

基于均值变换的 Particle Filter 实时跟踪算法

刘志明 韦 巍

(浙江大学 电气工程学院 系统科学与工程系 杭州 310027)

摘 要 提出一种基于均值变换(Mean Shift)的 Particle Filter 图像跟踪算法. 算法将目标的状态空间分解为位移子空间和形变子空间. 使用均值变换算法跟踪位移子空间变化, 获得目标的位置信息. 在此基础上使用 Particle Filter 跟踪形变子空间变化和补偿均值变换的跟踪误差. 由于均值变换算法跟踪的信息使 Particle Filter 跟踪的位移子空间大大缩小, 减少 Particle Filter 所需要的样本数, 使 Particle Filter 的实时性能提高, 而 Particle Filter 获得的形状信息补偿了均值变换算法对于形状跟踪的误差. 该算法比标准的 Particle Filter 算法具有更高的效率, 并拥有均值变换算法所不具备的形状跟踪能力. 实验结果证明算法的有效性和快速性.

关键词 图像跟踪, 均值变换, 粒子滤波, 实时跟踪

中图法分类号 TP391.4

Object Tracking Using Particle Filter Based on Mean Shift

LIU Zhi-Ming, WEI Wei

(Department of System Science and Engineering, Zhejiang University, Hangzhou 310027)

ABSTRACT

A novel particle filter based on mean shift is presented. The standard particle filter is supposed to track a target in a whole state space. The proposed algorithm decomposes the target state space into two independent subspaces: a displacement subspace and a deformation subspace. Mean shift algorithm embedded in the particle filter is utilized to track the state transfer in displacement subspace while the particle filter is used to track the transfer in deformation space. In this way, the particle filter tracks the target in a lower order subspace and thus its real-time performance is improved. On the other hand, the mean shift algorithm is granted a new capability to track the target deformation. The validity and efficiency of the new algorithm are demonstrated by a series of real time tracking experiments.

Key Words Visual Surveillance, Mean Shift, Particle Filter, Realtime Tracking

收稿日期: 2005-01-13; 修回日期: 2006-02-13

作者简介 刘志明, 男, 1980 年生, 硕士研究生, 主要研究方向为机器视觉、人工智能. E-mail: woodstocksn@gmail.com. 韦巍, 男, 1964 年生, 教授, 博士生导师, 主要研究方向为智能机器人、智能控制与智能系统、信息化系统的分析与集成.

1 引言

图像跟踪的手段可以分为自上而下和自下而上两类. D. Comaniciu, V. Ramesh 和 P. Meer 提出的均值变换算法(mean shift)^[1-2]是一种自下而上的图像跟踪方法. 该方法根据图像目标特征函数的归一化梯度信息, 寻找邻域内特征函数的极大值位置, 从而实现跟踪. 该算法的优点是速度快, 算法直观, 实时性强. 但该算法只能追踪图像的位置, 不能追踪图像的大小和旋转等形状的变化, 有较大的缺陷.

Isard 和 Blake 提出的 Condensation 算法^[3-4,9], 又称为 Particle Filter(PF)算法, 是一种自上而下的图像追踪方法. 该方法通过一组描述特征空间的样本和对应的权值来表达目标的状态, 通过这些样本对目标变化过程进行 Monte-Carlo 模拟, 根据样本和特征的符合度进行推理, 从而获得对目标状态跟踪能力. 该方法可以有效抵抗包括局部遮挡、快速运动变化等卡尔曼滤波器难以处理的情况, 有较强的鲁棒性和稳定性. 但是这种方法要得到较准确的结果, 需要较大的推理样本集, 样本集大小和状态空间维度呈指数关系.

综合以上两种算法, 本文对目标的状态空间进行子空间分解, 将目标分解为位移空间和形变空间, 在 PF 框架内引入均值变换算法作为 PF 的位移空间跟踪模型, 从而降低 PF 算法的特征空间维数, 减少 PF 推理所需要样本的数量. 同时利用 PF 补偿均值变换算法无法跟踪图像形变空间所引起的误差. 算法通过赋予 PF 和均值变换不同的图像模型, 充分利用目标的边缘和颜色分布信息. 在同等准确度的要求下, 该算法在时间复杂度上低于标准的 PF 算法. 对形状变化的目标跟踪的准确度上明显优于单纯的均值变换算法.

