

# 一种智能手机上的场景实时识别算法

桂振文<sup>1</sup> 陈靖<sup>2</sup> 刘越<sup>2</sup> 王涌天<sup>1,2</sup>

**摘要** 目前常用的 SIFT 和 SURF 识别算法存在匹配时间长、运算量大和内存占用多等问题,无法满足实时移动检索应用.针对这些问题,本文提出了一种智能手机上的实时识别算法,通过缩短特征点检测时间和降低尺度空间特征点定位的复杂度,保证识别的实时性和准确性.实验结果表明,本算法能有效地运行在普通的资源受限智能手机上,具有较好的通用性;同时能实现对场景的实时识别,消耗内存资源也较少,适合在实际应用中使用.

**关键词** 移动检索, SURF 算法, SIFT 算法, 智能手机

**引用格式** 桂振文, 陈靖, 刘越, 王涌天. 一种智能手机上的场景实时识别算法. 自动化学报, 2014, 40(1): 83–91

**DOI** 10.3724/SP.J.1004.2014.00083

## A Real-time Recognition Algorithm of Scenes on Smartphones

GUI Zhen-Wen<sup>1</sup> CHEN Jing<sup>2</sup> LIU Yue<sup>2</sup> WANG Yong-Tian<sup>1,2</sup>

**Abstract** Currently, the used SIFT and SURF algorithms cannot meet the demand of higher real-time identification applications, and these algorithms have a lot of problems, including a long matching time, a large amount of memory usage and computational complexity and so on. In this paper, we propose a method for real-time recognition on a smartphone, through shortening the time of feature point detection and reducing the complexity of feature point location on scale space to ensure real-time identification and accuracy. The experimental results show that this algorithm can effectively run on resource-constrained ordinary smartphone with good versatility. At the same time, it can achieve real-time recognition of the scene and consume less memory resources, so it is suitable for using in practical applications.

**Key words** Mobile retrieval, SURF algorithm, SIFT algorithm, smartphone

**Citation** Gui Zhen-Wen, Chen Jing, Liu Yue, Wang Yong-Tian. A real-time recognition algorithm of scenes on smartphones. *Acta Automatica Sinica*, 2014, 40(1): 83–91

图像识别是近年来计算机视觉、移动增强现实和图像处理等领域的研究热点之一,在图像拼接、三维重建和物体跟踪定位等具体应用中均发挥重要的作用.图像识别的基本原理是:首先对目标图像进行特征点检测,然后再对特征点进行描述,最后通过特征点匹配来确定目标图像对应样本库中最相似的图像.

目前,对于智能手机上大规模图像识别有两

种研究思路:一种思路是采用客户端-服务器(Client/server, C/S)和传感器的思想,在智能终端实现视频捕捉、特征点提取、特征点描述和 GPS 信息获取等简单算法,在 PC 端实现带 GPS 信息特征点匹配和误匹配点剔除等计算量和占用存储空间大的复杂算法,通过无线通信来实现特征点的发送和识别结果的接收.这种方法易受到无线网络的影响,在网络拥塞的情况下效果较差.如文献 [1] 通过在智能终端提取带 GPS 信息的视频帧,在服务器端进行识别,文献 [2] 通过采用 TF-IDF 模型和分层单词树(Hierarchical-VOC)的算法在 PC 端实现了大规模图像的快速识别.文献 [3] 通过采用机器学习支持向量机(Support vector machine, SVM)的方法实现大规模图像的高准确率的识别.另一种思路采用压缩简化的思想,降低描述符存储和匹配的复杂性,用位表示特征向量和用汉明距离来度量特征点的相似性,该方法随着压缩比率增大识别精度也会降低,需要其他辅助算法(包括空间一致性、几何一致性检查等算法)加以判断.如文献 [4] 通过 Hash 函数将高维的特征向量进行压缩,用 30 bit 的汉明码来表示高维特征,减少了存储空间,同时将度量描述符相似的欧拉空间转换到汉明空间,用一条汉明指

收稿日期 2012-07-09 录用日期 2013-03-19  
Manuscript received July 9, 2012; accepted March 19, 2013  
国家高技术研究发展计划(863 计划)(2013AA013802),国家自然科学基金(61072096),国家科技重大专项基金(2012ZX03002004)资助  
Supported by National High Technology Research and Development Program of China (863 Program) (2013AA013802), National Natural Science Foundation of China (61072096), and National Science and Technology Major Project of China (2012ZX03002004)

本文责任编辑 周杰  
Recommended by Associate Editor ZHOU Jie  
1. 北京理工大学计算机科学与技术学院 北京 100081 2. 北京市混合现实与新型显示工程技术研究中心,北京理工大学光电学院 北京 100081

1. Department of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081 2. Beijing Engineering Research Center for Mixed Reality and Advanced Display, School of Optoelectronics, Beijing Institute of Technology, Beijing 100081

令就能很快进行特征点匹配。

对于特征点检测,研究者提出了大量的特征点检测算法。例如: Mair 等<sup>[5]</sup>提出了 AGAST (Adaptive and generic accelerated segment test) 多尺度角点快速检测算法,通过建立像素相似度的二进制决策树来确定角点,该算法对图像角点具有可重复性。Leutenegger 等<sup>[6]</sup>提出 BRISK (Binary robust invariant scalable keypoints) 特征检测算法,通过在连续尺度空间检测相邻区域灰度变化显著性点作为稳定的局部特征来实现。Rosten 等<sup>[7]</sup>提出了 FAST (Features from accelerated segment test) 检测算法,通过比较圆形区域中心点的像素与周围点像素的亮暗关系确定特征点,检测速度快,占用内存资源较少,比较适合实时应用。

