

DOI:10.3969/j.issn.1673-4785.201311020

网络出版地址: <http://www.cnki.net/kcms/detail/23.1538.tp.20150402.1518.001.html>

# 一种基于参考点距离的 SIFT 特征点匹配算法

唐坤<sup>1</sup>, 韩斌<sup>2</sup>

(1. 东南大学 交通学院, 江苏 南京 210096; 2. 江苏科技大学 计算机科学与工程学院, 江苏 镇江 212000)

**摘要:**针对 SIFT 特征点匹配时间消耗大的问题,提出了一种基于参考点距离的 SIFT 特征点匹配算法—DRP 算法。该算法首先计算一次所有待匹配特征点到参考点之间的距离,对之进行快速排序并保存。然后计算待查询特征点到参考点的距离,并在已排序的距离中使用二分法搜索返回此距离的最近邻。最后以此最近邻为中心,在有限范围内搜索待查询特征点的近似最近邻。VGG 实验室 ACF 图片库的测试结果表明,相比于经典的 SIFT 算法,DRP 算法可以在不损失匹配效果的前提下,有效降低 SIFT 特征点匹配的时间消耗。

**关键词:**SIFT; DRP 算法; 特征点匹配; 最近邻; 参考点

**中图分类号:**TP319 **文献标志码:**A **文章编号:**1673-4785(2015)03-0376-05

中文引用格式:唐坤,韩斌.一种基于参考点距离的 SIFT 特征点匹配算法[J].智能系统学报,2015,10(3):376-380.

英文引用格式:TANG Kun, HAN Bin. A SIFT matching algorithm based on the distance to reference point[J]. CAAI Transactions on Intelligent Systems, 2015, 10(3): 376-380.

## A SIFT matching algorithm based on the distance to reference point

TANG Kun<sup>1</sup>, HAN Bin<sup>2</sup>

(1. School of Transportation, Southeast University, Nanjing 210096, China; 2. School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212000, China)

**Abstract:**To address the high time cost of feature point matching in scale invariant feature transform (SIFT), a new SIFT feature point matching algorithm based on the distance to reference point—DRP algorithm is put forward. Firstly, distances from the reference point to every feature point to be matched is computed using DRP algorithm. Then, these distances computed previously is ordered and saved in a dataset named as distance of ordering. Next, distances from the reference point to the feature point to be queried is also computed. After that, the nearest neighbor of the distance in distance of ordering is retrieved with binary search and returned as index of center. Finally, the nearest neighbor of feature point to be queried is searched one by one in a certain range whose center is index of center. It is proven by experiment tested on ACF (affine covariant features) pictures from VGG (visual geometry group) laboratory that DRP algorithm can effectively decrease the time cost of SIFT feature points matching without loss of matching results compared with the classical SIFT algorithm.

**Keywords:**scale invariant feature transform (SIFT); distance to reference point (DRP) algorithm; feature point matching; nearest neighbor; reference point

Lowe 提出的尺度不变特征变换(scale invariant feature transform, SIFT)<sup>[1]</sup>是一种鲁棒性强的图像局

部特征提取算法,具有对旋转、光照、尺度变化等保持不变性<sup>[2]</sup>,广泛应用于三维重建、目标识别、图像融合等领域<sup>[3]</sup>。SIFT 算法由特征点检测及描述与特征点匹配两部分构成,其中,SIFT 特征点匹配实质上可以转化为在高维空间中搜索特征点最近邻的问题,并且在很多情况下,只需要得到查询特征点的

收稿日期:2013-11-28. 网络出版日期:2014-04-02.

基金项目:国家自然科学基金资助项目(61374195);中央高校基本科研业务费专项资金资助项目;江苏省普通高校研究生科研创新计划资助项目(KYLX\_0180).

通信作者:唐坤. E-mail: tkpaperzc@sina.cn.

