

# 融合超像素与动态图匹配的视频跟踪

张君昌<sup>1,2</sup>, 周艳玲<sup>1</sup>, 万锦锦<sup>2</sup>

(1.西北工业大学 电子信息学院, 陕西 西安 710129; 2.光电控制技术重点实验室, 河南 洛阳 471000)

**摘要:**针对视频跟踪过程中目标的形变、遮挡、旋转和背景干扰问题,提出一种融合超像素与动态图匹配的视频跟踪方法。首先,采用融合局部熵特征的简单线性迭代聚类(simple linear iterative clustering, SLIC)方法经聚类分析生成超像素集合,使生成的超像素边缘贴合度更好。其次,采用图像分割(graph cuts)方法生成候选目标超像素集合,并融合在线支持向量机器学习算法(online SVM learning algorithm, LASVM)分类预测结果,使前景与背景分离的准确度更高。然后,充分利用目标的几何结构信息构建基于图模型的相似度矩阵,解决目标的形变和遮挡问题。理论分析与仿真结果表明:相比现有其他视频跟踪方法,新方法对跟踪过程中的遮挡和形变情况具有较强的鲁棒性,对一定程度的背景干扰和旋转问题跟踪效果良好。

**关键词:**目标追踪;信息融合;简单线性迭代聚类;超像素;图像分割

中图分类号:TP391

文献标志码:A

文章编号:1000-2758(2017)01-0133-05

随着计算机网络、数字通信和微电子技术的迅速发展,视频跟踪技术已经成为计算机视觉领域的重要研究方向之一。视频跟踪方法是围绕着如何解决视频跟踪过程中所遇到的问题而发展起来的。视频跟踪过程中常见的问题有形变、遮挡、旋转、背景杂波等。围绕这些问题,专家学者们提出一系列的视频跟踪方法。有的方法是根据目标与背景的差异性,通过二元分类解决视频目标跟踪问题;有的方法是根据目标的状态预测目标位置。这2类方法虽可以跟踪到目标,但是所需的数据量大,实时性不理想。因此,基于模板匹配的跟踪方法因其原理简单、易于实现、实时性好,逐渐成为根据视频目标跟踪技术的主流方法。

基于模板匹配的跟踪方法主要是通过一定的准则来寻找目标与候选模型之间的相似度来确定目标的最终位置。最经典的基于模板匹配的方法就是均值漂移算法,虽然经典的均值漂移方法运行速度快,在简单场景中的跟踪效果良好,但在场景复杂或目标运动状态变化迅速时跟踪效果变差。针对于此, Xu Yanming 等人<sup>[1]</sup>将严重分割的权重进行结合,意

在不降低实时性的条件下解决严重遮挡问题;为了减小相似背景对目标的干扰, Li Ning 等人<sup>[2]</sup>引入目标与环境的比例系数,以确保目标模型的准确更新。而且,在某些基于匹配的方法中可以根据跟踪情况实现核窗口宽度的自动调整。

然而,尽管有不少学者研究了目标跟踪过程中的遮挡或形变问题,但很少有人会同时考虑遮挡和形变问题。针对于此,本文提出了一种基于超像素与动态图匹配的视频跟踪方法,以超像素作为系统处理的基本单元,并结合图谱匹配进行视频目标跟踪。由于在构建图模型时融合了结构信息,因而实现了对目标的有效跟踪,并在处理目标形变、遮挡旋转和背景干扰等问题时具有较高的鲁棒性。

## 1 基于超像素的视频目标跟踪系统

本文采用基于模板匹配的方法,视频跟踪过程即是在连续帧间建立匹配的过程。它主要包含超像素生成、图像分割、图模型构建、图模型间的匹配、模

型更新等几个部分。原理框图如图1所示。首先采用融合信息熵的SLIC方法对输入的每一帧图像进行超像素生成,并将超像素作为后续处理的基本单元。其次采用graph cuts<sup>[3]</sup>图割算法,并融合在线分类器的分类预测结果,获得标签值为1的候选目标超像素集。然后采用图构建方法,运用候选目标超像素集构建出一个既包含节点信息又包含边缘信息

的、能有效呈现目标内部结构信息的候选图模型。最后,将前一帧更新后的图模型作为预测图模型,运用谱方法通过构建相似度矩阵实现图模型间的匹配,并根据匹配的结果准确定位目标的最终位置。在更新模块中,实时地更新颜色直方图特征、目标图模型节点,使跟踪算法保持健壮,能适应环境及目标自身的变化。

