计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191200104

基于特征聚类的轻量级图像搜索系统

王晓飞[1]　周　超[2]　刘利刚[1]
1 中国科学技术大学数学科学学院　合肥 230000
2 腾讯计算机系统有限公司　广东 深圳 518057
(wxf9545@mail.ustc.edu.cn)

摘　要　在图像搜索的场景中,由于搜索请求的随机性,为了提高搜索速度,搜索算法运行时需要把整个数据集预先载入到运行内存。由于运行内存价格远高于同容量的硬盘价格,降低运行内存自然可以大大降低图像搜索服务的成本,但如果直接对数据进行压缩,往往会极大地损失搜索精度。在这种情况下,文中提出了一种基于图像内容特征的分块式图像搜索框架。先利用神经网络的方法来预先提取图片特征,在不对特征进行量化压缩的前提下,采用一种启发式的聚类方法对数据进行分块,同时保证每个数据块的数据之间有一定的相似性。对于每个数据块,采用基于图结构的 HNSW 算法来构建索引子图以加速图片查询。在该框架下,通过控制查询时访问的数据块的个数,可以在保证精度的前提下大大减少算法所需要的运行内存容量。

关键词:图像检索;相似搜索;聚类;图像特征提取;近似最近邻匹配
中图法分类号　TP391

Lightweight Image Retrieval System Based on Feature Clustering

WANG Xiao-fei[1],ZHOU Chao[2] and LIU Li-gang[1]
1 School of Mathematical Sciences,University of Science and Technology of China,Hefei 230000,China
2 Tencent Computer Systems Co.,Ltd.,Shenzhen,Guangdong 518057,China

Abstract　In the scene of image search,due to the randomness of search request,in order to increase the search speed,it is often necessary to preload the entire data set into the running memory. Because the price of running memory with the same capacity is much higher than that of hard disk,reducing the running memory can greatly reduce the cost of image search service. However,if the data is compressed directly,the search accuracy will be greatly reduced. In this case,this paper proposes a content-based image search framework,which divides data set into groups. Firstly,the neural network is used to extract image features. On the premise of not compressing the features,a heuristic clustering method is used to group the data,ensuring that there is a certain similarity between the data of each data group. For each data group,HNSW algorithm based on graph structure is used to construct index to speed up image query. In this framework,by controlling the number of data blocks accessed during query,the running memory capacity required by the algorithm can be greatly reduced,under the premise of ensuring the accuracy.

Keywords　Image Retrieval,Similarity search,Clustering,Image feature extraction,Approximate nearest neighbor matching

1　引言

如今,随着图片获取技术的不断发展,每天都会新产生海量的图片数据。图片早已替代文字成为互联网上信息传播的重要方式。如何在这些海量数据中进行高效的相似图片搜索成为计算机视觉领域的一个重要问题。针对这个问题已经提出了很多算法[1-2],但是为了保证查询速度,这些算法在实际使用时需要将整个数据特征集从硬盘载入内存。我们知道,相同存储空间的内存成本远高于硬盘,在服务器上部署这类图像搜索算法时,面对不断增长的海量数据,我们常常会面临内存空间不够的问题。因此,如何有效地降低相似图片搜索算法运行时的内存成为图像搜索面临的重要挑战。

针对降低内存开销这一问题,本文提出了一种分块的图像搜索算法。在实际应用中,我们可以根据精度需求控制载入内存中的数据量,同时也可以保证较高的精度。该算法分为 3 步:特征提取、索引创建和图像搜索。特征提取的目标是利用卷积神经网络(Convolutional Neural Network,CNN)和 SPoC[3] 从图片中提取出用以描述图片特征的高维向量,所获得的特征向量之间的 L_2 距离在后续中将被用来衡量两张图片之间的相似度。文献[3]指出,这种对深度卷积特征累加求和生成的 SPoC 特征在图像的检索领域的表现非常出色。

为了保证搜索结果的精度,我们在第一步的核心目标是

到稿日期:2019-12-17　返修日期:2020-05-18　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61672482)
This work was supported by the National Natural Science Foundation of China (61672482).
通信作者:刘利刚(lgliu@ustc.edu.cn)

