

# 基于角特征和 Mean-shift 的车辆跟踪方法

何得平, 朱光喜, 赵广州

(华中科技大学武汉光电国家实验室, 武汉 430074)

**摘要:** 提出一种基于角特征点和 Mean-shift 的车辆跟踪方法, 对不断改变尺寸的车辆目标进行有效跟踪, 利用角特征点把模板目标构造成为若干个同尺寸的子窗口, 每个子窗口单独地按 Mean-shift 算法进行跟踪, 并寻找其相应的中心位置, 通过关联过程确定车辆目标在整个视场中的运行轨迹。实验结果表明, 该方法简单、可靠, 可以稳定地跟踪车辆目标。

**关键词:** 车辆跟踪; Mean-shift 算法; 跟踪窗口; 角特征点

## Vehicle Tracking Method Based on Corner Feature and Mean-shift

HA Dac-Binh, ZHU Guang-xi, ZHAO Guang-zhou

(Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074)

**【Abstract】** A vehicle tracking method based on corner feature and Mean-shift is presented for efficient tracking object that continuously changes in size. The same in size's kernel windows of model object are built based on corner features. The Mean-shift algorithm is carried out one by one. According to the tracking result, the object's final position is estimated. Experimental results show this method is simple, reliable, and can track the vehicle stably.

**【Key words】** vehicle tracking; Mean-shift algorithm; tracking window; corner feature point

### 1 概述

智能交通系统(Intelligence Transportation System, ITS)是未来交通发展的趋势。车辆跟踪是视频交通流量检测系统的一个重要组成部分。运动车辆跟踪的目的在于确定车辆的运动轨迹, 即要在前一帧检测到的车辆和当前帧检测到的车辆之间建立一一对应的关系。车辆跟踪方面存在的困难很大程度上是由实际环境中车辆大小、运动速度、运动轨迹、车辆颜色与背景颜色的相似程度、背景的稳定性以及视频数据所具有的特殊性、复杂性造成的。

过去十几年, 人们在车辆跟踪领域已经作了大量研究, 提出许多很好的算法。Mean-shift 算法是种有效的统计迭代算法, 通过迭代收敛于概率密度函数的局部最大值。在目标的实时跟踪领域起着非常重要的作用。文献[1-2]提出的 Mean-shift 算法以彩色直方图作为特征实现目标跟踪。文献[3]在计算目标特征分布直方图时, 用像素的位置转角及其到目标形心的归一化距离加权, 并将像素在局部坐标系下的特征转角作为新特征引入。虽然能够完成对运动车辆的可靠跟踪, 但是对一些应用场合, Mean-shift 算法还有许多需要进一步研究的问题, 如对于目标快速运动并快速增大或缩小尺寸的视频序列, 对连续 2 帧图像中的目标, 它们之间差别极大, 很难得到匹配。此外, 如果选择过小的窗口去跟踪就会丢失很多目标的重要信息, 反之, 如果选择过大的窗口, 那么会填入很多背景信息并且增加计算量。这些都会导致 Mean-shift 跟踪算法丢失目标。对于这些问题, 文献[4]利用视频前后 2 帧中目标的角点匹配估计刚性物体的仿射模型参数, 并由此参数来更新核窗宽, 但是该算法只适用于相邻帧间的运动很小的情况。文献[5]将多尺度图像的信息量度量方法引入到运动目标中, 根据信息量的变化自动选取跟踪窗的

大小, 但是只适合非刚性物体的跟踪。文献[6]提出特征适应和跟踪窗口尺度适应算法, 改进传统 Mean-shift 目标跟踪算法, 使其能跟踪存在明显尺度变化的目标。但是这些算法主要根据目标的轮廓来限制背景信息填入跟踪窗口, 以提高跟踪效果, 而它们未能解决同一个目标在连续 2 帧的尺度变化差异而导致搜索失败的问题。

在车辆跟踪时, 经常会出现车辆远离或逼近摄像头造成目标下一帧被放大或缩小而常常会导致目标的丢失的问题。为解决此问题, 依据车辆具有的明显几何特征和角特征点对于刚性物体能够很好地刻画其空间结构, 而且易于检测的特点, 本文以这些角特征点为中心把前一帧的模板目标构建成为若干个同尺寸的子窗口, 用每个子窗口单独按 Mean-shift 算法去跟踪, 便找到目标对每个子窗口的相应中心位置。将获得的点迹与当前时刻的角点进行比较并确定对应关系, 从而确定车辆目标在整个视场中的运行轨迹。