近似最近邻就足够了<sup>[4]</sup>。据此近年来提出了一些相关算法,其中,Lowe 提出的最优节点优先算法(best bin first, BBF)<sup>[5]</sup>在 K-D 树索引的基础上,使用优先队列和限制回溯查询次数来提高搜索最近邻的效率。va+file 算法<sup>[6]</sup>将数据集转换到 KLT 域,然后采用均匀量化划分来提高 va-file 算法<sup>[7]</sup>的过滤效率。iDistance 算法<sup>[8]</sup>对高维数据空间进行划分,使用 B+-tree 结果来组织参考点并搜索最近邻。LSH(locality sensitive hashing, LSH)算法<sup>[9-10]</sup>通过 Hash 函数将邻近高维点集映射到 Hash 表的同一个桶中,从而提高搜索最近邻效率。基于聚类分析的 B+-tree 算法<sup>[11]</sup>采用类 B+-tree 索引结构,通过更加细致的划分数据来加快搜索速度。MRSVQH<sup>[12]</sup>根据选择子向量的  $L_2$  范数来量化和散列特征向量,仅考虑具有与查询向量相同 Hash 值的集合,从而提高了搜索速度。

本文在以上研究的基础上,提出了一种基于参考点距离的 SIFT 特征点匹配算法—DRP 算法,并通过实验与经典的 BBF 算法相比,来验证基于 DRP 算法的 SIFT 特征点匹配效果。

## 1 DRP 算法

### 1.1 DRP 算法原理

设  $d$  维欧式空间  $\mathbf{R}^d$  中 2 个点  $\mathbf{P}_1 = (x_1, x_2, \dots, x_d)$ ,  $\mathbf{P}_2 = (y_1, y_2, \dots, y_d)$ , 两点对应的向量分别为  $\mathbf{V}_1 = [x_1 \ x_2 \ \dots \ x_d]$ ,  $\mathbf{V}_2 = [y_1 \ y_2 \ \dots \ y_d]$ 。则  $\mathbf{P}_1, \mathbf{P}_2$  两点之间的欧式距离  $D(\mathbf{P}_1, \mathbf{P}_2)$  为

$$D(\mathbf{P}_1, \mathbf{P}_2) =$$

$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_d - y_d)^2} =$$

$$\sqrt{|\mathbf{V}_1|^2 + |\mathbf{V}_2|^2 - 2 \mathbf{V}_1 \mathbf{V}_2} =$$

$$\sqrt{|\mathbf{V}_1|^2 + |\mathbf{V}_2|^2 - 2 |\mathbf{V}_1| |\mathbf{V}_2| \cos \langle \mathbf{V}_1, \mathbf{V}_2 \rangle}$$

$$|\mathbf{V}_1|^2 = x_1^2 + x_2^2 + \dots + x_d^2$$

$$|\mathbf{V}_2|^2 = y_1^2 + y_2^2 + \dots + y_d^2$$

$$\mathbf{V}_1 \mathbf{V}_2 = x_1 y_1 + x_2 y_2 + \dots + x_d y_d$$

(1)

式中:  $|\mathbf{V}_1|$ 、 $|\mathbf{V}_2|$  为向量  $\mathbf{V}_1$  与  $\mathbf{V}_2$  的模,  $\langle \mathbf{V}_1, \mathbf{V}_2 \rangle$  为向量  $\mathbf{V}_1$  与  $\mathbf{V}_2$  之间的夹角。

由式(1)可知,若  $|\mathbf{V}_1|$  确定且  $\mathbf{V}_1$  与  $\mathbf{V}_2$  的夹角一定时,当  $|\mathbf{V}_2| \rightarrow |\mathbf{V}_1| \cos \langle \mathbf{V}_1, \mathbf{V}_2 \rangle$  时,向量  $\mathbf{V}_1$  与  $\mathbf{V}_2$  对应的  $\mathbf{P}_1$  与  $\mathbf{P}_2$  两点之间的距离  $D(\mathbf{P}_1, \mathbf{P}_2)$  取最小值。

设  $d$  维欧式空间  $\mathbf{R}^d$  中的一个点集合  $F_d = \{\mathbf{P}_i | \mathbf{P}_i \in \mathbf{R}^d, i = 1, 2, \dots, n\}$ , 对于  $\forall \mathbf{P}_{\text{query}} \in \mathbf{R}^d$ ,