对于特征点描述,为了使局部区域的特征能够对旋转、尺度缩放、光照和轻量级的仿射变换保持不变,研究者提出了很多局部特征点描述算法,常见的有以下几种:尺度不变特征转换特征描述 (Scale-invariant feature transform, SIFT)<sup>[8]</sup>、快速鲁棒局部特征 (Speeded up robust features, SURF)<sup>[9]</sup> 和最近提出的 BRIEF (Binary robust independent elementary features) 特征描述<sup>[10]</sup>,在计算速度和识别率等方面都超过了 SIFT 和 SURF。但是这些算法普遍存在计算量大、消耗内存多和匹配时间长的问题,无法直接移植到智能终端。

对于特征点匹配,常用的匹配方式有逐个匹配的方法 (Brutal force algorithm, BFA)、通过聚类求最近邻 (Approximate neighbor nearest, ANN) 方法 (如 KD-Tree<sup>[11]</sup> 算法、K-means<sup>[12]</sup> 算法) 和基于贝叶斯后验概率的方法 (如 R-Tree<sup>[13]</sup> 算法、Ferns<sup>[14]</sup> 算法) 等。R-Tree 和 Ferns 算法通过对特征点进行后验概率判断最大可能性是属于哪类特征点,需要大量的学习样本才能生成准确率比较高的判定树,并且在训练的时候会占用大量内存,如果对于单个场景识别智能手机还能承受,如果对于多个场景该类算法不太适合。KD-Tree 和 K-Means 方式是通过聚类的方式形成判定树,减少样本的比较次数,也不需要额外的大量训练样本,比较适合智能手机端本身的硬件条件。

根据以上的算法衍生了一些在智能手机上的很好应用。文献 [15] 中通过改进 SIFT 算法和 Ferns 算法,在智能手机上实现了对自然场景的识别与跟踪,实时性较强,但是支持的场景数量有限。文献 [16] 中在智能手机上实现了基于 SURF 算法的一种匹配系统,实现方式比较容易,通用比较强,但是对高分辨的图像匹配的速度较慢。文献 [17] 中用 J2ME 开发了一种移动增强现实应用,能实现实时场景识别和叠加增强显示的三维物体,但缺点是该

应用是基于标志点的,对自然环境有影响。

基于以上问题,本文提出了支持多场景移动实时识别算法,通过分析智能手机本身的处理能力和内存容量,有针对性地设置图像的 Octave 和 Octave layers 数量,减少计算量;通过改进 FAST 提点算法,减少特征点提取时间;最后通过稳定点查找算法,减少特征点匹配的时间。实验结果表明,该算法在运行速度方面比 SIFT、SURF 算法提高了 40%,在内存的占用方面比 SIFT、SURF 算法少了 50%,在识别速度方面相对 SURF 算法提高了一倍多,但是比 SIFT 算法提高了一个数量级,因此更适合实时性较高的移动搜索。

## 1 智能手机终端的特征提取算法

### 1.1 智能手机平台

目前智能手机发展非常迅速,基本上都集成了高速处理芯片、大容量存储器、内置了百万像素的高清摄像头、蓝牙无线接口和 WIFI 无线局域网接口卡,甚至一些高端的设备中还集成了大量的传感器,如 GPS 定位传感器、磁力传感器和陀螺仪感应器等。这些不断提高的硬件设备,给计算机视觉和图像处理领域带来了新的发展空间,许多原来运行在 PC 上的计算机视觉算法,通过改进和优化,能有效地运行在智能终端的平台上。

但是,相对于现在装配了高速处理芯片和大容量存储芯片的服务器,智能手机还存在以下不足:较低的图像分辨率、缺少专门的图形加速器、没有浮点运算单元、尤其是低端处理器和低容量存储器等,限制了在智能手机上进行复杂运算和大规模计算。

针对智能手机存在的以上问题,本节后续部分的特征点检测和稳定点查找算法分别从计算量和内存空间两方面进行优化,使算法满足计算量小、占用空间低等特性,适合运行在智能手机上。

### 1.2 特征提取

在目标识别与物体检索的任务中,特征点检测算法是其首要的基础和关键步骤。特征点检测就是查找和定位图像中的具有代表性的像素点或者与周围像素迥异的斑点等。本文提出了改进的 FAST 提点算法和稳定点查找算法,适合智能手机平台的同时兼顾了提点算法的实时性。

#### 1.2.1 FAST 提点算法

FAST 检测算法由 Rosten 等<sup>[7,18]</sup>提出,特征点通过检测中心点与周围像素点的明暗程度来确定,简单的计算步骤使提点算法具备快速和高效性。

FAST 特征检测算法,通过给定一个以候选点为中心的 Bresenham 圆,对圆上的各个像素点逐一

检查, 如果有  $n$  个连续的点都比中心点亮 (或者暗), 则中心点被确定为一个亮特征点 (或者暗特征点). 暗点、相似点和亮点以式 (1) 来进行判别.  $P$  点是要判断的像素点坐标,  $c$  代表 Bresenham 圆上的顺时针标号的像素点,  $V_c$  代表该点的像素值,  $V_p$  代表中心点像素值,  $comp_c$  代表圆上的点与中心点的像素值比较结果,  $t$  为阈值. 图像纹理变化不一样, 阈值选择也不同. 通过给定一个阈值, 比较圆环上的像素值与中心点像素值大小来确定中心点是亮点还是暗点.

$$comp_c = \begin{cases} -1, & V_c < V_p - t \text{ (暗点)} \\ 0, & V_p - t < V_c < V_p + t \text{ (相似点)} \\ l, & V_p + t < V_c \text{ (亮点)} \end{cases} \quad (1)$$

图 1 所示  $P$  点是候选特征点, 根据式 (1), 将圆周围的像素点与中心点逐一进行比较. 在阈值为 9 的情况下, 若连续 9 个像素点比中心点亮, 则该中心点为 FAST 特征点.

