

彩色序列图像中实时运动目标跟踪方法

林宏基 叶政春 欧静
(福州大学数学与计算机科学学院, 福州 350002)

摘要 提出了一种彩色序列图像中的实时运动目标跟踪算法,该算法首先利用综合帧间差分法与背景差分法两种方法优点的动态背景更新算法来检测各种运动目标,在后续的图像序列中,利用运动检测算法来确定目标跟踪的起始点,并利用 Mean Shift 算法来跟踪运动物体;然后再更新 Mean Shift 的目标模板。实验结果表明,该算法能够克服 Mean Shift 算法对尺度变化的物体的跟踪效果较差且不能检测突然出现在图像序列中的物体的不足,快速准确地跟踪各种物体。

关键词 目标检测 目标跟踪 Mean Shift 算法

中图分类号:TP391.4 文献标识码:A 文章编号:1006-8916(2008)11-2181-06

Real-time Object Tracking in Color Image Sequences

LIN Hong-ji, YE Zheng-chun, OU Jing

(College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350002)

Abstract Real-time object tracking method is proposed in this paper. It provides the ability to track the moving object in color image sequences. Firstly, an improved motion detection method is used to acquire potential regions with moving object regions in the image. Then it tracks the object by combining the Mean Shift algorithm and moving detection algorithm. The experiment results show that the algorithm which can track the moving objects quickly and precisely, and the algorithm is robust to size changes of the moving objects and can track sudden appearance of objects.

Keywords motion detection, object tracking, Mean Shift algorithm

1 引言

运动目标跟踪是智能视频监控系统的-一个重要组成部分,是计算机视觉、图像处理和模式识别领域中具有挑战性的-重要课题。复杂的现场环境以及跟踪应用的实时性,对目标的跟踪算法都提出很高的要求,目前,比较常用的方法包括粒子滤波器^[1]、卡尔曼滤波器^[2]、Mean Shift 算法^[3]等。然而,这些算法普遍存在-一些不足。例如,传统的粒子滤波算法的退化现象,经过多次迭代后,大量粒子只集中了较小的权值^[4],它们对后验概率的估计几乎不起作用,浪费大量的计算时间。卡尔曼滤波器计算量大

大,而且很难设计通用的运动模型描述物体的自由运动,不适合实时运动目标的跟踪。Mean Shift 算法^[3]虽然存在着缺乏必要的模板更新算法,在目标有尺度变化时,可能出现跟踪失败的缺点。但是 Mean Shift 算法^[3]有以下几个优势:首先,算法计算量不大,在后-一帧目标区域与前一帧运动区域距离相差不大的情况下完全可以做到实时跟踪;其次,作为一个无参数估-算法,很容易作为一个模块和别的算法集成;另外,采用核函数直方图建模,对边缘遮挡,目标旋转,变形和背景运动不敏感。

本文采用运动检测和 Mean Shift 算法相结合的方法进行运动目标跟踪,该方法能够较好地克服传统相关算法中存在的计算量大和易受遮挡干扰的缺

基金项目:福建省教育厅基金资助项目(JB05043)

收稿日期:2008-02-28;改回日期:2008-07-15

第一作者简介:林宏基(1949~),男,副教授。主要研究方向为智能视频监控、网络数字媒体、网络视觉。E-mail:lhj057@163.com

点,同时也克服 Mean Shift 算法本身的不足。充分利用运动目标检测的结果,提高目标跟踪时的匹配效率,并采用提前阈值截止技术,缩小目标的搜索范围,实现了对目标的快速准确跟踪。

2 运动检测算法

2.1 常用的运动目标检测法

常用的运动目标检测方法有帧间差分法^[5]、背景减除法^[6,7]。帧间差分法是对一个极小时间间隔前后两帧图像采用基于像素灰度的差分,再用阈值化方法来提取图像中的运动区域,可描述为

$$M(x,y,t) = \begin{cases} 1 & |f(x,y,t) - f(x,y,t - \Delta t)| > \tau \\ 0 & |f(x,y,t) - f(x,y,t - \Delta t)| \leq \tau \end{cases} \quad (1)$$

