

一种复杂背景下的实时目标跟踪算法

邸男^{1),2)} 朱明¹⁾

¹⁾(中国科学院长春光学精密机械与物理研究所,长春 130033) ²⁾(中国科学院研究生院,北京 100039)

摘要 鉴于现有的均值漂移跟踪算法不能适应目标运动速度过快和全部遮挡,并且由于存在开方、除法等大量浮点运算,在定点 DSPs 数字信号处理器中应用难于满足实时跟踪要求。本文在核密度估计,均值漂移算法迭代权值, Bhattacharyya 系数等方面进行改进,给出优化的算法流程。不仅增强了目标与背景相似时搜索目标的能力,而且全部使用整点运算,计算时间满足实时跟踪要求。通过将优化算法与卡尔曼轨迹预测相结合,解决了目标运动速度过快和全部遮挡问题。目前该算法已经成功嵌入 TMS320C6416 硬件平台,实验结果表明,在混乱背景、目标发生旋转、遮挡、形状变化、运动速度过快等情况下,该算法能够保持连续稳定的实时跟踪。

关键词 均值漂移 实时跟踪 核函数 迭代权值 Bhattacharyya 系数 卡尔曼预测

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2008)09-1758-08

An Algorithm for Real-time Tracking Target in Complex Environment

DI Nan^{1),2)}, ZHU Ming¹⁾

¹⁾(Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033)

²⁾(Graduate School of Chinese Academy of Sciences, Beijing 100039)

Abstract The current mean-shift tracking algorithm is not suitable for the targets that move fast and with entire occlusions and is hard to realize real-time tracking when it is applied in the pointing digital signal processors due to much floating-point operation. This paper provides the optimal algorithm which contains some improvements of the kernel function, Mean Shift alternative weight and Bhattacharyya coefficient. These improvements not only enhance the capacity of tracking the object which resembles the background but also employ pointing operation to satisfy the need of the real-time tracking. Furthermore, the target moving fast and entire occlusions are resolved through combining the optimal algorithm with the Kalman forecast. Now, the algorithm is applied in the TMS320C6416 hardware system and successfully copes with clutter, target occlusions, revolution, scale variations and moving fast in the real-time tracking.

Keywords mean-shift, real-time tracking, kernel function, alternate weight, Bhattacharyya coefficient, Kalman forecast

1 引言

数字视频监控^[1]、无人侦察机战场侦察、火力控制系统以及巡航导弹末端制导都要求能对复杂环境中的运动目标进行实时跟踪。由于运动目标常常发生旋转、缩放、与背景过于相似、被遮挡等情况,这给跟踪算法的研究带来很大困难,算法既要适应目

标和背景的复杂变化,又要保证运算量小,满足实时要求。近年来,均值漂移算法^[2]以其无需参数、快速模式匹配的特性被广泛应用于目标跟踪领域。Comaniciu 提出使用核函数计算跟踪目标的彩色直方图,利用 Bhattacharyya 相似函数进行模板匹配,然后使用均值漂移算法找到相似函数的局部极大值点,最终确定目标跟踪区域^[3]。该方法能够解决摄像机运动、目标局部遮挡、目标尺度变化等干扰跟踪

基金项目:国家高技术研究发展计划(863计划)项目(2005AA778032)

收稿日期:2006-12-05;改回日期:2007-04-23

第一作者简介:邸男(1981~),女,实习研究员。现为长春光学精密机械与物理研究所机械电子专业在读博士研究生。主要研究方向为机器视觉和视频跟踪。E-mail:dinan0620@yahoo.com.cn

的问题。但由于涉及彩色直方图的计算,运算量比较大,不便于实时实现,同时仅适用于相邻帧目标位置变化不大的情况,没有解决目标运动过快和全部遮挡问题。江淑红针对文献[3]运算量大的问题提出解决方法,使用灰度直方图代替彩色直方图,同时改进均值漂移算法中的迭代权值,对于采样速率为每秒5帧的运动图像序列达到了很好的跟踪效果,但仍没有解决全部遮挡问题^[4]。本文以文献[4]为基础,进一步改进核函数、均值漂移算法中的迭代权值以及 Bhattacharyya 相似系数的计算,提出优化算法,大大简化了运算量。同时将优化算法与卡尔曼轨迹预测^[5]相结合,制定决策策略,解决了相邻帧目标位置变化大于目标大小的目标快速运动和全部遮挡问题,实现了连续稳定的序列图像实时跟踪。目前,算法已经成功嵌入硬件平台,实现了其在工程方面的应用。

2 跟踪算法

2.1 核函数的改进

由核密度估计方法^[6]知,初始模板的灰度直方图为

$$\hat{q}_u = C \sum_{i=1}^n K(\mathbf{X}_i^*, \mathbf{H}) \delta[b(\mathbf{X}_i^*) - u] \quad (1)$$

式中, $u = 1, \dots, m$; $i = 1, \dots, n$ 。 m 表示灰度级数, n 表示初始模板包含的像素个数。以图像左上角为坐标原点,水平向右为 x 方向,垂直向下为 y 方向, $\mathbf{X}_i = (x_i, y_i)$ 表示初始模板中第 i 个像素的坐标, $\mathbf{X}_c = (x_c, y_c)$ 表示初始模板中心像素的坐标,则 $\mathbf{X}_i^* = (x_i^*, y_i^*) = (x_i - x_c, y_i - y_c)$ 表示以 \mathbf{X}_c 为坐标原点第 i 个像素的相对坐标。 \mathbf{H} 为初始模板的带宽向量, $K(\mathbf{X}_i^*, \mathbf{H})$ 为核函数, $C = \frac{1}{\sum_{i=1}^n K(\mathbf{X}_i^*, \mathbf{H})}$ 是标准化常数,使得

