

面向视频监控的自动行人检测

李新江 龚 勋 李天瑞 赵 涛 熊 伟

(西南交通大学信息科学与技术学院 成都 610031)

摘 要 为了解决目前行人检测技术的检测速度和准确性之间的平衡问题,对基于视频的行人检测技术进行了研究,提出了利用 LUV 颜色空间信息与 C^4 行人检测算法相结合的视频自动行人检测方法(LUVC⁴)。首先利用 C^4 行人检测算法快速遍历视频的每帧图像,当得到的窗口置信度在可疑区间时,再进一步对该窗口做 LUV 颜色空间检测。如果两次检测的加权和分数满足阈值,则判别为行人。通过大量实验表明,该方法在检测速度几乎能达到 C^4 速度的同时,还能在 FPPI 为 0.1 时降低约 9% 的漏检率。

关键词 行人检测, LUV, C^4 , 置信度

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.12.057

Automatic Pedestrian Detection Based on Video Surveillance

LI Xin-jiang GONG Xun LI Tian-rui ZHAO Tao XIONG Wei

(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

Abstract To address the problem that the technologies of pedestrian detection can't achieve the balance between detecting speed and accuracy, this paper aimed to research on pedestrian detection under video surveillance. An automatic video pedestrian detection method (denoted as LUVC⁴) was proposed by combining LUV color space information and C^4 pedestrian detection algorithm. Firstly the C^4 algorithm is used to rapidly traversal each frame of the video image. The LUV color space is taken to detect this window further when the confidence score of detect window is in the suspicious interval. If the weighted sum of scores of the two detections satisfies the threshold, it is discriminated as a pedestrian. A large number of experiments show that the detection speed of the proposed method nearly reaches that of C^4 and it can greatly decrease the missrate about 9% when false positive per image equals to 0.1.

Keywords Pedestrian detection, LUV, C^4 , Confidence score

1 引言

行人检测是计算机视觉的一个重要研究领域,它可以定义为:判断输入图片(或视频帧)是否包含行人,如果有则给出位置信息。视频人数统计在客流量商业数据分析、公共安全等诸多方面有着重要的应用价值^[2]。当前比较主流的行人检测方法是基于统计训练的方法,即利用计算机对行人特征和非行人特征进行训练和学习,得到分类器,然后利用分类器对待测图像进行检测^[3-5]。Dalal 等在 2005 年提出的 HOG^[4](Histogram of Oriented Gradient)是当前使用最广泛的描述符,它刻画了图像局部梯度幅值和方向,但也存在特征维数高和计算时间开销大等缺点^[3]。Enzweiler 等^[6]和 Dollar^[7-9]等在公共数据集上比较了行人检测技术经典算法之间的性能差异,前者得出:HOG 比 Haar Wavelet 和 LRF(Local Receptive Fields)效果都好,后者在多种实验条件下得出:

Walk 等人^[10]提出的 MULTIFTR+MOTION 能达到相对好的效果。苏松志等^[2]把行人检测技术中的特征分为 3 类:底层特征、基于学习的特征和混合特征。

目前行人检测研究的重点主要集中在:(1)如何提取特征;(2)如何解决精确分类问题;(3)如何提高检测速度,达到实时性需求。虽然 Benenson 等^[11-13]采用了 GPU 来辅助加速,但其不能普遍适用。本文针对目前面向视频监控的行人检测技术中检测速度与准确性还不能有效平衡的问题,同时考虑到 C^4 ^[14]算法的描述符 CENTRIST^[1]不具备颜色信息和 C^4 算法具有几乎实时性的优点,提出了利用颜色空间信息 LUV^[15-17]与 C^4 行人检测算法相结合的视频自动行人检测方法。增加 LUV 颜色信息可以提高检测准确性,能有效解决检测速度和准确性之间的平衡问题,但相应增加了计算复杂度。

到稿日期:2013-06-25 返修日期:2013-08-16 本文受国家自然科学基金(61202191),中央高校基本科研业务费专项资金(SWJTU12CX095)资助。

李新江(1988-),男,硕士生,主要研究方向为行人检测、图像处理,E-mail:lxjbb777@163.com;龚 勋(1980-),男,副教授,硕士生导师,主要研究方向为认知图像处理、三维人脸建模、模式识别;李天瑞(1969-),男,教授,博士生导师,主要研究方向为数据挖掘与知识发现、粒计算与粗糙集、云计算;赵 涛(1987-),男,硕士生,主要研究方向为人脸特征点跟踪、图像处理;熊 伟(1985-),硕士生,主要研究方向为医学图像处理。

