文章编号 1004-924X(2009)10-2606-06

跟踪窗口自适应的 Mean Shift 跟踪

颜 佳,吴敏渊,陈淑珍,张青林
(武汉大学 电子信息学院,湖北 武汉 430079)

摘要:传统的 Mean Shift 跟踪算法在目标发生形变时会因跟踪窗不能动态改变尺寸而导致目标跟偏甚至跟丢,因此本文提出了一种新的跟踪窗口大小和方向自适应的改进算法。首先,采用跟踪窗口内协方差矩阵主分量分析法来计算跟踪目标的方向和尺寸大小;然后,联合相似性度量和卡尔曼滤波器来更新跟踪窗口的大小和方向倾角,使之适应目标的变化。实验显示,本算法可对不断旋转和缩放的运动目标进行准确实时跟踪,当目标尺寸在 35 pixel×17 pixel 到 176 pixel×80 pixel 之间变化时,平均处理时间为 17.45 ms/frame,表明改进的算法能够满足非刚体目标跟踪系统的要求。

关 键 词:Mean Shift;目标跟踪;主分量分析;形变目标;卡尔曼滤波器
中图分类号:TP391 **文献标识码**:A

Mean Shift tracking with adaptive tracking window

YAN Jia, WU Min-yuan, CHEN Shu-zhen, ZHANG Qing-lin

(*School of Electronic Information, Wuhan University, Wuhan* 430079, *China*)

Abstract: A new algorithm to estimate the scale and orientation of a tracking window is presented. The algorithm overcomes the problem that the traditional Mean Shift based tracking algorithm often fails when a deformable target is tracked because of the rigid tracking window. Firstly, the principal components of the variance matrix are adopted to compute the scale and orientation of tracking target, then the similarity measure and Kalman filter are used to update the tracking window. Experimental results show that the algorithm can be implemented in real-time and can adapt to changes of scale and orientation of the moving object. The average computing time is only 17.45 ms/frame with the object's scales varying between 35 pixel×17 pixel and 176 pixel×80 pixel. This algorithm can satisfy the requirements for tracking non-rigid objects.

Key words: Mean Shift; target tracking; principal component analysis; deformable object; Kalman filter

收稿日期:2008-11-22;**修订日期**:2008-12-23.
基金项目:国家 863 高技术研究发展计划资助项目(No. 2006AA040307)

1　引　言

运动目标实时跟踪在计算机视觉领域有着非常重要的应用。Mean Shift 算法是一种基于密度梯度的无参数估计方法[1],因其迭代次数少计算量小已经广泛应用在目标实时跟踪中[2-4]。但使用该方法对存在明显尺度变化和旋转运动的目标进行跟踪时,由于跟踪窗口尺寸固定而往往导致目标的跟偏甚至跟丢。

针对此问题,一种最直观的解决方法就是通过尝试不同的跟踪窗口分别计算对应的 Bhattacharyya 系数并选取最大 Bhattacharyya 系数所对应的窗口大小作为最佳跟踪窗大小,文献[4]就采取的类似作法。但文献[5]中已经指出该方法在对逐渐增大的目标进行跟踪时会失效。文献[5]中通过引入一个额外尺度核在尺度空间进行迭代,从而选取最佳的核函数窗宽,这样做不仅计算量大而且使用 Epanechnikov 核等价于在尺度空间中求平均,因此存在和文献[4]中一样的问题。文献[6]采用不对称核函数来扩展 Mean Shift 算法获得目标大小和方向,但该方法仅适用于轮廓特征不变的目标。文献[7]中通过前后两帧中目标的角点匹配来获得仿射变换参数从而更改核函数带宽,但该方法不适用于非刚性目标的跟踪。文献[8]中通过在下一帧中迭代更新上一帧中目标椭圆描述的五维参数从而更改目标尺度和角度,但该方法本质上和文献[5]是一样的,存在计算量大的不足。文献[9]提出了一种类似 EM(expectation maximization)的算法通过带宽矩阵来计算目标的中心位置和目标形状,但其中如何得到目标的大小和方向的计算公式并没给出。