$\sum_{u=1}^m \hat{q}_u = 1$ 成立。 δ 是 Kronecker delta 函数,即有

$$\delta[b(\mathbf{X}_i^*) - u] = \begin{cases} 1 & b(\mathbf{X}_i^*) = u \\ 0 & b(\mathbf{X}_i^*) \neq u \end{cases}$$

其中, $b(\mathbf{X}_i^*)$ 表示 \mathbf{X}_i^* 点的灰度值。

传统的直方图仅仅统计每个灰度值出现的概率,与灰度值所处的位置无关。核密度估计方法使用核函数对各个像素点的灰度值加权后再统计直方图,其中核函数的值与该像素点距中心像素点的距离成反比。此方法不仅统计每个灰度值的出现概率,同时引入像素点的空间位置信息,距离中心越远

的像素点对该灰度的直方图统计值贡献越小,反之越大。这与实际情况是相符合的,模板中越远离中心的点受到背景图像的影响越大,对模板直方图的贡献越小,反之越大。

文献[3]使用 Epanechnikov 核函数

$$K\left(\left\|\frac{\mathbf{X}_i^*}{\mathbf{H}}\right\|^2\right) = \frac{1}{2}c_d^{-1}(d+2)\left(1 - \left(\frac{x_i^*}{h_x}\right)^2 - \left(\frac{y_i^*}{h_y}\right)^2\right)$$

计算直方图。文献[4]将该核函数的系数 $\frac{1}{2}c_d^{-1}(d+2)$ 省略,得到新的核函数为

$$K\left(\left\|\frac{\mathbf{X}_i^*}{\mathbf{H}}\right\|^2\right) = 1 - \left(\frac{x_i^*}{h_x}\right)^2 - \left(\frac{y_i^*}{h_y}\right)^2 \quad (2)$$

从几何角度看,式(2)为椭圆抛物面方程。本文使用正四棱锥面方程

$$K(\mathbf{X}_i^*, \mathbf{H}) = h_x - |x_i^*| + h_y - |y_i^*| \quad (3)$$

代替式(2),以平面近似曲面,在不改变核函数单调递减性质的同时避免了式(2)的浮点运算和乘方运算,大大简化了计算量。

同样道理,跟踪模板灰度直方图为

$$\hat{p}_u(\mathbf{Y}) = C' \sum_{i=1}^{n'} K(\mathbf{X}_i^*, \mathbf{H}') \delta[b(\mathbf{X}_i^*) - u] \quad (4)$$

式中, \mathbf{Y} 为跟踪模板中心像素的坐标, \mathbf{X}_i^* 表示以 \mathbf{Y} 为坐标原点的第 i 个像素的相对坐标, \mathbf{H}' 为跟踪模板的带宽向量, $C' = \frac{1}{\sum_{i=1}^{n'} K(\mathbf{X}_i^*, \mathbf{H}')} = 1$ 成立的标准化常数, n' 为跟踪模板包含的像素个数。

2.2 初始模板直方图与跟踪模板直方图的距离

使用 Bhattacharyya 系数度量直方图向量 $\hat{\mathbf{Q}} = (\hat{q}_1, \dots, \hat{q}_m)^T$ 与 $\hat{\mathbf{P}}(\mathbf{Y}) = (\hat{p}_1(\mathbf{Y}), \dots, \hat{p}_m(\mathbf{Y}))^T$ 的相似性,即

$$\hat{\rho}(\mathbf{Y}) \equiv \rho(\hat{\mathbf{Q}}, \hat{\mathbf{P}}(\mathbf{Y})) = \sum_{u=1}^m \sqrt{\hat{q}_u \hat{p}_u(\mathbf{Y})} \quad (5)$$

Bhattacharyya 系数具有明确的几何意义,因为 $\hat{\mathbf{Q}}$, $\hat{\mathbf{P}}(\mathbf{Y})$ 都是密度函数,满足

$$\sum_{u=1}^m \hat{q}_u = \sum_{u=1}^m (\sqrt{\hat{q}_u})^2 = 1$$

$$\sum_{u=1}^m \hat{p}_u(\mathbf{Y}) = \sum_{u=1}^m (\sqrt{\hat{p}_u(\mathbf{Y})})^2 = 1$$

因此 $(\sqrt{\hat{q}_1}, \dots, \sqrt{\hat{q}_m})^T$ 和 $(\sqrt{\hat{p}_1(\mathbf{Y})}, \dots, \sqrt{\hat{p}_m(\mathbf{Y})})^T$ 都是单位球面向量,式(5)正表示两向量夹角的余弦值。两向量夹角越小,余弦值越大,相似度越大,当夹角为 0 时,相似度为 1。

定义距离函数

$$d(\mathbf{Y}) = \sqrt{1 - \rho(\hat{\mathbf{Q}}, \hat{\mathbf{P}}(\mathbf{Y}))} \quad (6)$$

文献[3]证明该距离满足距离公理。根据式(1)、式(4)可知,对于任意 $u = 1, \dots, m$, $\hat{q}_u, \hat{p}_u(\mathbf{Y})$ 为分段连续函数,由式(5)可知 $\hat{\rho}(\mathbf{Y})$ 为分段连续函数,因此距离函数 $d(\mathbf{Y})$ 连续,可以使用均值漂移算法搜索距离函数的极小值。