式中, Δt 为时间间隔, $f(x,y,t)$ 和 $f(x,y,t - \Delta t)$ 分别为时刻 t 和 $t - \Delta t$ 的图像。如果两帧图像对应像素灰度的绝对差值大于阈值 τ (用自适应阈值选取方法选取^[8]),则表明该像素是运动前景,则将其值置 1,否则为静态背景将其值置 0,可得二值图像 $M(x,y,t)$ 。

背景减除法则利用当前图像与预存的固定背景图像的差分,再取阈值来检测运动区域,可描述为

$$M(x,y,t) = \begin{cases} 1 & |f(x,y,t) - f(x,y,0)| > \tau \\ 0 & |f(x,y,t) - f(x,y,0)| \leq \tau \end{cases} \quad (2)$$

$f(x,y,t)$ 、 $f(x,y,0)$ 分别为时刻 t 图像和背景图像。

2.2 改进的运动目标检测法

背景减除法能检出完整的运动区域,但当背景变化时它将不能获得准确的运动区域。帧间差分法受光线变化影响小,但不能完全提取出运动物体所有相关的特征像素点,易产生空洞现象。并且它只检测在帧间相对运动的物体,不能检测出帧间相对静止的物体。这两种方法在性能上恰好互补,可结合两种方法的优点构造一种动态更新背景的模式,具体步骤如下:

(1) 在开始检测前,先预存一帧图像作为初始背景,其中可能包含运动目标。

(2) 按背景减除法做如下操作,把当前帧与背景相减,对结果进行阈值化处理得二值图像。

(3) 实时、动态地更新背景,更新过程中对检测出的前景和背景区域采用不同更新策略:

① 二值图像中的静态背景区域,把背景模块中对应静态背景处像素点的灰度值用当前帧中对应灰度值加权更新。

② 对于二值图像中的运动前景区域,为了兼顾处理运动目标永久移入、移出问题,为每一个像素点都设置一个运动时间累积计数器 T ,如果累积时间超过了设定的时间阈值 T_r 则认为该点处有物体永久移入或移出,此时把背景模板中对应像素点的灰度值直接用当前帧中对应像素点的灰度值来替代更新;否则只是增加 T 读数值,背景模型中对应的值保持不变。设 $f(x,y,0)$ 为背景图像,当前帧为 $f(x,y,t)$, $M(x,y,t)$ 为当前二值图像,若图片中总共有 $N \times M$ 个像素,当前帧每一个点 $f(x_i,y_i,t)$ 的背景动态更新策略可表示如下:

if($M(x_i,y_i,t) = 0$) //判断当前像素点是否为背景

{ $T[i] = 0$; //计数器清零

//动态更新背景, A 为更新系数

$f(x_i,y_i,0) = A * f(x_i,y_i,t) + (1 - A) * f(x_i,y_i,0)$; }

else { $T[i] += \Delta t$;

if($T[i] > T_r$) { $T[i] = 0$; //计数器清零

$f(x_i,y_i,0) = f(x_i,y_i,t)$; } } //更新背景

由实验结果可知,本算法可以得到较完整的运动物体的区域,而且能检测到帧间差分法(如图 1(c)所示)很难检测到的在相邻两帧间相对静止的目标(如图 1(c)右边的男士),而本文的算法(如图 1(d)所示)则可以得到。



图 1 本文提出算法与帧间差分法算法效果对比图

Fig. 1 Contrast experiments

3 传统的 Mean Shift 运动跟踪算法

在起始帧,通过运动目标算法确定一个包含运动目标的候选区域,该区域即为被跟踪的目标区域,这个区域也是核函数的作用区域,区域的大小等于核函数的带宽。假定是在彩色的视频序列图像中进行运动目标的跟踪,因此,图像像素的值是 RGB 颜色空间,按照直方图的方式将 RGB 颜色空间中的每个子空间 R、G、B 分成 K 个相等的区间,每个区间称为一个 bin,构成特征空间,特征空间中特征值的个数(也就是 bin 的个数)为 $m = K^3$ 。

在初始帧中的目标区域内的所有像素点,计算特征空间中的每个特征值的概率,称为目标模型的描述。在以后的每帧图像可能存在目标的候选区域中,对特征空间的每个特征值的计算称为候选目标模型描述。一般核函数选择 Epanechnikov 函数。利用相似性函数度量初始帧目标模型和当前候选帧模型的相似性,通过求相似性函数得到最大的关于目标的 Mean Shift 向量,这个向量即目标从初始位置向正确位置转移的向量,由于 Mean Shift 算法的收敛性。不断迭代计算 Mean Shift 向量,在当前帧中,最终目标会收敛到目标的真实位置(一个静止点),从而达到跟踪的目的。

