

基于局部微分光流的运动对象分割

雷震¹ 吴玲达² 李东¹

(装甲兵工程学院信息工程系 北京 100072)¹ (国防科技大学多媒体实验室 长沙 410073)²

摘要 运动对象分割是研究从场景的图像序列或视频中提取出运动目标的理论和方法,是计算机视觉中一个重要的研究方向,在军事和工业等领域有着广阔的应用前景。提出一种基于局部微分光流的运动对象分割算法。首先采用局部微分光流算法计算出场景的运动光流场并完成其初始分割,然后利用canny算子探测出对象的边缘信息并将其作为对光流场得到的运动信息的补充,从而分割出更为准确的运动对象。实验结果显示该方法具有良好的分割性能。

关键词 运动分割,局部微分光流,边缘探测,光流场

中图分类号 TP391 文献标识码 A

Moving Object Segmentation Based on Local Differential Optical Flow

LEI Zhen¹ WU Ling-da² LI Dong¹

(Department of Information Engineering, Academy of Armored Force Engineering, Beijing 100072, China)¹

(Multimedia Laboratory, National University of Defense Technology, Changsha 410073, China)²

Abstract Segmentation of moving objects is a systematic method on the extracting moving objects from image sequences or video. It is an important research direction of computer vision and has a widely future in military and industrial fields. A moving object segmentation method based on local differential optical flow was presented. Firstly, a local differential optical flow algorithm was adopted to calculate the moving optical flow field and the initial segmentation was finished. Secondly, the canny operator was utilized to detect edge information of objects, which is used as the supplement of moving information obtained by optical flow field, so as to segment more exact moving objects. The experimental results demonstrate the good segmentation performance of the proposed approach.

Keywords Moving segmentation, Local differential optical flow, Edge detection, Optical flow field

运动对象分割的主要目标是通过一定手段从视频或图像序列中分割出具有特定意义的运动对象实体。该技术在安全监控和视频图像的高层语义分析等领域应用广泛,同时也是计算机视觉领域的重要研究内容。

国内外学者提出了多种运动对象分割算法^[1-3],例如基于光流场的分割法、基于变化检测模板的分割法、时空联合分割法以及交互式分割法等等。其中,基于光流场的方法不仅携带了运动目标的运动信息,而且能够在不知道场景的任何先验知识的前提下检测出运动对象,因此该方法在运动对象分割领域有着广泛的应用。

普通的光流场算法很难得到完整准确的运动对象。为了弥补光流场得到的运动信息无法对目标边界进行准确定位这

一缺陷,本文采用Canny算子^[4,8]检测出的边缘信息作为对光流场得到的运动信息的补充手段,从而得到更加准确的运动对象分割结果。该方法包括Canny边缘探测、LK光流场提取及光流场分割融合边缘探测信息4个步骤,如图1所示。

1 局部微分光流场计算

基于局部微分度量的LK算法的精确度以及稳定性都很高,因此我们选取它作为光流场计算方法。

LK算法^[5]是由Lunas和Kanade提出的一种光流估算方法。该算法在当前像素点的邻域内按某一特定窗口函数进行加权平方和后求其最小值,以此来估算光流速度,每个像素点的运动矢量都限制于某个均匀的像素运动区域块B中,并可以看作一个运动点,其整体光流运动可以由中心像素点的运动替代处理:

$$v(x, y) = [v_x, v_y]^T = v, x, y \in B \quad (1)$$

区域块B内的误差函数在满足一般性约束方程时定义为:

$$\epsilon = \sum_{x, y \in B} \left(\frac{\partial E(x, y)}{\partial x} v_x + \frac{\partial E(x, y)}{\partial y} v_y + \frac{\partial E(x, y)}{\partial t} \right)^2 \quad (2)$$

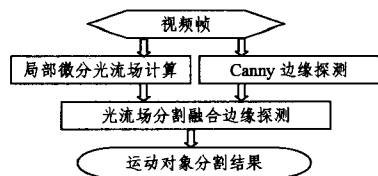


图1 运动对象分割算法原理图

到稿日期:2008-07-09 返修日期:2009-02-11 本文受国家自然科学基金(60473117),装甲兵工程学院创新基金(2008030602)资助。

雷震(1976-),男,博士,讲师,研究方向为多媒体信息处理, E-mail: williamjohnmail@126.com; 吴玲达(1962-),女,博士,教授,研究方向为多媒体信息系统和虚拟现实技术; 李东(1962-),男,副教授,研究方向为多媒体信息处理。

