

基于双帧图模型的视频物体分割

蒲松涛 查红彬[†]

北京大学机器感知与智能教育部重点实验室, 北京 100871; [†] 通信作者, E-mail: zha@cis.pku.edu.cn

摘要 通过将空间及时序信息有效结合, 提出基于双帧图模型的单目视频物体分割算法。首先, 通过手工交互得到初始帧的分割, 并据此训练获取前景物体及背景的颜色模型。然后, 利用双帧图模型融合当前帧的颜色信息以及由颜色差得到的空间和时序约束。根据观测到的颜色差异与物体运动的线性关系, 提出运动自适应的时序约束因子, 它能够随视频中物体运动变化自适应地调节。最后, 通过二值图割法计算当前帧的分割结果并更新颜色模型。利用双帧图模型可循序地对视频中的下一帧进行分割。实验结果证实, 提出的自适应时序约束因子可以提高物体分割结果的准确性和时序一致性, 量化指标表明此算法在视频物体分割中可获得更优结果。

关键词 视频物体分割; 双帧图模型; 时序约束因子; 图割

中图分类号 TP391

Video Object Segmentation via Two-Frame Graph Model

PU Songtao, ZHA Hongbin[†]

Key Laboratory of Machine Perception (MOE), Peking University, Beijing 100871;

[†] Corresponding author, E-mail: zha@cis.pku.edu.cn

Abstract The authors present a novel algorithm for object segmentation in monocular video sequences, which is based on proposed two-frame graph model. First, the bi-layer color models can be learnt from the interactive segmentation on the first frame. To sequentially extract the object on the next frame, the two-frame graph model is proposed, in which the color cues and spatio-temporal constraints are combined. Based on the observation that the amount of the object movement is approximate linearly related to the summation of the temporal contrasts between adjacent frames, the motion-adaptive temporal-constraint factor is proposed in order to control the weight of temporal constraints in the graph model. The simplification of the graph model increases the efficiency of the algorithm. Finally, each frame is segmented by binary graph cut and the color models are updated. Experimental results show that the adaptive temporal-constraint factor can improve the accuracy and temporal coherence of the results. Experiments on the public database indicate that the proposed algorithm is more efficient and effective.

Key words video object segmentation; two-frame graph model; temporal-constraint factor; graph cut

视频物体分割是将图像序列中的前景物体从背景中提取出来的技术。长期以来, 分割问题都是计算机视觉和模式识别研究领域的一个热点, 其中视频物体分割是一个难点, 近年来得到很多关注。在视频物体分割的基础上, 可以实现背景替换, 该技术已在电视节目及电影制作中得到广泛应用。

当前, 视频物体分割的方法主要有 4 类。第一

类是基于人工交互的方法^[1-3], 通过交互得到部分区域的前景或背景标记, 标记结果可用于学习颜色模型并提供强制约束, 从而实现视频中的前景物体的分割, 其结果准确性较高, 但依赖于手工干预。第二类是基于背景模型的方法^[4], 其中色度抠图是最为广泛应用的技术。此方法主要通过构建背景模型, 并采用背景差分的方式提取前景物体。其主要

难点在于如何获取或构建有效的背景模型以及设置合适的差分阈值。第三类是基于深度信息的方法,假定前景物体和背景环境存在深度差异,通过深度传感器^[5]、立体摄像机^[6-7]或基于运动三维建模^[8]获取场景的深度信息,并结合图像中的颜色信息对视频中的前景物体进行分离。第四类是基于时序运动的方法,获取当前帧的分割结果后,利用运动特性或时序约束来辅助下一帧的分割,如光流法^[9]。本文方法属于第四类,提出一个新的双帧图模型来融合运动及颜色信息,并结合前帧结果对当前帧进行前景提取。

在视频物体分割问题中,有三大挑战^[10]。首先是数据量庞大。相较于单幅图像视频增加了时间轴,使数据量成倍增加。为此,我们使用一个数据流算法,即对每帧图像仅处理一次并保留处理结果,从而实现高效的视频前景分割。其次是保持结果的时序一致性。人们在观看视频时,对于时序的不一致性极其敏感^[11],因此要求分割结果在帧与帧之间尽量保持一致。本文通过双帧图模型将空间约束和时序约束统一处理,从而在保证较高准确率的同时增强分割结果的时序一致性。最后是如何应对物体运动变化尤其是快速运动。在本文提出的双帧图模型中,引入自适应时序约束因子,该因子可以根据物体运动状态的变化自适应调节,使本文算法在物体运动变化时也能获得准确的分割结果。