2 均值变换算法^[1]

均值变换算法本质上是一种梯度下降的局部寻优方法^[1]. 算法目的是对一个 R_d 空间的 n 点集合 $\{x_i\}_{i=1,\dots,n}$ 进行多变量核密度估计, 这个核密度估计在图像上即是对目标进行建模, 当以 k 为核函数, 以 h 为窗口宽度, 则该集合的核密度函数的估计可以表达为

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{x-x_i}{h}\right\|^2\right), \quad (1)$$

记

$$g(x) = -k'(x), \quad (2)$$

$$M_h(x) = \left[\frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x \right]. \quad (3)$$

$M_h(x)$ 实际上是一个由 k 核获得的归一化后的密度梯度估计^[3]. 有了这个梯度信息, 就可以用梯度下降法更新 x , 实现对核密度函数的局部极值跟踪.

3 标准 Particle Filter 算法^[3-4]

该算法将跟踪的目标建模为一组表示状态空间不同参量组合的样本 s 以及样本对应的权重 π , 为了简化计算又引入一个累计概率 c , 于是目标的模型变为 $\{s_i^{(n)}, \pi_i^{(n)}, c_i^{(n)}, n=1, \dots, N\}$ (具体见文献^[3]).

算法通过利用 $t-1$ 时的旧样本集 $\{s_{i-1}^{(n)}, \pi_{i-1}^{(n)}, c_{i-1}^{(n)}, n=1, \dots, N\}$ 构建 t 时的新的样本集 $\{s_i^{(n)}, \pi_i^{(n)}, c_i^{(n)}, n=1, \dots, N\}$, 实现对目标的跟踪.

4 基于均值变换的 PF 算法

4.1 算法分析

PF 通过使用大量样本表达目标状态空间, 并利用这些样本在状态变化过程中的几率来表达目标状态变化的概率, 从而达到跟踪的目的. 这个方法带来的结果是 PF 所需要的样本数将随着目标状态空间维数的增加而指数性增长. PF 为了描述目标复杂的运动, 目标的运动模型不可避免的要复杂化, 而相应的状态空间的维数也会显著增加. 因此如何在不影响 PF 性能的前提下, 降低目标状态空间的维数成为提高 PF 实时性能的关键.

均值变换算法是一种跟踪点集一阶矩的方法. 从式(3) 我们可以推导出基于均值变换算法的梯度下降法:

$$x' = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x^{t-1}-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x^{t-1}-x_i}{h}\right\|^2\right)}, \quad (4)$$

其中 x' 表达了当前目标集的一阶矩, 在对目标估计的窗口不变的前提下, 可以根据式(4) 从前一时刻的数据递推获得当前的一阶矩. 但是该算法没有给出和目标窗口范围相关的变化信息. 因此在窗口范围改变的时候, 均值变换算法的前提无法满足. 针对这个问题, 文献^[1] 中提出使用窗口缩放适应的方法来估算目标窗口范围的变化. 这种方法没有从理

论上解决窗口范围变化的问题,对于各向异性的窗口,它相当于默认窗口的方向不变,而这在实际中往往是不合理的。

针对 PF 和均值变换算法存在的问题,本文提出一种状态空间分解的方法,同时解决 PF 的降维和均值变换窗口范围变化跟踪的问题。基于这个状态空间分解的思想,进一步将 PF 和均值变换算法进行结合,获得一种新的基于均值变换的 PF 图像跟踪算法。

4.2 状态子空间分解

对于目标表达,一种常用的方法是使用状态空间模型。对于非刚性物体跟踪,这个状态空间模型是一个高维模型,Blake 提出基于 b 样条插值的表达方法是一种较为常用的非刚性物体的表达方法。而在大多数跟踪情况下,目标往往可以使用一个抽象的规则图形代替,如椭圆或矩形。这些抽象图形的参数在某时刻中的表达构成目标在该时间点上的状态空间。