本文将主分量分析方法引入到 Mean Shift 框架中来,首先计算跟踪窗内基于像素权重的协方差矩阵,然后利用主分量分析方法来得到跟踪目标的方向和尺寸大小,最后采用卡尔曼滤波器结合相似性度量来更新跟踪窗口的大小和方向倾角,使之适应目标的变化。实验表明,跟踪窗口自适应的 Mean Shift 算法能很好的定位跟踪目标并且能实时运行。

2　Mean Shift 跟踪算法的缺陷

Mean Shift 算法是一种半自动化跟踪算法。首先通过手工选定目标,计算核函数加权下的跟踪窗口中的直方图分布。假设$\{y_i\}_{i=1,\cdots,n}$为目标区域中各个像素的位置,目标中心为 x_0,则目标的直方图分布为 $\hat{p}=\{\hat{p}_u\}_{u=1,\cdots,m}$,其中

$$\hat{p}_u(x_0)=C\sum_{i=1}^{n}G\left(\left\|\frac{y_i-x_0}{h}\right\|^2\right)\delta[b(y_i)-u], \tag{1}$$

G 为核函数,m 为特征空间中特征值的个数(本文 G 取高斯核函数,特征空间采用 HSV 色彩空间中的 H、S 信息,m 为 $16\times16=256$ 级),δ 为 Kronecker 函数,$b(y_i)$为像素 y_i 对应的特征值,C 为归一化系数,h 为核函数的带宽,一般设为跟踪窗口的一半。

其后的跟踪过程就是在序列图像中通过迭代寻找与目标最为相似的候选区域,相似性度量采用 Bhattacharyya 系数。目标模型的概率分布 $\hat{p}(x_0)$ 与候选区域概率分布 $\hat{q}(x)$ 的 Bhattacharyya 系数为:

$$\rho(x)=\sum_{u=1}^{m}\sqrt{\hat{p}_u(x_0)\,\hat{q}_u(x)}, \tag{2}$$

对应的迭代公式为:

$$x_1=\frac{\sum_{i=1}^{n}y_i w(y_i)G\left(\left\|\frac{y_i-x_0}{h}\right\|^2\right)}{\sum_{i=1}^{n}w(y_i)G\left(\left\|\frac{y_i-x_0}{h}\right\|^2\right)}, \tag{3}$$

x_1 为新的目标中心位置,其中

$$w(y_i)=\sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u(x)}{\hat{p}_u(x_0)}}\delta[b(y_i)-u], \tag{4}$$

其中,$w(y_i)$为权重函数。迭代过程就是不断计算(3)式,直至 Bhattacharyya 系数最大即定位为目标最终中心位置,停止迭代。

图 1 给出了传统 Mean Shift 算法(tracker I)对一段视频序列 Plane 进行跟踪的部分结果(初始跟踪窗口为手工选定其大小为 94 pixel×46 pixel)。由于传统 Mean Shift 算法中跟踪窗口的大小不变,如图 2 中的实线矩形框所示,所以核函数带宽 h 恒定。当目标尺寸逐渐变大时,固定带宽的核函数不能很好的描述目标模型会导致目标的跟偏甚至跟丢;当目标逐渐变小时,由于混入了

大量的背景信息会导致目标特征减弱影响跟踪走向。

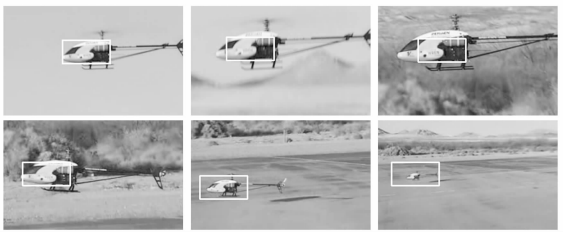

图 1　tracker Ⅰ对 Plane 视频序列的跟踪结果(第 1、56、141、219、271 和 322 帧)

Fig. 1　Tracking results of Plane sequence by using tracker Ⅰ (Frame 1, 56, 141, 219, 271, 322)

最理想的跟踪窗口应当不仅大小能自适应目标的变化,而且倾角也能适应目标的旋转运动,这样就能最逼近目标的真实形状,最少限度的包含背景信息。图 2 中虚线矩形框即为最理想的跟踪窗口。