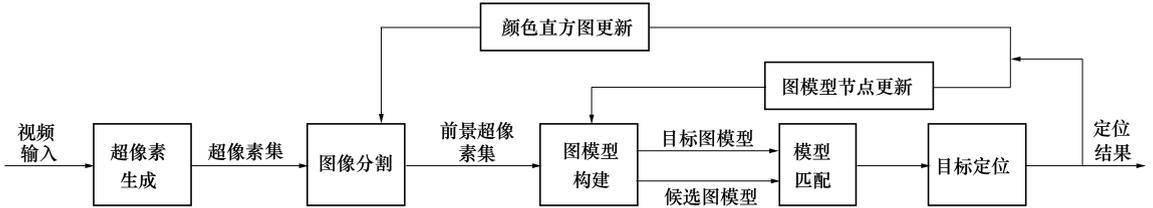


图1 原理框图

### 1) 基于熵的SLIC超像素生成方法

基于像素的视频跟踪方法由于计算量大,跟踪实时性很差。因此,本文采用一种基于超像素的视频跟踪方法。综合考虑计算代价、分割效果以及所需参数的复杂度,本文采用SLIC方法生成超像素。

SLIC方法生成超像素,是在以中心点为中心的某个邻域内根据距离值调整聚合中心,从而确定聚类中心点的位置。距离值的度量结合了颜色和空间信息,其度量方法如(1)式所示:

$$D_s = d_{lab} + \frac{m}{S}d_{xy} \quad (1)$$

式中, $D_s$ 是2个像素点的距离, $d_{lab}$ 和 $d_{xy}$ 分别表示的是颜色距离和空间距离;通过改变 $m$ 值的大小可以影响距离因素所占的比重; $S$ 为每个区域的边长,假设每幅图像有 $N$ 个像素,则 $S = \sqrt{N/K}$ 。

由于SLIC算法中运用lab色彩空间中的 $l$ 、 $a$ 、 $b$ 和像素点的位置信息来表达像素点的特征,在进行聚类时根据特征的相似性将像素点归为不同的超像素,所以特征选取的准确性对超像素的划分至关重要。而lab色彩空间中的 $l$ 表示亮度信息, $a$ 和 $b$ 表示颜色信息,亮度信息对像素点的描述并不是很明显,而且在后续跟踪的过程中,目标容易受到光照的影响。所以,本文采用信息熵信息代替亮度信息,进行超像素分割。图像的信息熵可以有效反映图像中的像素点与周围像素点的差异,进而将其与颜色信息与位置信息进行融合可以更有效地反映该像素点的特征,使得超像素对于背景干扰和目标旋转更具有鲁棒性。图像中选取以像素点为中心的邻域范

围求取信息熵值,作为当前点的熵信息。计算公式可以表示为:

$$h = -p_{ij} \log(p_{ij}) \quad (2)$$

式中, $p_{ij} = f(i,j) / \sum_{i=1}^3 \sum_{j=1}^3 f(i,j)$ , $f(i,j)$ 表示像素点 $(i,j)$ 处的灰度值。

不同像素点之间熵的距离可以表示为

$$d_h = |h_k - h_i| \quad (3)$$

式中, $d_h$ 表示2个熵之间的距离, $h_k$ 和 $h_i$ 分别表示超像素点 $k$ 和超像素点 $i$ 的熵。

距离 $D_s$ 可以重新表示为

$$D_s = \alpha d_{lab} + \beta d_h + \lambda d_{xy} \quad (4)$$

式中, $\alpha$ 、 $\beta$ 、 $\lambda$ 为常数,满足条件 $\alpha + \beta + \lambda = 1$ 。

### 2) 融合graph cuts和LASVM的图像分割方法

视频跟踪需要从超像素集合 $\{T_i\}$ 中挑选出标签为1的超像素集合 $\{T_i\}_{i=1}^{n_0}$ ,即生成候选超像素集。为了得到精确的候选超像素,本文采用graph cuts方法对超像素进行分离。同时,在能量函数项中融合LASVM<sup>[4]</sup>分类器的预测结果。LASVM是SVM分类器基础上发展起来的,能够更好地适应对连续变化的图像进行分类。这使得分割的结果更加精确,降低了误分割的概率。