考虑一个典型的状态空间 S :

$$S: [x, y, v_x, v_y, \theta, \lambda]^T, \quad (5)$$

其中, x, y, v_x, v_y 分别表示 x, y 方向的位置和速度, θ 表示旋转的角度, λ 表示等比例缩放参量。以上的状态参量是相互正交的。

考虑 S 空间中和位移相关的分量 x, y, v_x, v_y , 由于它们相互正交,以它们为基可以张成一个 S 的子空间——位移子空间 S_{disp} 。同理分量 θ, λ 可以张成另一个子空间 S_{defn} 。

因为 θ, λ 分别和 x, y, v_x, v_y 正交,所以子空间 $S_{\text{defn}}, S_{\text{disp}}$ 相互正交:

$$S_{\text{defn}} \perp S_{\text{disp}}. \quad (6)$$

根据式(6),目标在 x, y, v_x, v_y 构成的子空间 S_{disp} 中的变化不会在 S_{defn} 有投影,就是说在子空间 S_{disp} 和 S_{defn} 分别跟踪的结果不会互相影响。

两个子空间和状态空间的关系可以表达为

$$S_{\text{disp}} \oplus S_{\text{defn}} = S. \quad (7)$$

因此状态空间的状态向量表达可以由两个子空间的状态表达结合而得到。

从物理意义上看,这种划分是有广泛意义的,在图像跟踪中,一个物体的形状变化和它的运动轨迹变化联系在观察上是不显著的,因此可以认为它们之间是相互正交的子空间。对于一个广义的状态空间,同样可以将它分解为相互正交的位移子空间和形变子空间。

4.3 基于均值变换的 PF 算法

由于子空间的维数显然少于整个状态空间的维

数,因此对于 PF,由跟踪整个状态空间转变为跟踪状态空间的某个子空间,降低目标空间的维数,减少 PF 算法的样本数。对于形变子空间 S_{defn} ,由于它的变化容易受背景噪声和特征缺失的影响,因此在形变空间中的变化表现出高度的非线性。使用 PF 跟踪这个子空间的变化是一个合理的选择。

对于位移子空间 S_{disp} ,目标在空间上的变化所引起的可以观察的特征变化就是目标位置的变化。这种变化是对象点集的一阶矩变化。在满足帧间运动不过大的前提下,使用式(4)就可以跟踪目标在位移子空间的变化。而均值变换无法解决的窗口范围变化出现在形变子空间中,在时刻 t 中对位移空间没有影响。PF 在形变子空间中对目标跟踪的结果提供了形变子空间中变化的完整信息。利用这个信息,可以补偿均值变换算法由于点集二阶矩变化引起的跟踪误差,同时使均值变换算法获得跟踪目标形状变化的能力。

基于式(7),两个子空间的跟踪结果结合在一起就获得了整个状态空间的描述。这种方法就是基于均值变换的 PF 算法。下面是算法的基本步骤:

step1 划分目标的状态空间为位移子空间和形变子空间。

step2 对位移子空间进行均值变换跟踪。

step3 在位移子空间跟踪的结果处生成样本,使用 PF 算法跟踪目标的形变子空间。

step4 利用 PF 跟踪的结果补偿均值变换的误差。

根据 4.2 节的推导,对于形变子空间和位移子空间的跟踪是相互独立的。但在实际应用中,上述算法的 step2 使用均值变换进行位移跟踪的结果往往存在一定的误差。为了补偿这种误差对跟踪结果的影响,我们采用在均值变换跟踪结果的邻域内使用 PF 算法随机生成样本跟踪目标的方法。这种方法虽然在理论上增加 PF 算法跟踪的维数,但是由于均值变换结果的邻域取值范围相当有限。即使对于动作速度较快的目标,由于均值变换的跟踪结果已经非常接近目标的实际位置,因此 PF 只需要在位移空间的一个较小的子空间中进行搜索就可以满足跟踪的要求。因此即便使用这种补偿,新的算法也比标准的 PF 算法节约大部分跟踪位移空间的样本数目。