3.1 目标模型描述

由运动检测算法可检测到若干椭圆的区域,每个区域都代表一个运动物体,即被跟踪的目标区域。目标区域的中心为 \mathbf{x}_0 ,假设其中有个 n 像素用 $\{x_i\}_{i=1,\dots,n}$ 表示,那么对目标模型可以描述为在目标区域上所有特征值的概率值。目标模型的特征值 $u = 1, \dots, m$ (m 最大值为 $m = K^3$ 为特征向量空间中 bin 的个数, u 指第 u 个 bin),特征值估计概率密度

$$\hat{q}_u = C \sum_{i=1}^n k\left(\left\|\frac{\mathbf{x}_0 - x_i}{h}\right\|^2\right) \sigma[b(x_i) - u] \quad (3)$$

式中, $k(x)$ 为核函数的轮廓函数, h 为核函数的带宽,由于遮挡或者背景的影响,目标模型中心附近的像素比外物像素更可靠, $k(x)$ 对中心的像素给一个大的权值,而远离中心的像素一个小的权值。由于所得的目标区域为椭圆,而函数 $k(x)$ 中的 $\|(\mathbf{x}_0 - x_i)/h\|$ 的作用是为消除不同大小的目标计算时的影响,将椭圆表示的目标归一化为一个单位圆。 $\sigma(x)$ 是 Delta 函数。 $\sigma[b(x_i) - u]$ 总的作用是判断目标区域中的像素 x_i 的颜色值是否属于第 u 个

bin,若是则值 1,否则为 0。 C 是一个标准化的常量系数,使得 $\sum_{u=1}^m q_u = 1$, 因此

$$C = 1 / \sum_{i=1}^n k\left(\left\|\frac{\mathbf{x}_0 - x_i}{h}\right\|^2\right) \quad (4)$$

3.2 候选模型描述

运动目标在初始帧以后的每帧中可能包含目标的区域为候选区域,其中心坐标为 \mathbf{y} ,也是核函数的中心坐标,该区域中的像素用 $\{x_i\}_{i=1,\dots,n_h}$ (n_h 为候选区域中像素点的个数)表示对候选区域的描述可称为目标候选模型,候选模型的特征值为 $u = 1, \dots, m$ 的概率估计密度

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - x_i}{h}\right\|^2\right) \sigma[b(x_i) - u] \quad (5)$$

式中, $C_h = 1 / \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{x} - x_i}{h}\right\|^2\right)$ 是标准化的常量系数。

3.3 相似性函数

相似性函数描述目标模型与候选区域之间的相似程度,使用 Bhattacharyya 系数^[4]作为相似性函数,它是一种散度性测量,其直接的几何意义是两个向量间角度的余弦值。Comaniciu 在文献[9]中说明 Mean Shift 算法中 Bhattacharyya 系数优于其他相似性函数。其定义为

$$\hat{\rho}(\mathbf{y}) = \rho(\hat{p}(\mathbf{y}), \hat{q}) = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}) \hat{q}_u} \quad (6)$$

其值在 0 ~ 1 之间。 $\hat{\rho}(\mathbf{y})$ 的值越大,表示两个模型越相似,在当前帧中的不同的候选区域计算得到的候选模型,使得 $\hat{\rho}(\mathbf{y})$ 最大的候选区域即是在本帧中目标的位置。 \hat{q}_u 为目标模型属于第 u 个 bin 的概率, \hat{p}_u 是候选模型属于第 u 个 bin 的概率, $\hat{q} = (\hat{q}_1, \hat{q}_2, \dots, \hat{q}_m)$ 为目标区域的特征向量,而 $\hat{p}(\mathbf{y}) = (\hat{p}_1(\mathbf{y}), \hat{p}_2(\mathbf{y}), \dots, \hat{p}_m(\mathbf{y}))$ 为以 \mathbf{y} 为中心点的为候选区域特征向量。

