

文章编号: 1002-1582(2008)03-0404-04

基于显著性加权的 Mean Shift 跟踪方法^{*}

张恒, 李由, 李立春, 于起峰

(国防科技大学 航天与材料工程学院, 长沙 410073)

摘 要: 复杂背景下运动目标跟踪一直是计算机视觉的重要研究方向之一。针对这类问题, 提出了一种基于显著性加权的 Mean Shift 跟踪算法。首先根据目标上不同区域和背景的差别大小, 给目标每个区域赋予不同的权值。然后将权值与 Mean Shift 算法结合起来对目标进行跟踪。实验表明, 在复杂背景下, 新算法仍然可以有效、准确地跟踪运动目标。

关 键 词: 运动目标跟踪; 显著性加权; Mean Shift

中图分类号: TP391 文献标识码: A

A new significance weighted mean shift tracking method

ZHANG Heng, LI You, LI Li-chun, YU Qi-feng

(College of Aerospace and Materials Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: To track the moving target in the cluttering background is one of the most important research fields in computer vision. A new significance weighted mean shift tracking method is proposed. This new tracking algorithm sets the different weights to every parts of the target according to the variance between the target and the background, and tracking the target by the significance weighted mean shift algorithm (SWMS). Experimental results show that the new method can effectively and accurately track moving target in the cluttering background.

Key words: moving target track; significance weighted; mean shift

0 引 言

实时目标跟踪是视频监控、基于对象的视频压缩、巡航导弹末制导等计算机视觉应用中的关键技术。复杂背景下对指定运动目标进行稳健、有效地跟踪, 又是此关键技术中的一个极具挑战性的课题。近些年, 国内外学者提出了许多具有实际意义的运动目标跟踪算法。当像机静止不动时, 背景自适应减法^[1]可以有效地跟踪运动目标。Mean Shift 算法^[2]以其不需要参数, 不需要进行穷尽搜索等特性, 已经成功地应用在对实时性要求较高的目标跟踪领域中。扩展粒子滤波^[3]通过对多重假设的状态空间进行判断, 解决了目标部分遮挡的跟踪问题。但是当背景比较复杂时, 这些算法的跟踪效果并不理想。

本文在分析目标与背景差异和 Mean shift 跟踪算法的基础上, 提出了一种基于目标区域显著性加权的 Mean Shift 跟踪算法。目标中某些区域和背景差异越大, 说明其显著性越强, 这些区域对于跟踪算法的有效性、稳定性贡献也就越大。因此应对显著性较大的区域赋予较大的权值。新算法首先计算初始帧中选定目标的显著性权值。其次, 将颜色直方图每 d 个相邻颜色划分为同一颜色栅格, 从而得到 m 个直方图栅格。利用初始帧目标的显著性权值

计算颜色直方图栅格权值。最后, 利用显著性加权 Mean Shift 算法对选定目标进行跟踪 (Significance Weighted Mean Shift Tracking Method, 以下简称 SWMS)。由于短时间内目标的颜色特性变化不大, 所以跟踪过程中可以设定颜色直方图栅格权值不变。实验表明, 新跟踪算法可以稳健、有效地跟踪复杂背景下的运动目标。

1 初始帧中目标区域显著性权值的确定方法

1.1 选择目标内外区域

以指定目标为中心选取内外两个矩形区域, 分别称为内区域和外围区域。如图 1 所示, 包含目标的小矩形为内区域, 而大矩形和小矩形之间的区域为外围区域。

1.2 确定聚类数目及初始聚类中心

对内区域中每个像素点建立五维特征向量 $v = (I, j, r, g, b)$, 其中 i, j 为该像素点的纵、横坐标, r, g, b 为该点的红、绿、蓝颜色分量。图像点集 $S = \{v_1, v_2, v_3, \dots, v_n\}$ 中任意 v_i 一点的邻域定义以 v_i 为中心, r 为半径的超球, r 为预先选取的阈值, 则 v_i 邻域内的数据点密度定义为

* 收稿日期: 2007-05-30; 收到修改稿日期: 2007-08-02

E-mail: zhh-nudt@yahoo.com.cn

作者简介: 张恒(1979-), 男, 辽宁人, 长沙国防科技大学博士研究生, 从事运动目标跟踪研究。

$$D_i(v_i) = \sum_{j=1}^N u(r - \|v_i - v_j\|) \tag{1}$$

其中 $u(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$; N 为选取区域中像素点的总数。如果数据点 v_p 具有最大的分布密度, 即

$$D(v_p) = \max_{i=1}^N D(v_i), \text{ 那么第一个聚类中心为}$$
$$Z_1 = \sum_{v_j \in S_1} v_j D(v_j) / [\sum_{v_j \in S_1} D(v_j)] \tag{2}$$