尽可能详细地保存图片的特征，因此没有对这些特征向量提取主成分。同时，为了达到降低内存开销的目的，我们在索引创建这一步选择先使用二分聚类算法将数据分成若干簇，然后分别在这些簇上创建 HNSW 索引。基于二分聚类算法的特性，我们可以十分方便地增加簇的数量。同时，因为 HNSW 索引添加新元素时无需重新构建整个索引，所以扩充数据集时，可以直接在现有的索引上进行简单的插入操作，而无需重新构建整个索引。

实验结果显示，不进行内存压缩时，所提算法的精度可以达到 99.11%，而普通的 HNSW 方法只能达到 98.7%。在同样消耗正常内存 22%的情况下，所提算法的精度还能达到 98.27%，而运用了主成分法进行压缩的 HNSW 方法的精度此时只有 96.5%。

2 相关工作

2.1 图像特征提取

20 世纪 70 年代末，在传统的图像检索领域，最广泛使用的图像检索技术是基于文本的图像检索技术[1]。当时常用的做法是预先设定好图片对应的关键字或者描述内容的文本，这些文本可以用来描述图像的一些特征，如作者、年代、尺寸和流派等；再将图片和这些关键字作为对象和对象特征存储在数据库中，等到需要查询时，使用一些基于查询内容的特征进行精确匹配或者概率匹配。但是手动标注费时费力，自动标注[1]的质量又难以保证。随着图片采集方式的平民化，图片数量呈现爆炸性增长，为了解决图像搜索的问题，20 世纪 90 年代发展出了基于内容的图像检索技术。这种方法会基于图片的内容，利用一些算法来提取图片中的视觉特征，查询时会根据这些视觉特征和用户查询输入的相似度大小将结果返回给用户，文献[1-2]都是这一类的方法。在 CNN 之前，人们往往需要手工设计复杂的图片内容特征提取算法。CNN 问世后，基于这项技术的深度特征[4]由于在图像检索领域的效果远超传统方法，很快得到了大家的重视，最常见的此类特征有 VLAD[5]、Fisher 向量法[6]、三角嵌入法[7]、局部描述符子法[8]和哈希学习法[9]。

2.2 近似最近邻匹配

每张图片的特征就是一个高维向量，近似最近邻匹配算法（Approximate Nearest Neighbor，ANN）可以高效地在海量的向量中找到和输入向量最相近的几个。根据数据组织方式的拓扑结构，我们可以把 ANN 算法分成基于树结构和图结构两种。基于树结构的方法有对低维向量检索效率非常高的 KD-tree[10]和 Quadra tree[11]，这两种方法的主要思想是通过构造一个对 k 维空间的划分来加快搜索的速度。在 2-3 维的空间上，这两种算法的实际搜索时间复杂度十分接近于 $O(\log n)$，但是在一些特殊结构[12]上的最坏时间复杂度接近于 $O(kN^{1-\frac{1}{k}})$，其中 k 表示空间的维数。基于图结构的方法有 NSW[13]，HNSW[14]以及在 NSW 基础上发展出的索引量代搜索算法[15]。其中，NSW 建立了一个具有高速公路机制的类 Delaunay 图，其把数据中的所有向量构建成了一张连通图，并基于此图来搜索某个顶点的最近邻。HNSW 则更进一步地利用启发式的方法在原数据集上构建了一个分层的 NSW 结构。根据 Benchmark 上的 ANN 算法的测试[16]，基于图结构的 HNSW 算法在查询速度和精度上要优于其他算法。

2.3 聚类

K 均值聚类算法作为一种无监督学习算法被频繁地应用于数据挖掘和统计数据分析等数据相关领域。K 均值聚类算法最早在 1967 年由 MacQueen 提出[17]，并不断得到改进。为了找到全局最优解，二分 K 均值[18]算法先把数据全体看成一个簇，然后对这个簇进行一个 $k=2$ 的聚类，再不断地对误差平方和最大的簇执行这个过程，直到达到指定的 k。Canopy 聚类[19]提出预先设置距离阈值而无需指定 k 值，该算法规避了传统 K 均值聚类算法的问题最终结果对随机选择的初始的 k 个中心依赖性强的问题。为了规避同样的问题，K-means++[20]的基本思想是让这 k 个初始聚类中心尽可能地远，并引入了一个函数来描述每个点与最近的聚类中心的距离，并在依次选取 k 个中心时赋予这个距离较近的点更低的选中概率。