因此算法的 step3 描述变成:在位移子空间跟踪结果的邻域内生成样本,使用 PF 算法跟踪目标的形变空间。

从另一个角度考虑这个算法。算法相当于使用均值变换对整个搜索空间赋予不同的权重,根据空

间的权重分布来决定不同位置产生样本的多少. 权重较大的区域(均值变换的结果附近)生成较多的样本, 权重较小的区域(远离均值变换结果区域)生成较少的样本, 从而提高有效样本的比例. 这种方法在文献[4]和文献[5]中称为加权采样. 加权采样由于加入先验的知识, 减少模型的不确定性, 提高有效样本比例, 从而减少 Monte - Carlo 模拟需要的样本量. 在这个角度上理解, 这个算法类似于 ICondensation 方法^[4]. 但是 ICondensation 依赖的先验知识是固定和静态的, 是在跟踪框架以外进行. 而我们使用的算法给定的位置空间是通过内嵌于 PF 算法的均值变换实时更新, 并同时使用 PF 对其进行补偿. 这个方法可以称为动态的 ICondensation 算法.

4.4 图像建模

均值变换和 PF 都是通用算法, 可以使用多种图像模型. 本文将跟踪目标建模为一个椭圆区域, 因此状态空间表达为 $[x, y, axe1, axe2, \theta]$, 其中 $[x, y]$ 表示椭圆的中心位置, 它们构成状态空间的位移子空间, $[axe1, axe2, \theta]$ 分别表示长短半轴的长度和椭圆旋转过的角度, 它们张成状态空间的形变子空间.

本文分别使用颜色直方图模型和边缘轮廓模型作为均值变换和 PF 的模型. 从而实现两种图像模型的信息交互, 增加模型抗干扰的能力.

4.4.1 颜色直方图模型

针对图像跟踪问题, 使用类似于文献[1]的方法, 利用 HSV 颜色空间的颜色直方图分别对图像目标模板和候选目标建模, 候选和模板之间的距离使用 Battachaya 距离度量. 从而构建出对象目标的颜色直方图模型.

通过颜色直方图模型, 我们可以沿用文献[1]中使用的基于颜色直方图的均值变换模型.

4.4.2 边缘轮廓模型

根据 Blake 提出的边缘轮廓法线模型^[3,6,8], 构建候选目标轮廓上多个特征点 x 的法线, 在法线上进行一维搜索, 寻找边缘点. 假设背景噪声点的分布沿法线方向符合 Poisson 分布, 那么目标的真实量度将无偏地正态分布于特征点附近. 根据在各法线上得到的所有边缘点位置标量 $z = (z_1, \dots, z_M)$ 就可以得到候选轮廓的观察概率. 获得目标的观察模型如式(8), 其中 x 是表示在当前样本的状态参量构成的椭圆中的一组特征点集. 特征点集的选取一般使用等间隔的选取, 即在椭圆上等间隔的取点作为特征点:

$$p(z | x) \propto 1 + \frac{1}{\sqrt{2\pi\sigma q\lambda}} \sum_m \exp\left(-\frac{(z_m - x)^2}{2\sigma^2}\right). \quad (8)$$

4.4.3 基于边缘轮廓的 PF 算法

本文采用 Blake 在文献[3]中使用的算法. 即 CONDENSATION 算法. 算法用样本集 $\{s_i^{(n)}, \pi_i^{(n)}, c_i^{(n)}, n = 1, \dots, N\}$ 描述目标模型. 迭代过程主要分为 3 步:

1) 选择. 生成一个随机数 $r \in [0, 1]$, 用二分查找法寻找一个最小的 j 满足 $c_j^{(n)} \geq r$, 使 $S_i'^{(n)} = S_{j-1}^{(n)}$.