其中, S_1 为 v_p 邻域内的数据点集。找出第一个聚类中心后, 接着在集合 $\{S - S_1\}$ 里查找具最大密度函数值的数据点, 并将其邻域内的数据点的质心作为第二个聚类中心 Z_2 。重复这样的过程, 直至 $\{S - \sum_k S_k\}$ 为空集, 那么 k 为选取的目标区域的聚类数目, Z_i (其中 $i = 1, 2, 3, \dots, k$) 为初始聚类中心。

1.3 C-Mean 聚类^[4]及区域分割

根据初始聚类数目及聚类中心, 利用 C-Mean 聚类算法分别对内区域和外围区域进行聚类。利用形态学闭运算^[5]填充内区域聚类结果 S_i ($i = 1, \dots, k$) 中的细小“空洞”。利用序惯算法^[6]将空间不连通的同一聚类区域分割为不同区域, 最终形成的区域聚类结果记为 C_i ($i = 1, \dots, n$), n 为分割后的区域数目。

1.4 确定内区域中背景部分

假设分割好的区域如图 1 所示, 小正方形内部表示内区域, 而大小两个正方形之间的区域为外围区域。对经过 2.3 节分割后的内区域 (如区域 1 ~ 区域 7), 判断其内每个区域是否包含内区域矩形边缘。如果包含内区域矩形边缘, 则进行如下处理 (以区域 7 为例进行说明):

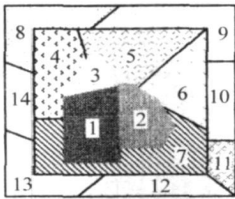


图 1 分割后内外区域示意图

(1) 判断区域 7 和外围区域中哪些区域相邻接, 这里为 10、11、12、13 号区域;

(2) 分别计算区域 7、10、11、12、13 的红、绿、蓝三种颜色的均值 M_r, M_g, M_b , 按公式 (3) 判断出区域 7 和这四个区域中哪个区域最为接近。
$$T = \min(|M_{r_7} - M_{r_i}| + |M_{g_7} - M_{g_i}| + |M_{b_7} - M_{b_i}|) \tag{3}$$
$$i = 10, 11, 12, 13$$

(3) 如果 T 大于预先指定的阈值, 则认为区域 7 应属于背景部分。

对内区域中所有分割区域进行判断, 将所有属于背景部分的区域划分为 C_{bg} 区域, 其权值 $W(x_i)$ 设为 0, x_i 表示内区域像素点坐标。

1.5 计算显著性权值

为了部分抑制光照变化的影响, 按公式 (5) 分别

计算内矩形区域中背景部分 C_{bg} 和每个目标分割区域 C_i ($i = 1, \dots, n_1$) 三个颜色分量的均值, 分别记为 $\overline{r_{C_{bg}}}, \overline{g_{C_{bg}}}, \overline{b_{C_{bg}}}$ 和 $\overline{r_{C_i}}, \overline{g_{C_i}}, \overline{b_{C_i}}$, n_1 为内区域中属于目标的区域总数。按公式 (4) 计算出每个目标分割区域 C_i 的权值 $W_q(x_i)$, x_i 表示内区域像素点坐标。

$$W_q(x_i) = 1 - \frac{1}{1 + |\overline{r_{C_i}} - \overline{r_{C_{lg}}}| + |\overline{g_{C_i}} - \overline{g_{C_{lg}}}| + |\overline{b_{C_i}} - \overline{b_{C_{lg}}}|} \tag{4}$$

其中

$$\left. \begin{aligned} \overline{r_C} &= \sum_{i \in C} \frac{r_i}{r_i + g_i + b_i} \\ \overline{g_C} &= \sum_{i \in C} \frac{g_i}{r_i + g_i + b_i} \\ \overline{b_C} &= \sum_{i \in C} \frac{b_i}{r_i + g_i + b_i} \end{aligned} \right\} \tag{5}$$

2 确定颜色直方图栅格权值

将内区域的颜色直方图每 d 个相邻的颜色划分为同一栅格, 从而得到 m 个直方图栅格, 其中 $d \times m = 256^3$ 。对于初始帧的内区域, 按公式 (6) 计算每个颜色直方图栅格的权值 $W_p(u)$ ($u = 1, \dots, m$), 该权值仅与颜色直方图栅格参数 u 有关, 而区域内像素坐标无关。其中函数 $b: x_i \rightarrow \{1, \dots, m\}$ 将坐标为 x_i 的像素映射到相应的颜色直方图栅格中, 而