3 算法流程

3.1 数据增强

为了增加特征提取时 CNN 的训练数据量并提高图片搜索的鲁棒性，首先需要对数据集进行扩充。首先，我们对图片添加服从零均值和方差 $\sigma=5.0$ 的高斯分布的线性变换来减弱高频特征对神经网络的影响；然后再将原图片和添加了噪声的图片分别进行绕中心的顺时针 $\frac{\pi}{2}$，π，$\frac{3\pi}{2}$ 旋转以及沿水平和竖直方向翻转。

3.2 特征提取

每一张彩色图片本质上都是一个高维向量，在进行搜索之前，我们需要提取里面重要的信息，也就是图片特征，作为搜索的对象。对于输入的每一幅图片 I，假设对 I 使用 CNN 进行推理，得到最后一个卷积层的输出为 $C\times W\times H$ 的三维张量 f。

另一方面，由于画面中心的部分往往更容易得到更多的关注，因此我们可以认为越靠近特征中心的部分往往越能体现图像的特征。基于这一点，我们可以采取以下的池化加权策略，令：

$$\psi_1(I)=\sum_{y=1}^{H}\sum_{x=1}^{W}\alpha_{(x,y)}f_{(x,y)} \tag{1}$$

其中，系数 $\alpha_{(x,y)}$ 仅与空间位置 x 和 y 相关，且在空间上满足高斯分布，即：

$$\alpha_{(x,y)}=\exp\left\{-\frac{\left(y-\frac{H}{2}\right)^2+\left(x-\frac{W}{2}\right)^2}{2\sigma^2}\right\} \tag{2}$$

最后，由于本文是用特征之间的 L_2 距离来衡量两张图片的相似程度，为了使最终的搜索结果同时符合余弦距离，还需要对 ψ_1 进行归一化。

$$\psi(I)=\frac{\psi_1(I)}{\|\psi_3(I)\|_2} \tag{3}$$

这样我们就得到了输入图片 I 的特征 $\psi(I)$。值得注意的是，ψ 的维度和 f 第一维的元素个数相同，为 C。本文采取和文献[3]相同的策略，对 $\psi(I)$ 提取主成分，起到节省数据存储空间的作用。

3.3 创建索引

在获取了所有图片的特征之后，需要创建一个索引，索引的创建可以极大地加速后续的搜索速度。根据 Benchmark 上各类算法的排名，基于图结构的 HNSW 算法的性能名列前茅，我们的算法也选择创建图结构的搜索索引。

本文的索引创建算法的思路如图 1 所示。其中，图 1(a)表示输入的图片集合，从中提取出图片特征。如图 1(b)所示，其中每一行对应图 1(a)中的一张图片。接下来对图 1(b)聚类分组得到图 1(c)。最后，对于每一个数据组，建立如图 1(d)所示的分层图结构索引。在提取完图片特征值以后，利用二分 K 均值聚类将接近的数据进行聚类分组，对于海量的输入数据，通过选择合适的 K 值，可以将分组之后的数据量控制在一个可接受的范围内。我们假设数据已经被分成了 m 个簇$\{G_i\}$，它们的质心为$\{C_i\}$，初始状态下 $m=1$，即所有的数据在同一个簇中。接下来，我们不断计算每个簇中所有元素与质心C_i距离的平方和。

$$SSE_i=\sum_{I\in G_i}\|\psi(I)-C_i\|^2 \tag{4}$$

其中，SSE_i的值越小，说明数据更集中，聚类效果也就更好。我们从中挑选出 SSE 值最大的簇G_k，对它进行 2 均值聚类，就获得了 $m+1$ 个簇$\{G_i\}$。不断对现有的簇重复以上过程，直到 $m=K$。

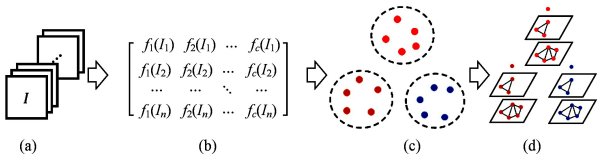


图 1 索引创建流程。

Fig. 1 Flow chart of index construction

最后一步是对于获取的每个簇G_i，构建分层的图结构索引$hnsw_i$。这个索引的结构如图 2 所示，最上层的红点是该结构的进入点，最下层则包含了G_i中的所有点，在相邻两层之间满足上层点的集合为下层点的集合的真子集。为了构建这个索引，我们需要不断重复以下的下插入过程。