2.3 使用均值漂移算法最小化距离函数

由式(6)可知,距离最小化相当于 $\hat{\rho}(\mathbf{Y})$ 最大化,将它在 $\hat{\mathbf{Y}}_{k-1}$ ($\hat{\mathbf{Y}}_{k-1}$ 为第 $k-1$ 帧目标的位置)处泰勒展开,假设相邻帧灰度直方图变化不大,认为 $|\hat{p}_u(\mathbf{Y}) - \hat{p}_u(\hat{\mathbf{Y}}_{k-1})|$ 很小,则只取泰勒一阶近似得:

$$\rho(\hat{\mathbf{Q}}, \hat{\mathbf{P}}(\mathbf{Y})) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{q}_u \hat{p}_u(\hat{\mathbf{Y}}_{k-1})} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(\mathbf{Y}) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{Y}}_{k-1})}} \quad (7)$$

将式(4)代入式(7)得:

$$\rho(\hat{\mathbf{Q}}, \hat{\mathbf{P}}(\mathbf{Y})) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{q}_u \hat{p}_u(\hat{\mathbf{Y}}_{k-1})} + \frac{C'}{2} \sum_{i=1}^{n'} \omega_i K(\mathbf{X}_i^*, \mathbf{H}') \quad (8)$$

其中迭代权值

$$\omega_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{Y}}_{k-1})}} \delta[b(\mathbf{X}_i) - u] = \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{Y}}_{k-1})}} \delta[b(\mathbf{X}_i) - u] \quad (9)$$

至此问题转化为将式(8)最大化。右边第1项与 \mathbf{Y} 无关,不再考虑。第2项的最大化就是密度最大化问题,只是多了权值 ω_i ,因此可以使用均值漂移算法^[7]确定第2项的局部极大值点,推导迭代位置公式为

$$\hat{\mathbf{Y}}_k = \frac{\sum_{i=1}^{n'} \mathbf{X}_i \omega_i g\left(\left\|\frac{\hat{\mathbf{Y}}_{k-1} - \mathbf{X}_i}{\mathbf{H}'}\right\|^2\right)}{\sum_{i=1}^{n'} \omega_i g\left(\left\|\frac{\hat{\mathbf{Y}}_{k-1} - \mathbf{X}_i}{\mathbf{H}'}\right\|^2\right)} \quad (10)$$

式中, $g(x) = -k'(x)$, $k'(x)$ 为核函数的轮廓函数, $\hat{\mathbf{Y}}_k$ 为第 k 帧迭代的新位置。式(10)的收敛性证明参见文献[8]。

由于选用 Epanechnikov 核函数,轮廓函数为

$$k(x) = \begin{cases} 1 - x & 0 \leq x \leq 1 \\ 0 & x > 1 \end{cases}$$

所以 $g(x) = -k'(x) = 1$,式(10)化为

$$\hat{\mathbf{Y}}_k = \frac{\sum_{i=1}^{n'} \mathbf{X}_i \omega_i}{\sum_{i=1}^{n'} \omega_i}$$

2.4 均值漂移算法中迭代权值的改进

用 $\tilde{\omega}_i = \omega_i^2$ 代替 ω_i ,根据式(9)得

$$\tilde{\omega}_i = \frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{Y}}_{k-1})} \delta[b(\mathbf{X}_i) - u]$$

对于某一个固定的灰度值 u ,当 $\hat{q}_u > \hat{p}_u(\hat{\mathbf{Y}}_{k-1})$ 时,也就是说初始模板中灰度值 u 出现的概率比较大时,灰度为 u 的像素是模板中的主要信息,此时 $\omega_i > 1$, $\tilde{\omega}_i = \omega_i^2 > \omega_i$,说明改进以后,初始模板中主要信息被赋予的权值比改进以前大;对于某一个固定的 u' ,当 $\hat{q}_{u'} < \hat{p}_{u'}(\hat{\mathbf{Y}}_{k-1})$ 时,初始模板中灰度 u' 出现的概率比较小,说明灰度为 u' 的像素是模板中的次要信息,此时 $\omega_i < 1$, $\tilde{\omega}_i = \omega_i^2 < \omega_i$,说明改进以后,初始模板中次要信息被赋予的权值比改进以前小。

综上所述,改进的 $\tilde{\omega}_i$ 使主要信息和次要信息对比更加鲜明,处理目标与背景相近时很有效,并且计算 $\tilde{\omega}_i$ 只需除法运算,避免了计算 ω_i 中的开方运算,进一步简化了计算量。

2.5 Bhattacharyya 系数的改进

2.2 小节中使用 Bhattacharyya 系数度量直方向量之间的相似性是利用了该系数连续的性质,可以进行泰勒展开,最终推导出均值漂移向量。在实际应用中,尝试使用其他相似性度量代替 Bhattacharyya 系数,希望在不改变相似性度量性能的前提下,尽量减少计算量。

度量直方向量相似性的方法有很多种,Swain 和 Ballard 的直方图相交算法^[9]是目前比较有效的度量方法之一。本文借鉴该方法,使用

$$\tilde{\rho}(\mathbf{Y}) = \sum_{u=1}^m \min(\hat{q}_u, \hat{p}_u(\mathbf{Y})) \quad (11)$$