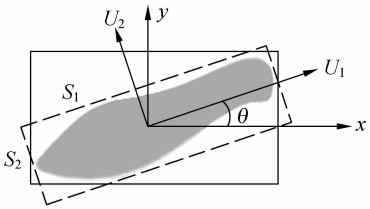

图 2　最佳跟踪窗口示意图

Fig. 2　Scheme of the best tracking window

3　主分量分析确定目标尺度和方向

主分量分析方法[10]已经广泛用于模式识别中的数据降维、压缩等处理。当计算主分量分析时,首先计算整个数据集的协方差矩阵,然后计算协方差矩阵的特征值。特征值对应的最大特征值将产生主分量。我们将该思想借鉴到 Mean Shift 框架中来,用于得到目标的尺度和方向表示。跟踪窗口内 n 个像素点之间的协方差矩阵为:

$$\boldsymbol{V}=\frac{\sum_{i=1}^{n}W(\boldsymbol{y}_i)(\boldsymbol{y}_i-\boldsymbol{x}_0)(\boldsymbol{y}_i-\boldsymbol{x}_0)^{\mathrm{T}}}{\sum_{i=1}^{n}W(\boldsymbol{y}_i)}\ , \tag{5}$$

其中,$W(\boldsymbol{y}_i)=w(\boldsymbol{y}_i)G\left(\left\|\frac{\boldsymbol{y}_i-\boldsymbol{x}_0}{h}\right\|^2\right)$为每个像素点的总权重。由于 $\boldsymbol{y}_i$ 是一个二维向量,所以 $\boldsymbol{V}$ 为一 2×2 矩阵。利用主分量分析求解特征方程

$$\boldsymbol{u}^{\mathrm{T}}\begin{bmatrix}V_{11} & V_{12}\\ V_{21} & V_{22}\end{bmatrix}\boldsymbol{u}=\lambda\ , \tag{6}$$

其中 $\lambda_1\geqslant\lambda_2$ 为特征值,对应的归一化特征向量 u_1 和 u_2 代表了像素点所组成的数据集分布的长主轴和短主轴方向。因此目标的方向 θ 由 u_1 决定,尺寸 s 由 λ_1 和 λ_2 决定,如图 2 所示。

$$\lambda_1,\lambda_2=\frac{V_{11}+V_{22}\pm\sqrt{(V_{11}-V_{22})^2+4V_{12}^2}}{2}\ , \tag{7}$$

$$\theta=\arcsin(u_{1x})=\arcsin\left(\frac{V_{12}}{\sqrt{(\lambda_1-V_{11})^2+V_{12}^2}}\right)\ , \tag{8}$$

$$s_i=k\sqrt{\lambda_i}\quad(i=1,2)\ , \tag{9}$$

式中,k 为系统级常数,实验中取 $k=4$。特别地,当目标水平时,$V_{12}=0$,$\theta=0$,$\lambda_1=V_{11}$,$\lambda_2=V_{22}$。因此当应用中已知跟踪目标始终平行于水平轴时可以强制 $V_{12}=V_{21}=0$。

4　卡尔曼滤波结合相似性度量更新跟踪窗口

由于跟踪窗口内不可避免地会包含背景信息,所以为了使跟踪窗口的变化更平滑。本节引入卡尔曼滤波来更新跟踪窗口。首先将 s_1、s_2 和 θ 作为卡尔曼滤波器的测量值输入,然后利用卡尔曼滤波器估计真实的 $\hat{s}_1$、$\hat{s}_2$ 和 $\hat{\theta}$ 作为下一帧的跟踪窗口参数。最优滤波估计方程为:

$$\begin{bmatrix}\hat{s}_1(k)\\ \hat{s}_2(k)\\ \hat{\theta}(k)\end{bmatrix}=\begin{bmatrix}\hat{s}_1(k-1)\\ \hat{s}_2(k-1)\\ \hat{\theta}(k-1)\end{bmatrix}+K(k)\begin{bmatrix}s_1-\hat{s}_1(k-1)\\ s_2-\hat{s}_2(k-1)\\ \theta-\hat{\theta}(k-1)\end{bmatrix}. \tag{10}$$

