低了计算量。但在聚类过程中只聚类一次, 聚类多次会使粒子多样性退化。

对于快速运动的目标, 目标模型可能在不同的环境可能会发生比较大的变化, 一直使用初始选定的目标模型, 可能会使误差不断增大, 所以在跟踪过程中不断更新目标模型。

具体的基于 Mean Shift 的粒子滤波视频目标跟踪算法如图 8 的流程图。

5. 实验及结果分析

本章将检验我们提出的算法的跟踪效果, 采用的视频基本信息如表 1, 将比较本章加入 Mean Shift 聚类的粒子滤波算法和没有加入 Mean Shift 聚类的粒子滤波算法的性能。

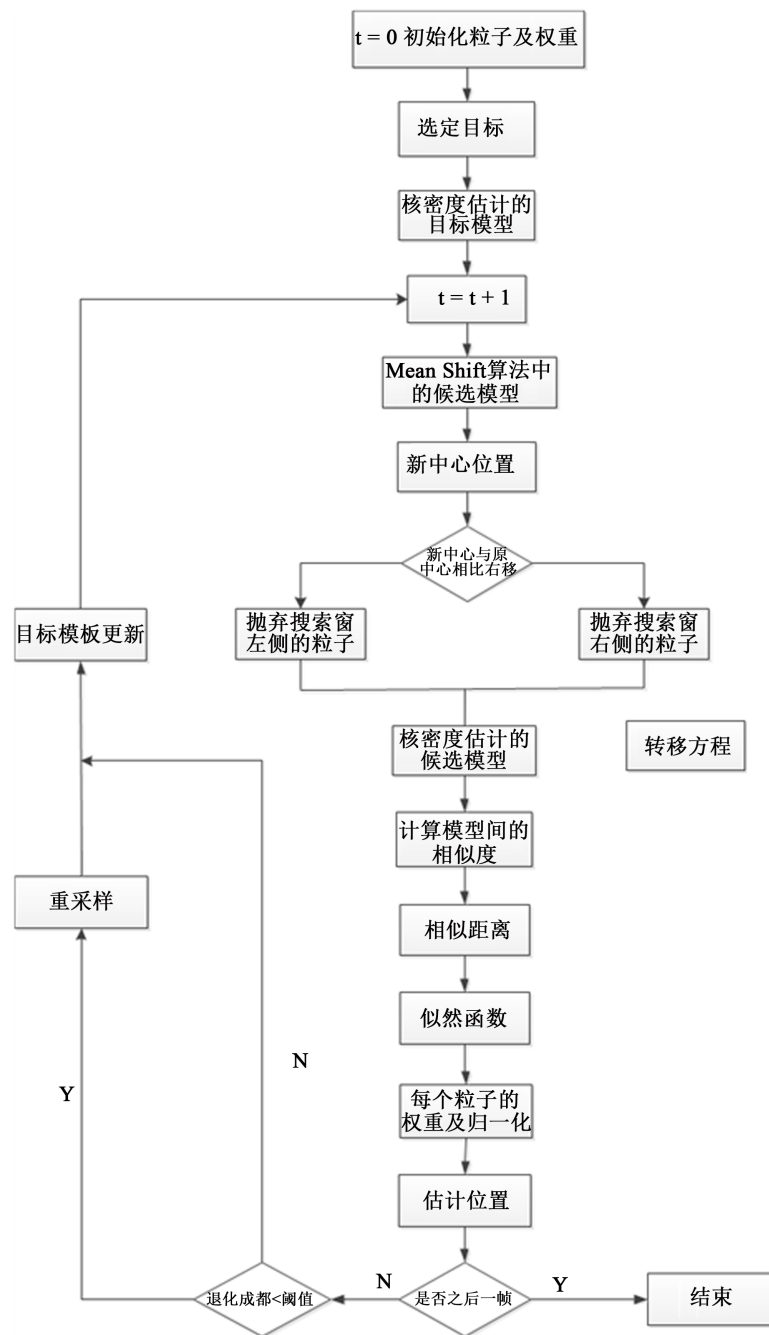


Figure 8. The flow chart of tracking algorithm of mean-shift and particle filter
图 8. Mean shift 粒子滤波跟踪流程图

Table 1. The information of the video

表 1. 视频信息

视频	视频
总帧数	551
帧大小	352 × 288
帧速率	25 帧/秒

首先在视频第一帧选中目标, 目标用蓝色矩形框表示, 在跟踪过程中蓝色矩形框会根据跟踪效果发生变化, 蓝色矩形框在跟踪效果不好时会逐渐变小, 蓝色矩形框消失表示没有跟踪到目标。初始设定 400 个粒子, 粒子用绿色的圆圈表示。为了观察的方便把视频的每一帧都编号, 用黄色数字表示。