能够使得上述误差函数达到最小的 $v(x, y)$ 值, 即满足 ϵ 对 x 方向和 y 方向运动矢量的一阶偏导数为 0 时的值就是块中心像素的运动估计矢量, 由此得到方程组:

$$\begin{cases} \sum_{x,y \in B} \left(\frac{\partial E(x,y)}{\partial x} \hat{v}_x + \frac{\partial E(x,y)}{\partial y} \hat{v}_y + \frac{\partial E(x,y)}{\partial t} \right) \cdot \frac{\partial E(x,y)}{\partial x} = 0 \\ \sum_{x,y \in B} \left(\frac{\partial E(x,y)}{\partial x} \hat{v}_x + \frac{\partial E(x,y)}{\partial y} \hat{v}_y + \frac{\partial E(x,y)}{\partial t} \right) \cdot \frac{\partial E(x,y)}{\partial y} = 0 \end{cases} \quad (3)$$

其中, \hat{v}_x, \hat{v}_y 是运动区域块 B 中心像素点的运动矢量估计值。求取运动矢量估计值的参考源码如下:

```
%% Lucas Kanade on the image sequence at pyramid step %%
function [Vx, Vy] = EstimateMotion (W, G1, half_window_size)
[m, n] = size (W);
Vx = zeros (size (W)); Vy = zeros (size (W));
N = zeros (2 * half_window_size + 1, 5);
for i = 1:m,
    l = 0;
    for j = 1-half_window_size:1+half_window_size,
        l = l + 1;
        N (l, :) = getSlice (W, G1, i, j, half_window_size);
    end
    replace = 1;
    for j = 1:n,
        t = sum (N);
        [v, d] = eig ([t(1) t(2); t(2) t(3)]);
        namda1 = d(1,1); namda2 = d(2,2);
        if (namda1 > namda2)
            tmp = namda1; namda1 = namda2; namda2 = tmp;
            tmp1 = v(:, 1); v(:, 1) = v(:, 2); v(:, 2) = tmp1;
        end
        if (namda2 < 0.001)
            Vx (i, j) = 0; Vy (i, j) = 0;
        elseif (namda2 > 100 * namda1)
            n2 = v(1,2) * t(4) + v(2,2) * t(5);
            Vx (i, j) = n2 * v(1,2) / namda2;
            Vy (i, j) = n2 * v(2,2) / namda2;
        else
            n1 = v(1,1) * t(4) + v(2,1) * t(5);
            n2 = v(1,2) * t(4) + v(2,2) * t(5);
            Vx (i, j) = n1 * v(1,1) / namda1 + n2 * v(1,2) /
namda2;
            Vy (i, j) = n1 * v(2,1) / namda1 + n2 * v(2,2) /
namda2;
        end
        N (replace, :) = getSlice (W, G1, i, j + half_window_size +
1, half_window_size);
        replace = replace + 1;
        if (replace == 2 * half_window_size + 2)
            replace = 1;
        end
    end
end
end
```

LK 算法是使用最小二乘法估计光流, 定义光流估计误差在邻域 Ω 上为:

$$\sum_{(x,y) \in \Omega} W^2(x) (I_x u + I_y v + I_t)^2 \quad (4)$$

其中, $W^2(x)$ 表示窗口权重函数, 它的解为:

$$U = (A^T W^2 A)^{-1} A^T W^2 B \quad (5)$$

该算法总是假设邻域 Ω 内每个像素点保持恒定且光流计算依赖于窗口权重函数 $W^2(x)$ 。

2 边缘信息融合

边缘检测算法^[6,7]通常通过对邻域内像素灰度求一阶导数、二阶导数及梯度来实现, 这些计算经过化简的结果称为算子, 考虑到 Canny 算子良好的稳定性和较高的信噪比, 本文中我们采用该算子进行边缘检测。Canny 算子的基本原理如下^[8]:

步骤 1 用高斯滤波器对图像进行平滑处理, 其中的高斯内核标准方差值 δ 由噪声干扰情况而定。实际应用中常采用沿图像的 x 和 y 方向各进行一次一维的高斯卷积运算来替代二维高斯卷积运算。

步骤 2 采用一阶偏导的有限差分计算梯度的幅值与方向, 常采用以下 2×2 大小的模板作为 x 和 y 方向偏微分的一阶近似:

$$P = \frac{1}{2} \times \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, Q = \frac{1}{2} \times \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad (6)$$