本文首先通过 GrabCut^[12]对视频的首帧进行二类分割,即前景物体和背景。根据获得的分割结果,使用 RGB 空间的混合高斯模型对前景和背景的颜色建模。为了提高算法效率,运用逐帧更新样本及间隔更新模型的策略。为了循序地分割新的一帧,基于双帧图模型的能量表示融合了颜色信息、时序和空间颜色差异以及前帧的分割结果。此外,双帧图模型的简化可以有效提高算法的运算效率,降低存储代价。通过分析分割中的空间边界及时序边界,定义了时序约束因子。基于物体运动与视频中时序颜色差异总和之间的近似线性关系,本文提出当前帧的物体运动量大小及自适应时序约束因子的估计方法。在自适应时序约束因子的帮助下,本文算法可以在物体运动变化的情况下,获得增强时序一致性的分割结果。

1 相关工作

在图像分割研究领域,图割法(Graph Cut)^[13]是

近年来广泛使用的方法。在手工交互辅助下获取前景及背景的颜色信息,通过构建单层图模型,估计像素间的平滑代价及与终端节点(前景或背景)的数据代价,利用 max-flow/min-cut 算法将图像中前景物体提取出来。GrabCut^[12]在图割法的基础上加以改进,可以实现更加便捷有效的单幅图像交互分割。在视频处理中,通过构建三维图模型并加入手工交互^[1-2],也可以实现视频中前景物体的分割。但是,该类方法效果依赖于手工交互,且需要较大的时间和存储代价,限制了实用性。本文提出的双帧图模型不仅包含像素水平的时序约束,并且实现了数据流式处理,从而使算法的存储消耗降低,提高了算法的运算效率。

在基于时序运动的视频物体分割方法中,光流估计被普遍用于获取当前帧的分割先验及增强分割结果的时序一致性^[14]。在光流估计得到的分割结果基础上,还可通过时序滤波对初始结果进行深度优化^[15]。但是,光流估计需要计算每个像素的运动,运算过程比较耗时并且往往在图像边缘区域及物体运动较快时会出现错误^[16]。本文方法摒弃光流估计,通过在双帧图模型中使用时序平滑约束增强结果的时序一致性,并提出自适应时序约束因子对应视频中的物体运动变化。

在视频处理中,时序颜色差异是一种普遍被利用的信息。在最新的研究中,时序颜色差异可以用于训练运动分类器。该分类器可以用来估计概率模型中的运动似然^[16]或生成“运动元”(motons)^[17]。此外,时序颜色差异还可用于估计前景物体的边界框^[18]。在这些方法中,时序颜色差异都作为一个单独的对象来处理。本文将时序颜色差异和空间颜色差异统一对待,在双帧图模型中提供时空平滑约束。

2 算法流程

本文提出一个数据流算法,循序地对每一帧图像进行单次处理,并使分割结果在后续处理过程保持不变。在对当前帧进行分割时,通过结合像素颜色、时空颜色差异以及前帧分割结果提供的信息,构建双帧图模型,最后通过二值图割法得到当前帧的分割结果。算法流程如图 1 所示,每个分割单元代表一次基于双帧图模型的分割。

图 1 中, I 表示视频的一帧, I^t 表示 t 时刻的图像。图像分割问题可认为是一个二类的标记问题,即计算每个像素 r 的标签 x_r 。 $x_r = \{1, r \in \text{前景}; 0, r \in \text{背景}\}$

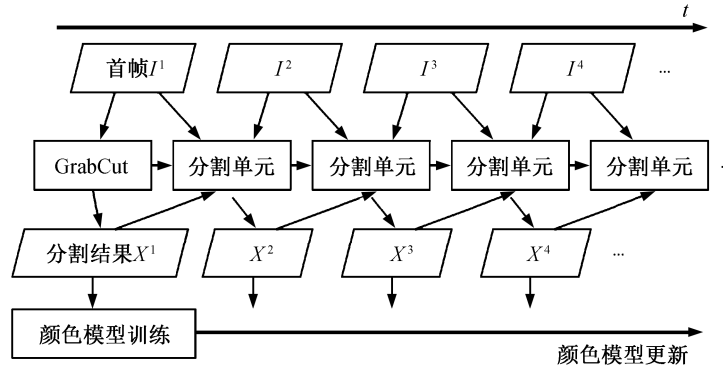


图 1 算法流程

Fig. 1 Framework of the algorithm

\in 背景}, t 时刻的图像对应的分割结果用 X^t 表示。

在算法的初始, 手工交互提供了初始的前景及背景信息, 并通过 GrabCut 来获取起始帧的分割结果, 作为整个算法的基础。得到首帧中各个像素的标签 x_r 后, 可以通过随机采样生成前景和背景的颜色样本。本文选用RGB空间的混合高斯模型分别对前景和背景颜色进行建模, 该模型可以通过 EM 算法训练得到。其中, 前景颜色模型可以表示为

$$p(I_r | x_r = 1) = \sum_{k=1}^K \omega_k^f N(I_r | \mu_k^f, \Sigma_k^f), \quad (1)$$