2) 预测. 本文使用一阶的回归模型:

$$S_i^{(n)} = \mathbf{A} S_i'^{(n)} + \mathbf{B} w_i^{(n)}, \quad (9)$$

其中, \mathbf{A} 在本文中是根据均值变换获得的位移矩阵, $w_i^{(n)}$ 是一组标准正态分布的随机量, $\mathbf{B}\mathbf{B}^T$ 就是一个可以调整的噪声矩阵.

3) 测量. 根据式(8)计算在新位置的观察概率:

$$\pi_i^{(n)} = p(z_i | x_i = S_i^{(n)}),$$

归一化使 $\sum_n \pi_i^{(n)} = 1$, 计算累加概率:

$$c_i^{(0)} = 0,$$

$$c_i^{(n)} = c_i^{(n-1)} + \pi_i^{(n)}, n = 1, \dots, N.$$

当 N 个样本构造完毕就完成了迭代.

4.5 实用算法

根据以上的模型, 就可以构建出均值变换 PF 的实际应用算法.

算法利用 $t-1$ 时的旧样本集 $\{s_{i-1}^{(n)}, \pi_{i-1}^{(n)}, c_{i-1}^{(n)}, n = 1, \dots, N\}$ 和目标位置估计 L_{t-1} 构建 t 时的新样本集 $\{s_i^{(n)}, \pi_i^{(n)}, c_i^{(n)}, n = 1, \dots, N\}$ 和目标位置估计 L_t .

第 1 帧时, 通过手动或其他目标检测方法获得目标起始位置, 构造目标的颜色直方图模板模型.

对于其他帧基本处理步骤如下:

step1 根据颜色模型进行均值变换跟踪, 获得在当前帧的目标起始位置预测 L'_t , 得到 $A = L'_t L_{t-1}^{-1}$.

step2 将 A 代入式(9)进行基于边缘轮廓的 PF 算法, 从旧样本集 $\{s_{i-1}^{(n)}, \pi_{i-1}^{(n)}, c_{i-1}^{(n)}, n = 1, \dots, N\}$ 中推理出该时刻的新样本集 $\{s_i^{(n)}, \pi_i^{(n)}, c_i^{(n)}, n = 1, \dots, N\}$.

step3 从新样本集中估计当前帧的目标位置 L_t .

5 实验设计和结果

算法在 P4 2.4G 256Mb 的 PC 上实现, 采用 Hauppauge bt878 图像卡实时采集数据进行实时的

图像跟踪. 图像区域大小为 320×240 , 目标的大小从 60×60 到 100×100 不等. 算法速度和目标区域大小没有显著关系. 在使用 500 个样本 (particle) 的条件下, 算法稳定运行在 25 帧/秒. 在起始时, 由人手决定跟踪的起始目标椭圆区域的位置和大小, 然后由算法完成自动跟踪. 图 1~图 4 分别选取图像跟踪过程中包括手部、头部等快速运动跟踪和局部遮挡、不规则变换等情况下的跟踪, 图 1 和图 2 同时也给出了使用单一均值变换算法和 PF 算法结果作为和新算法的比较.



图 1 手部跟踪

Fig. 1 Results of tracking hand



图 2 头部跟踪

Fig. 2 Results of tracking head

图 1 的上部表示使用单一的均值变换算法跟踪的结果. 其下部中椭圆表示使用均值变换的 PF 算法跟踪结果, 该算法使用 1 000 个样本 (particle).

从图 1 结果可以看出, 单一均值变换算法对位置的跟踪比较准确, 然而当目标大小发生变化时就无法准确跟踪. 使用基于均值变换的 PF 算法, 无论是对于手势的旋转或者是对于手掌的开合 (目标区域的大小不规则变化) 都可以稳定地进行跟踪.

图 2 上半部图像中椭圆代表使用标准 PF 算法跟踪结果, PF 采用了边缘轮廓模型, 采用一阶的 AR 模型作为运动模型, 下部图像中椭圆表示使用基于均值变换的 PF 算法跟踪. 两种算法均使用了 1000 个样本 (particle).