代替式(5)计算相似系数,可以避免式(5)的开方运算,减少计算量。

3 优化算法流程

假设第 $k-1$ 帧的跟踪模板中心为 $\hat{\mathbf{Y}}_{k-1}$,带宽向量 $\mathbf{H}_{k-1} = (h_x, h_y)$,尺度系数为 s_{k-1} 。求第 k 帧跟踪位置 $\hat{\mathbf{Y}}_k$,尺度系数 s_k ,带宽向量 $\mathbf{H}_k = (h'_x, h'_y)$ 。令 $s_i = 9s_{i-1}$, $\rho_{\text{temp}} = -1$,计数器 $\text{count} = 0$, $\hat{\mathbf{Y}}_i = (x_{y_i}, y_{y_i})$, $\hat{\mathbf{Y}}'_i = (x_{y'_i}, y_{y'_i})$ 。

(1)将式(3)代入式(1)计算 $\{\hat{q}_u\}_{u=1,\dots,m}$

$$\hat{q}_u = \sum_{i=1}^n (h_x - |x_i^*| + h_y - |y_i^*|) \delta[b(\mathbf{X}_i) - u]$$

$$C = \sum_{i=1}^n (h_x - |x_i^*| + h_y - |y_i^*|)$$

(2) $\hat{\mathbf{Y}}_t \leftarrow \hat{\mathbf{Y}}_{k-1}, \mathbf{H}_k \leftarrow \mathbf{H}_{k-1}$ 。

(3)将式(3)代入式(4)计算 $\{\hat{p}_u(\hat{\mathbf{Y}}_t)\}_{u=1,\dots,m}$

$$\hat{p}_u(\hat{\mathbf{Y}}_t) = \sum_{i=1}^{n'} \left(h'_x - \frac{|x_i - x_{y_t}| \times 10000}{s_t} + h'_y - \frac{|y_i - y_{y_t}| \times 10000}{s_t} \right) \delta[b(\mathbf{X}_i) - u]$$

$$C' = \sum_{i=1}^{n'} \left(h'_x - \frac{|x_i - x_{y_t}| \times 10000}{s_t} + h'_y - \frac{|y_i - y_{y_t}| \times 10000}{s_t} \right)$$

(4)计算迭代权值 $\{\tilde{\omega}_i\}_{i=1,\dots,n'}$

$$\tilde{\omega}_i = \frac{\hat{q}_u}{\hat{p}_u(\hat{\mathbf{Y}}_t)} \delta[b(\mathbf{X}_i) - u]$$

(5)计算新的目标中心位置

$$\hat{\mathbf{Y}}'_t = \frac{\sum_{i=1}^{n'} \mathbf{X}_i \tilde{\omega}_i}{\sum_{i=1}^{n'} \tilde{\omega}_i}$$

(6)判断是否收敛

如果 $|x_{y'_t} - x_{y_t}| \geq 1$ 或 $|y_{y'_t} - y_{y_t}| \geq 1$, $\hat{\mathbf{Y}}_t \leftarrow \hat{\mathbf{Y}}'_t$, 执行步骤3。

否则,计算 $\tilde{\rho}(\mathbf{Y}'_t) = \sum_{u=1}^m \min\left(\hat{q}_u, \frac{C \hat{p}_u(\mathbf{Y}'_t)}{C'}\right)$,

如果 $\rho_{temp} < \tilde{\rho}(\hat{\mathbf{Y}}'_t)$

则 $\rho_{temp} \leftarrow \tilde{\rho}(\hat{\mathbf{Y}}'_t)$

$\hat{\mathbf{Y}}_k \leftarrow \hat{\mathbf{Y}}'_t$

$s_{temp} \leftarrow s_t + 90s$

(7) $s_t += s_{k-1}$, $count ++$;

如果 $count < 3$, 执行步骤2;

否则,算法结束。

此时, $\hat{\mathbf{Y}}_k$ 为新的目标跟踪位置;新的尺度系数

$s_k = s_{temp}/100$;新的带宽向量为 $\mathbf{H}_k = s_k \mathbf{H}_{k-1}/1000$ 。

算法说明:

(1)以第1帧选取的模板中心位置 \mathbf{X}_c , 带宽向量 \mathbf{H} 分别作为 $\hat{\mathbf{Y}}_{k-1}, \mathbf{H}_{k-1}$ 的初值开始算法。

(2)对于尺度变化准则,文献[4]分别计算 $s_t = 0.9s_{k-1}; s_t = s_{k-1}; s_t = 1.1s_{k-1}$ 时的3个跟踪位置处的相似系数,选择使系数值达到最大的 s_t , 则新的尺度为 $s_k = 0.1s_t + 0.9s_{k-1}$ 。为了避免浮点运算,将 s_{k-1} 的初值赋予1000,步骤3中的分子乘以10000。将 $s_t = 9s_{k-1}; s_t = 10s_{k-1}; s_t = 11s_{k-1}$ 分别代入步骤2~步骤6计算3次($count < 3$ 表示循环计算三次),得到3个 \mathbf{Y}'_t , 然后在这3个位置分别计算 $\tilde{\rho}(\mathbf{Y}'_t)$, 选择使 $\tilde{\rho}(\mathbf{Y}'_t)$ 值达到最大的 s_t , 则新的尺度为 $s_k = (s_t + 90s_{k-1})/100$ 。新的带宽向量为 $\mathbf{H}_k = s_k \mathbf{H}_{k-1}/1000$ 。

