

Mean Shift 跟踪算法中尺度自适应策略的研究

左军毅 梁彦 赵春晖 潘泉 张洪才

(西北工业大学自动化学院, 西安 710072)

摘要 标准 Mean Shift 跟踪算法缺乏尺度自适应机制, 一种常见的尺度自适应策略是在上一帧尺度及上一帧尺度基础上增/减 10% 3 个尺度下执行 3 次标准 Mean Shift 算法来确定本帧的尺度。本文在一组典型场景下对这种方法进行了实验研究, 发现它存在两个缺陷, 即有时不能防止尺度在小于真实尺度处徘徊; 对快速尺度变化适应性差。其中任何一个缺陷都可能引起大的尺度定位偏差, 从而降低跟踪器的鲁棒性。在对上述缺陷深入分析的基础上, 修正了最优带宽的判别条件, 给出了自适应滤波器参数的设计方法, 从而得到了一种改进的尺度自适应算法。多种场景下的实验结果表明了这种算法的有效性。

关键词 均值漂移 目标跟踪 尺度自适应

中图分类号: TP391 文献标识码: A 文章编号: 1006-8961(2008)09-1750-08

Researches on Scale Adaptation Strategy in Mean Shift Tracking Algorithm

ZUO Jun-yi, LIANG Yan, ZHAO Chun-hui, PAN Quan, ZHANG Hong-cai

(College of Automation, Northwestern Polytechnical University, Xi'an 710072)

Abstract The standard Mean Shift tracking algorithm is lack of scale adaptation mechanism. A familiar scale adaptation strategy is to determine the scale in current frame by running the standard Mean Shift tracking algorithm three times respectively based on the previous scale, its 10% up and 10% off scales. In this paper, the algorithm was tested on numerous typical scenes and its two drawbacks are found: (1) sometimes it may be stuck in the scale smaller than the real scale; (2) it often responds poorly to rapid scale changes. Such drawbacks can introduce additional scale error and thus increase the risk of missing tracking. Through analyzing above drawbacks in detail, we propose the revised scale adaptation algorithm, in which the criterion of optimal bandwidth selection is modified and adaptive filtering parameter is introduced. Experiment results in numerous scenes show the effectiveness and efficiency of the improved algorithm.

Keywords mean shift, object tracking, scale adaptation

1 引言

作为匹配搜索类跟踪方法的代表, Mean Shift 跟踪算法^[1]一经提出就以其计算量小、对局部遮挡、目标变形、旋转变换有一定适应性等特点而受到了极大重视^[2-4]。这种算法中, 跟踪窗的尺寸作为参与计算的样本数量的决定因素显得非常重要, 其大

小应该随着目标影像实际尺寸的变化而变化。然而, 由于缺乏内在的尺度自适应机制, 这种算法的应用受到了很大限制。为弥补这一不足, 文献[1]提出在上一帧尺度及在上一帧尺度基础上增/减 10% 3 个尺度下执行 3 次 Mean Shift 跟踪算法, 找出 3 次执行结果中使相似性系数最大者对应的带宽为最优带宽, 再用 IIR 滤波器对最优带宽进行平滑处理, 进而得到本帧带宽的估计值。这种方法称为尺度增减

基金项目: 国家自然科学基金重点项目(60634030); 航空科学基金项目(2007ZC53037); 高等学校博士学科点专项科研基金项目(20060699032); 国家自然科学基金项目(60372085)

收稿日期: 2006-12-01; **改回日期:** 2007-04-13

第一作者简介: 左军毅(1975~), 男。西北工业大学自动化学院控制理论与控制工程专业博士研究生。主要研究领域为目标跟踪、图像与视频处理等。E-mail: zuojunyi@163.com

自适应算法。由于这种算法计算量小,便于执行,因而曾被文献多次引用^[4,5]。

Collins 指出尺度增减自适应算法有时候不能防止尺度在小于真实尺度下徘徊,当目标尺寸增大时,这种可造成尺度定位偏差进而引起空间定位偏差的现象尤为严重^[6]。该文通过实验验证了这个问题的存在性,并进一步指出其严重程度不仅和目标的运动形式有关,更重要的是和目标颜色特征在尺度空间上分布的均匀程度有关。另外,尺度增减算法对快速尺度变化适应性差,当目标尺寸的变化速度超过一定范围时,该方法必然会造成尺度跟踪的滞后。在对这两个问题的成因进行深入研究后,提出通过修改最优带宽的判别条件、选用自适应的滤波器参数两项措施对原算法进行改进。大量场景下的实验结果证明,改进算法能有效克服原算法的两个缺陷,而计算量却几乎没有增加。