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$$
$$W_p(u) = \frac{\sum_{i=1}^N \{W_q(x_i) \cdot \delta[b(x_i) - u]\}}{\sum_{i=1}^N \delta[b(x_i) - u]} \tag{6}$$
$$u \in \{1, \dots, m\}$$

3 SWMS 算法

3.1 计算初始帧的目标模板

$\{x_i\} i = 1 \dots, N$ 为目标窗口中第个像素的坐标, N 为目标窗口的像素点总数, 其中目标窗口中心为 x^* , m 为颜色直方图栅格总数。因此目标颜色直方图栅格特征向量中第 u 个分量为

$$q_u = C \sum_{i=1}^N k(\|x_i - x^*\|^2) W_q(x_i) \delta[b(x_i) - u] \tag{7}$$
$$u \in \{1, \dots, m\}$$

式中 $k(\|x\|^2)$ 是核函数 (通常为高斯核函数); $W_q(x_i)$ 为坐标 x_i 处像素的目标区域颜色显著性权值, 计算方法如 1.5 节。而 C 为归一化常数, 定义为

$$C = \frac{1}{\sum_{i=1}^N [k(\|x_i - x^*\|^2) W_q(x_i)]} \tag{8}$$

3.2 计算当前帧候选特征向量

$\{x_i\}_{i=1, \dots, N_h}$ 为当前帧中搜索窗口中第 i 个像素的坐标, N_h 为搜索窗口中像素总数, 其中搜索窗口中心为 y_0 , 类似(7)式, 计算当前帧中搜索窗口的颜色直方图栅格特征向量的第 u 个分量为

$$p_u(y_0) = C_h \sum_{i=1}^{N_h} k \left[\frac{\|x_i - y_0\|^2}{h} \right] W_p(u) \delta [b(x_i) - u]$$
$$u \in \{1, \dots, m\} \tag{9}$$

式中 x_i 为当前帧搜索窗口中第 i 个像素; $W(u)$ 为当前帧中第 u 个直方图栅格对应的权值; h 为核函数的带宽. 归一化常数 C_h 的计算方法类似于 C

$$C = \frac{1}{\sum_{i=1}^N \left[k \left[\frac{\|x_i - x^*\|^2}{h} \right] W_p(u) \right]} \tag{10}$$

3.3 相似性函数

用 Bhattacharyya 系数描述初始帧目标模板和当前帧候选向量的相似程度, 其定义为

$$\rho(y) \equiv \rho[p(y), q] = \sum_{u=1}^m \sqrt{q_u \circ p_u(y)} \tag{11}$$

式中 y 表示当前帧候选模板的中心. 式(11)的几何意义为 m 维单位向量 $(\sqrt{p_1}, \dots, \sqrt{p_m})$ 和 $(\sqrt{q_1}, \dots, \sqrt{q_m})$ 之间夹角的余弦值. 所以当两个向量夹角为 0 时, 两个向量最为相似.

3.4 当前帧跟踪结果求解

为了使(11)式最大, 在当前帧中, 以前一帧跟踪结果作为当前帧搜索窗口的位置, 设窗口中心为 y_0 , 在 y_0 邻域内寻找最优目标位置 y_1 , 反复迭代, 最后得到在当前帧目标的最优位置. 对(11)式在 $p_u(y_0)$ 处进行泰勒展开, 相似性函数可以近似表示为

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{q_u \circ p_u(y_0)} + \frac{C_h}{2} \sum_{u=1}^m \omega_u k \left[\frac{\|y - x_i\|^2}{h} \right] \tag{12}$$

其中

$$\omega_i = \sum_{u=1}^m \delta [b(x_i) - u] \sqrt{q_u / p_u(y_0)} \tag{13}$$

通过对相似性函数求最大值, 根据 Mean shift 向量迭代得到当前帧目标的新位置

$$y_{j+1} = \frac{\sum_{i=1}^{N_h} x_i \omega_i g \left[\frac{\|y - x_i\|^2}{h} \right]}{\sum_{i=1}^{N_h} \omega_i g \left[\frac{\|y - x_i\|^2}{h} \right]} \tag{14}$$

其中算法的收敛性将在附录中给出.

4 算法大致步骤

(1) 对第一帧图像选择要跟踪的目标, 计算相

应的显著性权值及其颜色直方图权值;

(2) 对后序帧, 利用 SWMS 算法对目标进行跟踪.