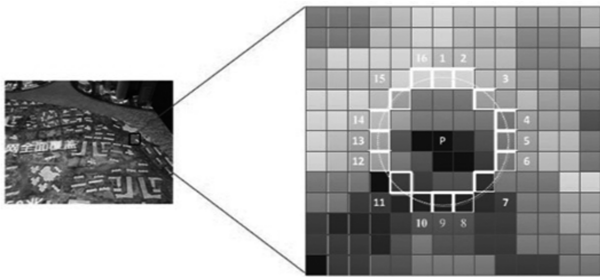


图 1 FAST 特征点检测  
Fig.1 FAST feature point detection

### 1.2.2 改进的 FAST 算法

FAST 特征点检测算法因为计算步骤比较简单和提取特征点的时间开销较少, 经常被用在实时跟踪系统. 然而 FAST 算法有一个重大的弱点, 即对图像的缩放不具有自适应性. 当图像尺度发生变化时 FAST 算法得到结果的重复性 (Repeatability) 较差. 本文提出改进的 FAST 特征检测算法使提取的特征点不仅具有图像位置信息, 还包括了尺度信息. 该算法流程图如图 2 所示.

**步骤 1.** 构建一组不同尺度的高斯滤波的图像, 剔除噪声引起的变化剧烈的像素点, 同组图像之间没有分辨率的差别.

**步骤 2.** 对不同尺度的图像进行降采样处理形成 3 阶高斯金字塔. 对不同的尺度进行不同的降采样处理: 尺度为 0 时图像不变, 尺度为 1 时进行一次降采样, 尺度为 2 时进行两次降采样, 这样便构成了不同分辨率的金字塔.

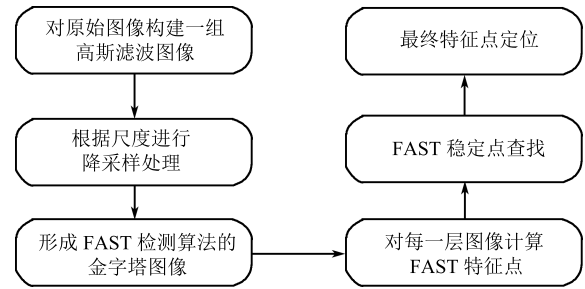


图 2 改进的 FAST 算法流程图  
Fig.2 Improved fast algorithm flowchart

**步骤 3.** 对不同阶的图像进行 FAST 特征点检测算法, 不同尺度的图像采用不同直径的 Bresenham 圆作为 FAST 检测算法的判定区域. 然后, 通过将各个层的 FAST 特征点还原到原始图像的像素点, 定位 FAST 稳定点. 最后, 得出带有 FAST 检测直径和尺度信息最终特征点.

构建图像金字塔的建造过程的高斯操作非常耗时, SIFT 和 SURF 算法就构建了稠密的尺度空间, 虽然对尺度具有不变性, 但是失去了对特征点提取的实时性. 本算法综合考虑了尺度的鲁棒性和特征提取的实时性, 仅用三次高斯滤波和三次降采样, 建立稀疏的尺度金字塔, 保证了对特征提取的实时性, 同时也具备一定的尺度鲁棒性. 改进的 FAST 算法的图像金字塔构建结果如图 3 所示.

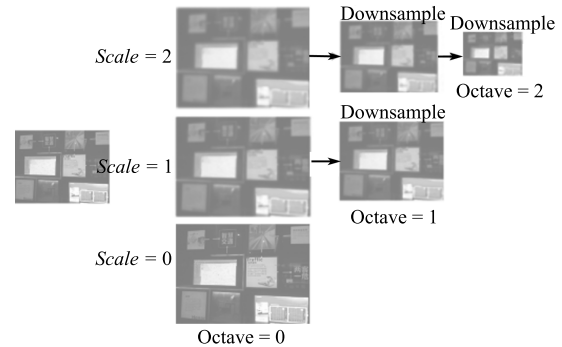


图 3 图像金字塔构建结果  
Fig.3 Image pyramid build results

### 1.2.3 特征点定位

建立好图像的尺度金字塔后, 用改进的 FAST 算法对各个阶层的图像进行特征点检测. 在文献 [7] 中经过测试得出特征点检测效果比较好的经验参数为 Bresenham 圆直径为 6 和  $n = 9$  ( $n$  为连续像素大于或者小于中心点像素个数). 本文算法中选取  $n$  为 9, 检测直径根据图像的高斯尺度参数进行相应的设置, 当高斯尺度  $Scale = 0$  时, Bresenham 圆的直径为  $d = 6$ ; 当高斯尺度  $Scale = 1$  时, Bresenham 圆的直径为  $d = 8$ ; 当高斯尺度  $Scale = 2$  时,

Brezenham 圆的直径为  $d = 12$ .

### 1.2.4 特征点定位

常见的特征点定位算法都是通过搜索尺度空间的局部三维极值点, 来确定局部特征点的位置和尺度. 如比较经典的 DOG 算子就是通过在 3 维  $(x, y, S)$  尺度空间中, 对每个采样点周围  $3 \times 3 \times 3$  的局部区域里进行非最大值抑制 (Non-maximum suppression); 再将采样点与相邻的 26 个像素点进行比较, 如果为极大值或极小值, 则确定为兴趣点; 最后通过 3 维 2 次函数拟合方法对兴趣点精确定位, 确定特征点位置. 该方法在 SIFT 算法中表现出非常好的效果, 但是也存在运算复杂和计算量大的问题, 对资源受限的智能终端不适用.