(3)在步骤1中,由于初始模板的标准化常数 $C = \sum_{i=1}^n (h_x - |x_i^*| + h_y - |y_i^*|)$ 为常量,在步骤4中仅使 $\tilde{\omega}_i$ 成倍变化,而步骤5对 $\tilde{\omega}_i$ 进行标准化时分子分母中的固定倍数可以约掉,因此 \hat{q}_u 中不再除以 C , 而是单独计算 C , 用于在步骤6中计算 $\tilde{\rho}(\hat{\mathbf{Y}}'_t)$ 。步骤3与步骤1相似。

(4)在步骤6中计算 $\tilde{\rho}(\hat{\mathbf{Y}}'_t)$ 时,由于 $\hat{q}_u, \hat{p}_u(\hat{\mathbf{Y}}_t)$ 没有除以标准化常数,因此式(11)变为

$$\tilde{\rho}(\hat{\mathbf{Y}}'_t) = \sum_{u=1}^m \min\left(\hat{q}_u, \frac{C \hat{p}_u(\hat{\mathbf{Y}}'_t)}{C'}\right)$$

(5)第 k 帧得到的 $\hat{\mathbf{Y}}_k, s_k, \mathbf{H}_k$ 作为第 $k+1$ 帧的初值继续进行序列图像的目标跟踪。

4 算法复杂度分析

假设各种运算的时间表示如下:一次加(减)法为 T_a , 一次乘法为 T_m , 一次除法运算为 T_d , 一次开方运算为 T_s 。目标位置收敛时的平均迭代次数为 N , 初始模板的像素数为 n , 灰度级数为 m 。

仅以一个尺度为例,比较本文与文献[4]算法的复杂度,如表1所示。

表1 算法复杂度比较

Tab. 1 The comparison of the algorithms' complexity

算法	计算 \hat{q}_u	计算 $\hat{p}_u(\hat{\mathbf{Y}}_t)$	计算 $\tilde{\omega}_i$	计算 \mathbf{Y}'_t	计算相似系数
本文	nmT_a	$NnmT_a$	NmT_d	$N(2nT_m + T_d)$	$m(T_m + T_d + T_a)$
文献[4]	nT_d	$NnmT_a$	$Nm(2T_m + T_d)$	$N(2nT_m + T_d)$	$m(T_m + T_s)$

因为加法的运算时间与乘法相比可以忽略,由表 1 得出本文算法复杂度为

$$(2Nn + m)T_m + (Nm + N + m)T_d \quad (12)$$

文献[4]算法复杂度为

$$(2Nn + 2Nm + m)T_m + (N + Nm + n)T_d + mT_s \quad (13)$$

式(13)减式(12)得 $2NmT_m + (n - m)T_d + mT_s$ 为本文简化的计算量。

在定点 DSPs 处理器中,由于只有专用的乘法器,没有除法器,因此乘法运算时间很短,几条指令就可以完成。但除法运算根据除数和被除数的特点,往往需要几十条指令完成,时间消耗比乘法运算大得多,如果涉及到浮点运算,时间开销会更大,这也是算法说明 2 中尺度系数不取小数的原因。

假设模板大小为 60×40 ,则 $n = 2400$,灰度级 $m = 16$,平均迭代次数取 $N = 4.19^{[3]}$,则本文简化了约 134 次乘法运算,2384 次除法运算,16 次开方运算,这仅仅是在一个尺度下的简化量。总的简化量为上述值的 3 倍,即约 402 次乘法运算,7152 次除法运算,48 次开方运算。

5 优化算法与卡尔曼轨迹预测的结合

根据算法说明 5 可知,优化算法是利用前一帧的目标位置和带宽向量在当前帧开矩形窗口,计算窗口图像的灰度直方图,开始迭代算法。如果当前帧所开的矩形窗口不包含目标信息,算法将失效。算法失效主要源于两种情况:一是当目标运动速度过快,相邻帧目标位置变化大于目标大小时,使用前一帧位置在当前帧所开矩形窗口不再包含目标信息,算法失效;二是目标被全部遮挡,显然矩形窗内也不包含目标信息,算法失效。

本文使用卡尔曼轨迹预测^[10]方法求出目标预测位置,通过比较预测位置与优化算法得到的位置,决策优化算法是否失效,并制定算法失效时采用的跟踪策略。

首先从第 1 帧开始为目标位置建立两个标准的矢量卡尔曼预测器: x 坐标矢量预测器和 y 坐标矢量预测器。以 x 坐标矢量预测器为例,假设第 $k - 1$ 帧的坐标值为 x_{k-1} ,预测第 k 帧的值 x_k ,递归矢量卡尔曼预测方程如下:

预测方程:

$$\hat{S}_k = A\hat{S}_{k-1} + M_k[x_{k-1} - CA\hat{S}_{k-1}] \quad (14)$$

预测修正矩阵:

$$M_k = AV_{k-1}A^T C^T [CAV_{k-1}A^T C^T + R]^{-1} \quad (15)$$

预测均方误差:

$$V_k = [I - M_k C][AV_{k-1}A^T + W] \quad (16)$$

其中, $A = \begin{bmatrix} 1 & T & T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$ 是状态转移矩阵, T 为帧间

时间间隔; $C = [1 \ 0 \ 0]$ 是量测向量; $R = \sigma_r^2$ 是

量测噪声协方差; $W = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_w^2 \end{bmatrix}$ 是系统噪声协

方差矩阵。本文设 $\sigma_r^2 = \sigma_w^2 = 1$ 。

通过目标在 1,2,3 帧的 x 坐标分量 x_1, x_2, x_3 得到矢量估计矩阵

$$\hat{S}_3 = \begin{bmatrix} x_3 \\ [x_3 - x_2]/T \\ [x_3 - 2x_2 + x_1]/T^2 \end{bmatrix}$$

