文章编号:1001 - 9081(2011)04 - 1047 - 03

doi:10.3724/SP.J.1087.2011.01047

一种时空域联合的机动视频目标精确跟踪方法

胡　波

(浙江东方职业技术学院 工程技术系,浙江 温州 325011)

(hubo9511@163.com)

摘　要:提出一种采用 Bhattacharyya 系数最大化并联合时空域信息的视频目标跟踪方法。时域通过卡尔曼滤波预测目标的运动信息,空域用 Camshift 算法精确匹配视频目标。由于运动目标机动性比较强,卡尔曼滤波预测的位置和真实位置存在较大的误差,容易导致下一步跟踪失败。采用基于 Bhattacharyya 系数的由粗到精的核匹配搜索方法,在卡尔曼滤波预测的位置基础上适当扩大搜索范围,通过 Bhattacharyya 系数最大化确定初始匹配窗口,再用 Camshift 算法精确匹配视频目标。实验证明该方法对机动快速运动目标具有很高的跟踪精度。

关键词:目标跟踪; Kalman 滤波; Camshift 算法; Bhattacharyya 系数; 匹配窗口

中图分类号: TP391.41　　**文献标志码**:A

Precise spatial-temporal tracking method for maneuvering video objects

HU Bo

(*Department of Engineering Technology, Zhejiang Dongfang Vocational and Technical College, Wenzhou Zhejiang* 325011, *China*)

Abstract: A video object tracking method combining the Bhattacharyya coefficients maximization with spatial-temporal information was proposed. The Kalman filter was used to predict the target movement information in the time domain, while in the space domain the target was precisely matched by using Camshift algorithm. Due to the strong maneuverability of the moving target, there will be a relatively big discrepancy between the predicted and true position, which will cause failure in tracking for the upcoming frame. To deal with this problem, a kernel matching approach based on Bhattacharyya coefficients was adopted in a way from rough to precise. The search scope was properly increased based on the position of the prediction, and the initial matching window was defined according to the Bhattacharyya coefficients maximization. Finally, the target was precisely matched by applying Camshift algorithm. The experimental results indicate that the proposed method is highly precise in tracking fast maneuvering moving target.

Key words: target tracking; Kalman filter; Camshift algorithm; Bhattacharyya coefficient; matching window

出于在公共安全监控、人机交互、智能交通、辅助驾驶以及军事等领域应用的需要,针对视频对象的跟踪技术已成为计算机视觉应用领域的一个研究热点。人们针对不同的应用环境提出了许多跟踪算法,如基于粒子滤波[1]、Kalman 滤波[2]、extended Kalman 滤波[3]、Unscented Kalman 滤波[4]、Mean Shift 算法[5-6]等。然而,单独使用这些算法普遍存在一些不足,如粒子滤波算法由于计算量大影响了跟踪的实时性;Kalman 滤波在处理非线性运动模型或非高斯噪声时误差比较大;extended Kalman 滤波和 Unscented Kalman 滤波均不能同时处理非线性非高斯情况;Mean Shift 算法缺乏必要的模板更新,当目标尺度变化时可能导致跟踪失败。综合考虑上述算法的优缺点,本文采用一种时空域联合的视频目标跟踪方法。在时域用 Kalman 滤波器预测目标的初始搜索窗口位置,考虑目标机动性比较强时 Kalman 预测的窗口位置与目标的真实窗口位置误差较大,因此在预测窗口基础上适当扩大搜索范围。通过比较 Bhattacharyya 系数[7]确定目标初始匹配窗口,同时在空域上用 Camshift 算法[8]对运动目标进行精确匹配。通过上述方法的改进,能够实现对视频序列中机动目标高精度的跟踪。