2 Mean Shift 跟踪算法及尺度增减自适应策略

Mean Shift 跟踪算法^[1]中,目标模型在初始化时被表示成加权直方图的形式: $\hat{q} = \{\hat{q}_u\}_{u=1, \dots, m}$, 其中,

$$\hat{q}_u = C_q \sum_{i=1}^{n_q} k(\|x_i/h_q\|^2) \delta[b(x_i) - u] \quad (1)$$

式中, $C_q = \frac{1}{\sum_{i=1}^{n_q} k(\|x_i/h_q\|^2)}$ 为归一化系数, $\delta(\cdot)$

为 Kronecker 函数, n_q 为目标模型窗口中像素的个数, $k(\cdot)$ 为选定的核轮廓函数, $x_i \in \mathbf{R}^2$ 为目标区域第 i 个像素相对于形心的坐标, $h_q \in \mathbf{R}^2$ 是 2 维带宽向量, x_i/h_q 表示 x_i 和 h_q 的对应分量之比构成的新的 2 维向量。 $b(x_i): \mathbf{R}^2 \rightarrow \{1, \dots, m\}$ 将坐标为 x_i 的像素按照其颜色特征映射到对应的直方图箱格中, u 为直方图箱格的索引。

若用 $\{x_i\}_{i=1, \dots, n_p}$ 来表示以 y 为形心的候选目标区域中像素坐标的集合, 其中 n_p 为像素个数, 则候选目标模型被表示为 $\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1, \dots, m}$ 。其中,

$$\hat{p}_u(y) = C_p \sum_{i=1}^{n_p} k(\|(y - x_i)/h_p\|^2) \delta[b(x_i) - u] \quad (2)$$

这里 $C_p = \frac{1}{\sum_{i=1}^{n_p} k(\|(y - x_i)/h_p\|^2)}$ 是归一化系数,

h_p 为当前核窗宽度。采用 Bhattacharyya 系数

$$\hat{e}(y) = \hat{e}[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (3)$$

来度量 \hat{q} 和 $\hat{p}(y)$ 的相似度, 使上式最大的 y 可通过 Mean Shift 迭代算法近似获得:

$$y_{k+1} = \frac{\sum_{i=1}^{n_p} x_i w_i g\left(\left\|\frac{y_k - x_i}{h_p}\right\|^2\right)}{\sum_{i=1}^{n_p} w_i g\left(\left\|\frac{y_k - x_i}{h_p}\right\|^2\right)} \quad (4)$$

式中, $g(x) = -k'(x)$

$$w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}}$$

在单帧内执行本套迭代算法, 可确定目标的形心位置, 将算法扩展到序列图像则可得 Mean Shift 跟踪算法。

用 Ω_h 表示在计算候选目标模型时权值不为零的像素的集合, 其大小由 h_p 决定; 跟踪窗内像素的集合用 Ω_e 表示; 参与计算的像素的集合用 Ω_n 表示; 实际目标所在区域的像素集合用 Ω_t 表示。参与计算的像素即为跟踪框中权值不为零的像素, 即 $\Omega_n = \Omega_h \cap \Omega_e$ 。

假设目标的形心位置是准确的, 若 $\Omega_n \supset \Omega_t$, 则必然有部分背景像素参与计算, 此时若目标附近的背景中存在和目标类似的颜色模式, 跟踪框可能会被其“吸引”导致目标丢失; 若 $\Omega_n \subset \Omega_t$, 则只有部分目标区域像素参与计算, 这加大了算法丢失快速运动目标的风险。为简便起见, 直接令 $\Omega_e = \Omega_h$, 即跟踪窗就是由权值不为零的像素组成, 则 $\Omega_n = \Omega_h$, 因此 Ω_n 也由 h_p 确定。于是问题简化为如何选择 h_p , 尽可能使 $\Omega_n = \Omega_t$ 。

文献[1]将核函数选为 Epanechnikov 核, 2 维核窗宽参数直接取为跟踪框高、宽尺寸的一半。即 h_p 的两分量和跟踪框高、宽只差一个常数 2, 因此下文中如无特别说明, 对两者不加区分。考虑到 h_p 是随跟踪帧数 k 变化的, 接下来的论述中用 h_k 代替 h_p 。于是尺度增减自适应算法可如下描述: 假设第 k 帧带宽参数为 h_k , 第 $k+1$ 帧, 分别在 3 个带宽 $(1 + 0.1\lambda_k)h_k$ (其中 $\lambda_k \in \{-1, 0, 1\}$) 下执行 3 次 Mean Shift 算法, 求得对应的相似性系数 $e_{0.9}$ 、 e_1 、 $e_{1.1}$, 并认为 3 次迭代