实验结果分析

视频是车辆在公路上行驶的视频序列, 在第一帧选定黑色的车子为跟踪目标, 目标一直在公路上行驶。选取 30 帧、105 帧、229 帧、316 帧、477 帧、551 帧观察跟踪效果。

从图 9、图 10 可以看出, 两种算法的前三帧跟踪效果差不多, 但后三帧基于 Mean Shift 的粒子滤波算法的跟踪效果要好一些。

除了从视频跟踪结果直观对比外, 还可以从三个跟踪指标评价跟踪算法的优劣, 一是运行时间; 二是运动轨迹; 三是误差距离。运行时间如表 2, 运动轨迹见图 10, 误差距离见图 11。

从表 2 可以看出 Mean Shift 的粒子滤波算法运行时间短, 主要原因是 Mean Shift 聚类丢弃了一些作用弱的粒子。

从图 11 可以看出, 在 229 帧处 y 方向的轨迹与真实的轨迹差距有些大, 原因是受到相同颜色车辆的干扰。在其他帧处本章基于 Mean Shift 的粒子滤波视频跟踪算法的轨迹更接近目标的真实轨迹。

从图 12 可以看出, 从总体上本章的算法离目标的真实位置更近。



Figure 9. The tracking results of particle filter
图 9. 粒子滤波跟踪效果

Table 2. Runtime of different algorithm
表 2. 两种算法的运行时间

跟踪方法	运行时间
粒子滤波	154.019s
Mean Shift 的粒子滤波	147.597s



Figure 10. The tracking results of mean-shift and particle filter

图 10. Mean shift 的粒子滤波跟踪效果

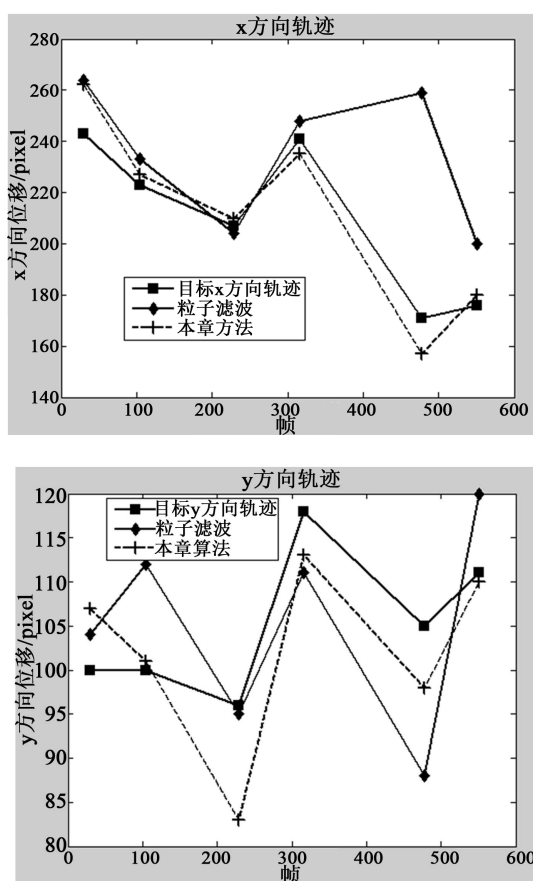


Figure 11. Comparison of trajectory between X direction and Y direction

图 11. X 方向和 Y 方向的轨迹比较

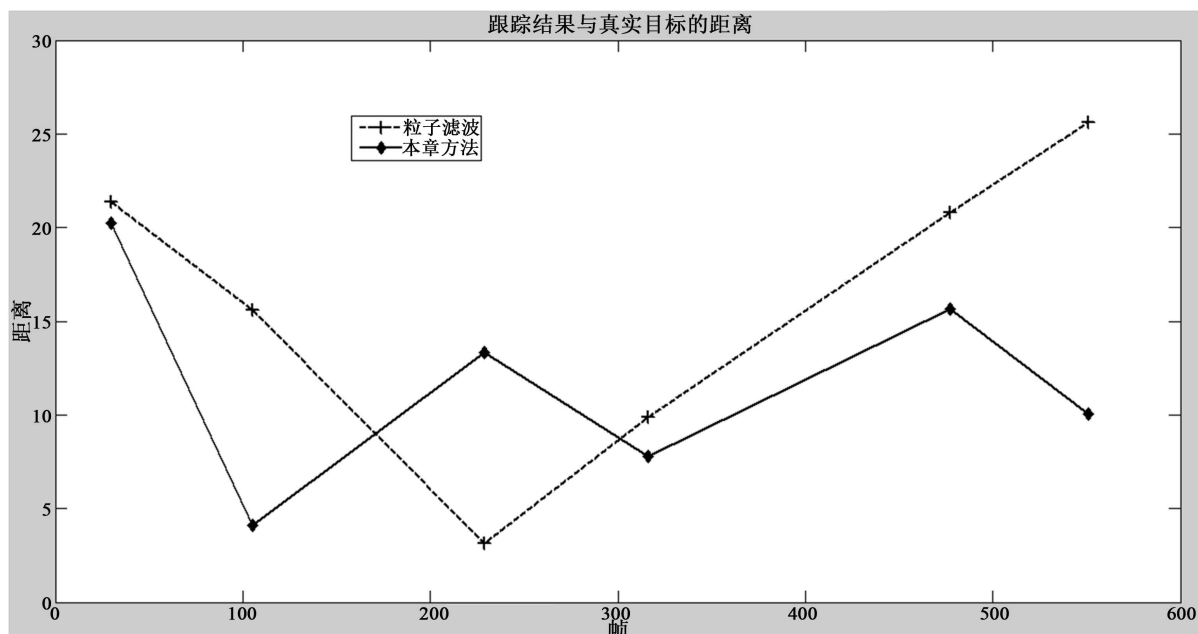


Figure 12. Error distance of two algorithms

图 12. 两种算法的误差距离

6. 小结

由于目标跟踪中有些粒子对目标位置的估计起到的作用微乎其微, 如果计算这些粒子的候选模型、候选模型与目标模型的相似度以及粒子的权重, 则浪费了很多时间。所以本文提出将 Mean 算法和粒子滤波算法相结合。对 Mean Shift 算法和粒子滤波算法的优点进行介绍, 说明将两者结合的优点。

由于 Mean Shift 算法会向目标的真实位置移动, 利用它的这一特点控制粒子的数量, 精简了计算, 提高了实时性。在估计出目标位置后对目标模型进行更新, 以适应目标的变化。

参考文献 (References)

- [1] Alam, I. (2015) Object Tracking in Video Sequences Using Information Fusion Principles. *IEEE Transactions on 5th Computer Science and Electronic Engineering Conference*, Colchester, UK, 17-18 September 2013, 146-151.
- [2] 胡世强, 敬忠良. 粒子滤波算法综述[J]. 控制与决策, 2005, 20(4): 361-365.
- [3] Tang, D. and Zhang, Y.J. (2011) Combining Mean-Shift and Particle Filter for Object Tracking. *International Conference on Image and Graphics*, Hefei, 12-15 August 2011, 771-776.