从图 2 的结果比较, 可以发现标准 PF 算法在头部晃动较快的时候, 会出现无法跟上目标变化的

情况. 这种情况在运动方向切换的位置 (由向左运动变成向右运动) 的过程最为明显, 原因是运动模型不能充分表达运动的往复性, 而当样本的数量无法表达整个状态空间的大范围变化时, 运动跟踪的误差变大. 新算法由于使用均值变换, 已经稳定跟踪目标的位置, 而且 PF 需要表达的目标状态子空间小, 因此使用同样的样本量已经成功跟踪目标.

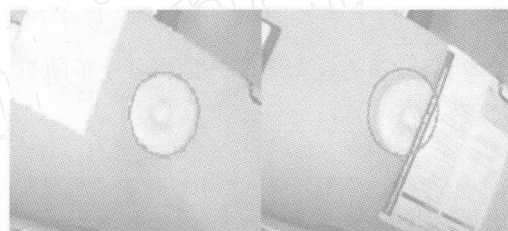


图 3 局部遮挡情况的跟踪

Fig. 3 Results of tracking with partial occlusions

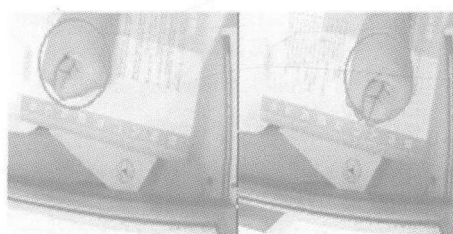


图 4 在复杂背景下的跟踪

Fig. 4 Results of tracking in complex background

图 3 和图 4 分别展示本文算法在较差的跟踪条件下的鲁棒性. 由于新算法结合颜色和边缘两类特征, 当其中一种特征不明显或者缺失时, 另一种特征仍然提供足够的信息支持跟踪. 在局部遮挡条件下, 颜色的分布信息受到影响, 但是利用局部的边缘信息, 算法还可以获得较好的效果. 对于复杂背景, 边缘信息受到背景噪声的干扰, 颜色信息提供了和环境区分的良好根据, 因此算法在这些情况下有较强的鲁棒性和稳定性.

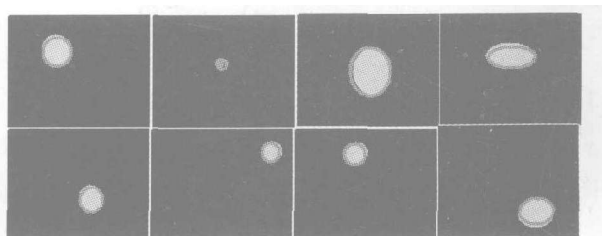


图 5 虚拟运动跟踪

Fig. 5 Results of tracking virtual movement

图 5 展示一个虚拟的跟踪测试. 实心的椭圆表示大小和轨迹随机运动的目标, 空心的椭圆表示使用基于均值变换的 PF 算法跟踪的结果. 图中小球的大小和轨迹根据式(10)给出:

$$\begin{aligned} x &= asint + b + rand_1(t), \\ y &= acost + c + rand_2(t), \\ are1_t &= are1_{t-1} + rand_3(t), \\ are2_t &= are2_{t-1} + rand_4(t), \\ \theta_t &= \theta_{t-1} + rand_5(t), \end{aligned} \quad (10)$$

其中 $f_i(t) = rand_i(t)$ 表示产生平均分布随机量函数.