所以, 本文提出一种简单的特征点定位算法, 该算法计算量小、容易实现并且满足尺度层间特征点的快速定位. 通过分析 FAST 算法对同一个图像不同尺度层的局部结构会提取许多重复的特征点, 首先将图像金字塔不同尺度层通过 FAST 特征检测算法提取的特征点, 还原到相应的原始图像上的特征点; 其次根据降采样的次数, 对特征点像素坐标进行逆变换; 再进行插值处理等在原始的图像中找到对应的特征点; 然后对相同分辨率不同尺度层上的特征点, 进行尺度空间定位, 最终确定出表征图像的稳定点. 尺度空间点定位如图 4 所示, 设  $P$  为  $C$  层上的特征点  $C_{(2,2)}$ ,  $A$ 、 $B$  是它的相邻尺度空间邻域, 如式 (2), 邻域的特征点矩阵为  $A_{3 \times 3}$ ,  $B_{3 \times 3}$ , 当像素点  $(x, y)$  为特征点时, 对应矩阵中该位置的值为 1.

$$A = \begin{bmatrix} A_{(1,1)} & A_{(1,2)} & A_{(1,3)} \\ A_{(2,1)} & A_{(2,2)} & A_{(2,3)} \\ A_{(3,1)} & A_{(3,2)} & A_{(3,3)} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{(1,1)} & B_{(1,2)} & B_{(1,3)} \\ B_{(2,1)} & B_{(2,2)} & B_{(2,3)} \\ B_{(3,1)} & B_{(3,2)} & B_{(3,3)} \end{bmatrix} \quad (2)$$

设定阈值  $threshold$  为该层 FAST 特征点数目阈值, 阈值设为 1000, 当该层特征点数目大于等于阈值时, 按照式 (3),  $P$  点与相邻层对应位置点进行与操作, 结果等于 1 为稳定点.

$$C_{(2,2)} \cap A_{(2,2)} \cap B_{(2,2)} \quad (3)$$

否则按照式 (4) 进行稳定点定位.  $P$  点与上下层特征点进行与操作, 结果等于 1 为稳定点.

$$C_{(2,2)} \cap (A_{(1,1)} \| A_{(1,2)} \| \cdots \| A_{(3,3)}) \cap (B_{(1,1)} \| B_{(1,2)} \| \cdots \| B_{(3,3)}) = 1 \quad (4)$$

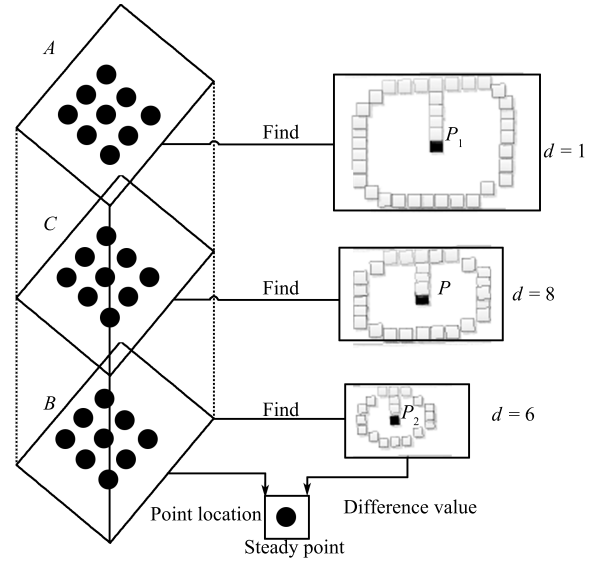


图 4 尺度空间特征点定位

Fig. 4 Scale space feature location

再对  $P$  点做如下处理: 对于相同坐标位置的点, 1) 如果都是亮点或都是暗点, 则选取最亮的或最暗的特征点, 并保存相应的尺度信息; 2) 如果是一个暗点和一个亮点, 则判断特征点与周围 Brezenham 圆上像素点的变化剧烈程度, 选取变化比较大的并保存尺度信息. 变化的剧烈程度通过中心点与满足其成为特征点的连续九个测试点像素灰度差的绝对值求和得到, 如式 (5) 所示.

$$S = \sum_{i=1}^9 |I_p - I_i| \quad (5)$$

其中,  $I_p$  代表 Brezenham 圆上中心点的像素值,  $I_i$  代表满足  $P$  点为特征点的连续 9 个点的像素值,  $S$  为特征点与周围像素的变化程度. 最后, 选取周围变化程度最大的那个特征点和尺度作为一个最稳定点的信息表征.

## 2 特征点的描述

SURF 描述子由 Bay 等<sup>[9]</sup> 提出, 充分利用了 Haar 小波响应和积分图像, 使特征描述符不仅具有尺度和旋转不变性, 而且对光照的变化也具有不变性. SURF 特征描述子包括特征点主方向的分配和特征描述符的计算.