2 LUV^C 行人检测算法简介

2.1 颜色空间

根据不同应用可以把颜色空间分为 3 大类: 计算机图像颜色空间、TV 系统颜色空间、CIE(International Commission on Illumination)颜色空间。计算机图像颜色包括 RGB、HSV 和 HSI 等, TV 颜色包括 YIQ、YUV 和 YCbCr 等, CIE 颜色包括 XYZ、LUV 和 Lab 等。

Wang 等^[16]在比较各种颜色空间用于行人检测之后, 得出结论(按降序排列): CIE-Lab、CIE-LUV、HSV、HSI 对于行人检测有很好的效果。

本文采用 LUV 颜色空间, 同样根据文献^[15]定义 4 种方向滤波, 如图 1 所示。在每种滤波器下计算邻近 cell(cell 的详细设置见 3.3 节)的直方图不相似度 $dissim(\cdot)$ (选用直方图相交方式进行匹配), 它定义为:

$$d(i, j, k) = dissim(cell_{i,j}, cell_{i_k, j_k}), k \in \{1, 2, 3, 4\} \quad (1)$$

其中, k 表示滤波器的种类, $cell_{i,j}$ 表示训练样本(或者检测窗口)中位置为 i 行 j 列的 cell 的颜色直方图, (i_k, j_k) 代表不同滤波器相对于 $cell_{i,j}$ 的位置偏移。

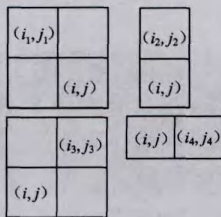


图 1 4 种方向滤波器

筛选出某一种方向滤波器下的最大不相似度, 见式(2)。

其中, 把 i 和 j 对应扩展成 I 和 J , 见式(3)和图 2。

$$H(i, j, k) = \max_{I, J} (d(I, J, k)), k \in \{1, 2, 3, 4\} \quad (2)$$

I, J 有下列规定:

$$\text{if } k = \begin{cases} 1, & I \in \{i, i+1\}, J \in \{j, j+1\} \\ 2, & I \in \{i, i+1\}, J \in \{j-1, j, j+1\} \\ 3, & I \in \{i, i+1\}, J \in \{j-1, j\} \\ 4, & I \in \{i-1, i, i+1\}, J \in \{j, j+1\} \end{cases} \quad (3)$$

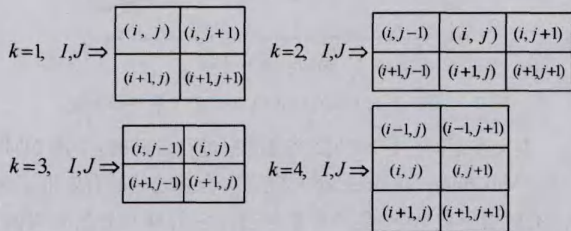


图 2 i, j 扩展成 I, J

最大不相似度标准化: 首先计算 i 行 j 列的 cell 具有的能量

$E(i, j) = \sum_{k=1}^4 H(i, j, k)^2$, 标准化公式如下:

$$N(i, j, k) = \frac{H(i, j, k)}{\sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} E(m, n)} \quad (4)$$

2.2 C⁴ 行人检测算法

2.2.1 计算 Sobel 图像^[14]

Sobel 图像能很好地反映原始图像的边缘信息, 如图 3 所示。

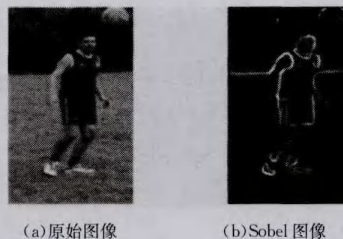


图 3

2.2.2 CENTRIST 描述符^[1]

Wu 等人提出的 CENTRIST 描述符, 即 CENSus Transform hISTogram, 最早应用于场景分类。其主要步骤如下:

对输入图像进行 CT(Census Transform)变换, CT 变换与 LBP(Local Binary Patterns)^[18]变换类似, 如图 4 所示。CENTRIST 描述符就是 CT 变换之后图像的直方图。

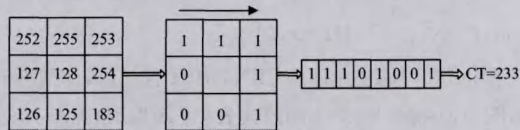


图 4 CT 变换示意图

所谓 C⁴ 就是 human contour using a cascade classifier and the CENTRIST visual descriptor, 具体见 3.1 节。