结果中使 Bhattacharyya 系数最大的带宽为 $k + 1$ 时刻的最优带宽 h_{k+1}^{opt} , 而第 $k + 1$ 帧的最终带宽 h_{k+1} 可表示为

$$h_{k+1} = (1 - \gamma)h_k + \gamma h_{k+1}^{opt} \quad (5)$$

γ 为平滑参数。若将跟踪窗高、宽的增减分别独立计算, 则算法可适用于目标变高宽比的情形。

3 尺度增减自适应算法存在的问题

尺度增减算法主要存在两个问题, 分别称为小尺度游荡问题和尺度跟踪滞后问题。前者是 Collins 首先提出的^[6]; 后者是在实验过程中发现的。本文在实验的基础上对这两个问题的成因进行了深入分析, 并进一步指出了其发生的条件, 从而为改进算法的提出提供了依据。

3.1 小尺度游荡问题

在图 1 所示的人脸序列中, 从第 1 帧开始到第 180 帧左右, 目标尺寸缓慢增大; 从 190 帧到 230 帧左右, 目标尺寸缓慢减小; 230 帧到 270 帧, 目标尺寸快速减小。用尺度增减算法来实时调整跟踪框的大小, 结果如图中白色线框所示, 从图 1(b)、(c) 可以看出, 跟踪框并没有随目标尺寸同步增大, 而是在小尺度(小于目标影像真实尺度)下徘徊, 这个问题称为小尺度游荡问题。文献[6]虽指出了该问题,

但并未对其作进一步分析。小尺度游荡引起的尺度定位误差不仅可能会进一步影响空间定位精度^[7], 还可能造成目标丢失。因为当目标影像尺寸较大时, 目标距摄像机一般较近, 其影像在图像中的移动速度也将较快, 此时, 小的跟踪窗往往会因为邻帧之间没有重叠区域而使跟踪器丢失目标。

尺度增减算法认为 3 种尺度中使相似性系数最大的尺度为最优尺度, 这意味着它总是假设目标的真实尺度发生在相似性系数取极大值处。为了检验这一假设, 用图 1(a) 中白色方框区域来构建目标模型, 在 118 帧, 以目标形心为中心, 分别在不同长、宽尺度下统计候选目标模型, 计算它和目标模型的相似性系数, 得到相似性系数随尺度变化的 3 维图(图 2)。可以看出, 当尺度大于真实尺度(图中白色“星号”处)时, 背景像素的融入使相似性系数迅速减小; 当尺度小于真实尺度时, 相似性系数不但没有减小, 反而增大了, 并且在某小尺度处达到峰值。事实上, 对于颜色特征在尺度空间上分布较均匀的目标, 在跟踪过程中由于光照等因素的影响, 很可能出现小尺度下的系数比真实尺度下的还要大的情形(假定目标的形心位置是准确的)。这说明尺度增减算法所依赖的基于相似性系数最大化的最优尺度判别准则有时不成立。明白了这一点, 尺度定位出现误差便不难理解。



图 1 尺度增减算法对变尺寸人脸的跟踪实例

Fig. 1 Example of tracking a face with changing scale using scale-plus-minus algorithm

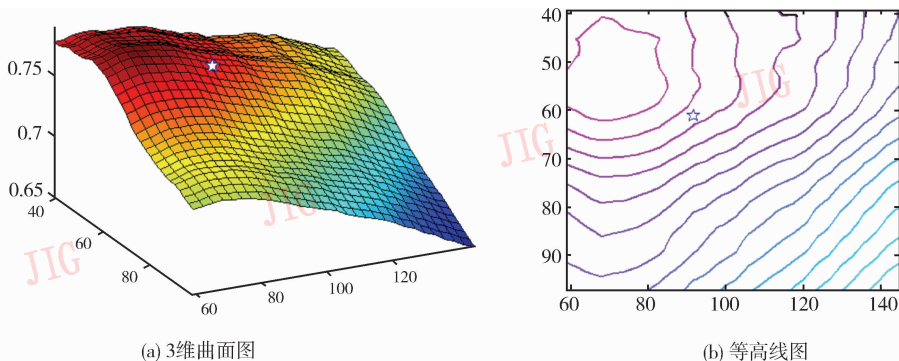


图 2 人脸序列 Bhattacharyya 系数随尺度变化的 3 维图

Fig. 2 3D figure of Bhattacharyya coefficient of the face sequence with respect to scale

对有些目标,上述假设却是满足或近似满足的。图 3 中,CD 光盘的颜色分布在尺度空间上分布极不均匀,其相似性系数随尺度变化的曲面如图 4 所示。显然其峰值点出现在目标真实尺度附近,对这类目标,小尺度游荡问题不存在跟踪结果如图 3 所示。

综上所述,尺度增减算法的性能对目标内部颜色特征的分布情况依赖性很大。既然我们不能对目标颜色分布提出过分的要求,那么如何才能克服或

者缓解尺度增减算法的这一不足呢?