基于graph cuts的图像分割方法中能量函数包含区域项和边界项,本文中将LASVM分类器的分类结果融合到区域项中,从而当能量函数达到最小时,可以得到更加精确的分割结果。

基于graph cuts图像分割方法中的能量函数包含区域项和边界项,本文中将LASVM分类器的分

类结果融合到区域项中,从而当能量函数最小化时,可以得到更加精确的候选目标超像素集合。

3) 图模型间的匹配将前一帧更新后的预测超像素和候选超像素分别以图的形式表示。其中,超像素表示图节点, $\xi$ -邻域内超像素之间的距离表示图的边缘,通过预测图模型和候选图模型之间的匹配实现目标跟踪。同时,预测图模型是通过不断更新目标图中的节点得到的,候选图模型是通过当前帧图像进行图像分割得到的。匹配的方式是通过构造相似度矩阵  $\mathbf{A}$ ,求解其最大特征值对应的特征向量。

在  $\mathbf{A}$  中,对角线元素表示预测图和候选图中节点之间的距离,非对角线元素表示预测图和候选图中对应边缘之间的关系。 $\mathbf{A}$  中的对角线元素  $\Omega_1(c_{i'v'})$  和非对角线元素  $\Omega_2(c_{i'v'}, c_{j'v'})$  分别表示如下:

$$\Omega_1(c_{i'v'}) = \exp\left\{-\frac{1}{\varepsilon_1^2}(D(f_i, f_{i'}))^2\right\} \quad (5)$$

式中,  $\varepsilon_1 = \frac{1}{\sqrt{2}}$ ,  $D(\cdot)$  表示目标图模型颜色特征  $f_i$  与候选图模型中颜色特征  $f_{i'}$  之间的卡方距离。 $f$  表示的是超像素的 HSV 颜色特征,与 RGB 颜色特征相比,HSV 更加符合人类视觉对色彩的感知。从视觉的角度看,该特征更加有利于外观颜色特征的表达。

$$\Omega_2(c_{i'v'}, c_{j'v'}) = \exp\left\{-\frac{1}{\varepsilon_2^2} \|(l_i - l_j) - (l_{i'} - l_{j'})\|_2^2\right\} \quad (6)$$

式中,  $\varepsilon_2 = \frac{r}{\sqrt{2}}$ ,  $r = \sqrt{W \cdot H / N_s}$ ,  $l_i$  与  $l_j$  表示目标图模型中两节点位置信息,  $l_i - l_j$  表示两节点之间的距离,  $l_{i'}$  与  $l_{j'}$  表示候选图模型中两节点的位置信息,  $l_{i'} - l_{j'}$  表示两节点之间的距离。因此,  $\Omega_2(c_{i'v'}, c_{j'v'})$  表示目标图与候选图中节点之间边缘的对应关系。该信息可以反映出目标发生一定形变时的边缘变化,从而在一定程度上,对形变具有鲁棒性。

矩阵  $\mathbf{A}$  中虽然包含了目标的外观颜色特征以及结构边缘信息,但是,将目标图中的任意一个节点与候选图中的每个节点都进行计算会造成  $\mathbf{A}$  的维度过大,而且也不具有实际意义,因此,通过  $k$ -最近邻法对待匹配的超像素点进行约束,对于每一个节点,只取 5 个距离最近的节点进行计算,进而对图中的节点进行限制。同时,为了进一步减小背景的干扰,对目标图和候选图中的对应节点之间的距离和角度进行了一定的阈值限定。

采用谱技术<sup>[6]</sup>求解矩阵  $\mathbf{A}$  的特征向量,谱技术与二次方程方法相比复杂性小,易于计算。公式表示如(7)式所示

$$\begin{aligned} \mathbf{x}^* &= \arg \max_x \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\ \text{s.t. } &\mathbf{x}^T \mathbf{x} = 1 \end{aligned} \quad (7)$$

求解出的最大特征值对应的特征向量  $\mathbf{x}^*$  代表了对应每一个对应成功的可能性  $P(c_{i'v'})$ ,但是由于所有可能性的总和为 1,所以,特征向量中的值并不是二元化的,为了将其二元化以便找出可能的目标超像素,本文中运用贪心算法将  $\mathbf{x}^*$  转化为只含有 0 和 1 的特征向量  $\mathbf{z}^*$ 。取出向量  $\mathbf{z}^*$  中值为 1 的元素集合,根据相应的位置进行加权定位。