3 LUV^C 实现行人检测

本文采用 108×36 作为检测窗口的大小和训练图像的大小, C⁴ 训练时采用线性 SVM 和 HIK SVM^[19], LUV 颜色空间分类器采用线性 SVM 训练。

3.1 C⁴ 快速检测

把窗口分成 9×4 个 block(每个 block 大小为 12×9), 把邻近的 2×2 个 block 作为一个超 block, 这与 HOG^[4]中特征提取时所设置的 cell 和 block 的关系一样, 如图 5 所示。因此每个窗口就有 8×3 个超 block, 所以得到的向量应为 $24 \times 256 = 6144$ 维。

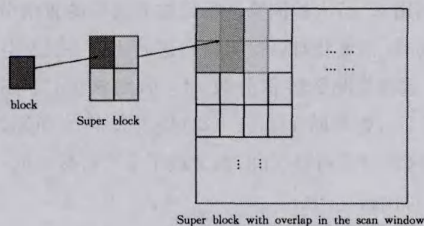


图 5 block 与超 block 以及窗口之间的关系

对于一个已经训练的线性分类器 $\omega \in R^{6144}$, ω 可以作为 $\omega_{i,j} \in R^{256}$ ($1 \leq i \leq 8, 1 \leq j \leq 3$) 的级联, 待检测窗口的特征向量 f (当然也看作 $f_{i,j}$ 的级联) 如果满足式(5), 则判定为行人。

$$\omega^T f = \sum_{i=1}^8 \sum_{j=1}^3 \omega_{i,j}^T f_{i,j} \geq \theta \quad (5)$$

用 (h, ω) 表示检测窗口, 则 block 表示为 $(h_s, \omega_s) = (h/9,$

$w/4$), 超 block 表示为 $(2h_s, 2w_s)$ 。

对于给定的图像, S 表示 Sobel 变换图像, C 表示 CT 变换图像, 检测窗口的左上角坐标为 (t, l) , 所以式(5)可以变换为:

$$\sum_{i=1}^8 \sum_{j=1}^3 \sum_{x=2}^{2h_s-1} \sum_{y=2}^{2w_s-1} \omega_{i,j}^{C(t+(i-1)h_s+x, l+(j-1)w_s+y)} \quad (6)$$

式中, $C(x, y)$ 表示 CT 图像的 (x, y) 坐标的像素, $\omega_{i,j}^k$ 表示 $\omega_{i,j}$ 的第 k 个部分, $x=2, \dots, 2h_s-1$ 是为了排除边界。

建立辅助图像 $A_{i,j}$ ($1 \leq i \leq 8, 1 \leq j \leq 3$), (x, y) 处的像素按式(7)设置。

$$A_{i,j}^{x,y} = \omega_{i,j}^{C(x,y)} \quad (7)$$

所以式(6)改写为:

$$\sum_{i=1}^8 \sum_{j=1}^3 \left(\sum_{x=2}^{2h_s-1} \sum_{y=2}^{2w_s-1} A_{i,j}^{t+(i-1)h_s+x, l+(j-1)w_s+y} \right) \quad (8)$$

式(8)与高效子窗口搜索法 (Efficient Subwindow Search, ESS)^[20] 相类似, 这里定义一个中间图像 $A(x, y)$:

$$A(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \omega_{i,j}^{C((i-1)h_s+x, (j-1)w_s+y)} \quad (9)$$

当 $n_x=8, n_y=3$ 时, 式(9)变为:

$$\omega^T f = \sum_{x=2}^{2h_s-1} \sum_{y=2}^{2w_s-1} A(t+x, l+y) \quad (10)$$

采用式(10)能快速计算 CENTRIST 描述符, 详细图解见图 6, 图 6(a) super block 中的值代表 CT 图像值, 图 6(b) super block 中的值代表在分类器 ω 中前者的 CT 值所对应的值。