1　时域 Kalman 预测搜索窗口

Kalman 滤波是一种对动态系统的状态序列进行线性最小方差估计的算法,近年来已经有学者通过 Kalman 滤波建立目标的运动模型[9-10]来完成视频目标的预测跟踪。本文通过矩形框来描述跟踪目标的对象区域,通过矩形框的中心和长宽来表示目标的形心位置以及轮廓边界。Kalman 滤波器就是通过不断更新当前帧运动目标的真实位置来预测下一帧矩形框中心位置,从而实现运动目标的跟踪。Kalman 滤波通过状态方程和观测方程来描述一个动态系统:

$$\boldsymbol{X}_k = \boldsymbol{A}_k\boldsymbol{X}_{k-1} + \boldsymbol{W}_{k-1} \tag{1}$$

$$\boldsymbol{Z}_k = \boldsymbol{H}\boldsymbol{X}_k + \boldsymbol{V}_k \tag{2}$$

其中:$\boldsymbol{A}_k$、$\boldsymbol{H}$ 分别表示状态转移矩阵和测量矩阵;$\boldsymbol{W}_k$、$\boldsymbol{V}_k$ 表示过程噪声和测量噪声,假设它们是相互独立且服从正态分布的高斯白噪声。

本文将视频目标在较短时间间隔 T 内的运动分解为在水平和垂直方向的匀加速直线运动,建立的状态向量 $[s_x(k)\quad v_x(k)\quad a_x(k)\quad s_y(k)\quad v_y(k)\quad a_y(k)]$ 表示目标在水平和垂直方向的位置变化,则目标的状态矩阵和观测矩阵为:

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & \frac{1}{2}T^2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

收稿日期:2010 - 10 - 12;**修回日期**:2010 - 11 - 13。　**基金项目**:浙江省教育厅(理)科研计划基金资助项目(Y200804700)。

作者简介:胡波(1982 -),男,浙江舟山人,助教,硕士,主要研究方向:视频信号处理、机器视觉。

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{4}$$

Kalman 滤波器的初始值为 $[s_x(0) \quad 0 \quad 0 \quad s_y(0) \quad 0 \quad 0]$，其中 $s_x(0)$，$s_y(0)$ 表示初始窗口的位置。

2 空域 Camshift 算法精确匹配视频目标

通过 Kalman 预测已经得到目标在当前帧的初始搜索位置，为了获得目标的精确位置需要进一步通过跟踪算法来实现，本文采用 Camshift 算法[11-12]来完成这一功能。Camshift 算法是对 Mean Shift 算法的改进，Mean Shift 算法通过计算窗口的零阶矩和一阶矩获得质心坐标，而 Camshift 算法通过计算二阶矩自适应地调整窗口的大小和轴向角。零阶矩和一阶矩如下表示：

$$\begin{cases} M_{00} = \sum\sum I_c(x,y) \\ M_{10} = \sum\sum xI_c(x,y) \\ M_{01} = \sum\sum yI_c(x,y) \end{cases} \tag{5}$$

其中：$I_c(x,y)$ 表示图像在 (x,y) 点处的概率密度，这里用原图像颜色分量[13]反向投影图的概率分布来表示。则质心位置表示为：

$$x_c = \frac{M_{10}}{M_{00}}, \quad y_c = \frac{M_{01}}{M_{00}} \tag{6}$$

假设在当前帧中用椭圆来精确匹配目标区域，计算二阶矩：

$$\begin{cases} M_{20} = \sum\sum x^2 I_c(x,y) \\ M_{02} = \sum\sum y^2 I_c(x,y) \\ M_{11} = \sum\sum xyI_c(x,y) \end{cases} \tag{7}$$

椭圆的长短轴，以及轴和水平方向的夹角为：

$$length = \sqrt{\cos^2\theta M_{20} + 2\cos\theta\sin\theta M_{11} + \sin^2\theta M_{02}} \tag{8}$$

$$width = \sqrt{\sin^2\theta M_{20} - 2\cos\theta\sin\theta M_{11} + \cos^2\theta M_{02}} \tag{9}$$

$$\theta = \arctan\frac{2\left(\frac{M_{11}}{M_{00}}\right)}{\frac{M_{20}}{M_{00}} - \frac{M_{02}}{M_{00}} + \sqrt{4\left(\frac{M_{11}}{M_{00}}\right)^2 + \left(\frac{M_{20}}{M_{00}} - \frac{M_{02}}{M_{00}}\right)^2}} \tag{10}$$