对应的背景颜色模型为 $p(I_r | x_r = 0)$ 。 f 代表前景, ω_k 表示混合高斯模型中第 k 个成分的权重, μ_k 和 Σ_k 分别表示其成分对应的颜色平均值及方差。本文中, 前景和背景颜色模型的混合成分数目 K 均设为 5。

EM 算法使用迭代求解的方式, 因此其时间代价较大, 基于提高算法效率的考虑, 在得到新的图像分割结果后, 仅仅更新了前景及背景的颜色样本。本文中, 每处理 10 帧之后, EM 算法才被用于更新前景和背景的颜色模型, 使之在视频场景变化中保持有效性。

3 双帧图模型

图 1 中每个分割单元都是通过二类图割来实现。与图割法一致, 给定一个视频序列 $V, V = \{I^1, I^2, \dots, I^n\}$, 本文通过最小化一个吉布斯(Gibbs)能量 E 得到前景物体分割结果 X :

$$X = \operatorname{argmin} E(X, V). \quad (2)$$

能量 E 主要由两个部分构成: 数据项和平滑项。数据项代表每个像素获得某一标签的代价, 通过图像的颜色信息来估计; 平滑项也称为对比项, 通过相邻像素的颜色差异来估计, 表示相邻像素具

有不同标签的代价。

与图割法相似, 本文提出的图模型中除终端节点外, 每个像素代表一个节点; 与图割法中的模型不同, 为了结合视频中的时序约束, 本文提出用双帧图模型对当前帧进行分割, 其中前帧图像的每个像素也构成图中的节点。双帧图模型中, 时序颜色差异被用于估计帧间像素节点间的权重。对于图像对 I^{t-1} 和 I^t , 其对应的时序平滑能量为

$$E_{s-t}(I^{t-1}, I^t) = \lambda_t \sum_{(r,s) \in \mathcal{E}_{t(t-1,t)}} E_s(r,s). \quad (3)$$

其中, $\mathcal{E}_{t(t-1,t)}$ 表示时间相邻帧 I^{t-1} 和 I^t 的所有对应位置像素对的集合。 $E_s(r,s)$ 表示像素节点 r 和 s 之间的平滑能量。

$$E_s(r,s) = |x_r - x_s| \cdot e^{-\beta d_{rs}}. \quad (4)$$

其中, $d_{rs} = \|I_r - I_s\|^2$ 表示 I_r 和 I_s 的颜色平方差。参数 $\beta = (2 \langle \|I_r - I_s\|^2 \rangle)^{-1}$, $\langle \cdot \rangle$ 表示在 \mathcal{E} 中所有像素对的颜色差异期望^[12]。

空间平滑能量表示为

$$E_{s-s}(I^t) = \lambda_s \sum_{(r,s) \in \mathcal{E}_{st}} E_s(r,s), \quad (5)$$

\mathcal{E}_{st} 表示图像 I^t 中所有空间相邻的像素对集合。

此外, 在双帧图模型中需要估计每个像素节点和终端节点之间的权重, 通过数据项来表示。对于当前帧, 可通过颜色似然来估计数据项能量:

$$E_{d-c}(I^t) = \sum_{r \in I^t} -\log p(I_r | x_r). \quad (6)$$

对于前景能量计算时, $x_r = 1$; 相应的背景能量计算时, $x_r = 0$ 。

在一个分割单元中, 之前帧的分割结果将会被保留, 因此之前帧的像素已被标记为“前景”或“背景”, 可以认为这些前帧标记结果提供了一个强制

约束,从而构成能量 $E_{d-c}(I^{t-1})$,对于前景,

$$E_{d-c}(I^{t-1} | x_r = 1) = \begin{cases} 0, & x_r^{t-1} = 1, \\ N_{\max}, & x_r^{t-1} = 0. \end{cases} \quad (7)$$