均方误差矩阵

$$V_3 = \begin{bmatrix} \sigma_r^2 & \sigma_r^2/T & \sigma_r^2/T^2 \\ \sigma_r^2/T & 2\sigma_r^2/T^2 & 3\sigma_r^2/T^3 \\ \sigma_r^2/T^2 & 3\sigma_r^2/T^3 & \sigma_w^2 + 6\sigma_r^2/T^4 \end{bmatrix}$$

然后根据预测方程式(14)~式(16)求出 M_4 、 \hat{S}_4 和 V_4 ,则 \hat{S}_4 的第 1 个元素为第 4 帧预测位置的 x 坐标。使用 \hat{S}_4, V_4 作为初值进行第 5 帧的预测。如此迭代计算,可以求出第 k 帧的预测位置,记作 $\tilde{Y}_k = (\tilde{x}_{y_k}, \tilde{y}_{y_k})$ 。

通过将预测位置 $\tilde{Y}_k = (\tilde{x}_{y_k}, \tilde{y}_{y_k})$ 和优化算法求得的位置 $\hat{Y}_k = (x_{y_k}, y_{y_k})$ 相比较,决策真实的目标跟踪位置。流程详述如下:

$$r(k) = |x_{y_k} - \tilde{x}_{y_k}| + |y_{y_k} - \tilde{y}_{y_k}| \quad (17)$$

表示预测位置和算法位置的距离。

假设一个域值 D ,

(1)如果 $r(k) \leq D$,说明算法迭代位置与预测位置相差不大,没有发生前述两种失效状况,采用 \hat{Y}_k 作为下一帧的初值继续进行序列图像跟踪。

(2)如果 $r(k) > D$,表示发生了失效情况,使用预测值 \tilde{Y}_k 作为优化算法的初值进行迭代,得到新的 \hat{Y}'_k ,代入式(17),结果记作 $r'(k)$ 。

①如果 $r'(k) \leq D$, 说明发生的第1种失效。由于帧间目标运动太快, 在前一帧位置开矩形窗口已经不包含当前帧的目标信息, 导致跟踪丢失。在预测目标位置开矩形窗口, 可以包含目标信息, 使得优化算法重新有效, 进而得到新的 \hat{Y}_k 与 \tilde{Y}_k 相差不大。此时以 \hat{Y}_k 作为下一帧的初值继续进行序列图像跟踪。

②如果 $r'(k) > D$, 说明发生第2种失效。由于目标全部遮挡, 即使在预测目标位置开矩形窗口, 也没有包含目标信息, 优化算法仍然无效, 得到的 \hat{Y}_k 与 \tilde{Y}_k 仍然相差很大。此时使用 \tilde{Y}_k 作为下一帧的初值进行序列图像跟踪。

6 实验结果

6.1 跟踪实例

目前, 算法已经成功嵌入 TMS320C6416 硬件平台, 并应用在中国科学院长春光学精密机械与物理研究所研制的小型电视跟踪器上, 进行了大量的外场实验。本文将 RGB 图像转化为灰度图像, 采用 $m = 16$ 灰度级。

图1显示了小车的跟踪效果, 该序列含43帧 720×576 pixels 大小的序列图像。使用恒定跟踪窗口(图中白色方框表示), 跟踪面积为 30×20 的单目标。本文采用改变采样速率的方法描述目标运动过快的情况。由图1可以看出, 在42帧以前, 图像采样速率为每秒5帧, 相邻帧小车中心位置 x 方向变化约9个像素, y 方向变化约7个像素。本文与文献[4]算法在小车发生旋转(图1(b)), 部分遮挡(图1(c))时, 都能够很好地完成跟踪任务。42帧以后, 将图像采样速率变为每秒2帧, 第42帧小车的中心位置为(158, 205), 第43帧为(125, 197), x 方向变化33个像素, y 方向变化8个像素。此时文献[4]使用第42帧的跟踪位置在第43帧图像内开矩形窗口已经不包含小车信息, 跟踪算法失效(见图1(d)右图)。本文在优化算法失效时, 使用卡尔曼滤波预测位置(140, 200)在第43帧图像内开矩形窗口包含了部分小车信息(图1(d)左图中黑色方框), 优化算法重新有效, 迭代到准确的跟踪位置(图1(d)左图中白色方框)。可见, 对于帧间目标位置变化超过目标大小的情况, 本文算法具有更好的适应性。

图2、3显示了大车的跟踪效果。该序列含570帧 720×576 pixels 大小的序列图像, 图像采样速率

为25 fps。使用变尺度跟踪窗口, 初始大小为 60×40 。由图2可以看出, 在43帧(图2(c))以前, 本文和文献[4]算法在大车发生部分遮挡(图2(b))时都能完成跟踪任务。43帧以后, 大车发生全部遮挡(图2(d)), 文献[4]算法失效。本文算法从第43帧开始使用卡尔曼轨迹预测位置, 保证了目标的顺利跟踪, 到第56帧(图2(e))时, 目标出现, 优化算法重新有效, 开始使用优化算法进行跟踪。由图3可以看出, 算法适应了目标发生旋转(图3(a)、(b)), 部分遮挡(图3(c)), 尺度变化(图3(d)), 窗口大小为 48×32 pixels)。综上所述, 本文算法适应了全部遮挡隐藏时间约为13帧的情况, 实现了对570帧序列图像的连续稳定跟踪。