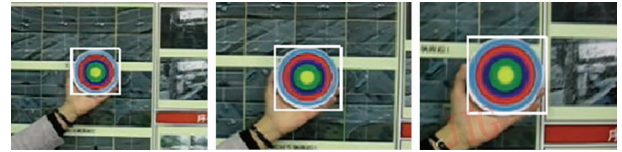


图 3 尺度增减算法对变尺寸 CD 光盘的跟踪实例
Fig. 3 Example of tracking a CD with changing scale using scale-plus-minus algorithm

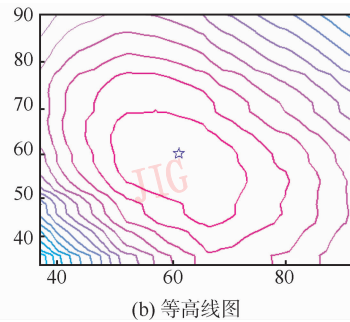
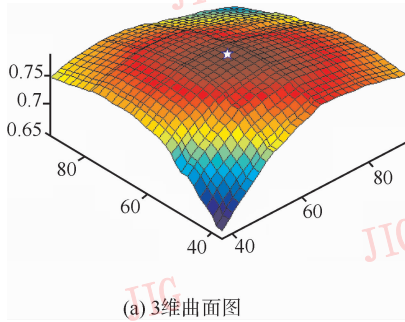


图 4 光盘序列的 Bhattacharyya 系数随尺度变化的 3 维图

Fig. 4 3D figure of Bhattacharyya coefficient of the CD sequence with respect to scale

3.2 尺度跟踪滞后问题

图 5 中,骑摩托车的人沿深度方向快速运动,尺度变化剧烈。从图中白色跟踪框可以看出,尺度增减算法不能使跟踪框及时随目标同步缩小,若干帧后,背景中灌木丛阴影被包含进跟踪框,而阴影正好和人衣服的颜色模式相似,在阴影的“吸引”作用下,跟踪框最终偏离了目标,造成目标丢失。把尺度不能随目标尺寸迅速调整的问题称为尺度跟踪滞后。



图 6 尺度增减算法对尺度快速增大的汽车的跟踪结果
Fig. 6 Example of tracking a car with rapid increase in scale using scale-plus-minus algorithm



图 5 尺度增减算法对尺寸快速减小的摩托车手的跟踪
Fig. 5 Example of tracking a rider with rapid decrease in scale using scale-plus-minus algorithm

当目标尺寸迅速增大时,尺度跟踪滞后现象同样存在。图 6 对小车的跟踪实例中,跟踪框不能及时随目标增大尺寸,造成了大的尺度定位偏差,这种偏差反过来又引起空间定位偏差,从图 6(b)、(c)可以看出,跟踪框的形心和目标形心并不重合。

开始目标尺寸迅速增大,则跟踪框尺寸也应从 h_k 开始增大。第 $k + 1$ 帧时,尺度增减算法为了适应这种变化会使 $\lambda_{k+1} = 1$,此时 $h_{k+1}^{opt} = 1.1h_k$,代入式(5)可知:

$$h_{k+1} = (1 + 0.1\gamma)^1 h_k$$

第 $k + 2$ 帧时, $h_{k+2}^{opt} = 1.1h_{k+1}$,代入式(5)有:

$$h_{k+2} = (1 + 0.1\gamma)^2 h_k$$

依次类推,第 $k + k'$ 帧时:

$$h_{k+k'} = (1 + 0.1\gamma)^{k'} h_k \quad (6)$$

同样,当尺度快速减小时(参数 λ 会连续取 -1)有:

$$h_{k+k'} = (1 - 0.1\gamma)^{k'} h_k \quad (7)$$

将 $h_{k+k'}$ 看作 k' 的函数,则式(6)、(7)构成了尺度增减算法对应的尺度自适应范围的两条边界。

为了便于说明问题,只讨论带宽 h_k 中的一个

尺度跟踪滞后是如何产生的呢?假如从第 k 帧

分量,取其初值为 31,按照文献[1]取 $\gamma = 0.1$,根据式(6)、(7)可得到两条表征尺度变化极限速度的曲线,如图 7 中的两条点线。两条点线之间的部分是尺度增减算法所能达到的区域。图 6 中小汽车影像在第 1 帧时长度为 31 个像素,将这个过程汽车影像长度的真实值用手工的方法标定,得到图 7 中的实线,可见它完全不在两点线间,尺度跟踪滞后现象便由此产生。增大 γ 的值固然可以缓解尺度跟踪滞后,比如 $\gamma = 1$ 时,尺度自适应的范围扩大到图 7 中两条点划线之间的区域(一般认为目标尺寸的单帧增加量不会超过 10%,因此这个区域是足够大的),但较大 γ 值会引起尺度平滑性变差。需要设计一种算法能够兼顾尺度的平滑性和尺度自适应的范围。

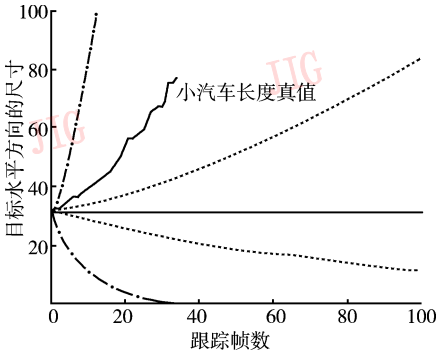


图 7 尺度跟踪滞后的成因图

Fig. 7 Illustration of scale lag

4 改进的尺度增减自适应算法

4.1 小尺度游荡问题的解决方案

大多数监控场景下,目标和背景的颜色分布总有一定差异甚至是明显的差异(这种差异有时还可以通过在线特征选择的方法^[3]增大),相似性系数必然会在真实尺度处以较大的变化率减小,可以利用这个有利条件通过设置合适的系数 α 来放宽尺度增大的判别条件,同时使尺度减小的判别条件变得更为苛刻,来解决小尺度游荡问题。

首先,取一个略小于 1 的系数 α 来对 3 个相似性系数重新赋值,即令 $e_{1.1} \leftarrow \alpha^{-1}e_{1.1}$, $e_{0.9} \leftarrow \alpha e_{0.9}$, $e_1 \leftarrow e_1$, 然后根据校正后的相似性系数来选择最优带宽,即

$$h_{k+1}^{\text{opt}} = \begin{cases} 1.1h_k & e_{1.1} \geq \max(e_1, e_{0.9}) \\ 0.9h_k & e_{0.9} > \max(e_1, e_{1.1}) \\ h_k & \text{其他} \end{cases} \quad (8)$$

考虑到对不同场景的适应性,将 α 设计成可变参数是适宜的。注意到目标和背景颜色特征的相似度粗略地反映了当前跟踪环境对小 α 值的“容忍”程度,因此可将 α 设计成二者相似度的函数,若相似度较大,则 α 取较大的值,以保证尺度不发散;反之, α 取较小的值,以充分利用背景和背景颜色特征的差异来解决小尺度游荡问题。

用 p° 来表示目标颜色分布直方图, q^b 表示当前目标周围背景的颜色分布直方图。与第 2 节介绍的颜色模型的不同之处在于此处的 p° 、 q^b 无需用距离加权。用 p° 和 q^b 夹角的余弦

$$\cos\theta_{pq} = \frac{\langle p^\circ, q^b \rangle}{\|p^\circ\| \|q^b\|} \quad (9)$$

来表示它们之间的相似度。式中, $\langle p^\circ, q^b \rangle$ 表示向量内积, $\|x\|$ 表示向量 x 的长度。则 α 的值由下式给出:

$$\alpha = 0.98 + 0.02\cos\theta_{pq} \quad (10)$$

大多数情况下,目标和背景的颜色特征总存在一定差异,按照式(10)求得的 α 总能使原算法的问题得到很好解决,同时保证尺度不发散。

式(9)中的 p° 及 $\|p^\circ\|$ 是固定的,只需在线调用;同时考虑到背景的颜色分布在邻帧间不可能发生太大变化,因此对 α 的更新可以每隔 5~8 帧执行一次,由此可见改进算法带来的附加计算量很小。

4.2 尺度跟踪滞后问题的解决方案

当目标宽、高尺寸同比例变化时,2 维带宽向量 h_k 的两分量只差一个常数;若不同比例时,可分别独立地计算 h_k 的两分量。为了简化描述,以下只研究 h_k 的一个分量,因此该小节中带宽用标量表示。