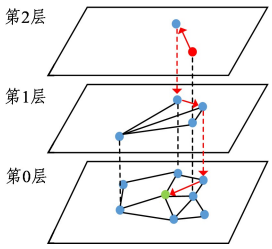


图 2 HNSW 索引结构图(电子版为彩色)

Fig. 2 Structure of HNSW index

对于一个待插入的新元素 q，首先用式(5)确定 q 可以达到的最高层编号 L。

$$L=\lfloor-\ln(k)\times m_L\rfloor \tag{5}$$

其中，k 为(0,1)内的随机数，m_L 为常量。接着从 $hnsw_i$ 的进入点开始，不断在本层当前点的相邻点内寻找距离 q 更近的点并更新当前点，直到这一过程无法继续进行。再沿着该点前往下一层。前往下一层之前，若当前层数小于或等于 L，还需将 q 插入本层并在 q 与找到的最近点之间建立双向边。然后重复以上过程直到到达最底层。

算法 1 是该插入过程的伪代码。其中，M_{max} 为每层每个元素的最大相邻元素个数，该值越大说明构建的图越稠密，后期查询的精度也越高，但相应地会消耗更多的查询时间；ef 为在图中寻找最近点时备选列表的大小，与 M_{max} 一样，该值越大，查询精度越高，速度越慢。

算法 1 插入算法

输入：($hnsw_i$, q, M_{max}, ef, m_L)
输出：插入了新元素 q 的 $hnsw_i$
1. W←∅ /* W 为当前已找到的最近的元素的集合 */
2. ep←$hnsw_i$的进入点
3. L←ep 所在层数 /* $hnsw_i$的最高层编号 */
4. l←⌊−ln(unif(0..1)) · m_L⌋/* 新元素所在层数 */
5. for l_c←Lto l+1 do
6. W←从 ep 出发l_c层与 q 最近的元素
7. ep←从 W 到 q 最近的 ef 个元素
8. for l_c←min (L,l) to 0
9. W←l_c层与 q 最近的前 ef 个元素
10. neighbors←W 中与 q 最近的 M 个元素
11. 在l_c层添加从 neighbors 到 q 的双向边
12. for e∈neighbors
13. eConn←l_c层 e 的相邻节点
14. if|eConn|>M_{max}
15. eNC←eConn 中与 e 最近的M_{max}个节点
16. 把 e 在l_c层的邻节点设置为 eNC
17. ep←W
18. if l>L
19. 把 $hnsw_i$的进入点设定为 q

3.4 *K* 邻近搜索

查询数据集中与元素 q 最接近的 K 个元素的过程和创建索引的过程十分类似。首先，我们根据每个簇的中心C_i和 q 的欧氏距离，即 $\|C_i-q\|_2$，从小到大对所有簇($hnsw_i$)排序，然后选取其中的前 P 个簇，使用算法 2 查询各个簇中与 q 最相近的 K 个元素。这样我们就获得了 KP 个元素，最后只需取其中的前 K 个即可。值得注意的是，$P=1$ 往往就能满足大多数时候的精度需求，这种情况下算法的内存开销也是最低的。并且如果想提高搜索精度，无需重新进行索引构建的过程，只需提高 P 值即可达到目的。

算法 2 搜索算法

输入：($hnsw_i$, q, K, ef)
输出：距离 q 前 K 个最近元素
1. W←∅/* W 为当前已找到的最近的元素的集合 */
2. ep←$hnsw_i$的进入点
3. L←ep 所在层数 /* $hnsw_i$的最高层 */
4. for l_c←L to 1 do
5. W←从 ep 出发l_c层与 q 最近的元素
6. ep←从 W 到 q 最近的 ef 个元素
7. W←最下层从 ep 与 q 最近的前 ef 个元素
8. 返回 W 中距离 q 最近的前 K 个元素