由此得到梯度幅值 M 和梯度方向 θ :

$$M(i, j) = \sqrt{P^2(i, j) + Q^2(i, j)} \quad (7)$$

$$\theta(i, j) = \arctan[Q(i, j)/P(i, j)] \quad (8)$$

步骤 3 对梯度值进行非极大值抑制。非极大值抑制是通过抑制梯度方向上所有的非屋脊峰值的幅值来细化梯度幅值 $M(i, j)$ 中的梯度幅值屋脊。

步骤 4 采用双阈值算法检测并连接边缘, 该算法对非极大值抑制图像设置了下限阈值 τ_1 和上限阈值 τ_2 ($\tau_2 > \tau_1$)。当 $M(i, j) > \tau_2$ 时, 则该 $M(i, j)$ 不变。如果当 $M(i, j) > \tau_1$ 时, 则将该 $M(i, j) = 0$ 。当 $\tau_2 > M(i, j) > \tau_1$ 时, 如果该 $M(i, j)$ 点的 8-邻域内有一点大于 τ_2 , 则该 $M(i, j)$ 不变, 否则该 $M(i, j) = 0$ 。

在求得光流场和边缘信息之后, 我们采用边缘信息融合光流场进行运动对象的分割。首先需进行光流场分割, 我们将模大于某一阈值的所有光流运动矢量保留下来, 并通过轮廓跟踪得到光流场区域的轮廓图。对原视频帧进行边缘提取后将边缘像素中属于轮廓周线邻域 δ 范围内的像素保留下来, 并结合该轮廓图内的边缘像素值, 这就得到最后的运动对象分割结果。具体过程描述如下:

假设通过光流场得到的 n 个运动矢量估计值集合为 $S_0 = \{s_1^x, s_2^x, \dots, s_{m-1}^x, s_m^x\}$, 通过光流场分割将矢量模大于某一阈值 T 的所有光流运动矢量保留下来得到另一包含 m 个运动矢量的集合 $S_e = \{s_1^y, s_2^y, \dots, s_{m-1}^y, s_m^y\}$, $m < n$ 且 $T = (|s_1^x| + |s_2^x| + \dots + |s_{m-1}^x| + |s_m^x|) / n \pm \gamma$, γ 为阈值微调参数, 接着我们采用 K 均值聚类算法在集合 S_e 的基础上通过迭代算法逐步优化聚类结果, 得到 k 个类型, 然后通过轮廓跟踪得到各光流场区域 $S_i, i = 1, 2, \dots, k$ 的轮廓图像集 $P_{outline} = \{p_1, p_2, \dots, p_{i-1}, p_i\}$ 。对原视频帧 F_i 进行边缘提取后得到边缘像素集合 $E = \{e_1, e_2, \dots, e_{k-1}, e_k\}$, 将该集合中属于轮廓周线邻域 δ 范围内的像素保留下来形成集合 $M_e = \{m_1^e, m_2^e, \dots, m_{i-1}^e, m_i^e\}$, 即 $|m_i^e - p_j| < \delta$, 这里 $M_e \subseteq E$ 且 $p_j \in P_{outline}$, 如果属于轮廓图 $P_{outline}$ 内的边缘像素集为 I_k , 则由当前两帧得到的运动对象为 $M_0 = M_e \cup I_k$ 。在提取出视频对象之后, 还需要在后续视频帧

中跟踪视频对象以完成视频对象的分割,好的视频对象分割算法不但具有空间准确性,也满足时间的稳定性。本文中我们采用后向投影法^[9]进行跟踪,该方法实际上是一种区域分类的方法,根据投影区域与前一帧视频对象的重叠面积来确定是否属于视频对象,其优势在于能适应视频对象的形变,缺点在于当前帧的每一分割区域都需要进行后向投影,导致运动估计的计算量过大,在实际操作上我们仅仅对当前帧中前帧视频对象附近的区域进行分割并对其后进行后向投影,这将在一定程度上降低运算量,有效地提高运算效率。

3 仿真实验

为验证上述算法的有效性,本实验采用两组具有代表性的图像序列进行实验。图2中的(a)和(b)为序列1中的连续两帧,图2(c)为根据两帧计算出的光流场,图2(b)相对于图2(a)只是中间的正方形发生了位置的平移,而摄像机不动且背景保持不变。图3(a)是采用本文分割算法在图2(c)光流场基础上得到的光流分割场,而图3(b)是对图2(b)进行Canny边缘探测得到的边缘探测图,图3(c)是采用本文算法得到的最终的运动对象分割结果,从图中我们可以看出,运动对象被清晰地检测出来,尽管右上角和底部偏左部分稍微有些缺失,不是一个完整的正方形。这是由于采用LK算法探测出的光流场在这些缺失领域的矢量模偏小而被过滤掉的缘故,并且LK算法只能得到稀疏的光流场,要想得到既致密又鲁棒的光流场还需要结合全局方法。

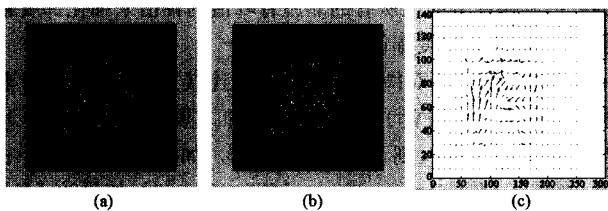


图2 序列1中的连续两帧((a)和(b))及其光流场(c)