为了兼顾尺度自适应的范围和平滑性,选用可变的滤波器参数 γ ,在目标尺度剧烈变化时增大 γ ;反之,减小它。 γ 是时变的,用 γ_k 表示 γ 在 k 时刻的值。

若 λ_k ($\lambda_k \in \{-1, 0, 1\}$) 连续取 1 或者 -1,即 $\lambda_k = \lambda_{k-1} = 1$ 或 $\lambda_k = \lambda_{k-1} = -1$,认为此时计算出的尺度跟不上目标影像实际尺寸的变化,滤波器的系数太小,可按式(11)第 1 式的经验公式增大它;当 $\lambda_k = \lambda_{k-1} = 0$ 时,说明目标影像尺寸变化很小,按式(11)第 1 式减小滤波器系数。当 $\lambda_k \neq \lambda_{k-1}$ 时,计算出的尺度和目标真实尺度相对比较接近,用式(11)第 2 式对 γ_k 进行修正。

$$\gamma_k = \begin{cases} \max\{0.1, \xi\gamma_{k-1} + (1-\xi)|\lambda_k|\} & \lambda_k = \lambda_{k-1} \\ \max\{0.1, \min\{10|e^{\frac{\dot{h}_k}{h_k}} - 1|, 1\}\} & \lambda_k \neq \lambda_{k-1} \end{cases} \quad (11)$$

式中, $\xi \in (0, 1)$ 为滤波器系数的更新因子, 其大小决定了 γ_k 对 λ_k 变化的敏感程度。

\hat{g}_k 是当 $\lambda_k \neq \lambda_{k-1}$ 时目标尺寸变化率的估计值, 计算时若直接采用两点求斜率的方法, 即 $\hat{g}_k = (h_k - h_{k_0}) / (k - k_0)$, 则 \hat{g}_k 可能会受到强噪声的污染。为此将 $(h_k - h_{k_0}) / (k - k_0)$ 看作 \hat{g}_k 的当前观测, 通过平滑处理来消除部分噪声, 即

$$\hat{g}_k = (1 - \eta)\hat{g}_{k-1} + \eta \frac{(h_k - h_{k_0})}{(k - k_0)} \quad (12)$$

式中, η 为平滑参数, k_0 为上一次 λ 值发生变化的时刻, h_{k_0} 为 k_0 时刻求得的目标尺寸。 η 的选择应充分考虑新观测值的噪声水平。当 $k - k_0$ 很小时, $\frac{(h_k - h_{k_0})}{(k - k_0)}$ 受噪声影响大, 相应的 η 应较小, 因此, 可将 η 设计成具有如下形式的自适应结构:

$$\eta = \max\left\{0, \min\left\{\frac{k - k_0}{C}, 1\right\}\right\} \quad (13)$$

C 为给定常数。

获得 \hat{g}_k 后, 令指数函数式(6)或(7)在原点处的导数等于 \hat{g}_k ,

$$\left. \frac{dh_{k+k'}}{dk'} \right|_{k'=0} = \hat{g}_k$$

将 $h_{k+k'}$ 的表达式代入上式, 有

$$h_k \ln(1 \pm 0.1\gamma)(1 \pm 0.1\gamma)^{k'} \Big|_{k'=0} = \hat{g}_k$$

可解得 $\gamma_k = 10 \left| e^{\frac{\hat{g}_k}{h_k}} - 1 \right|$, 并令 $\gamma_k \in [0.1, 1]$, 可得式(11)第2式。

由于式(11~13)每帧只需计算一次, 因此计算量几乎没有增加。

5 实验结果

5.1 实验结果比较

在不同场景下做了大量实验来验证两项改进措施的有效性。实验中滤波器系数的更新因子 $\xi = 0.97$, 式(13)中的 $C = 20$ 。

图8第1行是在尺度增减原型算法基础上引入第1项改进措施后的实验结果。当目标尺度缓慢变化时, 尺度定位效果很好, 这说明 α 的引入有效地解决了小尺度游荡问题; 在266帧左右, 目标尺度急剧缩小, 跟踪框不能及时和目标尺寸同步缩小, 即发生了尺度跟踪滞后现象, 同样的问题也出现在尺度增减原型算法中。图8第2行是在原算法上同时增加两项改进措施后的跟踪结果, 可见在整个过程中, 尺度定位精度很高。在 PV1.8GHz 的实验平台上, 对于这个长350帧的人脸序列, 改进算法的单帧平均执行时间是20ms, 可见实时性很强。