Camshift 算法能够根据计算得到的长短轴自适应地调整匹配窗口的大小。设长轴在水平和垂直方向的投影长度分别为 t_1 和 t_4，短轴在水平和垂直方向的投影长度为 t_2 和 t_3。则 t_1、t_2、t_3、t_4 按下式计算：

$$\begin{cases} t_1 = length \times \cos\theta \\ t_2 = width \times \sin\theta \\ t_3 = width \times \cos\theta \\ t_4 = length \times \sin\theta \end{cases} \tag{11}$$

匹配窗的长和宽计算如下：

$$\begin{cases} windows.length = \max(t_1, t_2) \\ windows.width = \max(t_3, t_4) \end{cases} \tag{12}$$

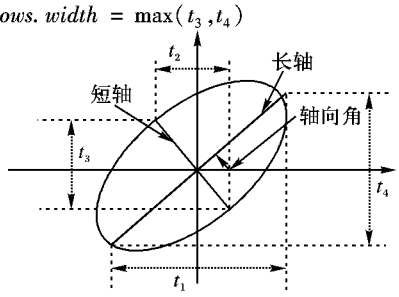
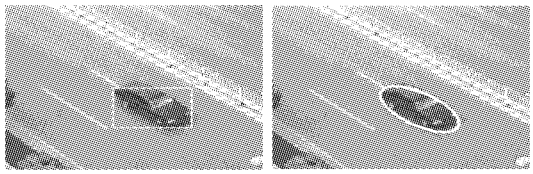


图 1 匹配窗口示意图

图 1 可以看出本算法能够根据长短轴在水平和垂直方向的投影自适应地调整匹配窗口的大小和轴向角，图 2(a)显示的矩形框只能大概确定车辆搜索窗口，而图 2(b)椭圆能够精确匹配车辆，提高了跟踪的鲁棒性。



(a) Mean shift跟踪窗口 (b) Camshift跟踪窗口

图 2 Mean shift 跟踪窗口和 Camshift 窗口对比

3 Bhattacharyya 系数确定初始匹配窗口

3.1 特征匹配方法

由于 Camshift 算法在跟踪目标时，必须保证在下一帧的运动目标不能完全脱离用 Kalman 预测的位置，也就是说预测位置和真实位置之间的目标在图像空间中必须有一部分是重叠的，否则传统的 Camshift 算法就会跟踪不上目标。事实上当运动目标速度较快时，两帧之间的运动目标在图像空间上完全有可能不重叠，而传统的 Camshift 算法只能在目标的预测窗口区域内迭代求解取得局部最优值，而得不到实际的跟踪位置，这就导致运动目标跟踪失效。为了解决这个问题，本文根据初始帧目标特征与当前帧目标特征之间的相似性，提出一种基于 Bhattacharyya 系数的由粗到精核匹配搜索方法。用 Bhattacharyya 系数来表示两者特征模型的匹配程度，按如下定义：

$$\rho(y) \equiv \rho[p(y), q] = \sum_{u=1}^{m}\sqrt{\hat{p}_u(y)\hat{q}_u} \tag{13}$$

其中 $\hat{q}_u$ 和 $\hat{p}_u(y)$ 分别表示初始帧和当前帧目标颜色概率分布，定义如下：

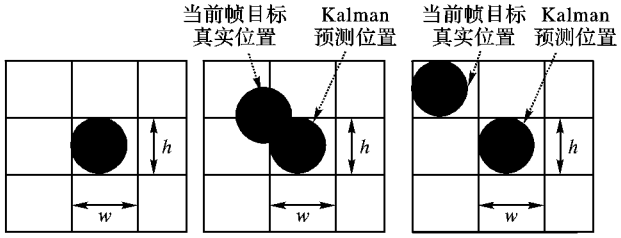
$$\hat{q}_u = C\sum_{i=1}^{n} k\left(\left\|\frac{t_i - d_0}{h}\right\|^2\right)\delta[b(t_i) - u] \tag{14}$$