相应可得背景能量为 $E_{d-c}(I^{t-1} | x_r = 0)$,其中 N_{\max} 代表一个极大值。该能量作用与图割法中的手工交互约束一致,即保证这些节点在图割后的标签不变。

综上所述,基于双帧图模型的能量可表示为

$$E(X, t) = E_{d-c}(I^{t-1}) + E_{d-c}(I^t) + E_{s-s}(I^t) + E_{s-t}(I^{t-1}, I^t) \quad (8)$$

其中 $E_{d-c}(I^{t-1})$ 和 $E_{d-c}(I^t)$ 为前帧节点和当前帧节点的数据项, $E_{s-s}(I^t)$ 和 $E_{s-t}(I^{t-1}, I^t)$ 为空间和时序平滑项。

图 2 为双帧图模型,上下两层节点之间的连线表示相邻帧间的连接权重,可以通过时序颜色差异 D^t 估计得到。同时,帧内节点间权重则通过空间颜色差异来计算。前帧的分割结果 X^{t-1} 可以认为是图模型中的强制约束,用黑色(背景)和白色(前景)节点表示。当前帧每个节点的数据能量由像素节点的颜色及前景和背景的颜色模型得到。

在双帧图模型中,由于前帧节点的分类标记不变,所以它在模型中的作用与终端节点一致。因此,

空间平滑约束实际上可以看做当前帧节点的附加数据项。图 3(a)为原始的图模型,图 3(b)为简化的模型。对其中一个当前帧节点而言,模型估计了其与前帧对应节点的平滑能量。由于前帧节点代表背景节点,因此可将该平滑能量转变为其与背景终端节点的数据能量。这样,式(8)中的能量可简化为

$$E(X, t) = E_{d-c}(I^t) + E_{d-t}(I^t) + E_{s-s}(I^t), \quad (9)$$

$$E_{d-t}(I^t | x_r = 1) = \begin{cases} 0, & x_r^{t-1} = 1, \\ E_{s-t}(I^{t-1}, I^t), & x_r^{t-1} = 0. \end{cases} \quad (10)$$

相应地,可以计算 $E_{d-t}(I^t | x_r = 0)$ 。通过简化图模型,节点的总数目由原来的 $2N+2$ 减少到 $N+2$ (N 代表单帧图像的像素数),从而减小了数据的存储代价,提高了算法的运算效率。