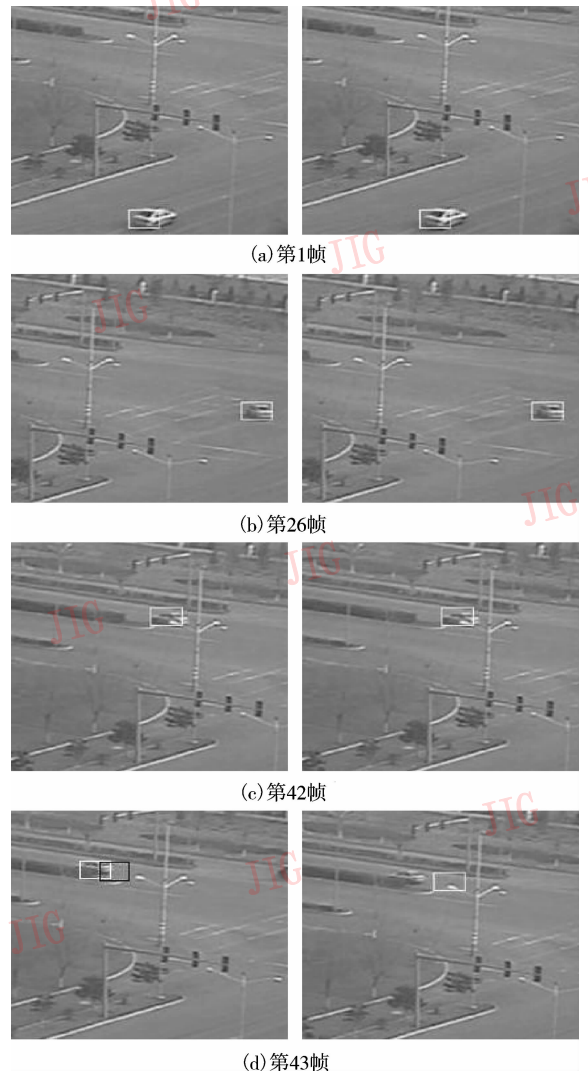


图1 小车跟踪: 本文跟踪结果(左)与文献[4]跟踪结果(右)比较

Fig. 1 Small car tracking, left column: using our algorithm, right column: using the algorithm in [4]

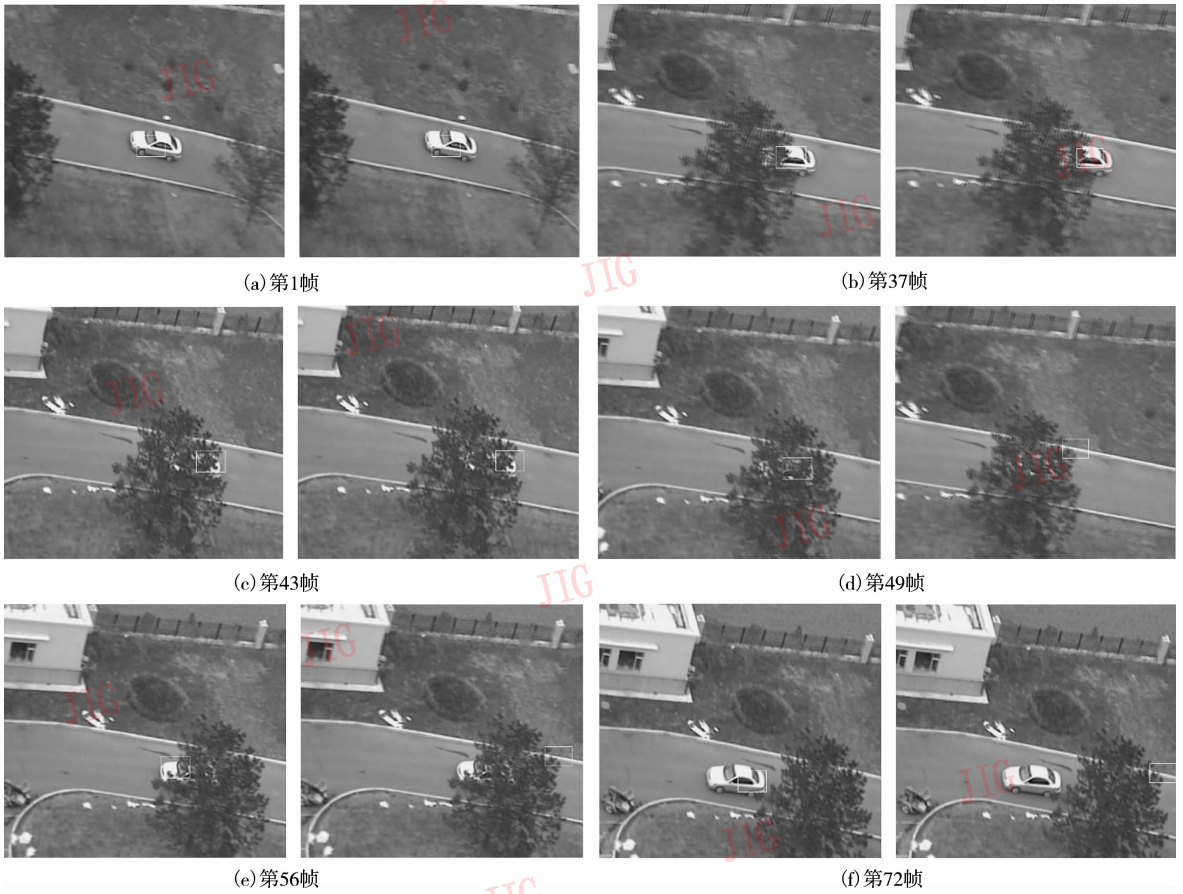


图 2 72 帧前大车跟踪(本文跟踪结果(左)与文献[4]跟踪结果(右)比较)