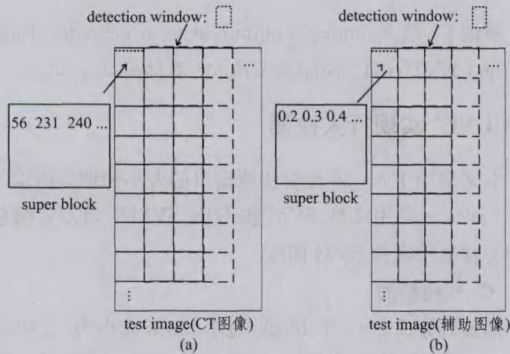


图 6 快速计算 CENTRIST 描述符

3.2 定义可疑区间

当利用 C^1 行人检测算法得到检测窗口的置信分数 s_1 在某个区间时, 就重新对该检测窗口进行 LUV 颜色空间检测, 若 LUV 重新检测的置信分数 $s_2 < 0$, 则该窗口判定为无行人, 若 $s_2 > 0$, 则判断 $1/2(s_1 + s_2)$ 是否大于 0, 详细过程见表 1。这个区间就是可疑区间, 本文取了 4 个可疑区间(用 sp 表示)进行对比。

$$|sp| \leq k (k=0, 1, \dots, 0.4) \quad (11)$$

表 1 判断流程

if $s_1 \in sp$
if $s_2 \leq 0$, 判为非行人
else if $1/2(s_1 + s_2) > 0$, 判为行人
else 判为非行人
else if $s_1 > sp_{max}$ 判为行人
else 判为非行人

3.3 LUV 颜色空间参数设置

在对训练样本(或者检测窗口)进行 LUV 颜色空间特征计算时, 把窗口划分成 12×4 个 cell, 每个 cell 大小为 9×9 , 由于每个 cell 的特征有 4 维, 则每个样本的特征维数为 $48 \times 4 = 192$ 维。因为需要对样本中每个 cell 进行滤波处理, 所以需要对样本扩展两个 cell 的宽度和高度。LUV 直方图如图 7 所示。

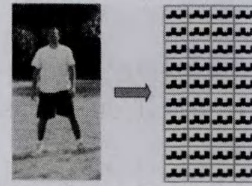


图 7 LUV 直方图

4 实验结果与分析

本文训练数据采用的是 INRIA^[4] 数据集, 选取正样本 2416 个, 负样本 12285 个, 如图 8 所示。测试数据集来源于 INRIA、PETS2009 两部分, 前者分辨率大部分是 480×640 , 后者分辨率为 720×576 。本文采用 2.93GHz 的 CPU 电脑。

在 INRIA 库上的结果如图 9 所示(其中 $k = sp_{max}$, $k=0$ 表示 C^1 算法)。



图 8 训练正样本来源于 INRIA 库

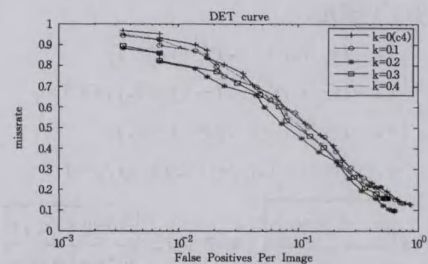


图 9 LUV C^1 与 C^1 在 INRIA 测试集上的评价对比

本文在对同一目标的多个检测框进行合并时, 采用 NMS (Non-Maximum Suppression) 算法^[8], 合并之后的有效框记为 R_d , 有效框分数 score 是所有参与合并它的框的分数的均值, INRIA 库中测试图像的标准位置记为 R_g , 若满足以下条件:

$$\frac{Area(R_d \cap R_g)}{Area(R_d \cup R_g)} > 0.5 \quad (12)$$

则判定该有效框属于 True Positive, 否则判定为 False Positive。一个 R_g 最多与一个 R_d 满足式(12), 设置不同阈值分数就可以得到 FPPI-miss rate 曲线。通过图 9 可以分析出, $sp_{max}=0.2$ 时效果最优。



图 10 C^4 与 $LUV C^4(k=0.2)$ 分别在 INRIA 测试集上的效果



图 11 C^4 与 $LUV C^4(k=0.2)$ 分别在 PETS2009_S2_L1_View006 测试集上的效果

本文算法($LUV C^4$)与 C^4 行人检测算法分别在 INRIA、PETS2009 库的实验对比结果如图 10、图 11 所示,结果表明本文方法在检测速度几乎能达到 C^4 速度的同时,还能大大减小漏检率。

结束语 本文分析了视频行人检测中检测速度和准确性之间的平衡问题的重要性,采用 LUV 颜色空间与 C^4 行人检测算法相结合进行行人检测,在 INRIA、PETS2009 库都取得了令人满意的检测效果。本文的工作还存在一定不足,下一步工作将研究快速提取特征以及特征精确分类等。