3.4 目标定位

为使 $\hat{\rho}(\mathbf{y})$ 最大,在当前帧先定位为前一帧中目标中心的位置 $\mathbf{y}_0 = \mathbf{x}_0$,从这一点开始寻找最优匹配的目标,其中心为 \mathbf{y} 。先计算目标候选模型 $\hat{p}(\mathbf{y}_0)$,对上式在 $\hat{p}(\mathbf{y}_0)$ 处进行泰勒展开可得到 Bhattacharyya 系数近似值为

$$\rho(\hat{p}(\mathbf{y}_0), \hat{q}) = \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}_0) \hat{q}_u} +$$

$$\frac{C_h}{2} \sum_{i=1}^{n_h} w_i k(\|(\mathbf{y} - x_i)/h\|^2) \quad (7)$$

式中, $w_i = \sum_{u=1}^m \frac{\sqrt{\hat{q}_u}}{\hat{p}_u(\mathbf{y}_0)} \sigma[b(x_i) - u]$, 式(7)中只有 $\frac{C_h}{2} \sum_{i=1}^{n_h} w_i k(\|(\mathbf{y} - x_i)/h\|^2)$ 项随 \mathbf{y} 变化。所以分析此项, 令

$$f_{n,k} = \sum_{i=1}^{n_h} \frac{C_h}{2} w_i k(\|(\mathbf{y} - x_i)/h\|^2) \quad (8)$$

就是在有权值 w_i 时的核密度估计, 通过计算 Mean Shift 向量, 在每次的 Mean Shift 迭代中, 目标区域的中心位置都会由 \mathbf{y}_0 移动到新位置

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g(\|(\mathbf{y} - x_i)/h\|^2)}{\sum_{i=1}^{n_h} x_i g(\|(\mathbf{y} - x_i)/h\|^2)} \quad (9)$$

其中, 轮廓函数 $g(x) = -k'(x)$, 这样目标区域由初始位置逐步移动到真实的目标位置。

4 综合利用目标检测的跟踪算法

传统的 Mean Shift 算法从当前帧先定位为前一帧中目标中心的位置, 从这一点开始寻找最优匹配的目标, 由于匹配搜索没有充分利用到运动目标检测的结果, 还需要在初始的搜索中心位置附近进行多次模板匹配, 计算复杂度较高, 而且匹配容易陷入局部最优; 如果相邻帧间的物体运动速度较快, 预测得到目标位置和目标的实际位置相差较远时, 将无法或者要经过很长时间才能搜索到目标。本文充分利用运动目标检测的结果, 根据检测出来的运动目标距离预测位置远近的优先顺序, 对各个运动目标进行匹配搜索, 大大减少了计算量。另外, 传统的 Mean Shift 算法缺乏必要的模板更新算法, 当目标有尺度变化时, 可能跟踪失败。为了检测尺度变化的运动目标, 考虑将运动检测到的目标按比例变化将提取到的候选目标与最终目标做比较。具体步骤如图 2 所示。

(1) 在开始跟踪前, 先取一些帧进行运动目标检测, 得到后面要跟踪的运动物体。

(2) 在要跟踪的后续帧中, 利用运动检测算法得到跟踪匹配的候选区域, 计算该候选区域与目标区域的相似度, 并从中选择相似度最大的且大于某一特定阈值 $T1$ (通过实验取 0.8) 的物体, 作为该物体在这一帧中的延续, 如果该相似度值与前一帧得到的相似度