需要在  $F_d$  中搜索  $\mathbf{P}_{\text{query}}$  的最近邻。设定一个参考点  $\mathbf{P}_r$ , 则对于每一个需要搜索的  $\mathbf{P}_{\text{query}}$ ,  $\mathbf{P}_{\text{query}}$  与参考点  $\mathbf{P}_r$  构成向量  $\mathbf{P}_r \mathbf{P}_{\text{query}}$  是确定的(长度和方向),若  $F_d$  中所有点与设定参考点  $\mathbf{P}_r$  构成的向量  $\mathbf{P}_r \mathbf{P}_i$  与  $\mathbf{P}_r \mathbf{P}_{\text{query}}$  之间的夹角都为  $\theta$  时,则使向量  $|\mathbf{P}_r \mathbf{P}_i| \rightarrow |\mathbf{P}_r \mathbf{P}_{\text{query}}| \cos \theta$  的点  $\mathbf{P}_i$  就是  $\mathbf{P}_{\text{query}}$  的最近邻。

若集合  $F_d$  中所有特征点与设定参考点  $\mathbf{P}_r$  构成的向量  $\mathbf{P}_r \mathbf{P}_i$  与  $\mathbf{P}_r \mathbf{P}_{\text{query}}$  之间的夹角不都为  $\theta$  时,即不相等时,上述结论并不一定成立。如图 1 的三维空间所示。

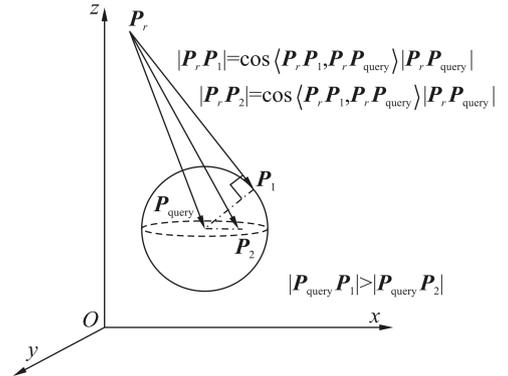


图 1 不同夹角时的最小距离

Fig. 1 The minimum distance of different angle

尽管  $|\mathbf{P}_r \mathbf{P}_1| = \cos \langle \mathbf{P}_r \mathbf{P}_1, \mathbf{P}_r \mathbf{P}_{\text{query}} \rangle |\mathbf{P}_r \mathbf{P}_{\text{query}}|$ , 而  $|\mathbf{P}_r \mathbf{P}_2| > \cos \langle \mathbf{P}_r \mathbf{P}_2, \mathbf{P}_r \mathbf{P}_{\text{query}} \rangle |\mathbf{P}_r \mathbf{P}_{\text{query}}|$ 。但是,  $\mathbf{P}_2$  相对于  $\mathbf{P}_1$  更靠近  $\mathbf{P}_{\text{query}}$ 。

为了尽可能的得到  $\mathbf{P}_{\text{query}}$  的最近邻,本文将搜索范围扩大到集合  $F'_d$  中的所有点。如图 2 所示。

$$F'_d = \{\mathbf{P}_i | R_{\text{th}}^- < |\mathbf{P}_r \mathbf{P}_i| < R_{\text{th}}^+, R_{\text{th}}^-, R_{\text{th}}^+ \geq 0\}$$

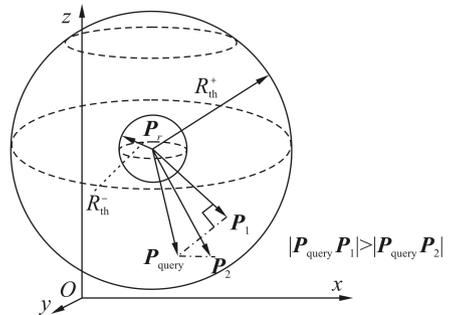


图 2 在一定区域内搜索最近邻

Fig. 2 Search the nearest neighbor in a specified area

若将搜索范围扩大到以  $\mathbf{P}_r$  为球心,  $R_{\text{th}}^-$  为半径的球外,  $R_{\text{th}}^+$  为半径的球内区域,则可得到  $\mathbf{P}_{\text{query}}$  的真正最近邻  $\mathbf{P}_2$ 。随着  $R_{\text{th}}^-$  的减小和  $R_{\text{th}}^+$  的增大,搜索到  $\mathbf{P}_{\text{query}}$  的真正最近邻的概率也随之增大。由此可知,当  $F_d$  中所有点与设定参考点  $\mathbf{P}_r$  构成的向量  $\mathbf{P}_r \mathbf{P}_i$  与  $\mathbf{P}_r \mathbf{P}_{\text{query}}$  之间的夹角不相等时,只需要选择一个合适的搜索范围  $R_{\text{th}}^-$  与  $R_{\text{th}}^+$ , 仍然可以搜索到  $\mathbf{P}_{\text{query}}$  的真正最近邻。