其中:$\hat{s}_1(k-1)$、$\hat{s}_2(k-1)$ 和 $\hat{\theta}(k-1)$ 为 $k-1$ frame 时的跟踪窗口参数,滤波增益 $K(k)$ 为:

$$K(k)=P(k-1)\times(P(k-1)+R(k))^{-1}, \tag{11}$$

$$P(k)=(1-K(k))\times P(k-1)+W(k)\ . \tag{12}$$

取 $P(0)=100$,$W=1$,$\hat{s}_1(0)$、$\hat{s}_2(0)$和 $\hat{\theta}(0)$分别为初始跟踪窗的宽度、高度和倾角。

一般假设 $R(k)$为高斯白噪声,本文将相似性度量采用的 Bhattacharyya 系数 $\rho(x)$ 和 $R(k)$联系起来,取 $R(k)$为:

$$R(k)=\sqrt{1-\rho(x)}\ . \tag{13}$$

从式(10),(11),(13)可以看出,Bhatta-

charyya 系数 $\rho(x)$越大,$R(k)$越小时,测量值 s_1、s_2 和 θ 的权值越大,真实值就更接近于测量值。这样做就能使新的跟踪窗口在原有窗口与 s_1、s_2 和 θ 决定的窗口中达到一个最佳平衡,既避免了不更新又避免了过度更新。

计算得到的 $\hat{s}_1(k)$、$\hat{s}_2(k)$和 $\hat{\theta}(k)$即作为 $k+1$ frame 时 Mean Shift 算法所需的跟踪窗口的宽度、高度和倾角。

5　实验结果及分析

采用本文算法(tracker II)对 Plane 视频序列进行跟踪的部分结果如图 3 所示(其中 $\hat{s}_1(0)=94$,$\hat{s}_2(0)=46$,$\hat{\theta}(0)=0$)。可以看到,虽然视频中飞机先逐渐变大然后逐渐变小并伴有方位的微动,但此时跟踪窗口已能自适应的改变自身的大小(第 191 帧时达到最大值:176 pixel×80 pixel;第 322 帧时最小:35 pixel×17 pixel)和方位。图 4 给出了 tracker II 对另一视频序列 Car 的跟踪结果(其中 $\hat{s}_1(0)=53$,$\hat{s}_2(0)=25$,$\hat{\theta}(0)=0$),该视频中目标旋转幅度较大,但 tracker II 依旧表现出了良好的性能。

图 3　tracker II 对 Plane 视频序列的跟踪结果(第 1、56、141、219、271 和 322 帧)

Fig. 3　Tracking results of Plane sequence by using tracker II (Frame 1, 56, 141, 219, 271, 322)

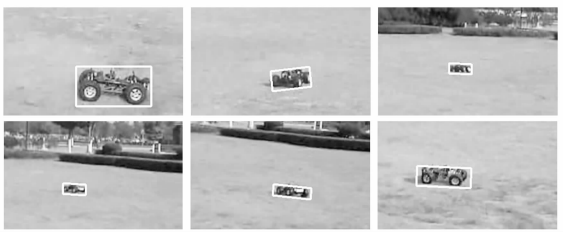

图 4　tracker II 对 Car 视频序列的跟踪结果(第 1、40、79、112、188 和 219 帧)

Fig. 4　Tracking results of Car sequence by using tracker II (Frame 1, 40, 79, 112, 188, 219)

由于加入了估计目标尺度和方向的计算量,所以 tracker II 比 tracker I 处理时间略多,图 5 给出了两种算法对 Plane 序列的处理时间曲线图(算法均在 Intel Pentium D 的 CPU,1 G 内存配置的电脑上用 VC 6.0 编程实现)。从图 5 可以看出,随着目标尺寸的变大计算量也随之增加,这是由于参与计算的像素点增多的原因,特别地,当镜头晃动时由于目标移位较大所以计算量陡增,如第 187 frame、第 235 frame 处;当目标尺寸变小时,tracker II 计算量比 tracker I 还要小,这是因为 tracker I 此时的跟踪窗口包含了太多的背景像素影响了 Mean Shift 的迭代。计算可知,前 322 frame tracker I 平均处理时间为 11.19 ms/frame,tracker II 为 17.45 ms/frame,所以 tracker II 在适当的计算量下不仅提高了跟踪性能而且同样能够满足跟踪中的实时性要求。图 6 给出了 tracker II 对 Car 序列的处理时间曲线图,平均处理时间为 8.20 ms/frame。