5 实验结果

5.1 权值计算结果

图 2 为复杂背景下飞机目标权值计算示意图. 图 2(a)为原始图像, 其中小矩形表示包含目标的内区域, 大矩形和小矩形之间表示外围区域. 图 2

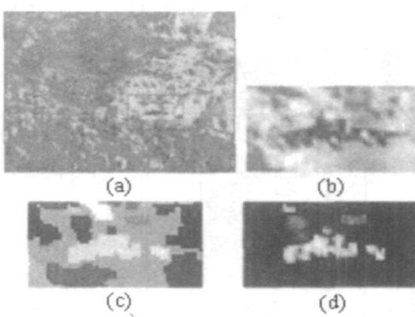


图 2 复杂背景图像中目标权值计算示意图

(b)表示内区域放大图像. 图 2(c)为内区域图像经过 1.2 节、1.3 节和 1.4 节处理后的结果, 这里用不同的灰度代表不同的聚类区域. 图 2(d)为经过 1.5 节计算出来的内区域权值结果, 其中越接近白色权值越高, 这里可以看出, 权值大的区域相对于背景的差别越大, 与人类视觉反映一致.

5.2 实际图像跟踪结果

利用最小二乘匹配跟踪法^[7](LSM)、Mean Shift 跟踪算法(MS)、粒子滤波跟踪算法(PF)以及本文提出的 SWMS 跟踪算法, 对三组序列图像中的运动目标进行跟踪, 其中用红色矩形框表示跟踪结果.

5.2.1 复杂背景下运动飞机跟踪结果(1)

该组实验序列图像中, 飞机相对于背景差别较大. 图 3 表示为序列图像中第 0、20、40、60 帧的跟踪结果, 其中 A 组为 LSM 跟踪结果, B 组为 MS 跟踪结果, C 组为粒子滤波算法跟踪结果, D 组为 SWMS 跟踪结果. 结果表明, LSM 算法和 MS 算法均跟踪失败. 粒子滤波算法开始可以有效地跟踪目标, 但是后来也跟踪失败. 而 SWMS 算法依然可以一直有效、准确地跟踪运动目标.

5.2.2 复杂背景运动飞机跟踪结果(2)

该组实验序列图像中, 飞机相对于背景差别较小. 图 4 表示为序列图像中第 0、8、16、24 帧的跟踪结果, 其中 A 组为 LSM 跟踪结果, B 组为 MS 跟踪结果, C 组为粒子滤波算法跟踪结果, D 组为 SWMS 跟踪结果. 结果表明, LSM 算法和 MS 算法均跟踪失败. 粒子滤波算法开始可以有效地跟踪目标, 但是后来也跟踪失败. 而 SWMS 算法依然可以有效地跟踪运动目标.

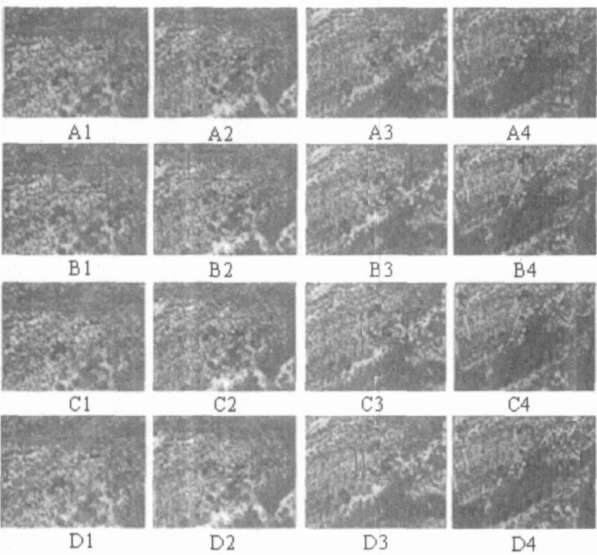


图 3 复杂背景下运动飞机跟踪结果比较 (1)

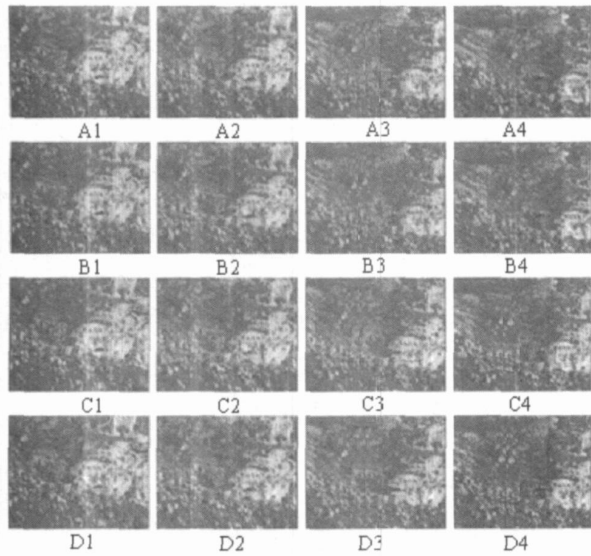


图 4 复杂背景下运动飞机跟踪结果比较 (2)