### 1.2 DRP 算法描述

设  $d$  维欧式空间  $\mathbf{R}^d$  中的 2 个点集合:

$$Q_d = \{P_i | P_i \in \mathbf{R}^d, i = 1, 2, \dots, m\}$$

$$Q'_d = \{P_q | P_q \in \mathbf{R}^d, q = 1, 2, \dots, n\}$$

对于  $\forall P_q \in Q'_d$ , 需要在集合  $Q_d$  中搜索其最近邻。引入一个参考点(可随机选取)  $P_r \in \mathbf{R}^d$ , 若  $P_{nearest}$  ( $P_{nearest} \in Q_d$ ) 是  $P_q$  的最近邻, 则  $P_{nearest}$  应该出现在  $P_q$  的附近, 即向量  $P_r P_q$  与向量  $P_r P_{nearest}$  之间的夹角  $\langle P_r P_q, P_r P_{nearest} \rangle$  应该趋向于零, 此时  $|P_r P_{nearest}| = |P_r P_q| \times \cos \langle P_r P_q, P_r P_{nearest} \rangle \rightarrow |P_r P_q|$ 。因此, 当  $|P_r P_{nearest}| = |P_r P_q|$  时, 点  $P_{nearest}$  为查询点  $P_q$  的最近邻。由此, 对于  $\forall P_q \in Q'_d$ , 不需要逐一计算  $P_q$  与  $Q_d$  集合中所有点的距离, 而只要计算  $P_q$  与参考点  $P_r$  之间的距离  $D_{rq}$ , 然后在点集合  $Q'_d$  中搜索, 就可能得到  $P_q$  的真正最近邻, 即  $P_q$  的真正最近邻落在集合  $Q'_d$  所表示的范围中。

$$Q'_d =$$

$$\{P_i | d_{rq} - \Delta d_{th}^- < |P_r P_i| < d_{rq} + \Delta d_{th}^+, P_i \in Q'_d\}$$

以图 3 所示的三维空间为例, 如果搜索向量  $P_q$  的最近邻, 只需要在  $P_r$  为球心, 半径为  $d_{rq} - \Delta d_{th}^-$  的球以外和半径为  $d_{rq} + \Delta d_{th}^+$  以内的空间搜索, 就可以得到  $P_q$  的真正最近邻。

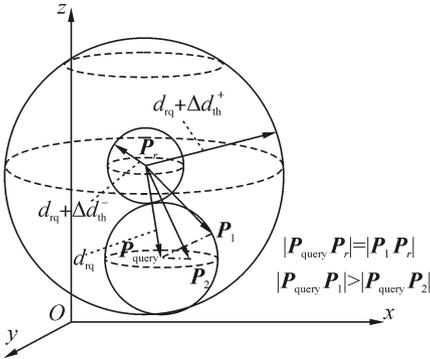


图 3 DRP 算法搜索最近邻

Fig. 3 Search the nearest neighbor with DRP algorithm

基于以上分析, 在向量集合  $F_d$  中搜索向量  $V_q$  的最近邻的步骤如下:

- 1) 在  $d$  维欧式空间  $\mathbf{R}^d$  中任意选取一点  $P_r$  作为参考点, 设定一个阈值  $th$  (正整数)。
- 2) 对于任一属于集合  $F_d$  中的点  $P_i$  ( $i = 1, 2, \dots, m$ ), 计算其与参考点  $P_r$  之间的距离, 保存于数组  $D_{ri}$  中, 并建立  $D_{ri}$  与  $P_i$  的映射。
- 3) 使用快速排序算法对  $D_{ri}$  中的距离按升序 (或降序) 进行排序, 得到  $D_{ri}^{order}$ 。
- 4) 从待匹配点集合  $F'_d$  中任取一待匹配点  $P_{query}$ , 计算  $P_{query}$  与参考点  $P_r$  之间的距离  $D_{qr} = |P_r P_{query}|$ 。