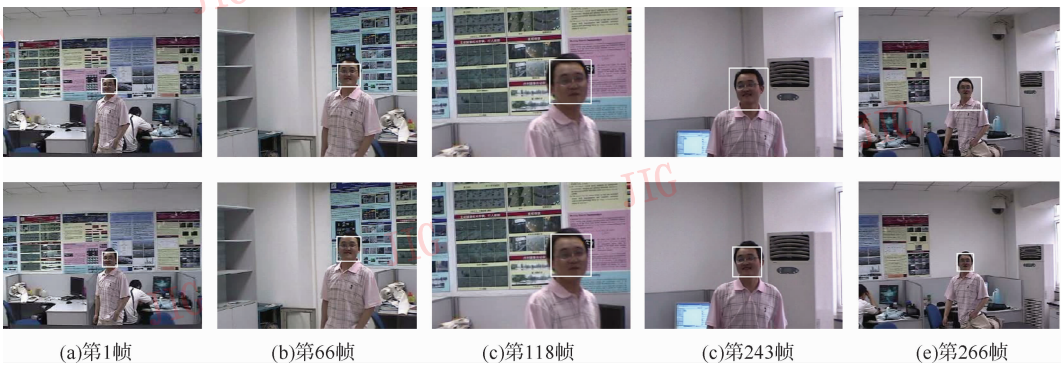


图8 尺度增减改进算法对人脸序列的跟踪结果

Fig. 8 Tracking the face using improved scale-plus-minus algorithm

图9给出了改进算法对摩托车序列的跟踪结果, 和原算法跟踪结果(图5)相比, 算法对快速尺度变化的反应速度加快了, 跟踪框能迅速随目标缩小, 背景中的相似颜色模式没有机会进入其中, 从而有效提高了算法的鲁棒性。

图10是改进算法对小车序列的跟踪结果。和图6原型算法的实验结果相比, 改进算法尺度定位



图9 改进算法对摩托车手的跟踪结果

Fig. 9 Tracking results of the rider using improved algorithm



图 10 改进算法对小汽车的跟踪结果

Fig. 10 Tracking results of the car using improved algorithm

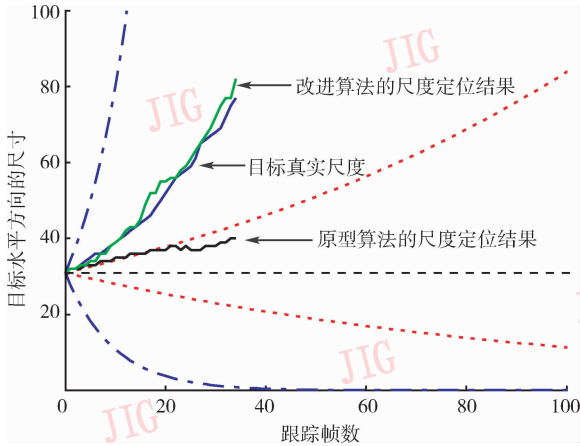


图 11 汽车的尺度变化

Fig. 11 The scale change of the car

精度要高得多。尺度定位精度的提高反过来引起空间定位精度的提高。从图 10(b)、(c)可以看出,跟踪框的形心和目标形心几乎重合。分别画出尺度增减原型算法、改进后的算法计算得到的小车长度随帧数的变化曲线,分别对应于图 11 中的黑线和绿线,如前所分析,黑线严格位于两条红虚线之间,严重偏离了真值曲线(实蓝线),并且在某些帧处错误地指示出目标尺度在减小;绿线和蓝线位置很接近,表明本文算法获得的目标尺寸很接近于目标真实尺寸。

5.2 参数分析

算法所涉及的两个关键参数也如期望的那样产生自适应的调节作用。在图 8 的人脸序列前 170 帧,系数 α 的变化情况如图 12 所示。当目标逐渐由复杂的海报背景过渡到单一的墙面背景时,目标和背景的相似度降低,因此 α 的值有所减小。但从总体上讲,这个序列中目标和背景的颜色特征差异还是比较大的, α 的值没有超过 0.984。然而在图 13 的序列中,目标和背景的颜色相似度较大,算法在不修改任何参数的情况下,仍能保证尺度不发散,且能有效缓解小尺度游荡问题。跟踪过程中参数 α 取值都在 0.991