### 2 Mean-shift 算法

对在  $n$  维欧氏空间  $X$  中的有限序列  $S$  来说, 样本数据点  $x \in X$  处的样本均值定义为

$$m(x) = \frac{\sum_{s \in S} K(s-x)w(s)}{\sum_{s \in S} K(s-x)w(s)} \quad (1)$$

其中,  $K$  为核函数;  $w$  为样本的权值函数。差值  $m(x)-x$  被称为 Mean-shift 矢量, 反复将数据点朝着 Mean-shift 矢量方向

**作者简介:** 何得平(1973-), 男, 博士研究生, 主研方向: 图像压缩和处理, 智能交通系统; 朱光喜, 教授、博师生导师; 赵广州, 博士

**收稿日期:** 2009-11-06 **E-mail:** hadacbinh@yahoo.com

移动直至收敛的过程被称为 Mean-shift 算法。具体步骤如下：

**步骤 1 计算模板特征模型：**

$$q_u(y_0) = C_h \sum_{i=1}^{n_h} k \left( \left\| \frac{y_0 - x_i}{h} \right\|^2 \right) \delta [b(x_i) - u] \quad (2)$$

其中,  $y_0$  为目标模板的中心;  $\{x_i\}$ ,  $i=1,2,\dots,n_h$  是模板中的各个像素位置; 函数  $b$  是  $R^2 \rightarrow \{1,2,\dots,m\}$  映射, 把相应位置像素的颜色进行  $m$  级量化;  $\delta$  是 delta 函数;  $C_h$  是归一化常数。

**步骤 2 在当前帧初始位置处计算候选目标特征模型：**

$$p_u(y_1) = C_h \sum_{i=1}^{n_h} k \left( \left\| \frac{y_1 - x_i}{h} \right\|^2 \right) \delta [b(x_i) - u] \quad (3)$$

其中,  $y_1$  是当前帧跟踪窗口中心位置。

**步骤 3 计算数据点  $x_i$  的权值：**

$$w_i = \sum_{u=0}^{m-1} \frac{q_u}{p_u} \delta [b(x_i) - u] \quad (4)$$

**步骤 4 计算候选目标新位置：**

$$y_2 = \frac{\sum_{i=1}^{n_h} x_i w_i g \left( \left\| \frac{y_2 - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i g \left( \left\| \frac{y_2 - x_i}{h} \right\|^2 \right)} \quad (5)$$

**步骤 5** 如果  $\|y_2 - y_1\| < \varepsilon$  或达到了最大迭代次数, 则停止, 否则  $y_1 \leftarrow y_2$  转到步骤 2。 $\varepsilon$  的选择应使  $y_1$  与  $y_2$  的间距小于 1 个像素。

### 3 算法实现

本文提出的基于 Mean-shift 车辆跟踪方法需要做如下几个工作: 背景的构建, 道路识别和车辆检测, 角特征点的提取, 子窗口的构建, Mean-shift 算法的跟踪及关联过程。

#### 3.1 背景构建、道路识别和车辆检测

本文采用文献[7]提供的基于区域生长处理的运动车辆提取方法, 进行初始背景构建、道路识别和车辆检测, 如图 1 所示。

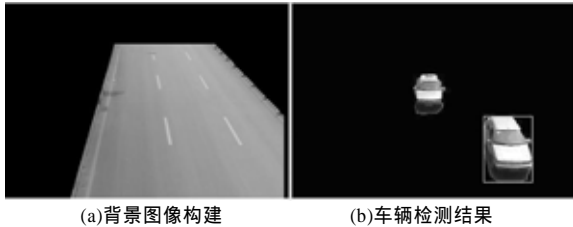


图 1 背景图像构建和车辆检测结果

#### 3.2 角特征点提取

由于角特征点对于刚性物体能够很好地刻画其空间结构, 且易于检测, 因此本文采用车辆的角特征点进行跟踪。考虑到角特征点往往非常凌乱, 且数目众多, 因此, 需要进行一次预处理, 即通过膨胀算法将部分角特征点合并成一个角特征块, 然后利用细化算法将其收缩至中心点, 以便准确地提取出所需要的角特征点。