5) 使用二分查找法从  $D_{ri}^{order}$  中快速查找  $D_{ri}$  的最近邻  $D_{rj}$ , 并得到  $D_{rj}$  对应的点  $P_j^{nearest}$  及其索引  $index(j)$ 。

6) 对  $D_{ri}^{order}$  中索引范围在  $[index(j) - th, index(j) + th]$  对应的所有点中, 利用穷举搜索法查找  $P_{query}$  的最近邻。

在步骤 6) 中涉及到距离的计算, 为进一步提高算法的效率, 均采用如下所述的距离累计算法。当  $d$  维点  $P_x$  与  $P_{query}$  的前  $i$  ( $i \in [1, d]$ ) 维的欧式距离已经超过上一次比较的最小距离时, 无论后面  $d - i$  维的距离情况如何, 将必定大于已经产生的最小距离。因此, 也无需计算它们后面  $d - i$  维的距离, 直接剔除跳出, 进行下一特征点的比较。

## 2 SIFT 特征点快速匹配算法

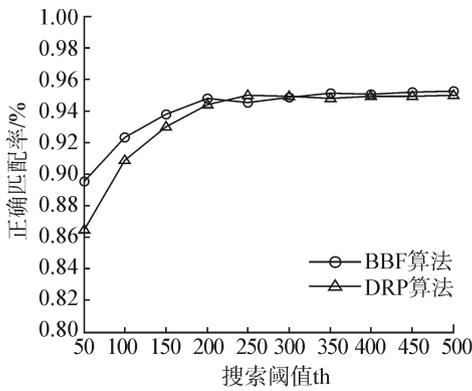
SIFT 算法提取的图像特征点包括极大值与极小值 2 种类型, 显而易见, 只有极值类型相同的特征点才可能正确匹配。因此, 本文在特征点匹配之前进行特征点类型的判断, 仅当 2 个特征点具有相同类型时, 才进行接下来的匹配运算, 否则直接跳出, 转而比较下一个特征点。这在一定程度上可以提高匹配的效率。

对于 2 幅含有同一场景的图像  $I_q(x, y)$  和  $I_s(x, y)$ , 本文介绍的 DRP 算法按如下步骤进行。

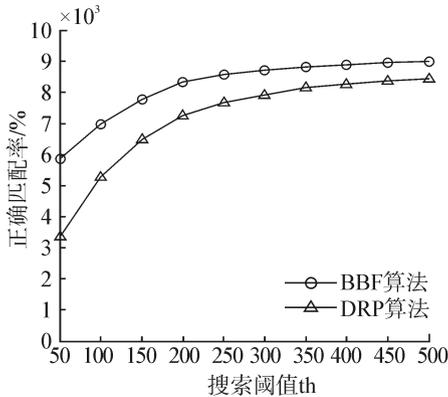
- 1) 使用 SIFT 算法分别提取图像  $I_q(x, y)$  与图像  $I_s(x, y)$  的特征点, 生成的特征点构成的集合分别为  $P_q^{128}$  和  $P_s^{128}$ 。
- 2) 对于  $\forall P_i \in P_q^{128}$ , 使用本文介绍的 DRP 算法在集合  $P_s^{128}$  中搜索得到查询点  $P_i$  的最近邻特征点  $P_{nearest}$ , 对应的最近邻距离为  $d_{nearest}$ , 次近邻特征点  $P_{nearest2}$ , 对应的次近邻距离  $d_{nearest2}$ , 若比值  $d_{nearest}/d_{nearest2}$  小于已知的阈值  $r_{th}$ , 则认为  $P_i$  与  $P_{nearest}$  两点之间正确匹配。
- 3) 采用随机抽样一致性原则 (random sample consensus, RANSAC) 算法剔除误匹配的特征点。