之上,其变化情况如图 14 所示,可见本文算法能够根据跟踪环境自动选取合适的 α 值。

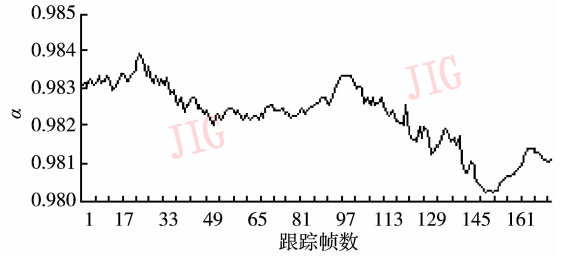


图 12 人脸跟踪过程中参数 α 的曲线

Fig. 12 The parameter α in face tracking



图 13 目标和背景颜色分布相似度较大情况下的跟踪结果

Fig. 13 Tracking result in the case that object share the similarly color distribution with background

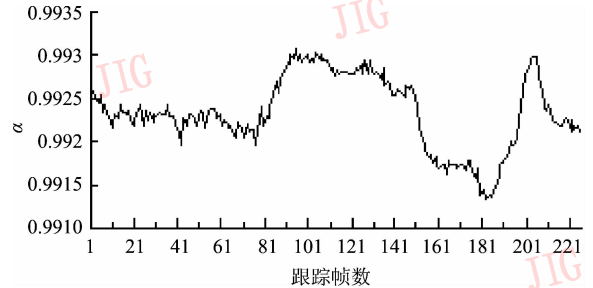


图 14 行人跟踪过程中参数 α 的曲线

Fig. 14 The parameter α in pedestrian tracking

滤波器系数能根据目标尺寸变化快慢及时调整。对于人脸序列, γ 的变化情况如图 15 所示。大约在 230 帧以后, γ 取值较大,这正好对应于视频序列中目标的一次快速尺度变化。

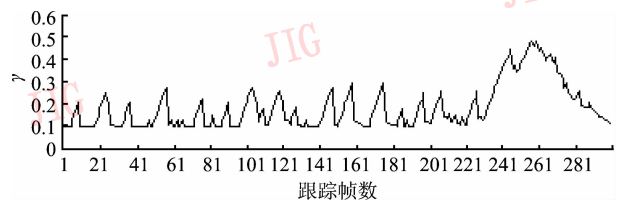


图 15 人脸跟踪过程中 γ 的曲线

Fig. 15 The parameter γ in face tracking

实验中发现 C 的取值在 10 到 25 之间变化或者 ξ 取值在 0.95 附近变化对实验结果几乎没有影响,即

算法对参数 C 和 ξ 的选择有较强的鲁棒性。这主要是因为参数 C 和 ξ 的引入实现了滤波器参数 γ 的自适应,从而以一种很间接的方式影响着算法的性能。

6 结 论

Mean Shift跟踪算法中,尺度自适应问题解决的好坏直接关系到这种算法的鲁棒性。不幸的是,从计算量、精度等方面综合考虑,目前还没有一种解决方法让人满意。在已知的方法中,尺度增减自适应算法由于执行简单、计算量小而倍受关注。然而,其固有的小尺度游荡问题和尺度跟踪滞后问题使其实用性大打折扣。本文在对这两个问题的成因深入分析的基础上,通过修改最优带宽的判别条件、选用自适应的滤波器参数两项措施对原算法进行了改进,实验结果表明,这种改进策略是有效且高效的有很高的应用价值。

参考文献 (References)

- 1 Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, **25**(5): 564 ~ 577.
- 2 Yang C J, Duraiswami R, Davis L. Efficient spatial-feature tracking via the Mean-Shift and a new similarity measure [A]. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition [C], San Diego, USA, 2005:293 ~ 300.
- 3 Collins R, Liu Y X. Online selection of discriminative tracking features [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, **27**(10):1631 ~ 1643.
- 4 Deguchi K, Kawanaka O, Okatani T. Object tracking by the Mean-Shift of regional color distribution combined with the particle-filter algorithm [A]. In: Proceedings of the 17th International Conference on Pattern Recognition [C], Cambridge, UK,2004:506 ~ 509.
- 5 Adam A, Rivlin E, Shimshoni I. Robust fragments-based tracking using the integral histogram [A]. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition [C], New York, USA,2006: 798 ~ 805.
- 6 Collins R. Mean-Shift blob tracking through scale space [A]. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition [C], Madison, Wisconsin, USA, 2003:234 ~ 240.
- 7 Peng Ning-song, Yang Jie, Liu Zhi, *et al.* Automatic selection of kernel-bandwidth for Mean-Shift object tracking [J]. Journal of Software, 2005, **16**(9): 1542 ~ 1550. [彭宁嵩, 杨杰, 刘志等. Mean Shift跟踪算法中核函数窗宽的自动选取[J]. 软件学报, 2005, **16**(9): 1542 ~ 1550.]