参 考 文 献

- [1] Wu J, Rehg J M. CENTRIST: A visual descriptor for scene categorization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1489-1501
- [2] 苏松志, 李绍滋, 陈淑媛, 等. 行人检测技术综述[J]. 电子学报, 2012, 40(4): 814-820
- [3] Zhu Q, Yeh M, Cheng K, et al. Fast human detection using a cascade of histograms of oriented gradients[C]//Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision Pattern Recognition, 2006: 1491-1498
- [4] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision Pattern Recognition, 2005: 886-893
- [5] 孙昫, 刘富强, 李志鹏. 基于空间梯度直方图的行人检测算法[J]. 中国图象图形学报, 2008, 13(10): 1825-1828
- [6] Enzweiler M, Gavrilu D M. Monocular pedestrian detection: Survey and experiments[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(12): 2179-2195
- [7] Dollár P, Wojek C, Schiele B, et al. Pedestrian detection: A benchmark[C]//Proceedings of the 2009 IEEE Conference on Computer Vision Pattern Recognition, 2009: 304-311
- [8] Dollár P, Wojek C, Schiele B, et al. Pedestrian detection: An evaluation of the state of the art[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(4): 743-761
- [9] Dollár P, Belongie S, Perona P. The fastest pedestrian detector in the west[C]//BMVC 2010. Aberystwyth, UK, 2010
- [10] Walk S, Majer N, Schindler K, et al. New features and insights for pedestrian detection[C]//IEEE Proceedings of the 2010 IEEE Conference on Computer Vision Pattern Recognition, 2010: 1030-1037
- [11] Benenson R, Mathias M, Timofte R, et al. Fast stixel computation for fast pedestrian detection [C] // Proceedings ECCV, 2012: 11-20
- [12] Benenson R, Mathias M, Timofte R, et al. Pedestrian detection at 100 frames per second[C]//Proceedings of the 2012 IEEE Conference on Computer Vision Pattern Recognition, 2012: 2903-2910
- [13] Benenson R, Timofte R, Van Gool L. Stixels estimation without depth map computation[C]//IEEE ICCV Workshops, 2011: 2010-2017
- [14] Wu J, Geyer C, Rehe J M. Real-time human detection using contour cues[C]//Proceedings of the 2011 IEEE International Conference on Robotics and Automation(ICRA), 2011: 860-867

- [15] Wang Q, Pang J, Liu G, et al. Color Maximal-Dissimilarity Pattern for Pedestrian Detection[C]//Proceedings of the 2012 21st International Conference on Pattern Cognition. 2012;1952-1955
- [16] Wang Q, Pang J, Qin L, et al. Justifying the importance of color cues in object detection: a case study on pedestrian[M]// The Era of Interactive Media. Springer, 2013;387-397
- [17] Van De Sande K E, Gevers T, Snoek C G. Evaluating color descriptors for object and scene recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32 (9): 1582-1596
- [18] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 971-987
- [19] Wu J. A fast dual method for HIK SVM learning[M]//Computer Vision-ECCV 2010. Springer, 2010, 552-565
- [20] Lampert C H, Blaschko M B, Hofmann T. Efficient subwindow search; A branch and bound framework for object localization [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(12): 2129-2142

(上接第 237 页)

方下载的 FaCT++、Pellet、RacerPro 的推理引擎时间对 RQ2 (将 RQ1 的最后一个条件去掉) 的分类进行对比。对每个查询测试了 10 次, 计算平均响应时间。经过测试, FaCT++ 在分类的推理运用上比 RacerPro 和 Pellet 效率更高, 对比时间如表 4 所列。在数据量较大时, 本文实验与 FaCT++ 进行对比, 对比时间如表 5 所列。

表 4 RQ2 各推理机分类推理时间对比(单位 s)

RDF 三元组数(个) 推理机	5000	10000	20000	50000
FaCT++	0.086	0.237	0.529	1.109
Pellet	21.951	86.542	134	261.276
RacerPro	3.626	6.62	9.366	15.098
TICM	3.442	4.821	5.475	6.576

表 5 RQ2FaCT++ 和本文分类推理的时间对比(单位 s)

RDF 三元组数(个) 推理机	100000	200000	500000	1000000	2000000	3000000
FaCT++	4.583	37.053	311.67	1256.357	内存溢出	内存溢出
TICM	8.069	11.56	29.224	32.449	76.404	410.556