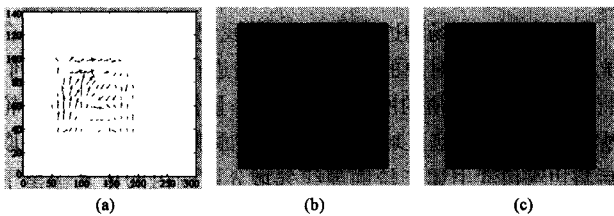


图3 光流分割场(a)、边缘探测图(b)及运动对象分割结果(c)

图4中的(a)和(b)为序列2中的连续两帧,图4(c)为根据两帧计算出的光流场,主要运动信息在光流场中被有效地凸显出来。与序列1的不同之处在于图4中的两帧只是在树前方的摄像机发生了平移,而近处的树和远处的房子并没有运动,所以看起来好像是图中的树发生了位置的移动。图5(a)是采用本文分割算法在图4(c)光流场基础上得到的光流分割场,该图背景是比较复杂的,但光流分割场仍然比较准确地把离摄像机不远处的树给勾勒出来,同时弱化掉了其背景信息。而图5(b)是对图4(b)采用Canny算子探测到的边缘,图5(c)是采用本文算法得到的最终的运动对象分割结果,图中结果显示采用本文算法有效地分割提取出了运动对象。

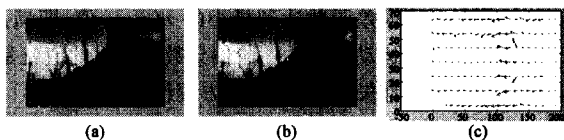


图4 序列2中的连续两帧((a)和(b))及其光流场(c)

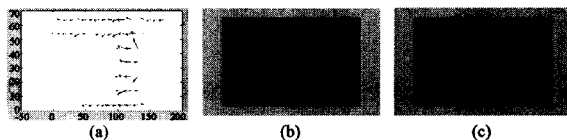


图5 光流分割场(a)、边缘探测图(b)及运动对象分割结果(c)

实验中进行边缘提取时我们同时尝试了其它算子,如Prewitt算子^[7]。结果发现:从和光流分割场信息融合提取运动对象的角度考虑,Canny算子的效果是最好的。这是因为后者并不只是简单地依赖梯度运算来决定像素是否为边缘点,而是综合考虑其他像素的影响并采用边界跟踪来确定边缘点。

结束语 当摄像机不动而运动对象发生运动或摄像机围绕某对象运动时,单独采用普通光流场算法或边缘探测算法均不能准确地分割运动对象。本文在采用LK算法提取并分割光流场的基础上结合Canny边缘探测信息有效地分割出了运动对象,该算法经实验证明是有效的。

需要指出的是,光流场算法计算复杂,这在一定程度上影响了其实时性,并且在运动变化较大的场合,常规的光流场算法无法准确有效地计算出光流运动场。因此如何提高光流场算法的效率并使得算法更具实用性,如何从时空域的角度有效地减小对象大幅运动的差异,仍需做进一步研究。

参考文献

- [1] DeMenthon D. Spatio-Temporal Segmentation of Video by Hierarchical Mean Shift Analysis[C]// Statistical Methods in Video Processing Workshop. 2002
- [2] Meier T, Ngan K N. Automatic Segmentation of Moving Objects for Video Object Plane Generation[J]. IEEE Trans. Circuits System for Video Technology, 1998, 8(5): 525-538
- [3] 张坤华,王敬儒,张启衡. 复杂背景下扩展目标的分割算法研究[J]. 红外与毫米波学报, 2002, 21(3): 233-237
- [4] Canny J. A computational approach to edge detection [J]. IEEE Trans. on PAMI, 1986, 8(6): 679-698
- [5] Lucas B D, Kanade T. An iterative image registration technique with application to stereo vision[C]// Proceedings of the 7th International Joint Conference on Artificial Intelligence. Vancouver, 1981: 674-679
- [6] 连静,王珂. 样条小波自适应阈值多尺度边缘检测算法研究[J]. 系统仿真学报, 2006, 6(18): 1473-1477, 1482
- [7] 王爱民,沈兰荪. 基于矢量Prewitt算子的多尺度彩色图像边缘检测方法[J]. 中国图象图形学报, 1999, 12: 1024-1028
- [8] 贾云得. 机器视觉[M]. 北京: 科学出版社, 2000: 97-100
- [9] Gu C, Lee M C. Semantic video object tracking using region-based classification [C] // Proceedings of ICIP98. Chicago, 1998: 643-647