为了明晰地阐述时序平滑约束与空间平滑约束的关系,下文仍将时序约束作为一个平滑项来看待。

4 自适应时序约束因子

在双帧图模型中, λ_s 和 λ_t 分别表示空间平滑能量及时序平滑能量的权重。式(5)表示空间平滑能量的总和与 3 个因素相关:权重 λ_s 、分割边界的数

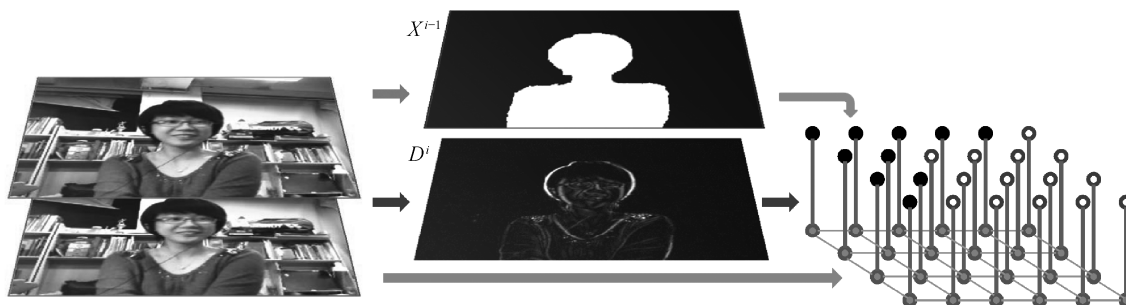


图 2 双帧图模型
Fig. 2 Two-frame graph model

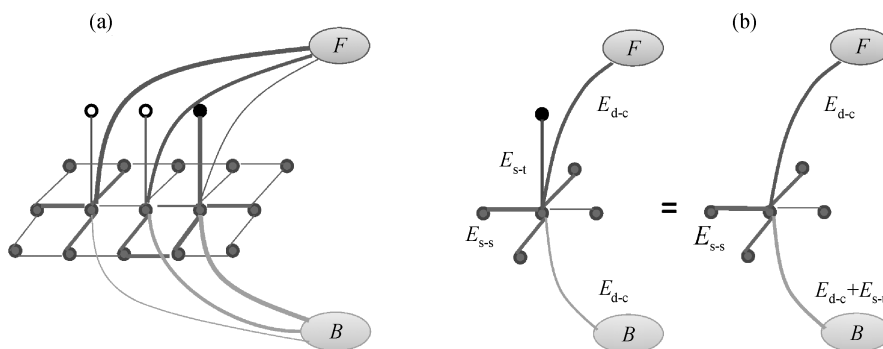


图 3 双帧图模型的简化
Fig. 3 Simplification for two-frame graph model

目和边界节点对间的颜色差异。如图 4 第 2 行所示, 在一个视频中, 前景物体的轮廓长度一般变化不大, 通过选用恒定的 λ_s , 使空间平滑能量的大小主要依赖于边界颜色差异, 保证提取的物体轮廓尽量位于颜色差异较大的像素对之间。但是, 如图 4 第 3 行所示, 在一个视频中时序边界的大小并不是几乎不变的, 它与物体的运动紧密相关。当物体运动较快时, 时序分割边的数目较大; 当物体运动较慢时, 时序分割边的数目较小; 当物体处于静止时, 时序分割边的数目为 0。因此, 为了保证时序平滑能量能够准确地反映时序颜色差异, 在双帧图模型中, 时序平滑能量的权重 λ_t 需要根据视频中的物体运动自适应调节。

在双帧图模型中, λ_t 的大小反映时序约束的强弱, 本文称之为时序约束因子。如果将模型中的时序约束因子 λ_t 保持恒定, 会导致两种错误的出现: 一是在物体运动量较小时, 颜色数据项及空间平滑项的权重增大, 削弱了时序平滑约束, 出现分割结果的欠一致性; 二是在物体运动较快时, 时序平滑项的作用更为明显, 削弱了颜色及空间颜色差异提供的约束, 导致分割结果的过一致性。这两种错误均降低了算法结果的准确率。

因此, 在物体保持静止或运动缓慢时, 应当增大时序约束因子来增强时序约束, 使当前帧的分割与前一帧结果尽量保持一致; 当物体运动加快时应减小时序约束因子来削弱时序约束, 提升颜色信息及空间平滑项在分割中的作用。因此, 为了确定约束因子的大小, 首先需要估计物体在当前帧的运

动量。

在分割问题中, 物体的运动可以通过相邻帧的轮廓差异和 $\sum_r |x_r^t - x_r^{t-1}|$ 来表示, 它恰恰代表了时序分割边的数目。我们观测到时序轮廓差异与视频中帧间时序颜色差异存在近似的线性相关, 即当相邻帧的颜色差异总和越大, 物体往往运动越快, 其对应的轮廓差也将越大, 反之亦然。如图 5 所示, 在两个测试视频(AC 和 MS)上, 时序颜色差异总和与相邻帧轮廓差之间的相关性系数均接近 1, 表明了二者的统计线性关系。

基于物体运动和时序颜色差异总和的线性关系, 物体当前帧的运动量(即时序轮廓差异)可以通过下式计算得到:

$$D(t) = D_c(t) \frac{Ds_x(t-1)}{Ds_c(t-1)}, \quad (11)$$

其中 $D_c(t) = \sum_r |I^t - I^{t-1}|$, 表示图像 I^t 与图像 I^{t-1} 的时序颜色差异总和; $Ds_c = \sum_{t'=2}^t D_c(t')$, 为前帧累计颜色差异总和; $Ds_x = \sum_{t'=2}^t \sum_r |x_r^{t'} - x_r^{t'-1}|$, 表示前帧分割得到的前景轮廓时序差异总和。

双帧图模型中的自适应时序约束因子可表示为

$$\lambda_t(t) = \lambda_s \cdot (D_{ref} / D(t))^\kappa, \quad (12)$$

其中, D_{ref} 表示相邻帧轮廓差的参照值, 在本文中设定为首帧中物体轮廓长度的一半, 空间平滑参数 λ_s 和常参数 κ 的默认值分别为 150 和 2。 $\lambda_t(2)$ 即前两帧图像对应的时序约束因子, 默认等同于 λ_s , 在之后的分割单元中, 时序约束因子均通过式(12)计算得到。



图 4 空间及时序分割边界真值

Fig. 4 Ground-truth of spatial and temporal cut boundaries

5 实验结果

实验采用的视频数据来源于本文第一作者用 iPhone 手机拍摄的视频和公共数据集^①。在 2.67 GHz 台式机上,用 Matlab 编译本文算法,对单帧大小为 320×240 的视频的处理速度可以达到约 8 fps。首先检验自适应时序约束因子的有效性。在双

