

基于并行框架的鲁棒自适应前景检测算法^①

陈文竹¹, 陈岳林², 蔡晓东³, 华娜³

¹(桂林电子科技大学 机电工程学院, 桂林 541004)

²(桂林电子科技大学 信息科技学院, 桂林 541004)

³(桂林电子科技大学 信息与通信学院, 桂林 541004)

摘要: 视频监控数据 TB 级的增长, 从海量视频数据中高效准确的分离出视频监控场景中的运动物体, 是计算机视觉领域的研究重点和挑战. 提出了基于云平台的视频数据处理的并行计算框架及一种改进的基于混合高斯模型(GMM)的自适应前景提取算法, 通过对混合高斯分布的自适应学习和在线 EM(期望最大化)算法获得最优参数组合, 并将改进算法融合到视频处理并行计算框架. 实验结果表明, 该方法不但能大大提高视频处理的效率, 并对复杂环境下准确提取前景目标也有良好的鲁棒性.

关键词: 视频监控; 并行计算; 混合高斯模型; 自适应学习; 在线 EM 算法

Robust Adaptive Foreground Detection Algorithm Based on Parallel Framework

CHEN Wen-Zhu¹, CHEN Yue-Lin², CAI Xiao-Dong³, HUA Na³

¹(Electromechanical Engineering College, Guilin University of Electronic Technology, Guilin 541004, China)

²(Institute of Information Technology, Guilin University of Electronic Technology, Guilin 541004, China)

³(School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: Video surveillance data is increasing quickly, it's a challenge to separate out moving objects from a massive video data in the field of computer vision. The article designs and implements a Cloud-based distributed video processing framework, and proposes an improved adaptive foreground extraction algorithm based on gaussian mixture model(GMM). The method obtains the optimal parameters by adaptive learning gaussian distribution and online EM(Expectation Maximization) algorithm, and it fuses the improved algorithm to distributed video processing framework. The experiment shows that the method can not only greatly improve the efficient of video processing but also accurate extract foreground targets under complex environment, and it has good robustness.

Key words: video surveillance; distributed computing; GMM; adaptive learning; online EM algorithm

大数据不再是一个新术语, 基于 Hadoop 云平台的并行计算框架能有效的处理结构化数据, 如文本等, 但对于非结构化的视频图像数据, 由于数据之间的耦合性, 海量视频数据的并行化处理面临挑战. 然而, 在计算机图形学领域, 视觉算法的计算量相对较大, 对算法效果上的少量改进, 也会以相当大的复杂度为代价, 当数据为海量时, 计算资源的瓶颈往往限制了复杂算法的应用. 文献[1,2]提出了基于 Hadoop 的视频分布式转码方案, 它们只是对视频图像在结构上进行处理,

并没有实现基于视频内容的挖掘算法. 前景检测在视频分析中是一个非常重要的工作, 它旨在自动分离视频监控背景场景中的运动物体, 后续的流程, 如跟踪和行为识别, 都极大地依赖于分割结果. 在过去的几年里, 大量的工作都致力于这一主题, 并且在不同场景的应用上提出了许多研究方法. 文献[3,4]总结了这些方法. 在这些方法中, 混合高斯模型(GMM)对复杂场景具有一定的适应性和鲁棒性因而常作为视频监控中前景提取的方法, 文献[5]提出了基于混合高斯模型

① 基金项目: 广西自然科学基金(2013GXNSFAA019326)

收稿时间: 2014-07-29; 收到修改稿时间: 2014-09-23

的运动目标检测方法,但固定的参数更新速率和固定的参数值无法快速适应复杂场景的变化. 文献[6]提出了一种改进的运动目标检测方法,在 GMM 训练阶段采用 EM(期望最大化)算法对高斯分布参数进行最大似然估计,使初始化后高斯分布参数很快收敛. 文献[7]提出了动态控制高斯分布的方法,结合场景的变化控制高斯分布数,减少了算法运算时间,但经验参数的设置对高斯分布数的更新有很大影响. 将改进的混合高斯背景建模算法与并行计算框架结合,在满足前景检测鲁棒性的同时,不但能提高前景提取的实时性,而且能有效解决单台计算机处理海量数据能力不足的问题.