Fig. 2 Big car tracking in the frames before 72, left column: using our algorithm, right column: using the algorithm in [4]



图 3 72 帧后大车本文算法的跟踪结果

Fig. 3 Big car tracking in the frames after 72: using our algorithm

6.2 性能比较

本文在 32bits 定点, 1GHz 时钟的 TMS320C6416 定点数字信号处理器环境中, 采用双 DSPs 并行运算系统, 一个 DSPs 用于优化算法计算目标位置, 另一个 DSPs 计算卡尔曼轨迹预测位置, 因此卡尔曼轨迹预测的运算时间不计入优化算法内。表 2、表 3 分别比较了本文优化算法与文献[4]算法对于图 1、图 2 所示序列图像的计算时间。

表 2 小车定尺度计算时间比较

Tab. 2 The time comparison of tracking the small car with the invariable scale

算法	目标大小 32 × 20 pixels	
	第一帧计算 \hat{q}_u 时间	其他帧迭代时间
本文	77 μ s	200 μ s ~ 1 ms
文献[4]	973 μ s	500 μ s ~ 2.5 ms

表3 大车变尺度计算时间比较

Tab.3 The time comparison of tracking the big car with the variable scales

算法	目标大小 60 × 40 pixels	
	第1帧计算 \hat{q}_u 时间 (μs)	其他帧迭代时间 (ms)
本文	295	2.5 ~ 12
文献[4]	4 114	20 ~ 75

由表2可以看出,在跟踪窗口不变的情况下,与文献[4]相比,第1帧计算模板直方图 \hat{q}_u 的时间减少了约12倍。即使在背景复杂,发生遮挡等情况下,本文算法最多约用1ms,比文献[4]约减少了2.5倍,并且随着选取模板的变大,简化的效果会更明显。在工程应用中,对于每秒25帧的采样速率,要求在20ms内完成目标跟踪任务及其他图像处理过程,可见本文算法占用时间资源很少,为嵌入更多的图像处理算法奠定了基础。

由表3可以看出,在大目标变尺度跟踪序列中,与文献[4]相比,第1帧计算模板直方图 \hat{q}_u 的时间减少了约14倍。迭代过程中,本文算法最多约用12ms,比文献[4]约减少了6倍,可以在20ms内顺利完成实时跟踪任务。

7 结论

针对现有的均值漂移跟踪算法不能适应目标运动速度过快、全部遮挡以及难于满足实时跟踪要求的缺点,本文在核密度估计,均值漂移算法迭代权值,Bhattacharyya系数等方面进行改进,提出优化算法,同时将算法与卡尔曼轨迹预测相结合,解决了目标运动速度过快和全部遮挡问题。实验结果表明,与现有算法相比,本文算法不仅大大降低了运算量,运算时间满足硬件系统实时跟踪要求,同时在混乱背景、目标发生旋转、遮挡、形状变

化、运动速度过快等情况下,该算法仍能够保持连续稳定的实时跟踪。

参考文献 (References)

- Collins R, Lipton A, Fujiyoshi H, et al. Algorithms for cooperative multisensor surveillance [J]. Proceedings of the IEEE, 2001, **89**(10): 1456 ~ 1477.
- Cheng Y. Mean shift, mode seeking and clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, **17**(8): 790 ~ 799.
- Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, **25**(5): 564 ~ 575.
- Jiang Shu-hong, Zhang Jian-qiu, Hu Bo. An image tracking algorithm based on object center distance-weighting [A]. In: Proceedings of the 12th National Conference on Image and Graphics [C]. Beijing: Tsinghua and Springer Publishing Company, 2005: 364 ~ 367. [江淑红,张建秋,胡波.目标中心距离加权的图像跟踪算法[A].见:第十二届全国图像图形学术会议论文集[C].北京:清华大学出版社与Springer联合出版,2005:364~367.]
- Cheng Guang, Liu Wei-dong, Wei Shang-jun, et al. Methods for estimation and prediction of maneuverable target based on kalman filtering [J]. Computer Emulating, 2006, **23**(1): 8 ~ 10. [成光,刘卫东,魏尚俊等.基于卡尔曼滤波的目标估计和预测方法研究[J].计算机仿真,2006,23(1):8~10.]
- Rosenblatt M. Remarks on some nonparametric estimates of a density function [J]. Annals of Mathematical Statistics, 1956, **27**(6): 832 ~ 837.
- Comaniciu D, Meer P. Mean shift: a robust approach toward feature space analysis [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, **24**(5): 603 ~ 619.
- Li Xiang-ru, Wu Fu-chao, Hu Zhan-yi. Convergence of a mean shift algorithm [J]. Journal of Software, 2005, **16**(3): 365 ~ 374. [李乡儒,吴福朝,胡占义.均值漂移算法的收敛性[J].软件学报,2005,16(3):365~374.]
- Swain M J, Ballard D H. Color indexing [J]. International Journal of Computer Vision, 1991, **7**(1): 11 ~ 32.
- Xu Jian, Duan Zhe-min. Kalman filter in target tracking [J]. Computer Emulating, 2005, **22**(11): 120 ~ 122. [徐剑,段哲民.卡尔曼预测器在目标相关跟踪中的应用[J].计算机仿真,2005,22(11):120~122.]