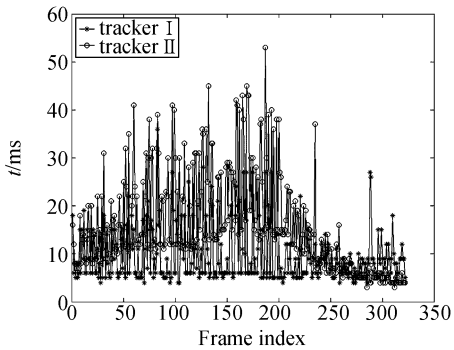

图 5　tracker I 和 tracker II 对 Plane 序列的处理时间

Fig. 5　Computing time of tracker I and tracker II for Plane sequences

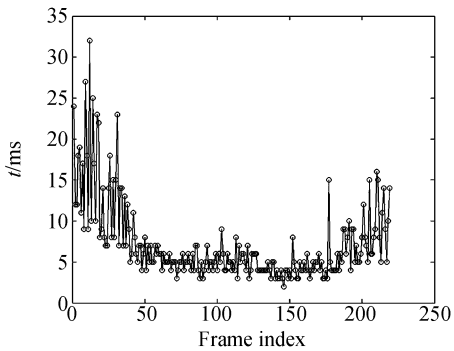

图 6　tracker II 对 Car 序列的处理时间

Fig. 6　Computing time of tracker II for Car sequences

当然，文献[3]指出由于 Mean Shift 算法仅在核函数带宽内寻找最佳位置所以当目标运动速度过快时往往会丢失目标，此时就需要辅助其他跟踪手段，譬如运动估计。而只要在准确定位目标位置的前提下，本文改进算法对目标形变速度没有明显限制。

6 结　论

本文针对原始 Mean Shift 算法不能很好的对缩放和旋转目标进行跟踪的缺陷，提出了跟踪窗口自适应的改进办法：首先计算跟踪窗内基于像素权重的协方差矩阵，然后利用主分量分析方法来得到目标的方向和尺寸大小，最后采用卡尔曼滤波器结合相似性度量来最终更新跟踪窗口作为下一帧的跟踪窗口。通过对视频序列的测试，证明本文方法在对存在明显尺度变化和旋转运动的目标进行跟踪时跟踪窗口能够很好的适应目标的变化，跟踪准确并且平均处理时间为 17.45 ms/frame（目标尺寸 35 pixel×17 pixel—176 pixel×80 pixel），能够保证跟踪的实时性。然而，本文暂未考虑目标的遮挡问题，当目标变形并且伴有遮挡和干扰时如何准确定位目标是进一步需要研究的问题。

参考文献：

[1] FUKANAGA K, HOSTETLER L D. The estimation of the gradient of a density function, with applications in pattern recognition [J]. *IEEE Trans. On Information Theory*, 1975,21(1):32-40.

[2] 孙中森，孙俊喜，宋建中，等. 一种抗遮挡的运动目标跟踪算法 [J]. 光学 精密工程, 2007,15(2):267-171.
SUN ZH S, SUN J X, SONG J ZH, *et al*.. Anti-occlusion arithmetic for moving object tracking [J]. *Opt. Precision Eng.*,2007,15(2):267-271. (in Chinese)

[3] YILMAZ A, SHAFIQUE K, SHAH M. Target tracking in airborne forward looking infrared imagery [J]. *Int'l Journal of Image and Vision Computing*, 2003,21(7):623-635.

[4] COMANICIU D, RAMESH V, MEER P. Kernel-based object tracking [J]. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003,25(5):564-575.

[5] COLLINS R T. Mean shift blob tracking through scale space [C]. 2003 *IEEE Computer Society Conference or Computer Vision and Pattern Recognition (CVPR03)*,2003,2:234-240.