本文提出了基于 Hadoop 云平台的视频并行处理框架及基于混合高斯模型自适应在线学习的前景提取方法. 首先,深入分析视频数据结构,设计并实现使非结构化视频数据结构化,便于 Hadoop 并行处理的相关视频处理接口,通过并行计算提高计算机视觉算法处理的速度. 其次,通过对高斯分布的在线学习,提高了对复杂场景的适应性,并结合在线 EM 算法对高斯分布参数的实时训练,使分布参数能较快的收敛,加快背景建模的速度.

1 Hadoop分布式系统及混合高斯模型

1.1 Hadoop 分布式系统

Hadoop 是开源的分布式框架,本文核心框架基于 Hadoop2.X,其主要由分布式文件系统(HDFS)和基于 YARN 架构的分布式计算系统(MapReduce)组成.

HDFS 在 Hadoop 体系结构中处于最基础的地位,实现海量数据的存储,并提供了数据的备份机制,能扩展容量,使存储数据具有高度的容错性和可扩展性. 对数据的操作主要是大规模的流式读取和小规模的随机读取,因而视频数据可以不加任何处理上传到 HDFS 上.

MapReduce 是 Google MapReduce 的开源实现,文献[8]详细介绍了 MapReduce 并行计算技术及其发展. 基于 YARN 架构 MapReduce 框架如图 1 所示. 用户向 ResourceManager(资源调度管理器)提交计算任务, ResourceManager 在一个 NodeManager(节点管理器)上产生一个 ApplicationMaster, ApplicationMaster 向 ResourceManager 申请计算资源, ResourceManager 根据资源状况及分配策略,分配合适的 Container(资源)给 ApplicationMaster, ApplicationMaster 得到资源后,管理这些资源完成计算任务. 同时 ResourceManager 不

断的搜集 NodeManager 的使用情况. MapReduce 以函数方式提供 Map 与 Reduce 函数来完成分布式计算,利用一个键值对<key, value>集合完成输入输出. 数据通过序列化和反序列化完成与键值对的交互.

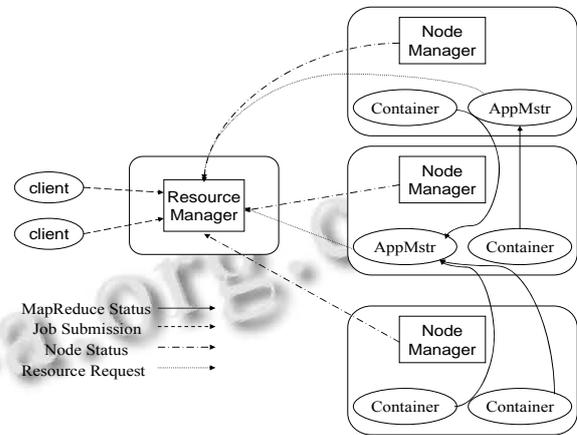


图 1 基于 YARN 架构的 MapReduce 架构图

HDFS 与 MapReduce 相互配合,完成高度容错和高吞吐的海量数据解决方案.

1.2 混合高斯模型

图像中每个像素点的值短时间内都是围绕与某一中心值一定距离内分布,根据统计规律,如果数据点足够多,这些点呈高斯分布. 混合高斯模型是指像素点值存在多个中心位置,每个位置产生一个高斯分布. 如果像素点偏离中心值较远,则为前景,如果像素值偏离中心值很近(在一定的方差范围内)则为背景. 通过对背景建模,分离前景和背景,从而达到运动物体检测的目的.

混合高斯模型对图像中的每一个像素点定义 m 个模型表示其所代表的信息. 每个模型用一个高斯函数表示. 则可以认为图像中像素点 X 在时刻 t 的概率为:

$$P(X_t) = \sum_{i=1}^m \omega_{i,t} \times g(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

$\omega_{i,t}$ 表示权重 $\omega_{i,t} > 0$ 且 $\sum_{i=1}^m \omega_{i,t} = 1$, $\mu_{i,t}$ 表示均值, $\Sigma_{i,t}$ 为协方差矩阵, $g(X_t, \mu_{i,t}, \Sigma_{i,t})$ 为高斯分布的概率密度函数:

$$g(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{i,t}|^{\frac{1}{2}}} e^{-\frac{1}{2} \Delta^2} \quad (2)$$

n 表示图像的通道数,单通道图像 $n=1$,多通道图像 $n=3$,假设三个通道颜色信息相互独立,协方差矩阵表示为 $\Sigma_{i,t} = \sigma_{i,t}^2 I$, I 为单位矩阵, Δ 代表马氏距离,

$$\Delta = \sqrt{(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_t - \mu_{i,t})}$$