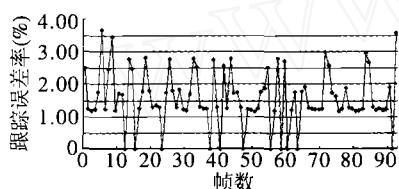


图 6 位置跟踪误差

Fig. 6 Position tracking error

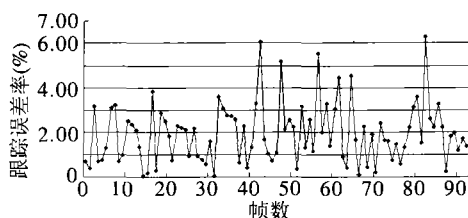


图 7 形状跟踪误差

Fig. 7 Shape tracking error

使用下式度量跟踪的位置误差 E_{position} 和形状误差 E_{Shape} :

$$\begin{aligned} E_{\text{position}} &= \frac{\sqrt{(x' - x)^2 + (y' - y)^2}}{\sqrt{x^2 + y^2}}, \\ f_{\text{proj}}(a, b, c) &= |a \sin c| + |b \cos c|, \\ E_{\text{Shape}} &= \left| \frac{f_{\text{proj}}(are1, are2, \theta) - f_{\text{proj}}(are1', are2', \theta')}{f_{\text{proj}}(are1, are2, \theta)} \right|. \end{aligned} \quad (11)$$

其中 $(x, y, are1, are2, \theta)$ 表示目标的实际参数, $(x', y', are1', are2', \theta')$ 表示算法获得的跟踪结果参数. 图 6 和图 7 给出跟踪的位置误差和形状误差结果.

6 结束语

实验结果证明新算法的快速性和准确性. 在实

验中, 我们发现算法速度对目标区域大小并不敏感, 当目标增大到 100×100 像素的情况下, 图像跟踪的速度仍然可以保持在 25fps. 同时由于该实验不依赖于光流等运动信息, 因此这个算法可以应用在摄像机运动的场合, 如机器人漫游等情况. 该算法的缺点和一般的跟踪算法类似, 主要是启动过程需要手动设置, 因此下一步的工作是加入适当的目标检测方法实现整个过程的自动化.

参考文献

- [1] Comaniciu D, Ramesh V, Meer P. Real-Time Tracking of Non-Rigid Objects Using Mean Shift // Proc of the IEEE Conference on Computer Vision and Pattern Recognition. Hilton Head, USA, 2000. I: 142-149
- [2] Zivkovic Z, Kröse B. An EM-Like Algorithm for Color-Histogram-Based Object Tracking // Proc of the International Conference on Computer Vision and Pattern Recognition. Washington, USA, 2004. I: 798-803
- [3] Isard M, Blake A. Contour Tracking by Stochastic Propagation of Conditional Density // Proc of the 3th European Conference on Computer Vision. Cambridge, UK, 1996. I: 343-356
- [4] Isard M, Blake A. Icondensation: Unifying Low-Level and High-Level Tracking in a Stochastic Framework // Proc of the 5th European Conference on Computer Vision. Freiburg, Germany, 1998. I: 893-908
- [5] Wu Y, Huang T S. A Co-Inference Approach to Robust Visual Tracking // Proc of the 8th IEEE International Conference on Computer Vision. Vancouver, Canada, 2001. II: 26-33
- [6] Wu Y, Yu T, Hua G. Tracking Appearance with Occlusions // Proc of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Madison, USA, 2003. I: 789-795
- [7] Li P H, Zhang T W, Pece A C E. Visual Contour Tracking Based on Particle Filters. Image and Vision Computing, 2003. 21(1): 111-123
- [8] Wu Y, Hua G, Yu T. Switching Observation Models for Contour Tracking in Clutter // Proc of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Madison, USA, 2003. I: 295-302
- [9] Nummiaro K, Koller-Meier E, van Gool L. An Adaptive Color-Based Particle Filter. Image and Vision Computing, 2003. 21(1): 99-110
- [10] Nait-Charif H, McKenna S J. Tracking Poorly Modelled Motion Using Particle Filters with Iterated Likelihood Weighting // Proc of the Asian Conference on Computer Vision. Jeju Island, Korea, 2004: 156-161
- [11] Birchfield S. Elliptical Head Tracking Using Intensity Gradients and Color Histograms // Proc of the IEEE Conference on Computer Vision and Pattern Recognition. Santa Barbara, USA, 1998: 232-237