车辆的角特征点可以理解为  $X,Y$  方向上的边缘线条的交点, 对此可采用 Sobel 算子进行提取。通过 Sobel 算子处理后得到的图像中含有大量的拐角点, 必须对其进行膨胀处理以便将邻近的拐角点合并。本文采用一个  $2 \times 2$  的正方形图块作为膨胀处理的结构元素, 经过膨胀操作可以得到一定数量的区域块, 并且这些区域块中包含角特征点。作为一种合理的考虑, 不妨认为该区域块的中心点就是角特征点。为此还需要对所有的区域块加以细化, 使其收缩至一个点, 这需要利

用细化技术。

细化技术常被用来获取区域的骨架。其中, 最常用的中轴变换(Medial Axis Transform, MAT)就是种用来确定物体骨架的细化技术。当然, 此时所要获得的不是整个区域的骨架, 而是其中心点, 因此, 本文对中轴变换进行一些修改, 以便直接获得其中心位置, 具体算法可参见文献[8]。图 2 描述了角特征点的提取结果。

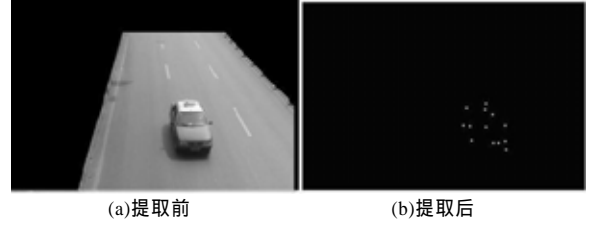


图 2 角特征点的提取结果

#### 3.3 目标跟踪与关联过程

在前一帧目标的角特征点  $O^{t-1}_i$  确定后, 对模板目标以角特征点为中心构建若干个同尺寸(如  $5 \times 5$ )的子窗口。这些子窗口按行列顺序去排列, 若下一个窗口的中心位置位于前面的某一窗口就取消该窗口, 然后用这些子窗口单独按照 Mean-shift 算法进行搜索该窗口在当前帧的中心点  $I'_i$ 。最后用这些中心点  $I'_i$  通过关联过程与当前帧目标的角特征点  $O'_i$  进行比较并确定对应关系。其大致步骤为: 先在当前帧的分割后车辆序列中提取目标的角特征点  $[O'_i]_k$  ( $k$  表示为第  $k$  个车辆区域), 并按顺序排列。用前一帧的角特征点搜索到的中心点  $I'_i$  集合与当前帧的角特征点  $[O'_i]_k$  集合一对一做比较, 若  $I'_i$  集合的一个点与  $[O'_i]_k$  集合的一个或多个点的距离较小(小于设定的某个阈值), 可以认为它们是同一目标的角特征点, 将此点  $I'_i$  的位置放入邻近点  $[E'_i]_k$  集合中, 并记录它们的相互关系; 如两点的距离较大(大于设定的某个阈值), 则认为两者没有什么联系, 继续对前一帧的角特征点搜索到的中心点  $I'_i$  集合中的下一个点做再次比较判断, 重复进行, 直至对前一帧的角特征点搜索到的中心点  $I'_i$  集合的所有数据都做比较。最后根据邻近点集合中的点数确定目标的最终位置, 即若

$$M = \frac{|\sum E'_i|_k}{\sum I'_i} \times 100\% \quad T$$

则认为第  $k$  个区域是所跟踪的目标。

### 4 实验结果与分析

利用 VC6.0++ 语言编写程序在 Pentium 1.46 GHz PC 上对目标不断增大和缩小尺寸的视频序列进行了跟踪测试。为评价车辆跟踪算法的性能, 用传统 Mean-shift 算法和本文算法针对不断增大和缩小尺寸的车辆目标在不同的工作环境、不同的交通场景做大量实验。图像大小为  $352 \times 240$  像素, 角特征点的跟踪窗口大小为  $5 \times 5$ 。对 RGB 分量分别进行 16 级量化处理得到的核直方图大小为  $16 \times 16 \times 16$ , 实验结果见表 1。

表 1 不同光照环境及交通场景下流量检测评价结果

评价指标	光照充分的环境		光照不足的环境	
	高速公路	城市交通	高速公路	城市交通
实际流量	300	300	300	300
检测流量(传统 MS 算法)	295	287	279	268
检测流量(本文算法)	298	293	289	278
检测精度(传统 MS 算法)/(%)	98.3	95.7	93.0	89.3
检测精度(本文算法)/(%)	99.3	97.6	96.3	92.6

(下转第 200 页)