当新的观测值出现混合模型参数需要更新, 将 X_t 同存在的 m 个高斯分布使用马氏距离比对如下:

$$|X_t - \mu_{i,t}| < \lambda \delta_{i,t} (\lambda \text{取} 3) \quad (3)$$

若满足条件, 则参数更新如下:

$$\mu_{i,t+1} = (1 - \rho) \mu_{i,t} + \rho X_t \quad (4)$$

$$\delta_{i,t+1}^2 = (1 - \rho) \delta_{i,t}^2 + \rho (X_t - \mu_{i,t})^T (X_t - \mu_{i,t}) \quad (5)$$

$$\omega_{i,t+1} = (1 - \alpha) \omega_{i,t} + \alpha \quad (6)$$

α 和 ρ 代表更新速率和更新因子, 对于未匹配的分分布参数, 均值和方差不变但权重减少. 如果像素值没有匹配的分分布, 需要将优先级最低的高斯分布用一个新分布替代: $\omega_{i,t+1} = \omega_{init}$, $\delta_{i,t+1} = \delta_{init}$.

要准确提取前景, 背景模型需要选择最能代表背景的分分布, 背景的改变相对于前景没有那么频繁, 因此代表背景的分分布应有较大的权重和较小的方差. 将 m 个分分布按 $\omega_{i,t+1} / \sigma_{i,t+1}^2$ 降序排列选择前 B ($0 < B < m$) 个分分布构成新的背景模型:

$$B = \arg \min_b \left(\sum_{i=1}^b \omega_{i,t+1} > Th \right) \quad (7)$$

Th 表示决定组成背景的最小信息量的阈值.

2 视频并行处理框架及改进的GMM算法

2.1 视频并行处理框架

如 1.1 节所述, Map 与 Reduce 函数完成分布式计算, 利用 $\langle \text{key}, \text{value} \rangle$ 集合完成输入输出. 数据需要在并行框架下传递, $\langle \text{key}, \text{value} \rangle$ 必须能够进行序列化, 序列化将结构化的对象转为字节流以便于通过网络进行传输或写入持久存储过程. Hadoop 通过 Writable 接口对数据进行序列化. Hadoop 实现了一系列的 Writable 类型, 如 Text, IntWritable, ByteWritable, DoubleWritable 等, 但对于视频图像等复杂的非结构化数据, 需要创建一个新的 Writable 类型. 视频数据解码后是由一帧一帧连续的图像数据组成, 对视频的处理, 实际上是对每一帧图像的处理, 本文实现了新的序列化方法 Image. Image 实现将输入流重构为图像数据, 并将图像数据转化为二进制流, 主要通过重写 readFiles() 和 write() 方法完成. readFiles() 读取图片信息, write() 写出图片信息.

MapReduce 编程过程如下: InputFormat 完成数据输入规范检查, 对数据输入进行分片, 并通过 RecordReader 将数据记录逐一读出转化为 Map 阶段的

$\langle \text{key}, \text{value} \rangle$ 输入. 经过 Map 阶段和 Reduce 阶段处理, OutputFormat 完成数据的输出规范, 依靠 RecordWriter 明确写入 HDFS 中的具体格式. 对于视频流的处理, 需要定制对应的输出格式. 基于视频流处理的 MapReduce 流程如图 2 所示: VideoInputFormat 类完成对数据的划分, FrameRecordReader 类对视频块解码, 并将解码帧转化为 $\langle \text{Text}, \text{Image} \rangle$ 类型的键值对作为 Map 阶段输入, 其中 Text 为文件块名和帧数组成的字符串, Image 为值类型. MapReduce 阶段完成计算机视觉算法实现. 本文针对视频监控数据, 完成前景提取, 目标跟踪, 目标检测算法, 最后将检测结果以图片形式保存到分布式文件系统 HDFS 中, 因此定义 ImageRecordWriter 及 ImageOutputFormat 类, 将输出键值对转化为图片保存.