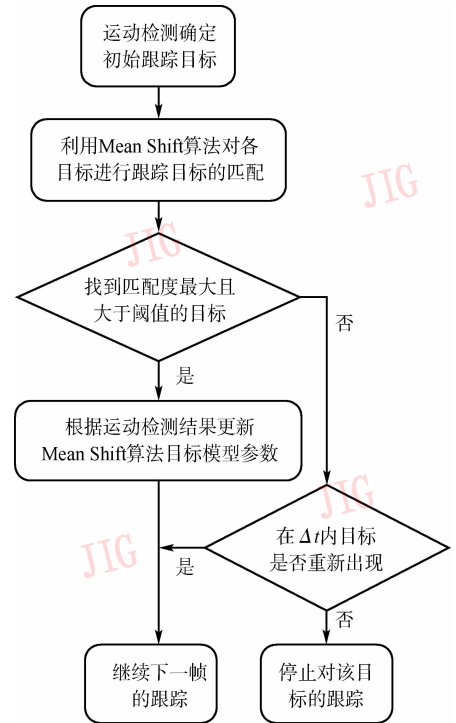


图 2 运动目标跟踪流程图

Fig. 2 Flowchart of object tracking

值差大于阈值 $T2$ (通过实验取 0.2), 可认为该物体已发生变化, 重新更新该物体的目标模型。

(3) 对于运动检测得到的非前一帧延续的运动物体, 可认为它是新出现的运动物体, 将它放入到跟踪物体缓冲池中, 在后面的帧中对它进行跟踪。

(4) 对于一个物体在连续几帧内都没有出现, 如果在随后的 Δt 时间内还没出现, 就认为它已永久移出图像序列中, 将它移出目标跟踪缓冲池, 不再跟踪。

5 实验结果

为了验证算法的有效性, 在 VC++ 6.0 平台下开发了实验程序, 进行目标跟踪。机器的配置为 P4 2.4G, 256M 内存, 采集图像的分辨率为 352×288 的图像。下面是跟踪过程中抽取的有代表性的几帧图像。

实验中, 目标的定位主要使用外接矩形框, 目标的区分使用数字标注。在实验中, 出现 3 个目标。第 25 帧, 第 37 帧中, 目标 1 和 2 相向运动; 在第 45 帧中, 目标 1 离开了场景; 在第 58 帧中, 目标 2 被树枝挡住 (如 58 帧红色圈出部分), 造成目标 2 的短暂消失, 同时又有一个新的目标 3 进入场景; 在第 70 帧, 第 83 帧中, 目标 2 重新出现, 系统继续对目