[6] YILMAZ A. Object tracking by asymmetric kernel Mean Shift with automatic scale and orientation selection [C]. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, 2007:1-6.

[7] 彭宁嵩，杨杰，刘志，等. Mean Shift 跟踪算法中核函数带宽的自动选取 [J]. 软件学报, 2005,16(9):1542-1550.
PENG N S, YANG J, LIU ZH, *et al*.. Automatic selection of Kernel-bandwidth for Mean Shift object tracking [J]. *Journal of Software*, 2005,16(9):1542-1550. (in Chinese)

[8] 张恒，李立春，于起峰. 尺度方向自适应 Mean Shift 跟踪算法 [J]. 光学 精密工程, 2008,16(6):167-173.
ZHANG H, LI L CH, YU Q F. Scale and direction adaptive Mean Shift tracking algorithm [J]. *Opt. Precision Eng.*, 2008,16(6):167-173. (in Chinese)

[9] ZIVKOVIC Z, KROSE B. An EM-like algorithm for color-histogram-based object tracking [C]. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, 2004:798-803.

[10] HASTIE T. *TIBSHIRANI R. The Elements of Statistical Learning* [M]. 1st ed. America: Springer,2001.

作者简介：

颜　佳(1983－)，男，湖北天门人，博士研究生，2005 年于武汉大学获得学士学位，主要从事目标跟踪、机器视觉等方面的研究。E-mail：yanjiaapple@tom.com

吴敏渊(1964－)，男，湖北武汉人，副教授，1984 年，1989 年分别于武汉测绘科技大学获得学士、硕士学位，主要从事图像处理、机器视觉等方面的研究。E-mail：wmy@eis.whu.edu.cn

张青林(1983－)，男，湖北襄樊人，博士研究生，2005 年于武汉大学获得学士学位，主要从事高速图像处理等方面的研究。E-mail：zhang_qinglin83@yahoo.com.cn

导师简介：

陈淑珍(1946－)，女，湖北武汉人，教授，博士生导师，1970 年于武汉大学获得学士学位，主要从事图形图像处理等方面的研究。E-mail：szchen@whu.edu.cn

●下期预告

微纳 V 槽脆/塑性域切削的 3D 激光检测及评价

谢　晋[1]，韦　凤[1]，田牧纯一[2]

(1.华南理工大学机械与汽车工程学院，广州 510640；2.北見工業大学微纳加工学研究室，日本北見)

针对微纳米级功能 V 槽微细加工及评价困难的问题，采用单点金刚石切削方法在超精密机床上对光学玻璃进行 V 槽的微纳尺度加工，且利用非接触激光检测技术展现 V 槽的加工形貌，用以分析 V 槽的微纳米尺度加工的可行性以及找出如何评价 V 槽加工精度的方法。首先，采用单点金刚石在光学玻璃上进行 V 槽的微纳尺度切削试验。然后，利用 3D 激光超精密检测仪器检测加工的 V 切痕，构建微 V 槽切痕的形貌图，建立 V 槽形状误差 PV 值和 V 槽尖角圆弧半径的评价模式。最后，分析在微纳尺度加工中切除深度与 V 槽角度的形成机理以及切削深度对 V 槽形状误差及其尖角圆弧半径的作用机制。结果表明，在亚微米级尺度加工中存在一个脆/塑性域切除加工状态转变的临界切削深度 0.386 μm。在切削深度＜0.386 μm 的塑性域切削中，金刚石刀具尖角形状可以复制到工件表面，形成深度＜0.386 μm、形状误差 PV 值约 0.103 μm 的 V 槽。此外，V 槽形状误差在塑性域切除加工中始终保持不变，但在脆性域切除加工中随着切削深度增大而逐渐剧烈加大。而且，V 槽尖角圆弧半径在塑性域切削中随着切削深度减小而减小，但切削深度还需控制在临界成型界线以下，达到 0.365 μm 以下，才能形成尖角半径为 0.182 μm 的完整 V 槽。因此，利用非接触激光检测的 3D 数据可以建立 V 槽形状误差 PV 值和尖角圆弧半径的参数模型，用于 V 槽加工精度和微细程度的评价。