主方向的计算通常通过以特征点为中心、以  $6s$  ( $s$  为特征点的尺度) 为半径的圆形区域, 对图像进行 Haar 小波响应运算. 使用  $\sigma$  为  $2s$  的高斯加权函数对 Haar 小波的响应值进行高斯加权, 然后设计以特征点为中心、张角为  $\pi/3$  的扇形滑动窗口, 以步长为 0.2 弧度, 旋转这个滑动窗口, 并对滑动窗口内

图像 Haar 小波响应值  $dx$ 、 $dy$  通过式 (6) 和式 (7) 进行累加, 得到一个矢量  $(m_w, \theta_w)$ :

$$m_w = \sum_w dx + \sum_w dy \quad (6)$$

$$\theta_w = \arctan \frac{\sum_w dx}{\sum_w dy} \quad (7)$$

主方向为最大 Haar 响应累加值所对应的方向, 也就是最长矢量所对应的方向, 如式 (8), 即:

$$\theta = \theta_w | \max\{m_w\} \quad (8)$$

特征描述符的计算, 是在一个矩形区域来计算 Haar 小波响应. 以特征点为中心, 沿主方向选取  $20s \times 20s$  的正方形区域, 并将该区域旋转到主方向, 将该区域划分成  $4 \times 4 = 16$  个子区域, 在每一个子区域有  $5s \times 5s$  像素. 使用尺寸为  $2s$  的 Haar 小波对子窗口图像进行其响应值计算, 共进行 25 次采样, 分别得到沿主方向的  $dy$  和垂直于主方向的  $dx$ . 再以特征点为中心, 对  $dy$  和  $dx$  进行高斯加权计算, 其  $\sigma$  为  $3.3s$ . 最后, 分别对每个子块的响应值进行统计. 设子块向量为  $v$ , 则  $v$  的值如式 (9) 所示.

$$v = \left[ \sum dx, \sum |dx|, \sum dy, \sum |dy| \right] \quad (9)$$

因为有  $4 \times 4$  个子块, 所以特征描述子共由  $4 \times 4 \times 4 = 64$  维特征向量组成.

### 3 目标匹配

本文的目标匹配算法包含两部分, 分别为特征向量的匹配和匹配目标的查找. 目标物体通过最近邻 SURF 特征向量的查询, 来实现特征点的匹配; 根据匹配特征点的数目和匹配特征点占训练样本图像的比重来决定是否查找到成功匹配的样本.

1) SURF 特征匹配: 训练样本的特征点通过 Kd-Tree 的方式进行组织, 特征匹配时, 通过查询 Kd-Tree 树寻找与目标 SURF 特征向量距离最近的样本特征向量, 当二者之间的距离小于某个阈值, 则接受这一对匹配点, 认为特征匹配.

2) 目标查找: 将当前目标的所有特征向量都在样本库中进行最近邻查找, 记录匹配的特征点对  $m(s_i, d_j)$ , 表示当前目标第  $i$  个特征向量与样本库第  $j$  幅图像中的特征点匹配. 匹配结果为  $sam_j$  (表示第  $j$  个样本图像与目标物体相匹配特征向量的个数), 如式 (10). 在确定目标图像与样本图像是否相似时, 可以用匹配特征点数目和匹配率来衡量. 通过设定特征点匹配数目的阈值和匹配率来进行判断. 设  $e$  为当前目标特征向量的总个数,  $n_{\max}$  为样本图

像最大匹配点数, 如式 (11),  $r$  为样本图像的匹配率, 如式 (12),  $thd1$  为特征点匹配数目阈值,  $thd2$  为特征点匹配率阈值.

$$sam_j = \sum_{i=1}^e m(s_i, d_j) \quad (10)$$

$$n_{\max} = \max(sam_1, sam_2, \dots, sam_j, \dots) \quad (11)$$

$$r = \frac{n_{\max}}{e} \quad (12)$$

当  $n_{\max} > thd1$  和  $r > thd2$  同时满足时, 表明两图像的匹配程度较高, 可认为当前目标与样本库中特征点数目匹配最多的图像相似, 样本图像即为对应的匹配目标.

## 4 实验验证

### 4.1 实验条件

实验用到的硬件平台如下所述: 智能手机 (HTC A9191), CPU (1024 MHz), RAM (1.5 GB), OS (Google Android 2.3), MicroSD (32 G). 实验数据采用两个数据集: 1) 我们用智能手机拍摄的上海电信信息馆的 120 个场景图像, 其中的一组图像如图 5 所示; 2) 国际标准 UKBench<sup>[19]</sup>, 包含 2550 个不同的场景, 每个场景包括 4 个从不同角度拍摄的图像, 一共 10 200 幅图像, 其中的一组图像如图 6 所示.

### 4.2 算法性能

在图像识别算法中, 通常从训练时间、训练内存占用、识别时间和识别精度四个方面来衡量算法的优越性. 训练时间是从开始训练到训练结束消耗的时间; 训练内存占用是训练阶段内存消耗的最大值; 识别时间是从开启捕获视频开始到智能终端出现结果的时间; 识别精度为智能终端成功识别的次数与总识别次数的比值. 本文在智能手机上分别实现了提出的算法、SIFT 算法和 SURF 算法, 并从上面提出的四个方面对算法进行对比. 每次实验任选 20 个



图 5 智能手机拍摄的数据集  
Fig. 5 Data sets by smartphone



图 6 UKBench 数据集

Fig.6 Data sets on UKBench

不同的场景进行样本训练, 每张训练图像的分辨率都统一转换为 320 像素 × 240 像素, 实验取每组样本图像 10 次测试的平均值.