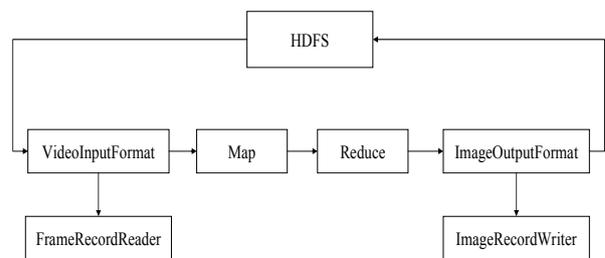


图 2 基于视频流处理的 MapReduce 流程图

2.2 改进的 GMM 算法

2.2.1 混合高斯模型的自适应学习

混合高斯模型中, 高斯分分布数 m 为一固定值, 对复杂变化的场景, 固定的分分布数往往不能适应场景的变化. 对快速变化变化的区域, 有较多变化的前景目标, 需要更多的高斯模型适应场景, 设置混合高斯模型数量过大, 又会消耗大量的计算时间, 影响算法的实时性. 自适应高斯混合模型能鲁棒地处理动态场景和在视频监控中重复出现的运动物体, 但这一方法仍需要手动设置初始化经验参数.

本文提出混合高斯模型的自适应学习方法. 通过对当前观测值的扰动进行学习来评估高斯分分布数, 将混合高斯模型数 m 迭代计算实时更新. 具体步骤如下:

初始化阶段令混合高斯模型数 $m=1$, 权值为 1, 均值和方差采用平均值初始化, 采用前 N 帧各像素点的值计算均值和方差, 能更准确的获得背景信息, 减少后续迭代计算的工作量, N 值可取 30 到 50 之间.

当 t 时刻一组新的观测出现, 计算当前观测值与已有模型的马氏距离如下:

$$D = \frac{|X_t - \mu_{i,t}|}{\delta_{i,t}} \quad (8)$$

设定阈值 $T_\lambda = 3$, 当 D 小于 T_λ , 满足背景模型条件, 按 EM 算法^[9]自适应更新, 当最小 D 值大于 T_λ 时, 当前观测值远离模型均值, 对下一时刻, 建立一个新的高斯分布模型如下:

$$m = m + 1 \quad (9)$$

$$\omega_{i+1,t+1} = \frac{1}{t+1} \quad (10)$$

$$\mu_{i,t+1} = X_{t+1} \quad (11)$$

$$\delta_{i,t+1}^2 = \sum_{i=1}^m \frac{\omega_{i,t} g(X_t, \mu_{i,t}, \sum_{i,t})}{P(X_{t+1})} \delta_{i,t}^2 \quad (12)$$

其中方差值为 t 时刻方差的后验概率权值均值。

更新后的混合高斯模型归一化处理如下:

$$\omega_{i,t+1} = \frac{t}{t+1} \omega_{i,t} \quad (13)$$

2.2.2 自适应的高斯混合模型参数估计

高斯分布参数为固定值, 需要设置合理的更新率 α 对高斯混合模型参数进行调整, 更新率较大造成运动目标的丢失, 更新率过慢又会影响背景建模的速率。

EM 算法^[9]是一种能够保证在搜索空间收敛到局部极大值的迭代方法, 它由两个步骤组成: E 步估计隐含变量, M 步估计其它参数, 交替将极值推向最大。EM 算法适应于数据成批处理, 对于连续更新的视频文件, 需要使用在线 EM 算法处理, 文献[10]提供了具体推到过程及理论。由于高斯分布的动态变化, 需将在线 EM 算法改写如下:

$$\omega_{i,t+1} = \omega_{i,t} + \frac{1}{t}(p_{i,t+1} - \omega_{i,t}) \quad (14)$$

$$\mu_{i,t+1} = \mu_{i,t} + \frac{p_{i,t+1}}{t \times \omega_{i,t}}(X_{t+1} - \mu_{i,t}) \quad (15)$$

$$\delta_{i,t+1}^2 = \delta_{i,t}^2 + \frac{p_{i,t+1}}{t \times \omega_{i,t}}[(X_{t+1} - \mu_{i,t})^2 - \delta_{i,t}^2] \quad (16)$$

其中 $p_{i,t+1}$ 为 X_{t+1} 的后验概率密度表示如下:

$$p_{i,t+1} = \frac{\omega_{i,t} g(X_t, \mu_{i,t}, \sum_{i,t})}{P(X_{t+1})} \quad (17)$$