帧图模型框架下,使用恒定值时序约束因子及自适应时序约束因子的结果见图 6。图中第一行表示输入视频中的若干帧,第二、三行分别为时序约束因子设定为较大恒定值(增强时序一致性)及较小恒定值(增强颜色模型作用及空间约束)时的分割结果,第四行为使用自适应时序约束因子的分割结果。结

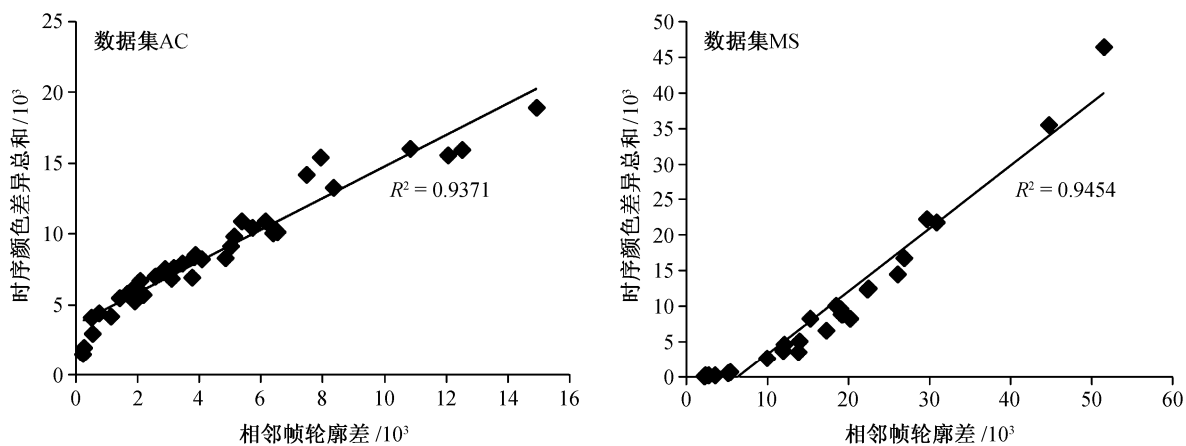


图 5 时序颜色差异总和与相邻帧轮廓差之间的线性关系

Fig. 5 Linear relationship between the summation of temporal color differences and the difference of adjacent extracted objects



图 6 不同时序约束因子下的分割结果

Fig. 6 Results with different temporal-constraint factors

① <http://research.microsoft.com/en-us/projects/i2i/data.aspx>

果表明,前景物体运动加快时,恒定较大的时序约束因子增强了时序约束,削弱了颜色项及空间平滑项的作用,前景及背景变化区域更倾向于与前帧对应的像素标签保持一致,导致分割结果出现过一致性错误,同时前帧中的错误结果在时序一致性增强中得以延续。时序约束因子恒定较小时,分割中时序约束的作用减小,使前后帧的分割结果出现较大差异,导致欠一致性错误。本文提出的自适应时序约束因子可根据物体的运动速度来自适应地调节,在物体运动较慢时增强分割结果的时序一致性,当物体运动较快时提高颜色信息和空间颜色差的作

用,保证了分割结果的准确性。实验结果还表明,本文算法不依赖于背景模型,可适用于动态背景情况。

实验中,分割结果的误差可通过单帧图像中错误标记像素的数目与图像像素总数的百分比表示:

$$\text{误差} = \frac{\sum_r |x_r - x'_r|}{N} \times 100\%, \quad (13)$$

其中 x'_r 表示像素 I_r 标记的真值, N 表示单帧图像的像素数目。误差值越小表明该物体分割算法的效果越好。

图 7 为在 3 个视频序列上使用恒定及自适应时

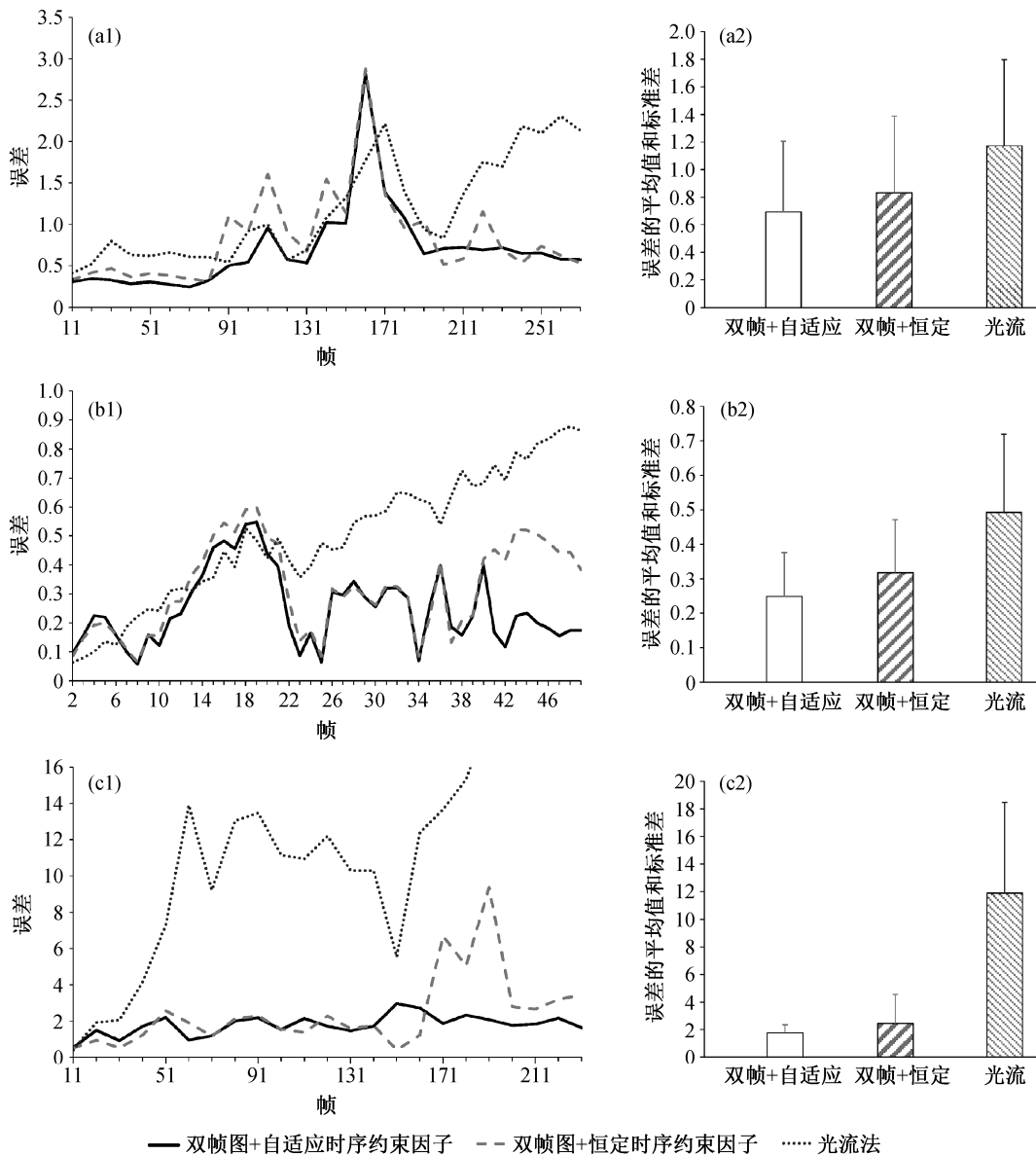


图 7 在 3 个视频序列上的同光流法的结果对比

Fig. 7 Comparison with optical flow based method on three typical videos

序约束因子时,本文方法与光流法的结果比较,其中误差为错误标记像素数与图像大小比值。图 7(a)中物体运动较为缓慢,由于光流估计可以得到较为准确的结果,因此 3 种方法均可以得到较为理想的结果。图 7(b)视频中物体运动较快,导致光流法的结果误差随着帧数逐渐递增。图 7(c)视频中物体运动显变差,而使用时序约束因子的双帧模型可以获得更优的分割结果。图 7 表明,在减少时间代价的同时,本文提出的双帧图模型还可有效地利用视频中的时序约束,在不同视频数据上的表现都更为优异。物体运动变化时,使用自适应时序约束因子的结果误差更小。

在公共数据集上的实验结果如图 8 和表 1 所示。图 8 为本文算法在部分视频数据上的分割结果。表 1 为在数据集上的误差中值比较。通过与最新研究^[17]比较,10 组数据中有 7 组本文算法的误差中值更小,表明分割结果更加准确。并且,本文算法不需要运动模型的训练过程,具有更强的实用