4 实验结果与分析

4.1 算法参数

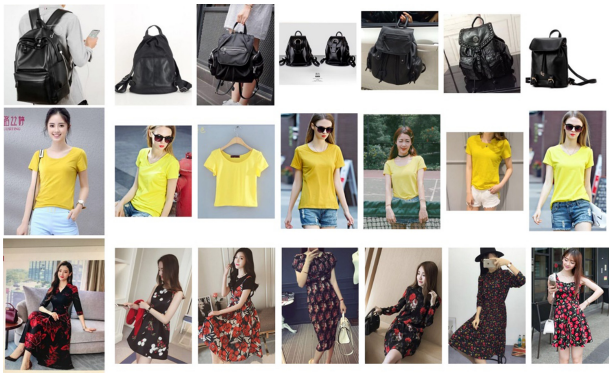
我们的卷积神经网络采用了文献[21]中提出的模型来进

行特征值提取。该模型的实现借助了 Caffe[22] 库。训练前，所有的图片都会预先被缩放到 512×512 像素。最终，这个网络最后一层卷积层输出的大小 $C\times W\times H=1\,792\times 32\times 32$。提取完特征后，我们没有对向量进行压缩，而是直接传入 SPoC 进行下一步操作，因此最终传入 SPoC 的特征向量维数固定为 1 792。

构建 HNSW 索引时，我们取最大相邻元素个数 $M_{\max}=64$，备选元素的列表大小 $ef=40$，搜索时取备选元素列表大小 $ef=16$。

4.2 结果分析

以下实验都是在我们收集的 300 万实际场景的数据集和 10 万查询集上进行的，提取完这些图片的特征后，对于每个查询集的数据，我们先用枚举法在数据集中找到最接近的 3 个结果。如果待测试的方法搜索得到的最佳结果在枚举法找到的 3 个结果中，则认为本次查询的结果是“正确”的，最终用正确查询的比例作为该方法的最终精度。图 3 给出了我们数据集中部分图片的搜索结果。



注：每行最左侧为查询时的输入图片，右侧是用我们的方法在数据集中找到的最相似的 6 个结果

图 3 样例搜索结果

Fig. 3 Sample search results

表 1 列出了所提方法与之前提到的被广为采用的主成分方法对于同样的图像特征在内存消耗和精度上的比较。其中内存消耗是相较于无压缩的正常方法的。从前两行可以看出，与提取 512 维的主成分的 HNSW 方法相比，所提方法在 $K=10$，$P=3$ 时的精度比主成分方法略高且消耗的内存只有 33%。后两行显示出通过调整参数 P 和主成分方法的压缩程度，在同样消耗约 22% 的内存时，所提方法的精度已经远超提取 256 维的主成分方法。

表 1 几种方法之间的比较

Table 1 Comparison between several methods

（单位：%）

Algorithm	Memory Cost	Accuracy
PCA-512	42	98.4
Our Method $K=10,P=3$	33	98.92
PCA-256	22	96.5
Our Method $K=10,P=2$	21	98.27

在内存充裕的情况下，我们往往希望获得更高的搜索精度，表 2 列出了所提方法与 Benchmark[16] 上 3 种常见方法的精度比较，可以看出，在这种情况下，所提方法可以获得高达 99.11% 的搜索精度。

表 2 不同方法最高精度

Table 2 Maximum accuracy of different methods

Algorithm	Accuracy/%
HNSW	98.7
BK-tree	97.8
KD-tree	95.4
Our Method $K=10,P=8$	99.11

对于所提算法，在构建索引时采用不同的分组数 K 和在进行查询时使用不同的 P 都会导致精度的变化。本文统计了不同的 K 和 P 时算法精度的表现，如表 3 所列。可以看出，当 $P=3$ 或者 4 时，算法的查询精度就可以达到一个接近 $P=K$ 时的水平，完全可以满足实际场景的需要。

表 3 不同参数下的算法精度

Table 3 Accuracy with different parameters

（单位：%）

	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
$P=1$	95.35	95.97	94.64	95.44	93.37
$P=2$	98.57	98.44	98.36	98.25	98.27
$P=3$	98.85	98.76	98.93	98.52	98.92
$P=4$	98.90	98.78	98.99	98.64	99.05
$P=5$	98.92	98.80	99.02	98.64	99.09
$P=6$	98.92	98.80	99.02	98.64	99.09
$P=7$		98.80	99.02	98.64	99.10
$P=8$			99.02	98.64	99.11
$P=9$				98.64	99.11
$P=10$					99.11

结束语 本文提出了一种基于 HNSW 算法的轻量级图像搜索算法，将庞大的数据集利用二分聚类的方法进行管理，通过动态地调整查询邻近聚类的数量来降低内存消耗或者提升查询精度，并在我们所收集的图像数据集上获得了很好的效果。我们的算法在使用 22% 内存的情况下，可以保持 98.27% 的精度，并在内存充足的情况下，可以达到 99.11% 的精度。

值得注意的是，所提图片特征提取算法并不是一个通用算法，对于以商品为主的数据集，它确实可以很好地用 L_2 距离来表示图片之间内容的相似性，对于其余类型的数据集，可能需要选择别的算法提取的特征来搭配本文的检索算法。

参 考 文 献

[1] COX I J, MILLER M L, MINKA T P, et al. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments[J]. IEEE Transactions on Image Processing, 2002, 9(1): 20-37.