其中：d_0 为搜索窗口的中心坐标；$t_i(t_{ix}, t_{iy})$ 表示该窗口中第 i 个像素的坐标；n 表示像素总个数；$k(x^2)$ 表示核函数；h 表示核函数的带宽；函数 b 和 δ 的作用是判断第 i 个像素的灰度或颜色值是否属于特征值 u；C 是一个归一化系数，使得 $\sum_{u=1}^{m}\hat{q}_u = 1$，其中 m 表示特征个数。同理，$\hat{p}_u(y)$ 定义与 $\hat{q}_u$ 类似。显然当 $\hat{\rho}(y)$ 越大时匹配程度就越高，即当前帧目标与初始帧目标最相似。

3.2 初始匹配窗口确定

具体的匹配过程如图 3 所示，对于图 3(a)中的运动目标，在原始跟踪窗口宽度和高度分别为 w 和 h 的基础上，适当扩大初始搜索区域进行粗略搜索。假设以 Kalman 预测位置所在窗口为中心，新的搜索窗口的宽度和高度分别扩大为原始跟踪窗口的 m 和 n 倍，即分为 $m \times n$ 个网格，在这个新的搜索窗口内对目标进行粗略定位，这里 m 和 n 的值都取 3。图 3(b)是目标运动速度不快的情况，逐一比较每个网格和初始帧目标的 Bhattacharyya 系数，系数最大的认为该网格和初始帧目标所在跟踪窗口的颜色概率分布相似性最大，把这个网格作为比较准确的初始匹配位置。这里因为目标运动速度不快，前后帧存在较大区域重合，因此初始匹配窗口仍然是中心位置，确定初始匹配位置后再利用 Camshift 算法确定其精确位置。同样，对于图 3(c)考虑的是运动目标速度较快的情况，利用 Bhattacharyya 系数对当前帧运动目标进行粗定位时会得到目标的初始匹配位置是左上角的网格，以这个位置为出发点，再利用 Camshift 算法求解其精确位置。如果目标在

两帧之间的运动范围更大,则可以适当增加网格数来扩大搜索范围,例如 5×5 网格、7×7 网格等,这样就解决了大范围机动运动目标的跟踪问题。



(a) 原始跟踪窗口　(b) 运动速度不快情况　(c) 运动速度过快情况

图 3　核匹配搜索

4　实验结果及讨论

为了实现精确跟踪的目的,本实验用一段可以精确测量其轨迹的小球视频序列作为研究对象。视频每帧图像大小 800×600,序列长度 75 帧,所有的算法均在 VC++ 6.0 环境中实现。本文主要研究机动快速运动目标的跟踪效果,将球体做一定高度的平抛运动,球体下落后与桌面发生碰撞并反弹,一共经历了 6 次反弹运动,目标在每次反弹时刻的速度变化非常剧烈,可以作为机动运动特性,运动轨迹如图 4 所示。

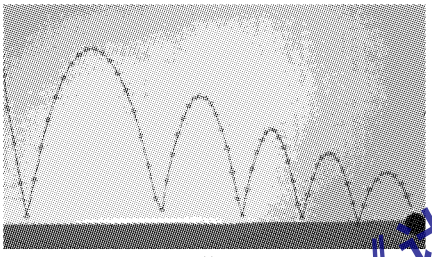
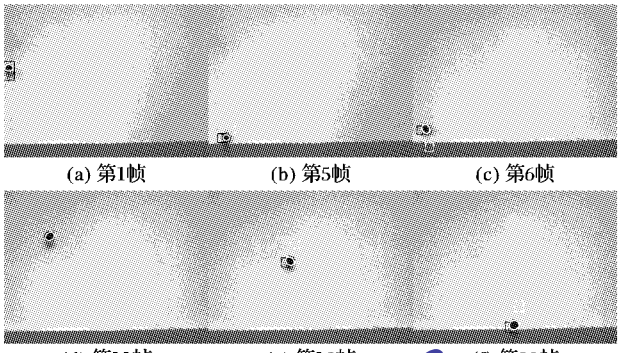