标 2 进行正确跟踪。实验表明,本文提出的方法与传统的 Mean Shift 算法相比能够检测到各种运动物体,即使在物体发生形状变化或者突然被遮挡消失

一段时间后再出现也能够被很好地跟踪。同时在跟踪过程中,系统平均每 130ms 处理一帧图像,基本达到实时的要求。

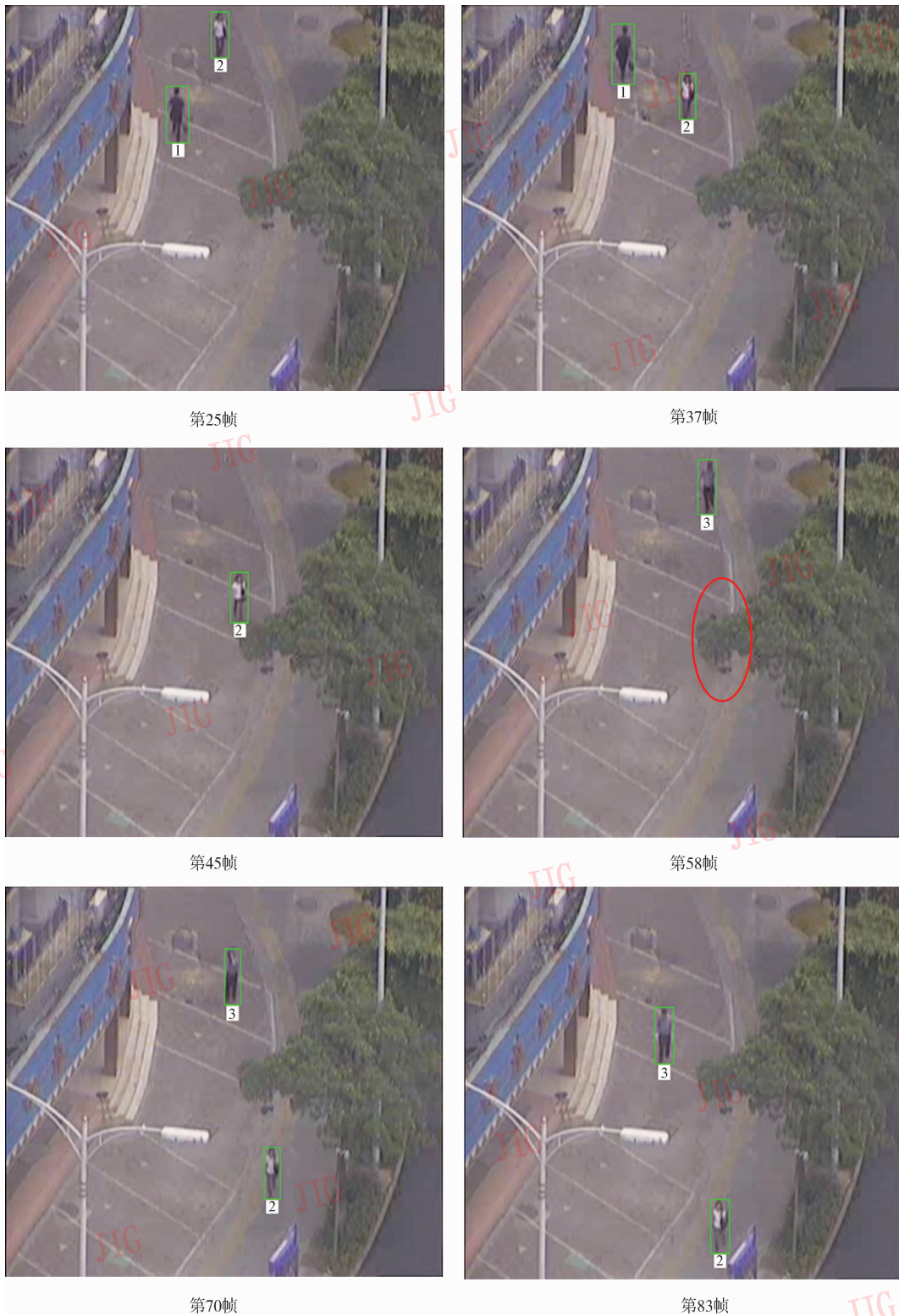


图 3 部分单运动目标跟踪的结果

Fig. 3 Some detection results of single object tracking

6 结 论

本文提出了一种新方法用于描述彩色图像的空间特征和进行相似性获取。无序矩阵反映了图像的空间位置关系,并且它对旋转和平移等位置的相对变化具有不变性。相似性测量算法中既考虑了颜色的分布特性,也考虑到位置因素的影响。和其他方法相比,本文的算法推导简单,对图像没有特殊的限制和预处理要求。实验结果表明,这种方法是有效的,可以在很大程度上提高获取的精确度,并且在位置旋转和平移不变性方面也达到了预期的结果。由于计算量最大的图像库中待测图像直方图的计算可以预先在离线状态下完成,本方法的运算效率较高。在下一步的工作中,准备用更大的图像数据库来测试本文中的算法,并改进无序矩阵的推导方法,更进一步提高整个获取算法的运行效率。

参考文献 (References)

- 1 Doucet A, Gordon N, Krishnamurthy V. Particle filter for state estimation of jump Markov linear systems[J]. IEEE Transactions on Signal Processing, 2001, **49**(3):613 ~ 624.
- 2 Xia Li-min. Image restoration based on particle filter [A]. In: Proceedings of 7th International Conference on Signal Processing[C], Beijing, 2004:1084 ~ 1087.
- 3 Comaniciu D, Ramesh V, Meer P. Real-time tracking of non-rigid objects using Mean Shift [A]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition[C], Hilton Head Island, South Carolina, USA, 2000:142 ~ 149.
- 4 Jiang Bao-an, Lu Huan-zhang. Particle filter for target tracking [J]. Radar Science and Technology, 2003, **1**(3),170 ~ 178. [江宝安, 卢焕章. 粒子滤波器及其在目标跟踪中的应用[J]. 雷达科学与技术, 2003, **1**(3),170 ~ 178.]
- 5 Mech R, Wollborn M. A noise robust method for segmentation of moving objects in video sequences [A]. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing [C], Munich, Germany, 1997: 41 ~ 45.
- 6 Magee D R. Tracking multiple vehicles using foreground background and motion models [J]. Image and Vision Computing, 2004, **22**(2): 143 ~ 155.
- 7 Naoya Ohta. A statistical approach to background subtraction for surveillance systems [A]. In: Proceedings of Eighth IEEE International Conference on Computer Vision[C], Vancouver, BC, Canada, 2001: 481 ~ 486.
- 8 Liang Yu-feng, Wilder Joseph. Real-time face tracking [A]. In: Proceedings of Machine Vision Systems for Inspection and Metrology SPIE[C], Boston, MA, USA, 1998:149 ~ 156.
- 9 Comaniciu. Kernel-based object tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, **25**(2): 564 ~ 577.