- [4] 张颖颖, 王红娟, 黄义定. 基于 Mean Shift 和粒子滤波的行人目标跟踪方法[J]. 计算机与现代化, 2012(3): 40-43.
- [5] 马丽, 常发亮, 乔谊正. 基于均值漂移算法和粒子滤波算法的目标跟踪[J]. 模式识别与人工智能, 2006, 19(6): 787-793.
- [6] 顾幸芳, 茅耀斌, 李秋洁. 基于 Mean Shift 的视觉目标跟踪算法综述[J]. 计算机科学, 2012, 39(12): 16-22.
- [7] Comaniciu, D. and Meer, P. (2002) Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 603-619. <https://doi.org/10.1109/34.1000236>
- [8] Comaniciu, D., Ramesh, V. and Meer, P. (2000) Real-Time Tracking of Non-Rigid Objects Using Mean Shift. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, 15 June 2000.
- [9] Maggio, E. and Cavallaro, A. (2005) Multi-Part Target Representation for Color Tracking. *Proceeding of IEEE International Conference on Image Proceeding*, Genova, 14 September 2005. <https://doi.org/10.1109/ICIP.2005.1529854>
- [10] Zhang, X. and Ming, Y. (2010) An Improved Mean Shift Tracking Algorithm Based on Color and Texture Feature. *Proceeding of the 2010 International Conference on Wavelet Analysis and Pattern Recognition*, Qingdao, 11-14 July 2010, 38-43. <https://doi.org/10.1109/ICWAPR.2010.5576453>

-
- [11] Du, K., Ju, Y. and Jin, Y. (2012) Mean Shift Tracking Algorithm with Adaptive Block Color Histogram. *IEEE Transactions on Special Fund for Basic Scientific Research in the Central University*, Yichang, 21-23 April 2012, 2692-2695. <https://doi.org/10.1109/CECNet.2012.6202074>
- [12] He, J.W. and Yang, Y.Y. (2014) Multi-Iterative Tracking Method Using Mean Shift Based on Kalman Filter. *IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Guilin, 5-8 August 2014, 22-27. <https://doi.org/10.1109/ICSPCC.2014.6986144>

期刊投稿者将享受如下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org