图 4　球体运动轨迹

由于本实验中目标在垂直方向上速度变化比较快,而水平方向不是很大,经过多次实验最后选择 5×3 的核匹配窗口。目标的跟踪效果如图 5 所示,其中白色矩形框表示用卡尔曼滤波器预测位置所在的搜索窗口,黑色矩形框表示经过 Bhattacharyya 系数计算后得到的当前帧初始匹配窗口,白色的椭圆表示当前帧 Camshift 精确匹配窗口。球体在运动过程中速度变化最剧烈的是与桌面发生碰撞时刻,此时速度方向几乎以 180°发生改变,卡尔曼滤波器预测误差最大。观察图 5 中 5 帧和 6 帧反映的就是球体发生碰撞前后时刻跟踪情况,此时卡尔曼滤波器预测位置所在的窗口和真实目标之间不存在任何重叠,它们之间的位置相差接近 2 个窗口高度。如果 Camshift 直接用这个预测位置作为初始匹配窗口的位置则无法计算得到最优匹配窗口,从而导致跟踪失败。通过计算 Bhattacharyya 系数使得初始匹配窗口和目标已经部分重叠,在此基础上再用 Camshift 进行迭代计算将会得到最优窗口。

如果直接将卡尔曼预测位置所在的窗口作为 Camshift 跟踪算法的初始匹配窗口,其跟踪效果如图 6 所示。在球体运动到最高点(第 13 帧)时确定跟踪窗口,发现 18 帧时预测位置所在窗口与球体只有很少一部分是重叠的,因此 Camshift 不能很好地跟踪。在 24 帧时预测位置所在窗口与目标完全不重合,Camshift 跟踪完全失败。

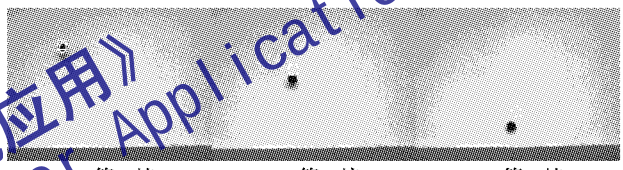
图 7 显示了 Kalman 预测、Bhattacharyya 系数粗定位和最终跟踪结果三者和真实位置的误差比较。从图 7(a)可以发现 Kalman 预测的位置和真实位置误差不大,这是由于球体在水平方向速度变化不大,因此通过 Bhattacharyya 系数粗定位并没有显示明显的优势。从图 7(b)可以发现 Kalman 预测误差较大而且误差曲线有 6 个时刻发生突变,误差变化范围达到 -90 ~ 80(单位:像素),这是由于球体在垂直方向分解为自由落体运动因此每个时刻速度变化很大,而且球体和桌面发生 6 次碰撞时运动速度达到最大并且方向发生了剧变,Bhattacharyya 系数粗定位优势就非常明显。观察 Bhattacharyya 系数粗定位误差曲线其误差范围在 -30 ~ 26(单位:像素),比 Kalman 预测误差明显减少。本文方法最终的跟踪误差范围在 -8.5 ~ 9(单位:像素),只有 Kalman 预测误差的 10%。