性。此外,本文算法在数据集中的两组存在相机抖动及严重遮挡的挑战视频(“GTTS09”和“GTTS10”)上的测试结果同样优秀,误差的中位数均小于 2%。

6 结论

本文提出一个用于单目视频物体分割的新算法。该方法使用交互的方式来确定第一帧的分割,并以此作为初始参照,除此以外,该方法实现了全自动处理,基于数据流方式可高速处理庞大的视频数据。本文提出的双帧图模型有效地结合了图像序列提供的时序及空间约束,以及颜色和前帧分割信息,最终通过二值分割来实现前景提取。通过简化双帧图模型,可以减少算法的空间复杂度,提高算法的运算效率。

基于物体运动量与时序颜色差异总和的近似线性关系,本文提出运动自适应时序约束因子,该因子可以随物体运动的变化动态调节。实验结果表



图 8 在测试视频序列的分割结果

Fig. 8 Results on three test sequences

表 1 在 10 组测试视频上的分割误差
Table 1 Segmentation error for ten test sequences %

测试序列	文献[17]	本文误差
41	0.80	0.30
43	0.02	3.26
50	1.31	0.70
51	1.06	1.01
54	0.33	0.41
56	0.93	2.65
58	0.79	0.21
60	6.33	1.36
IU	2.56	1.53
JM	0.27	0.17

明, 通过引入自适应时序约束因子, 增强了分割结果的准确性和时序一致性。通过与光流法的比较, 本文算法具有更高的效率及效果。在公共数据集上的量化比较表明, 本文算法可以在视频物体分割中获得更准确的结果。

参考文献

- [1] Li Y, Sun J, Shum H Y. Video object cut and paste. *ACM Transactions on Graphics*, 2005, 24(3): 595–600
- [2] Wang J, Bhat P, Colburn R A, et al. Interactive video cutout. *ACM Transactions on Graphics*, 2005, 24(3): 585–594
- [3] Bai X, Sapiro G. A geodesic framework for fast interactive image and video segmentation and matting // *IEEE International Conference on Computer Vision*. Rio de Janeiro, 2007: 1–8
- [4] Sun J, Zhang W, Tang X, et al. Background cut // *Proceedings of European Conference on Computer Vision*. Graz, 2006: 628–641
- [5] Wang L, Gong M, Zhang C, et al. Automatic real-time video matting using time-of-flight camera and multi-channel poisson equations. *International Journal of Computer Vision*, 2012, 97(1): 104–121
- [6] Kolmogorov V, Criminisi A, Blake A, et al. Bilayer segmentation of binocular stereo video // *IEEE Conference on Computer Vision and Pattern Recognition*. San Diego, 2005: 53–60
- [7] Yang W, Zhang G, Bao H, et al. Consistent depth maps recovery from a trinocular video sequence // *IEEE Conference on Computer Vision and Pattern Recognition*. Providence, 2012: 1466–1473
- [8] Yu T, Zhang C, Cohen M, et al. Monocular video foreground/background segmentation by tracking spatial-color Gaussian mixture models // *IEEE Workshop on Motion and Video Computing*. Austin, 2007: 23–24
- [9] Bai X, Wang J, Simons D, et al. Video snapcut: robust video object cutout using localized classifiers // *ACM SIGGRAPH*. New Orleans, 2009: 1–11
- [10] Wang J, Cohen M F. Image and video matting: a survey. *Foundations and Trends in Computer Graphics and Vision*, 2007, 3(2): 97–175
- [11] Villegas P, Marichal X. Perceptually-weighted evaluation criteria for segmentation masks in video sequences. *IEEE Transactions on Image Processing*, 2004, 13(8): 1092–1103
- [12] Rother C, Kolmogorov V, Blake A. “GrabCut”: interactive foreground extraction using iterated graph cuts. *ACM Trans Graph*, 2004, 23(3): 309–314
- [13] Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans Pattern Anal Mach Intell*, 2004, 26(9): 1124–1137
- [14] Chuang Y Y, Agarwala A, Curless B, et al. Video matting of complex scenes. *ACM Trans Graph*, 2002, 21(3): 243–248
- [15] Bai X, Wang J, David S. Towards temporally-coherent video matting // *MIRAGE’11*. Seattle, 2011: 63–74
- [16] Criminisi A, Cross G, Blake A, et al. Bilayer segmentation of live video // *CVPR*. New York, 2006: 53–60
- [17] Yin P, Criminisi A, Winn J M, Essa I A. Bilayer segmentation of webcam videos using tree-based classifiers. *IEEE Trans Pattern Anal Mach Intell*, 2011, 33(1): 30–42
- [18] Liu Dingming, Zhao Jieyu. Spatio-temporal video object segmentation using moving detection and graph cut methods // *ICNC*. Shanghai, 2011: 1859–1862