[2] KIRANYAZ S, GABBOUJ M. Hierarchical CellularTree: An Efficient Indexing Scheme for Content-Based Retrieval on Multimedia Databases[J]. IEEE Transactions on Multimedia, 2007, 9(1): 102-119.

[3] BABENKO A, LEMPITSKY V. Aggregating Deep Convolutional Features for Image Retrieval[J]. arXiv: 1510.07493v1, 2015.

[4] BABENKO A, SLESAREV A, CHIGORIN A, et al. Neural codes for image retrieval[C]// European Conference on Computer Vision. Cham: Springer, 2014: 584-599.

[5] JEGOU H, DOUZE M, SCHMID C, et al. Aggregating local descriptors into a compact image representation[C]// Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010.

[6] PERRONNIN F, LIU Y, JORGE S, et al. Large-scale image retrieval with compressed Fisher vectors[C]// CVPR. IEEE, 2010.

[7] HERVÉ J, ANDREW Z. Triangulation Embedding and Democratic Aggregation for Image Search[C]// In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14). IEEE Computer Society, Washington, DC, USA, 3310-3317.

[8] SYED S H, MIROSLAW B. Improving Large-Scale Image Retrieval Through Robust Aggregation of Local Descriptors[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(9): 1783-1796.

[9] DO T T, TAN D K L, PHAM T T, et al. Simultaneous Feature Aggregating and Hashing for Large-Scale Image Search[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017.

[10] BENTLEY J L. Multidimensional binary search trees used for associative searching[J]. Com. of ACM, 1975, 18(9): 509-517.

[11] FINKEL R A, BENTLEY J L. Quad trees: a data structure for retrieval on composite keys[J]. Acta Inf. 1974, 4(1): 1-9.

[12] LEE D T, WONG C K. Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees[J]. Acta Inf. 1977, 9(1): 23-29.

[13] MALKOV Y, PONOMARENKO A, LOGVINOVA, et al. Approximate nearest neighbor algorithm based on navigable small world graphs[J]. Information Systems, 2014, 45: 61-68.

[14] MALKOV Y A, YASHUNIN D A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, PP(99).

[15] MATTHIJS D, ALEXANDRE S, HERVE J. Link and Code: Fast Indexing with Graphs and Compact Regression Codes [C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018.

[16] MARTIN A, BERNHARDSSON E, FAITHFULL A. ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms[C]// International Conference on Similarity Search and Applications. 2017.

[17] MACQUEEN J. Some Methods for Classification and Analysis of Multi Variate Observations[C]// Proc. of Berkeley Symposium on Mathematical Statistics & Probability. 1965.

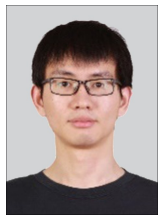
[18] LI Y, CHUNG S M. Parallel bisecting k-means with prediction clusteringalgorithm [J]. Journal of Supercomputing, 2007, 39(1): 19-37.

[19] MCCALLUM A, NIGAM K, UNGAR L H. Efficient clustering of high-dimensional data sets with application to reference matching[C]// Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining. DBLP, 2000.

[20] ARTHUR D, VASSILVITSKII S. k-means++: the advantages of careful seeding[C]// Eighteenth Acm-Siam Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics. 2007: 1027-1035.

[21] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv: 1409.1556, 2014.

[22] JIA Y, SHELHAMER E, DONAHUE J, et al. Caffe: Convolutional architecture for fast feature embedding[C]// Proceedings of the 22nd ACM International Conference on Multimedia. ACM, 2014: 675-678.



WANG Xiao-fei, born in 1995, postgraduate. His main research interests include image processing and model processing.

LIU Li-gang, born in 1975, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include geometry modeling, computational fabrication and shape analysis.