2.3 并行框架与算法的融合

根据 Hadoop 云平台数据流的处理特征, 数据的并行处理是基于数据并行处理串行的模式, 视频并行处理框架封装了数据的并行分片、解码及容错处理、负载均衡等细节, 可以把大数据量的计算自动地分发到

各个节点执行。因此, 计算机视觉算法能简单的融入视频并行框架, 而不必考虑视频图像数据复杂的并行分布策略。

对于海量监控视频的前景提取, 视频并行处理框架读取 HDFS 中视频块文件, 视频分片并通过 FrameRecordReader 类完成视频解码, 并将解码视频帧通过 <key, value> 的形式循环读入作为 Map 阶段的输入, map() 函数阶段完成前景提取、检测跟踪算法, 完成任务后, 输出存入分布式文件系统 HDFS(本文不需要考虑输出图片的合并, 故不需设置 Reduce 操作)。

3 实验结果对比分析

3.1 并行处理框架效率测试

测试并行视频处理框架的高效性, 设计实验研究并行视频处理框架处理效率。实验采用 3 台联想主机 ubuntu32 位系统, 配置为: 4 核 CPU, 内存为 4G, 1T 硬盘, Hadoop2.2.0 框架, 选取摄像头固定的情况下, 某一户外场景中所拍摄的监控视频。视频大小为 720*480, 帧率为 25 帧/s。

对不同大小的视频数据在单机模式及两个子节点和三个子节点下完成运动目标检测所耗时间记录如表 1 所示。

表 1 视频处理所耗时间

视频文件大小(M)	单机模式(s)	两个子节点(s)	三个子节点(s)
240	80	88	68
360	91	94	77
720	171	168	133
960	361	305	172
1240	424	344	187

比较并行模式同单机模式下处理不同大小视频的提速比如图 3 所示, 横坐标为视频块大小, 纵坐标为单机模式所耗时间与并行模式下不同子节点的比值。

由表 1 与图 3 实验结果可知当视频文件较小时, 单机模式视频处理效率与并行模式下两个子节点处理视频所需时间大致相等, 甚至低于并行模式视频处理所需时间, 三节点的处理效率高与前者。随着数据量的增加, 并行处理模式处理效率高与单机模式。实验表明, 并行框架适用于视频大数据的处理, 在视频数据海量时, 增加节点数目会提高 hadoop 的视频处理并行框架的效率。

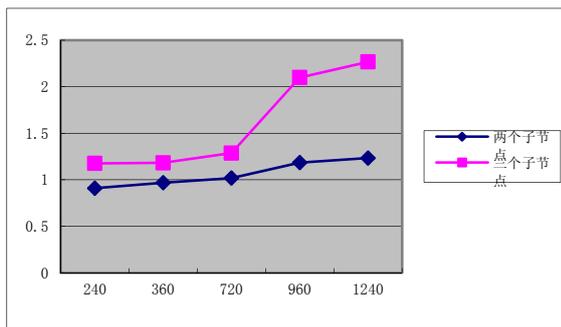


图 3 提速比值图

3.2 目标检测测试

Hadoop 平台由 java 语言编写, JavaCV^[11]是一款开源的视觉处理库,是对各种常用计算机视觉库的封装后的一组 jar 包. 本文算法是在 JVM 下实现. 为验证本文方法在复杂场景下的鲁棒性,在 eclipse 和 JavaCV 平台下对 GMM 方法和本文方法进行前景检测实验. 选取 WallFlower 数据集^[12]数据进行测试. 视频序列大小为 160*120,帧率为 25 帧/s.

前景检测的性能指标有:前景检测正确率 Fgr 、错误率 Fge 和检测精度 Pre . 计算方法如下:

$$Fgr = \frac{TP}{TP + FN} \times 100\% \quad (18)$$

$$Fge = \frac{FP}{FP + TN} \times 100\% \quad (19)$$

$$Pre = \frac{TP}{TP + FP} \times 100\% \quad (20)$$

其中, TP 为正确前景点数, FP 为错误前景点数, TN 为正确背景点数, FN 为错误背景点数. Fge 越大,表示前景图像噪声越多, Fgr 与 Pre 越大表示前景检测性能越好.