(a) 第1帧　(b) 第5帧　(c) 第6帧

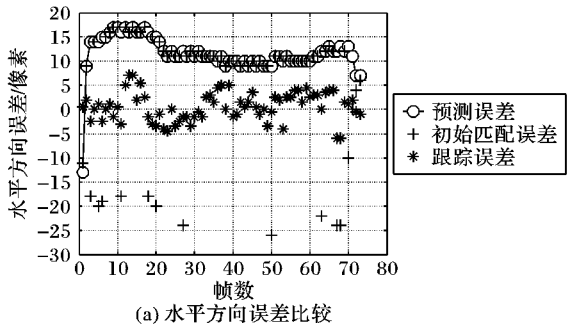
(d) 第11帧　(e) 第18帧　(f) 第23帧

图 5　基于 Bhattacharyya 系数核匹配跟踪效果图

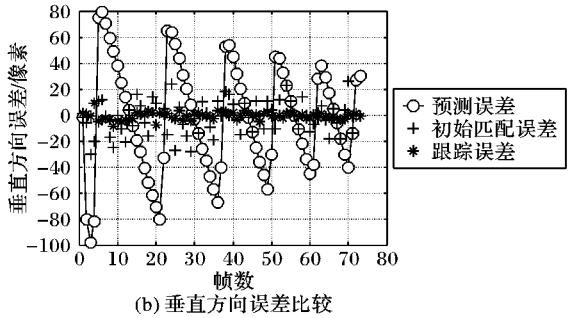


(a) 第13帧　(b) 第18帧　(c) 第24帧

图 6　没有 Bhattacharyya 系数核匹配跟踪效果



(a) 水平方向误差比较



(b) 垂直方向误差比较

图 7　跟踪误差比较

讨论机动运动目标的跟踪精度,按如下定义:

$$\sigma = \frac{\sum_{k=1}^{N}\left[(r_x(k) - T_x(k))^2 + (r_y(k) - T_y(k))^2\right]^{1/2}}{N} \tag{15}$$

其中:$T(k)$、$r(k)$ 分别表示第 k 帧目标真实位置和各种跟踪方法得到的位置,N 表示视频序列总长。表 1 显示了用

(下转第 1061 页)

5　结语

本文通过对 Bresenham 算法的分析,得出了逆向生成直线的类 Bresenham 算法,并进一步提出了将求斜率在(0.5,1]上的直线的位移码转化为求斜率在[0,0.5]的直线的位移码的算法。该算法只有加法和乘法运算,适合硬件实现,生成的直线与 Bresenham 算法一致,且能够大大减少生成直线的计算量,提高直线绘制速度。

在第 1 类直线中的 $y_0 = 1$ 和第 2 类直线中的 $x_0 - y_0 = 1$ 时,基于像素链的算法会退化为单步的 Bresenham 算法。在未来的工作中,我们将继续完善直线绘制算法。

参考文献:

[1] BRESENHAM J E. Algorithm for computer control of a digital plotter[J]. IBM Systems Journal, 1965, 4(1): 25 – 30.

[2] GARDNER P L. Modifications of Bresenham's algorithm for display [J]. IBM Technical Disclosure Bulletin, 1975, 18(5): 1595 – 1596.

[3] WU X, ROKNE J G. Double-step incremental generation of lines and circles[J]. Computer Vision, Graphics, and Image Processing, 1987, 37(3): 331 – 344.

[4] GRAHAM P, IYENGAR S S. Double and triple-step incremental linear interpolation[J]. IEEE Computer Graphics and Applications, 1994, 14(3): 49 – 53.

[5] BAO P G, ROKNE J G. Quadruple-step line generation[J]. Computers & Graphics, 1989, 13(4): 461 – 469.

[6] GILL G W. N-step incremental straight-line algorithms[J]. IEEE Computer Graphics and Applications, 1994, 14(3): 66 – 72.

[7] BRESENHAM J E, GRICE D G, PI S C. Run length slice algorithms for incremental lines[J]. IBM Technical Disclosure Bulletin, 1980, 22(8B): 3744 – 3747.

[8] BOYER V, BOURDIN J J. Auto-adaptive step straight-line algorithm[J]. IEEE Computer Graphics and Applications, 2000, 20(5): 67 – 69.

[9] 郑宏珍, 赵恢. 改进的 Bresenham 直线生成算法[J]. 中国图象图形学报, 1999, 4(7): 606 – 609.

[10] 刘晶. 改进的 Bresenham 直线生成算法[J]. 计算机应用与软件, 2008, 25(10): 247 – 249.

[11] 贾银亮. Bresenham 直线生成算法的改进[J]. 中国图象图形学报, 2008, 1(13): 158 – 161.

[12] 刘晶. 快速直线生成算法[J]. 金陵科技学院学报, 2007, 23(3): 9 – 12.

[13] 苗兰芳. 自适应多步位移码直线绘制算法[J]. 软件学报, 2002, 13(4): 637 – 642.