#### 4.2.1 提点算法占用时间和内存对比

图 7 是三种算法对手机拍摄数据集的一张图像提点结果. 从图中可以看出, 本文算法提点数量比较集中. 表 1 统计了算法的提点数量、消耗时间和占用

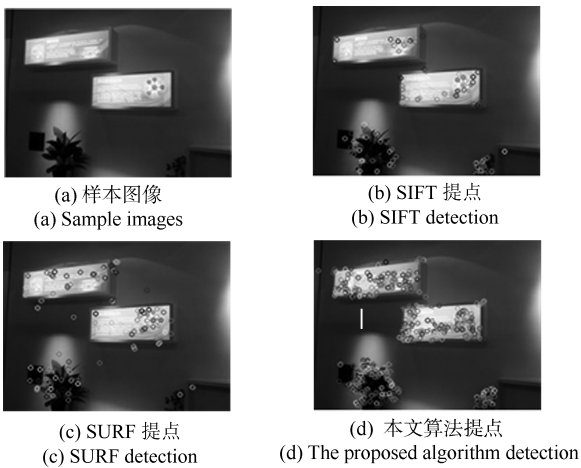


图 7 样本图像和提点效果

Fig.7 Sample images and detection results

表 1 提点数量与时间和内存开销对比

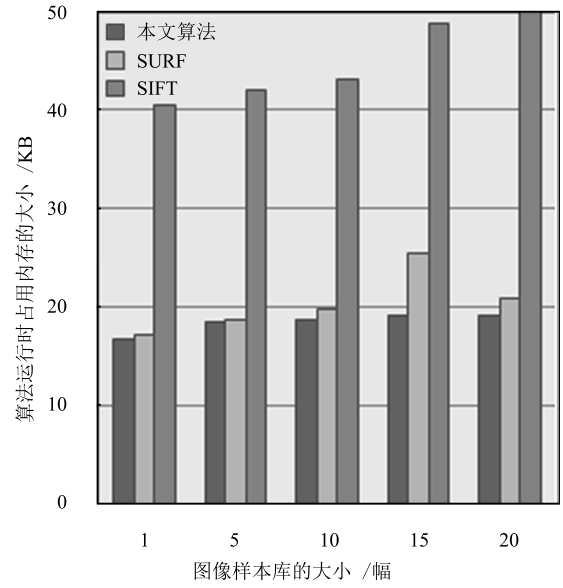
Table 1 Total points with time and memory comparison

算法名称	特征点数	时间消耗 (ms)	内存消耗 (KB)
本文算法	147	271	16 851
SURF 算法	125	1 282	17 231
SIFT 算法	118	1 937	41 631

内存. 由表 1 得知, 本文算法不仅提取的特征点数量最多, 而且消耗时间和内存最少, 相比其他两种算法更高效.

#### 4.2.2 图像训练占用时间和内存对比

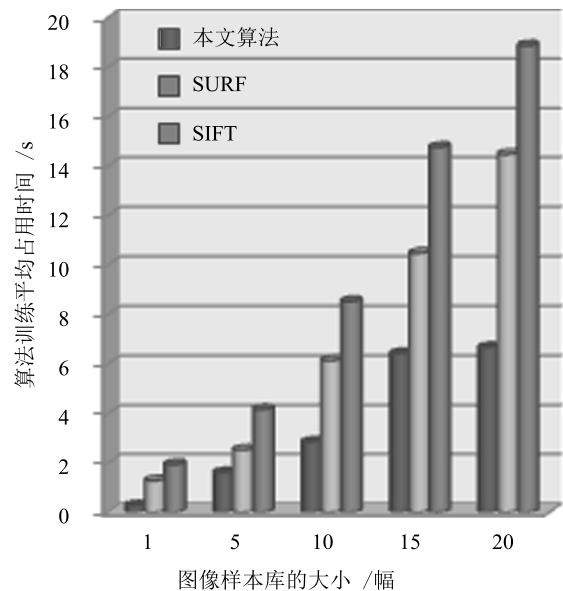
图 8 是算法训练所占用时间和内存对比. 从图 8 (a) 得出, 本文的算法占用内存最低, SURF 算法位居第 2, 略高于本文算法, SIFT 算法占用内存量



图像样本库的大小 / 幅

(a) 内存对比

(a) Memory comparison



图像样本库的大小 / 幅

(b) 时间对比

(b) Time comparison

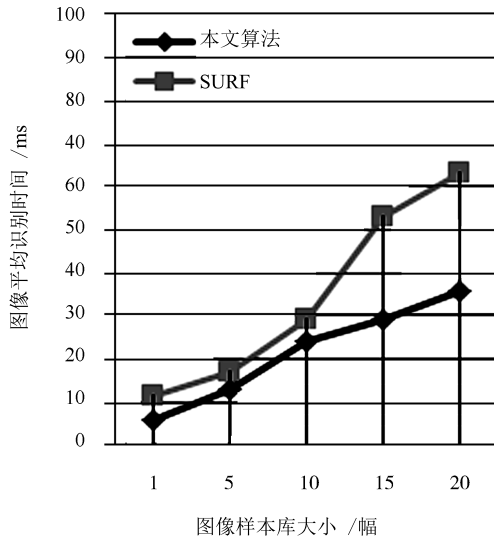
图 8 算法训练占用内存和时间对比

Fig.8 The memory and time comparison on algorithm training

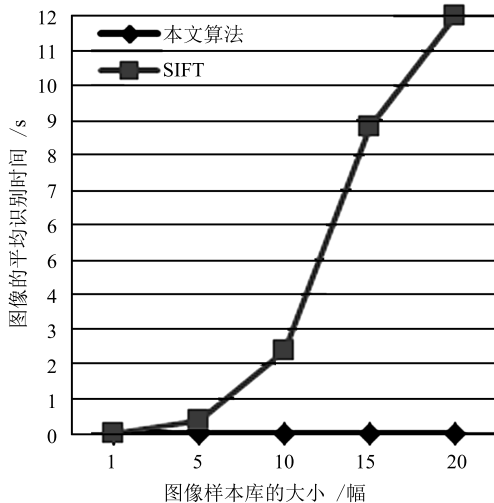
最大，比本文算法两倍还要多。从图 8 (b) 得出，本文算法时间消耗最少，明显低于 SIFT 和 SURF 算法，SURF 算法大约为本文算法的 5 倍，SIFT 算法大约为本文算法的 7 倍多。实验结果证明，本文算法在线训练内存占用最少和时间消耗最低。