#### 4) 目标更新

为了适应目标的动态变化,保证跟踪的准确性,需要对颜色直方图、预测图节点进行更新。颜色直方图的更新主要是保留初始帧的颜色信息,以便于准确跟踪。

更新预测图中节点时,将图中节点分为 3 种状态:①节点  $i$  不与预测图中的任意节点匹配,或到预测图中任意节点的几何距离满足一定的阈值条件,则被视为新生节点;②节点  $i$  能够成功匹配其他节点则被视为稳定节点;③如果一个节点在连续 5 帧都没有与其他节点匹配,则被视为死亡节点。

当一个目标矩形框确定之后,新生节点将会被添加到预测图中,稳定节点将会被保留在预测图中,死亡节点将会从预测图中删除,进而实现预测图模型中节点的更新。

## 2 仿真与结果分析

实验是通过 C++ 语言在 VS2012 上编程实现的,计算机的配置为 Intel(R) Core(TM) CPU 2.20Hz, 32 位 Window7 系统,内存为 2GB。所选用的数据主要是具有形变、遮挡、旋转以及一定的背景相似性特点的数据,同时将本文的跟踪算法与当前跟踪领域的一些主要跟踪算法进行比较(如 CT 算法, Frag 算法, MIL 算法, TLD 算法以及 DGT 算法<sup>[6]</sup>),通过定性和定量分析,可以验证该算法的优越性。

#### 1) 定性分析

定性分析主要是分析不同算法产生的跟踪框对目标跟踪的准确程度。本文中选用 2 组典型的视频数据(bolt, lemming)进行定性分析,具体跟踪结果

如图2所示。针对上述6种算法,分别用不同的线型表示,其中CT和MIL算法分别与Frag和TLD线型一样,但是CT和MIL算法颜色为灰色,Frag和TLD颜色为黑色。所选的视频数据中,目标主要经历了形变、遮挡、旋转以及相似背景干扰等变化。在bolt视频序列中,跟踪目标为位于图中间跑的较快的运动员,#0128帧目标经历了不同程度的背景干扰及沿轨道旋转的变化,CT算法、MIL算法、TLD算法以及Frag算法因为太过于注重外在信息而忽略了内部结构信息,因此目标丢失。而本文方法更能准确跟踪目标。在lemming视频序列中,跟踪目标为矩形框较集中的棕色玩具,#0320帧背景对目标

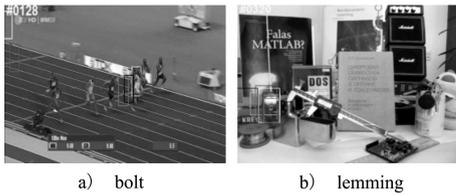


图2 不同算法在不同视频序列上的跟踪结果

产生相似性的干扰,同时出现严重遮挡,由于熵信息的运用,对比可以看出本文方法的跟踪框的准确性明显好于其他跟踪算法。但本文方法可以准确的跟踪到目标。

## 2) 定量分析

在定量分析中,本文采用2种方法对跟踪算法进行定量分析,一种是中心误差准则,它可以定义为目标实际位置与手动标注位置之间的距离的均方根。由于当误差距离太大时,跟踪矩形框已经基本偏离目标,不具有研究意义。所以,本文仅统计中心误差距离小于25的帧图像的平均中心误差作为判断依据。计算结果如表1所示;另一种方法是成功跟踪的帧数,即成功率大于0.5的帧数目,其中,跟踪的成功率可以定义为  $SC = \frac{area(R_T \cap R_C)}{area(R_T \cup R_C)}$ ,  $R_T$  和  $R_C$  分别为跟踪算法和手动标注的目标矩形框,  $area$  则表示目标跟踪状态的面积。当  $SC$  大于0.5时,则表明成功的跟踪了目标。统计结果如表2所示。从表2可以看出,本文方法的跟踪性能优于其他算法。

表1 中心误差率

视频序列	CT	Frag	MIL	TLD	DGT	本文方法
basketball	89.11	13.02	91.92	213.86	5.92	5.90
bolt	363.80	183.38	393.54	90.92	7.66	6.93
david3	88.66	13.55	29.68	208.00	6.39	6.38
lemming	32.25	126.87	12.06	15.99	6.15	6.15
football1	20.71	15.70	5.62	45.36	11.30	6.26