[14] 蔺想红, 张田文. 自适应多基元直线绘制算法[J]. 计算机辅助设计与图形学学报, 2006, 18(8): 1136 – 1141.

[15] 叶晓彤, 邓云. 基于对角行程的直线生成算法研究[J]. 计算机应用, 2008, 28(9): 2270 – 2273.

(上接第 1049 页)

Camshift 算法、Camshift 算法结合卡尔曼预测(CK)和本文方法(BhaCK)对球体运动视频序列的跟踪精度对比结果,可以看到本文方法具有明显优势。

表 1　跟踪精度比较

算法	跟踪精度
Camshit	21.8871
CK	17.2559
BhaCK	3.7629

5　结语

本文提出一种从时域卡尔曼预测到空域 Camshift 精确匹配的视频目标跟踪方法。当运动目标机动性比较强时卡尔曼滤波器建立的运动模型与目标真实的运动模型不一致,因此导致单步预测误差值较大,容易使下一步 Camshift 跟踪失败。采用一种在卡尔曼预测窗口基础上适当扩大搜索范围然后再用 Bhattacharyya 系数核匹配方法确定初始匹配窗口,在这个窗口中再用 Camshift 算法精确跟踪视频目标的方法。通过实验表明本文方法能够有效地定位运动目标在每一帧的初始位置,解决了目标前后帧空间上不存在重叠的跟踪难点,取得了令人满意的跟踪精度。

参考文献:

[1] ZHOU S K, CHELLAPPA R, MOGHADDAM B. Visual tracking and recognition using appearance-adaptive models in particle filters [J]. IEEE Transactions on Image Processing, 2004, 13(11): 1491 – 1506.

[2] 虞旦, 韦巍, 张远辉. 一种基于卡尔曼预测的动态目标跟踪算法研究[J]. 光电工程, 2009, 36(1): 52 – 56.

[3] ZHE DONG, ZHENG YOU. A novel extended Kalman filter for a class of nonlinear systems [J]. Progress in Natural Science, 2006, 16(9): 912 – 918.

[4] JULIER S J, UHLMANN J K. Unscented filtering and nonlinear estimation [J]. Proceedings of the IEEE, 2004, 92(3): 401 – 422.

[5] COMANICIU D, MEER P. Mean shift: A robust application toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(5): 603 – 619.

[6] 朱胜利, 朱善安, 李旭超. 快速运动目标的 Mean shift 跟踪算法 [J]. 中国图象图形学报, 2005, 33(5): 66 – 70.

[7] SOHAIL K M, UMAR I M, SAQUID S M, *et al.* Bhattacharyya coefficient in correlation of gray-scale objects[J]. Journal of Multimedia, 2006, 1(1): 56 – 61.

[8] 卢璇, 雷航, 赫宗波. 联合多特征的自动 CamShift 跟踪算法 [J]. 计算机应用, 2010, 30(3): 650 – 652.

[9] 胡波, 陈恳, 徐建瑜, 等. 基于 Kalman 预测和 Mean-shift 算法的视频目标跟踪[J]. 光电子·激光, 2009, 20(11): 1517 – 1522.

[10] WENG S-K, KUO C-M, TU S-K. Video object tracking using adaptive Kalman filter[J]. Journal of Visual Communication and Image Representation, 2006, 17(6): 1196 – 1197.

[11] WANG ZHAO-WEN, YANG XIAO-KANG, XU YI, *et al.* Camshift guided particle filter for visual tracking[J]. Pattern Recognition Letters, 2009, 30(4): 407 – 413.

[12] 赵祎, 穆志纯, 刘克. 基于肤色及轮廓信息的人耳实时跟踪 [J]. 中国图象图形学报, 2006, 11(7): 949 – 953.

[13] NUMMIARO K, KOLLER-MEIER E, GOOL V L. A color-based particle filter [C]// Proceedings of 1st International Workshop on Generative-Model-Based Vision. Copenhagen: European Conference on Computer Vision, 2002: 53 – 60.