### 4.2.3 图像识别时间和精度对比

图 9 和图 10 为在手机拍摄数据集上的测试结果。从图 9 (a) 可以看出，本文算法在样本库较少时



(a) 本文算法与 SURF 算法对比  
(a) Comparison between the proposed algorithm and SURF



(b) 本文算法与 SIFT 算对比  
(b) Comparison between the proposed algorithm and SIFT

图 9 三种算法识别时间对比

Fig.9 Recognition time comparison among three algorithms

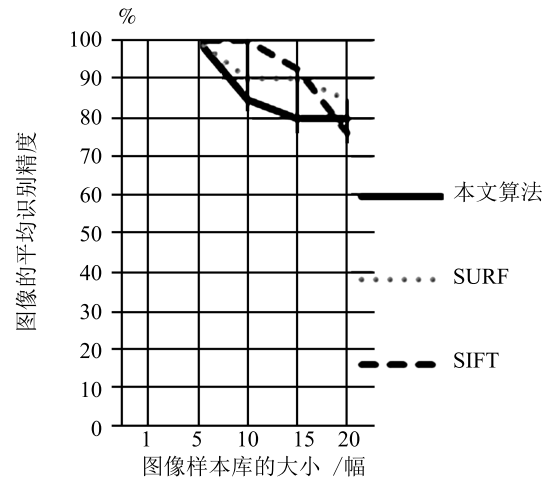


图 10 三种算法的识别精度对比

Fig.10 Recognition accuracy comparison among three algorithms

与 SURF 算法识别时间相当，但是在样本库增大到 15~20 张时变化比较明显，SURF 算法平均识别时间明显高于本文算法，甚至为其 2 倍。从图 9 (b) 可以看出，本文算法在样本库较少时与 SIFT 算法识别时间相当，但是在样本库逐渐增大时，本文算法平均识别时间在 1s 以内，而 SIFT 算法平均高出一个数量级。实验结果表明，本文的算法识别时间最短，平均在 1s 以内，实时性较好。

图 10 是图像平均识别精度对比。由图 10 可以看出，在样本库较少时，算法都有较好的识别精度；在样本库增大到 10 时，算法的识别率都保持在 80% 以上；但是随着样本库的持续增大，识别精度都有下降的趋势，其中 SIFT 算法下降最明显，SURF 算法也是持续下降，而本文算法下降到接近 80% 的识别率时趋于稳定。实验结果表明，本文算法在不同大小的图像库中识别率比较稳定。

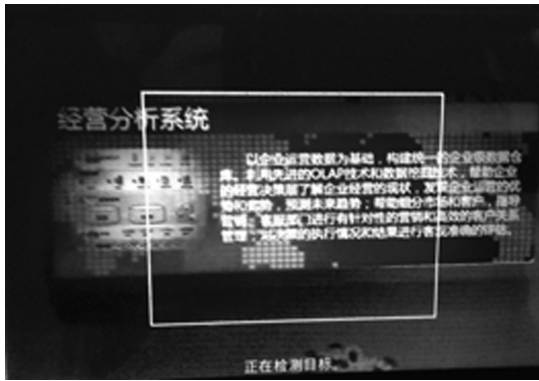
表 2 是在 UKBench 数据集上的测试结果。从表 2 可以看出，本文算法的识别时间明显优于其他两种算法，始终保持在 1s 以内；识别精度虽然低于逐个匹配的 SIFT 算法，但是与 SURF 算法相当，仍旧较高。因此，在公开的标准数据集上的实验再次验证了本文算法的有效性，既能进行快速识别，同时又保持了较高的识别精度。

### 4.3 算法应用

将本文的算法应用在上海电信馆的移动检索系统中，通过智能手机对展馆的产品进行识别，系统主界面如图 11 所示。图 11 (a) 为启动界面，开启智能手机摄像头截取视频帧。将摄像头对准待检索物体，通过识别来获取该物体在数据库中的增强信息，实现移动检索。图 11 (b) 为识别成功后将增强信息渲染在目标物体上的界面。

表 2 UKBench 数据集图像测试结果  
Table 2 UKBench datasets images test results

UKBench	5 幅		10 幅		15 幅		20 幅	
	时间 (ms)	精度 (%)	时间 (ms)	精度 (%)	时间 (ms)	精度 (%)	时间 (ms)	精度 (%)
本文算法	17	99.2	26	88.7	36	82.5	78	79.8
SURF 算法	20	99.3	30	92.6	58	87.8	113	76.5
SIFT 算法	70	99.6	274	98.3	9000	94.3	12 448	80.3



(a) 启动界面

(a) The start interface



(b) 增强显示界面

(b) Augmented display interface

图 11 电信馆移动检索系统

Fig. 11 Telecom museum mobile retrieval system

## 5 结论

本文提出了一种移动实时识别的算法,并应用于电信场馆的移动检索系统中。在特征检测方面,通过构建高斯模糊和降采样的多尺度图像金字塔,用改进的 FAST 检测算法对各尺度层进行特征点检测和弱尺度空间查找算法定位最终特征点;在特征描述方面采用了 SURF 64 维的特征描述算法;在目标匹配方面采用了 KD-Tree 特征点匹配和目标查找的算法。实验证明,该算法对智能手机的实时识别是非常有效的。本文算法也有不足之处,不能支持大规