## 3 实验结果与分析

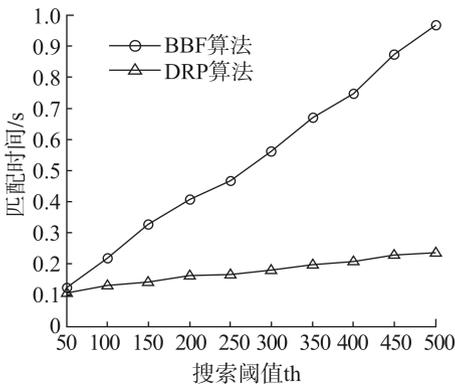
本文实验采用的代码由 GitHub 上 Rob Hess 维护的 Opensift 库<sup>[13]</sup>改进而来。实验采用的测试图片来自牛津大学 VGG 实验室的 Affine Covariant Features (ACF) 图片库<sup>[14]</sup>。实验硬件环境为 CPU Inter (R) Core (TM) 2 T6600, 主频 2.20 GHz, 内存 2 GB DDR3 RAM; 软件环境为 OS Windows 7 32 bit、IDE Microsoft Visual Studio 2010、GTK+2.24 图形工具包和 Opencv2.4 机器视觉库。



(a) 特征点正确匹配率与搜索阈值 th 的关系



(b) 正确匹配特征点总数与搜索阈值 th 的关系



(c) 平均匹配时间与搜索阈值 th 的关系

图 4 DRP 算法与 BBF 算法实验对比结果

Fig. 4 The comparison results of DRP algorithm and BBF algorithm

为了验证 DRP 算法的有效性,本文进行了在不同搜索阈值  $th$  下, DRP 算法与经典 BBF 算法的对比实验。简便起见,所有实验中的参考点均采用坐标原点,即  $P_r = (x_r^j)$ ,  $x_r^j = 0, j = 1, 2, \dots, 128$ 。其中,交叉验证两算法的最近邻与次近邻距离阈值比  $r_{th}$  保持为 0.56。BBF 算法中除搜索阈值 KDTREE\_BBFF\_MAX\_NN\_CHK5 变化外,其他所有参数保持默认。实验中搜索阈值  $th$  以 50 为步长在  $[50, 500]$  范围内变化,在不同搜索阈值  $th$  条件下,对 Bikes、

Trees、Graf、Bark、Boat、Leuven 6 组图片每组中的  $Img1/Img2, Img1/Img3, Img1/Img4$  3 对图片各进行 10 次 SIFT 特征点匹配实验。统计得到特征点的正确匹配率、正确匹配特征点总数以及平均每幅图像匹配时间与搜索阈值  $th$  的关系如图 4 所示。

由图 4(a) 可知,在特征点正确匹配率方面, DRP 算法与 BBF 算法具有相同的趋势。当搜索阈值  $th$  小于 200 时,随着  $th$  的增大而增大;当搜索阈值  $th$  大于 200 时,特征点正确匹配率很难得到进一步的改善,趋于稳定。总体来看,这 2 个算法在特征点匹配率的表现上相差无几。由图 4(b) 可知,在正确匹配特征点总数方面, DRP 的算法略逊于 BBF 算法,但是,随着搜索阈值  $th$  的增大,两者的差距逐渐缩小。由图 4(c) 可知,在匹配时间上, DRP 算法与 BBF 算法都随着搜索阈值  $th$  的增大而呈线性增长趋势,但是, BBF 的增长速率显著大于 DRP 算法,由此可知,在匹配时间上, DRP 算法较 BBF 算法具有明显的优势。综上所述,相对于经典的 BBF 算法,本文介绍的 DRP 算法能获得满意匹配效果的同时,有效降低匹配时间。

## 4 结束语

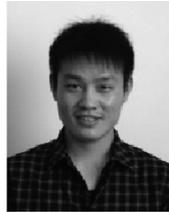
本文介绍了一种基于参考点距离的 SIFT 特征点匹配算法,并使用牛津大学 VGG 实验室的 ACF 图片库进行 SIFT 特征点匹配实验。实验结果表明,与经典 BBF 最近邻搜索算法相比,使用本文介绍的 DRP 算法进行 SIFT 特征点匹配,不仅可以获得满意的匹配效果,并且可以有效地降低匹配时间消耗,提高匹配速度。尤其在搜索阈值  $th$  较大的情况下, DRP 算法在时间成本上较经典的 BBF 算法具有明显的优势。另外, DRP 算法还有一些可以改进的地方,如对提取的 SIFT 特征点分布情况进行分析、参考点选择方法的分析等都可能进一步提高特征点的匹配效率。对于不同的特征点分布情况,如何选取一个合适的参考点,从而尽可能改善匹配效果是下一步的工作重心。