表2 成功跟踪的帧数

视频序列	CT	Frag	MIL	TLD	DGT	本文方法
basketball	191	505	200	18	708	712
bolt	2	37	4	51	302	307
david3	88	205	172	26	217	229
lemming	909	542	1083	793	1279	1281
football1	6	25	58	29	25	59

总体看来,本文方法和DGT算法的性能较为优越。但进一步分析可以看出,与DGT算法相比,本文方法由于增加了熵信息,对相似背景的干扰具有一定的鲁棒性,因此提高了匹配的精度,增加了成功匹配的超像素数目,如表2所示。但由于熵信息着重于提高了匹配成功的超像素数目,对中心点的准确性仅有略微提高。因此,单就计算中心误差而言,本文方法略优于DGT算法,如表1所示。总之,本

文方法对整体性能有所提高。

## 3 结论

本文通过充分利用超像素以及信息熵的优点,将需要跟踪的目标特征充分的表现出来,使其具有较强的鲁棒性。本文通过图模型的构建充分表现出超像素点之间的边缘关系,通过预测图模型和候选

图模型之间的边缘以及颜色特征之间的匹配,使得目标不仅在发生形变和遮挡的情况下具有鲁棒性,同时对旋转和背景干扰也具有较好的处理效果。对

比实验结果表明,本文方法与其它跟踪算法相比跟踪效果更好。

## 参考文献:

- [1] Xu Yanming. An Improved Mean-Shift Moving Object Detection and Tracking Algorithm Based on Segmentation and Fusion Mechanism[C]//IEEE Conference on Systems Process and Control, 2013:224-229
- [2] Li Ning, Zhang Dan, Gu Xiaorong, et al. An Improved Mean Shift Algorithm for Moving Object tracking[C]//IEEE Conference on Electrical and Computer Engineering, 2015: 1425-1429
- [3] Yuan Jun, Tang Shuming, Wang Fei, Zhang Hong. A Robust Road Segmentation Method Based on Graph Cut with Learnable Neighboring Link Weights[C]//IEEE Conference on Intelligent Transportation Systems, 2014: 1644-1649
- [4] Antoine Bordes, Seyda Ertekin, Jason Weston, et al. Fast Kernel Classifiers with Online and Active Learning[J]. Journal of Machine Learning Research, 2005, 6(3):1579-1619
- [5] Egozi A, Keller Y, Guterman H. A Probabilistic Approach to Spectral Graph Matching[J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2013, 35(1):18-27
- [6] Cai Z, Wen L, Lei Z, et al. Robust Deformable and Occluded Object Tracking with Dynamic Graph[J]. IEEE Trans on Image Processing, 2014, 23(12): 5497-509

## Video Tracking Method Jointing Superpixel and Dynamic Graph Matching

Zhang Junchang<sup>1,2</sup>, Zhou Yanling<sup>1</sup>, Wan Jinjin<sup>2</sup>

(1.School of Electronics Information, Northwestern Polytechnical University, Xi'an 710072, China)  
(2.Science and Technology on Electro-Optic Control Laboratory, Luoyang 471000, China)

**Abstract:** Focusing on the problem of target deformation, occlusion, rotation and background interference, a video tracking method jointing superpixels and dynamic graph matching was proposed in this paper. Firstly, superpixels were generated by the simple linear iterative clustering analysis method integrating the local entropy feature, so that we can get superpixels that the edge fit better. Secondly, the candidate target superpixels was generated by graph cuts method, in which the LASVM classifier was combined with the graph guts method in order to make the separation of foreground and background more accurately. Thirdly, when the graph modle was constructed, we make full use of the geometric information of the target to solve the problem of the occlusion and deformation effectively. Meanwhile, the constraints were introduced to reduce the dimension of the affinity matrix, so that the computational complexity was reduced. Theoretical analysis and simulation results show that our method has strong robustness and better tracking accuracy when deals with the occlusion and deformation, and the proposed method is good in dealing with the target rotation and a certain degree of background interference, compared with currently other video tracking methods.

**Keywords:** target tracking; information fusion; simple linear iterative clustering; superpixels; graph cuts