模场景的识别。下一步的工作是研究描述符的压缩和高精度匹配,在智能手机端实现对大规模场景的实时识别。

## References

- Chen D M, Baatz G, Koser K, Tai S S, Vedantham R, Pylvanaine T, Roimela K, Chen X, Bach J, Pollefeys M, Girod B, Grzeszczuk R. City-scale landmark identification on mobile devices. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. Washington, USA: IEEE, 2011. 737–744
- Wang X Y, Yang M, Cour T, Zhu S H, Yu K, Han T X. Contextual weighting for vocabulary tree based image retrieval. In: Proceedings of the 2011 International Conference on Computer Vision. Barcelona, Spain: IEEE, 2011. 209–216
- Muralidharan R, Chandrasekar C. 3D object recognition using multiclass support vector machine — K-nearest neighbor supported by local and global feature. *Journal of Computer Science*, 2012, 8(8): 1380–1388
- Heo J P, Lee Y, He J F, Chang S F, Yoon S E. Spherical hashing. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence, USA: IEEE, 2012. 2957–2964
- Mair E, Hager G D, Burschka D, Suppa M, Hirzinger G. Adaptive and generic corner detection based on the accelerated segment test. In: Proceedings of the 2010 European Conference on Computer Vision (ECCV). Berlin, Germany: Springer, 2010. 183–196
- Leutenegger S, Chli M, Siegwart R Y. Brisk: binary robust invariant scalable keypoints. In: Proceedings of the 2011 IEEE International Conference on Computer Vision. Barcelona, Spain: IEEE, 2011. 2548–2555
- Rosten E, Drummond T. Machine learning for high-speed corner detection. In: Proceedings of the 9th European conference on Computer Vision. Berlin, Heidelberg: Springer-Verlag, 2006. 430–443
- Low D G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004, 60(2): 91–110
- Bay H, Ess A, Tuytelaars T, Van Gool L. SURF: Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 2008, 110(3): 346–359
- Calonder M, Lepetit V, Ozuysal M, Trzcinski T, Strecha C, Fua P. BRIEF: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, 34(7): 1281–1298

- 11 Yuan Z W, Wang Y H. Research on K nearest neighbor non-parametric regression algorithm based on KD-tree and clustering analysis. In: Proceedings of the 4th International Conference on Computational and Information Sciences. Chongqing, China: IEEE, 2012. 298–301
- 12 Koonsanit K, Jaruskulchai C, Eiumnoh A. Parameter-free K-means clustering algorithm for satellite imagery application. In: Proceedings of the 2012 International Conference on Information Science and Applications. Suwon, Korea: IEEE, 2012. 1–6
- 13 Yamashita T, Yamauchi Y, Fujiyoshi H. Human detection for multiple pose by boosted randomized trees. In: Proceedings of the 1st Asian Conference on Pattern Recognition. Beijing, China: IEEE, 2011. 229–233
- 14 Ozuysal M, Calonder M, Lepetit V, Fua P. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, **32**(3): 448–461
- 15 Wagner D, Reitmayr G, Mulloni A, Drummond T W, Schmalstieg D. Real-time detection and tracking for augmented reality on mobile phones. *IEEE Transactions on Visualization and Computer Graphics*, 2010, **16**(3): 355–368
- 16 Chen W C, Xiong Y G, Gao J, Gelfand N, Grzeszczuk R. Efficient extraction of robust image features on mobile devices. In: Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. Nara, Japan: IEEE, 2007. 287–288
- 17 Gu J, Mukundan R, Billinghurst M. Developing mobile phone AR applications using J2ME. In: Proceedings of the 23rd International Conference on Image and Vision Computing New Zealand. Christchurch, New Zealand: IEEE, 2008. 1–6
- 18 Rosten E, Drummond T. Fusing points and lines for high performance tracking. In: Proceedings of the 10th International Conference on Computer Vision. Beijing, China: IEEE, 2005. 1508–1515
- 19 Nister D, Stewenius H. Scalable recognition with a vocabulary tree. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. New York, USA: IEEE, 2006. 2161–2168



**桂振文** 北京理工大学计算机学院博士研究生. 主要研究方向为计算机视觉, 图像处理和移动增强现实.

E-mail: quizhenwen1983@bit.edu.cn

(**GUI Zhen-Wen** Ph. D. candidate at the School of Computer Science, Beijing Institute of Technology. His research interest covers computer vision,

image processing, pattern recognition, and augmented reality.)



**陈靖** 博士, 北京理工大学光电学院副教授. 主要研究方向为模式识别和增强现实. 本文通信作者.

E-mail: chen74jing29@bit.edu.cn

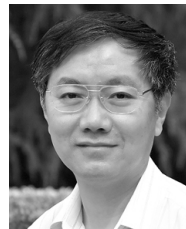
(**CHEN Jing** Ph. D., associate professor at Beijing Institute of Technology. Her research interest covers pattern recognition and augmented reality.

Corresponding author of this paper.)



**刘越** 博士, 北京理工大学光电学院教授. 主要研究方向为虚拟现实与增强现实. E-mail: liuyue@bit.edu.cn

(**LIU Yue** Ph. D., professor at Beijing Institute of Technology. His research interest covers virtual reality and augmented reality.)



**王涌天** 博士, 北京理工大学计算机学院教授. 主要研究方向为计算机视觉, 虚拟现实与增强现实.

E-mail: wyt@bit.edu.cn

(**WANG Yong-Tian** Ph. D., professor at Beijing Institute of Technology. His research interest covers computer vision, virtual reality, and aug-

mented reality.)