5.2.3 四种算法平均处理时间比较

对于两组图象, 分别用 MS、LSM、PS 和本文算

法进行运动目标跟踪, 在初始条件相同的条件下, 算法平均处理时间如表 1 所示。其中对于 MS、LSM 和 PS 算法的处理时间取前若干帧跟踪上目标的时间进行计算。可以看出本文提出的 SWMS 算法可以满足实时处理的要求, 而 LSM 和 PS 算法耗时较长, MS 虽然耗时较短但跟踪很容易失败。

表 1 四种算法计算量比较

算法	Ms 算法	最小二 乘法	粒子滤波算法 (30 个粒子)	本文 算法
第一组图象平均处理时间/ms	16.5	47.8	150.7	17.3
第二组图象平均处理时间/ms	17.5	40.2	200.4	18.6

6 结 论

本文提出一种基于显著性加权的 Mean Shift 跟踪算法, 新算法根据目标不同区域和背景的差别, 对目标区域赋予不同的权值。新算法仅对初始帧进行权值计算, 后序帧的跟踪计算效率较高。为了验证新算法实验中, 将最小二乘匹配跟踪法 (LSM)、Mean Shift 跟踪算法 (MS) 以及本文提出的 SWMS 跟踪算法跟踪结果进行了比较。实验表明, 新跟踪算法可以稳健、有效地跟踪复杂背景下运动目标。

参考文献:

[1] Stauffer C, Grimson W E L. Adaptive background mixture models for real-time tracking [J]. IEEE Computer Vision and Pattern Recognition, 1999, 10: 246—252.
[2] Comaniciu D, Ramesh V, Meer P. Real-time tracking of Non-rigid objects using mean shift [J]. IEEE Computer Vision and Pattern Recognition, 2000, 11: 142—149.
[3] Isard M, Blake A. Condensation: conditional density propagation for visual tracking [J]. International Journal of computer Vision, 1998 29(1): 5—28.
[4] 孙即祥, 等. 现代模式识别 [M]. 长沙: 国防科技大学出版社, 2002 31—34.
[5] Castlem an K R. Digital image processing [M]. 北京: 电子工业出版社 2001. 408—410.
[6] 贾云得. 机器视觉 [M]. 北京: 科学出版社, 2000. 36—37.
[7] 张祖勋, 张剑清. 数字摄影测量学 [M]. 武汉: 武汉大学出版社, 1997. 180—186.

(上接第 403 页)

影响也较小, CCD 焦面组件不会因为 CCD 成像电路组件的温度波动而超出工作范围。说明 MLI 和隔热垫等被动热控措施起到了作用, 达到了对 CCD 焦面组件预期的热控效果。

4 结 论

CCD 焦面组件是航天成像传感器热设计中的重要组成部分, 其热控效果的好坏直接关系到空间光学遥感器的成像质量。本文对 CCD 焦平面组件进行了热设计, 采取了以被动热控为主, 主动热控为辅的热控策略, 并通过 CAE 分析验证了其瞬态和温度补偿时的热控效果。数据表明, 对 CCD 焦面组件

所做的热控措施是合理的, 达到了预期效果。对空间环境的各类光学传感器 CCD 焦面组件的热设计有一定的指导和借鉴作用。

参考文献:

[1] 李国强, 贾 宏. CCD 组件的热分析和热试验 [J]. 航天返回与遥感, 2003, 24(3): 15—18.
[2] 陈荣利, 耿利寅, 等. 空间相机的热分析和热设计 [J]. 光子学报, 2006, 35(1): 154—156.
[3] 阎桂荣, 郭舜. 航天器热控制 [M]. 北京: 科学出版社, 1998.
[4] 丁延卫, 卢铿. 遥感器 CCD 驱动器热设计及其在摄像过程中的温度变化 [J]. 光学技术, 2003, 29(2): 172—176.
[5] 吴清文, 卢泽生, 等. 空间光学遥感器热分析 [J]. 光学精密工程, 2002, 10(2): 205—208.
[6] 牛晓明. 空间光学遥感器的热响应分析及热控 [J]. 光学精密工程, 1998, 6(6): 74—78.
[7] 陈恩涛, 卢铿. 空间遥感器 CCD 组件热设计 [J]. 光学精密工程, 2000, 3(6): 522—524.