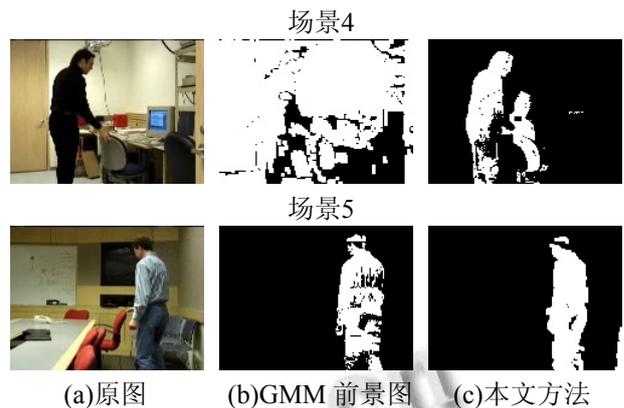


图 4 前景检测结果

表 2 前景检测性能分析对比

性能指数	场景类型(%)					
	场景 1	场景 2	场景 3	场景 4	场景 5	
GMM	Fgr	81.8	66.9	70.1	24.6	87.4
	Fge	18.0	41.1	33.8	77.6	13.0
	Pre	78.3	65.4	67.8	27.2	78.3
本文方法	Fgr	89.0	92.3	87.8	67.8	94.0
	Fge	11.7	8.1	13.4	32.1	9.7
	Pre	71.3	74.3	75.8	65.3	78.1

分别对不同场景下的图像序列提取前景目标,将混合高斯背景建模与改进方法进行比较,对比两种方法的优劣. 图 4 为视频序列前景检测结果.

根据前景检测提取的二值化图像,计算不同场景下的前景检测的正确率,错误率,和检测精度如表 2 所示.

由图 4 和表 2 可知相对于 GMM 方法本文方法在前景检测正确率和检测精度方面都更占优势. 从检测速度上, GMM 的检测平均速率为 18 帧/s, 本文方法的检测平均速率为 21 帧/s, 本文方法高于 GMM 算法. GMM 方法由于更新率和模型数量固定,导致背景未完全更新,不能准确提取前景目标. 本文方法通过对高斯混合模型各参数的自适应动态控制,能更准确的提取出前景目标.

4 结语

本文从两个方面对前景检测算法进行了改进. 首先,本文根据视频数据结构的特点,设计实现了基于 Hadoop 的非结构化视频数据处理并行计算框架,通过并行计算提高视频处理的效率. 其次,高斯分布参数

的自适应动态学习及在线 EM 算法,提高了前景检测算法的准确性和鲁棒性.未来的研究中,有关数据倾斜,资源管理调度及并行处理框架数据节点稳定性方面还有待优化.

参考文献

- 1 杨帆,沈奇威.分布式系统 Hadoop 平台的视频转码.计算机系统应用,2011,20(11):80-85.
- 2 Kim MJ, Cui Y, Han SH, Lee HK. Towards efficient design and implementation of a hadoop-based distributed video transcoding system in cloud computing environment. International Journal of Multimedia and Ubiquitous Engineering, 2013, 8(2): 213-224.
- 3 Brutzer S, Hoferlin B, Heidemann G. Evaluation of background subtraction techniques for video surveillance. Computer Vision and Pattern Recognition, 2011: 1937-1944.
- 4 万纓,韩毅,卢汉清.运动目标检测算法的探讨.计算机仿真,2006,23(10):221-226.
- 5 Stauffer C, Grimson WEL. Adaptive background mixture models for real-time tracking. Computer Vision and Pattern Recognition, 1999, (2): 246-252.
- 6 Liu Z, Huang KQ, Tan TN. Foreground object detection using top-down information based on EM framework. Image Processing, 2012, 21(9): 4204-4217.
- 7 李百惠,杨庚.混合高斯模型的自适应前景提取.中国图象图形学报,2013,18(12):1620-1627.
- 8 应毅,刘亚军.MapReduce 并行计算技术发展综述.计算机系统应用,2014,23(4):1-7.
- 9 王爱平,张功营,刘方.EM 算法研究与应用.计算机技术与发展,2009,9(19):108-110.
- 10 Pernkopf F, Bouchaffra D. Genetic-based EM algorithm for learning Gaussian mixture models. Pattern Analysis and Machine Intelligence, 2005, 8(27): 1344-1348.
- 11 <https://code.google.com/p/javacv/>.
- 12 Toyama K, Krumn J, Brumitt B, Meyers B. Wallflower: principles and practice of background maintenance. International Conference Vision. 1999, 1(1): 255-361.