通过实验对比发现, 在数据量较小时, 本文的效率不如 FaCT++, 但随着数据量的增大, 本文的效率会比 FaCT++ 高。本文的方法适用于用复杂集合运算定义的类的分类推理, 从实例出发, 只对需要推理的规则进行推理, 不产生中间结果; 而其他推理引擎在推理的过程中会对其他规则也进行推理, 在推理的过程中产生大量的中间结果, 之后再对结果进行过滤, 因而当数据量增大时, 时间效率会越来越差。但本文的方法只适用于个体分类推理, 如要完成其他类型的推理, 可与其他推理引擎嵌套使用。

4.3 TCRA 算法分析

采用查询处理时间来评价本方法中分类推理模型的效率。假设数据集中实例个数为 M , RDF 三元组数为 P , 每个三元组按照 RDFS 的 5, 7, 9, 11 规则进行扩充, 设时间为 Q , 查询句子中的所有 C_i^k 中 n 叠加得 N , 最坏的情况下要遍历树上所有的节点才能判断此实例是否属于这个类, 时间复杂度为 $O(PQ+MN)$ 。最好的情况下, 只要匹配第一个节点就可以判断此实例不属于这个类, 时间复杂度为 $O(PQ+M)$ 。

结束语 本文在描述逻辑的基础上提出基数限制集合的表示方式, 并在 OWL DL 语言中扩展了集合基数限制标签 `kCardinalitySetsOf`, 实现了类运算中除交集、并集、补集以外的基数 k 集合表示法, 并根据基数 k 集合的运算思想将分类运算转化成树型分类推理算法 (TCRA), 通过个体在 TCRA 中的遍历匹配, 判断个体的类型。本文以 RDF 数据格式存储

简历中的实体信息, 实现人才知识库的存储和分类。实验结果表明, 本文的方法在准确率和查询效率上达到了良好的效果, 能帮助用人单位更快、更准地找到所需要的信息, 提高了人才分类的效率。本文的 TCRA 本质上是一种两类个体分类器, 对于两类分类, 它的时间复杂度是线性的, 但是如果进行多类个体分类则必须构造多个 TCRA, 类别数目越多则分类时间越多, 当查询条件更改时需要重新建立推理模型。

参考文献

- [1] Tsarkov D, Horrocks L Optimised Classification for Taxonomic Knowledge Bases[C]// Proceedings of the 2005 International Description Logic Workshop (DL 2005). Manchester, UK, 2005
- [2] 王海龙, 马宗民, 严丽, 等. 支持模糊数据类型表示的模糊描述逻辑 F-SHOIQ(G)[J]. 计算机学报, 2009, 2(8): 1511-1524
- [3] 程经纬, 马宗民, 严丽, 等. 模糊描述逻辑知识库查询蕴涵的判定方法[J]. 计算机学报, 2012, 35(4): 767-785
- [4] Stoilos G, Stamou G, Pan J. Fuzzy Extensions of OWL: Logical Properties and Reduction to Fuzzy Description Logics[J]. International Journal of Approximate Reasoning, 2010, 51(6): 656-679
- [5] Stoilos G, Stamou G. Reasoning with fuzzy extensions of OWL and OWL 2[J]. Knowledge and Information Systems, 2013, 40(1): 205-242
- [6] Bobillo F, Delgado M, Gómez-Romero J. DeLorean: A reasoner for fuzzy OWL 2[J]. Expert Systems with Applications, 2012, 39: 258-272
- [7] 李言辉, 豫宝文, 陆建江, 等. 支持数量约束的扩展模糊描述逻辑复杂性研究 [J]. 软件学报, 2006, 5(17): 968-975
- [8] 邓志鸿, 唐世渭, 张铭, 等. Ontology 研究综述[J]. 北京大学学报, 2002, 35(5): 730-738
- [9] 石莲, 孙吉贵. 描述逻辑综述[J]. 计算机科学, 2006, 33(1): 194-197
- [10] Tsarkov D, Horrocks L. FaCT++ Description Logic Reasoner: System Description[C]// Proceedings of the International Joint Conference on Automated Reasoning (IJCAR'06). 2006; 292-297
- [11] Haarslev V, Moller R. Racer: A Core Inference Engine for the Semantic Web Ontology Language (OWL)[C]// Proceedings of the International Workshop on Evaluation of Ontology-based Tools. 2003; 27-36
- [12] Sirin E, Parsia B, Grau B, et al. Pellet: A practical OWL-DL reasoner[J]. Journal of Web Semantics, 2007, 5(2): 51-53
- [13] Guo Y, Pan Z, Heflin J. LUBM: A benchmark for OWL knowledge base systems[J]. Journal of Web Semantics, 2005, 3(2/3): 158-182