## 参考文献:

- [1] LOWE D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60(2): 91-110.
  - [2] 吴伟交, 王敏, 黄心汉, 等. 基于向量夹角的 SIFT 特征点匹配算法 [J]. 模式识别与人工智能, 2013, 26(1): 123-127.
- WU Weijiao, WANG Min, HUANG Xinhan, et al. SIFT feature matching algorithm based on vector angle. Pattern

- Recognition and Artificial Intelligence, 2013, 26(1): 123-127.
- [3] KRYSTIAN M, CORDELIA S. A performance evaluation of local descriptors[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(10): 1615-1630.
- [4] GIONIS A, INDYK P, MOTWANI R. Similarity search in high dimensions via hashing[C]//Proceedings of the 25th International Conference on Very Large Data Bases. San Francisco, USA, 1999: 518-529.
- [5] BEIS J S, LOWE D G. Shape indexing using approximate nearest-neighbor search in high-dimensional spaces[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. San Juan, Puerto Rico, 1997: 1000-1006.
- [6] FERHATOSMANOGLU H, TUNCEL E, AGRAWAL D, et al. Vector approximation based indexing for non-uniform high dimensional data sets[C]//Proceedings of the International Conference on Information and Knowledge Management. McLean, USA, 2000: 202-209.
- [7] WEBER R, SCHEK H J, BLOTT S. A quantitative analysis and performance study for similarity-search methods in high dimensional spaces[C]//Proceedings of the International Conference on Very Large Databases. New York, 1998: 194-205.
- [8] JAGADISH H V, OOI B C, TAN K L, et al. IDistance: an adaptive B<sup>+</sup>-tree based indexing method for nearest neighbor search[J]. ACM Transactions on Database Systems, 2005, 30(2): 364-398.
- [9] AUCLAIR A, COHEN L D, VINCENT N. How to use SIFT vectors to analyze an image with database templates[C]//Proceedings of the 5<sup>th</sup> International Workshop on Adaptive Multimedia Retrieval. Paris, France, 2008: 224-236.
- [10] SLANEY M, CASEY M. Locality-sensitive hashing for finding nearest neighbors[J]. IEEE Signal Processing Magazine, 2008, 25(2): 128-131.
- [11] 张军旗, 周向东, 王梅, 等. 基于聚类分解的高维度量空间索引 B<sup>+</sup>-tree[J]. 软件学报, 2008, 19(6): 1401-1412.
- ZHANG Junqi, ZHOU Xiangdong, WANG Mei, et al. Cluster splitting based high dimensional metric space index B<sup>+</sup>-tree[J]. Journal of Software, 2008, 19(6): 1401-1412.
- [12] 杨恒, 王庆, 何周灿. 面向高维图像特征匹配的多次随机子向量量化哈希算法[J]. 计算机辅助设计与图形学学报, 2010, 22(3): 494-502.
- YANG Heng, WANG Qing, HE Zhoucan. Multiple randomized sub-vectors quantization hashing for high-dimensional image feature matching[J]. Journal of Computer-Aided Design & Computer Graphics, 2010, 22(3): 494-502.
- [13] LOWE D. OpenSIFT—An open-source SIFT library[EB/OL]. [2013-11-15]. <http://robwhess.github.io/opensift/>.
- [14] The Visual Geometry Group. Affine covariant features[EB/OL]. (2007-07-15) [2013-11-15]. <http://www.robots.ox.ac.uk/~vgg/research/affine/index.html>.

#### 作者简介:



唐坤,男,1988年生,博士研究生,主要研究方向为数字图像处理、智能交通。



韩斌,男,1968年生,教授,博士,主要研究方向为数字图像处理、智能检测、并行计算。