



(12)发明专利

(10)授权公告号 CN 104021234 B

(45)授权公告日 2017.04.19

(21)申请号 201410305838.1

(22)申请日 2014.06.30

(65)同一申请的已公布的文献号

申请公布号 CN 104021234 A

(43)申请公布日 2014.09.03

(73)专利权人 中国科学院自动化研究所

地址 100190 北京市海淀区中关村东路95号

(72)发明人 郭勤振 曾智 张树武

(74)专利代理机构 中科专利商标代理有限责任公司 11021

代理人 宋焰琴

(51)Int.Cl.

G06F 17/30(2006.01)

G06K 9/62(2006.01)

(56)对比文件

CN 101894130 A,2010.11.24,

CN 101710334 A,2010.05.19,

CN 102508910 A,2012.06.20,

US 2011194737 A1,2011.08.11,

审查员 曹丽冉

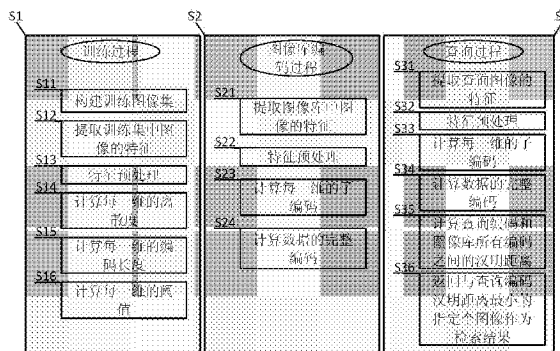
权利要求书3页 说明书7页 附图3页

(54)发明名称

一种基于自适应位分配哈希算法的大规模图像库检索方法

(57)摘要

一种基于自适应位分配哈希算法的大规模图像库检索方法,包括:从待检索图像库中选取部分图像作为训练集,提取训练集的GIST特征;利用PCA对训练集的特征数据进行投影,并对训练数据计算每一维的离散度;根据不同维的离散度,自适应地分配不同的位数来编码数据;根据每一维的编码长度和阈值编码每一维,得子编码,拼接成对数据的完整编码;对查询图像和待检索图像库中特征数据的处理与训练过程对应,分别计算待检索图像和查询图像特征的哈希编码;计算两者之间的汉明距离,由此快速检索相似图像。本发明具有很好的普适性,能很好地保持原始特征数据的近邻结构,利用哈希方法对数据进行编码,既减少了数据的存储空间又提高了查询的检索效率。



1. 一种基于自适应位分配哈希算法的大规模图像库检索方法,包括下列步骤:

步骤S1: 训练步骤,从待检索图像库中选取部分图像作为训练集,计算训练集的编码长度 c_i 和编码阈值 t_i^j ,其中 $1 \leq i \leq d, 1 \leq j \leq (c_i+1)$, d 为维度;其中所述训练步骤S1中根据每一维度的离散度自适应地为每一维分配不同的位数,第 i 维的编码长度 c_i 计算公式如下:

$$c_i = \begin{cases} \left[k \cdot \frac{\lambda_i}{\sum_{n=i}^d \lambda_n} + 0.5 \right] & i=1 \\ \left[(k - \sum_{i=1}^{i-1} c_i) \cdot \frac{\lambda_i}{\sum_{n=i}^d \lambda_n} + 0.5 \right] & i \geq 2 \end{cases};$$

其中, k 是总的编码长度, λ_i 为每一维的离散度;

步骤S2: 待检索图像库编码步骤,根据训练步骤得到的编码长度 c_i 和编码阈值 t_i^j 计算待检索图像的哈希编码 $H(y_r)$,其中 y_r 为待检索图像库中第 r 幅图像的GIST特征;

步骤S3: 查询步骤,先根据训练步骤得到的编码长度 c_i 和编码阈值 t_i^j 计算查询图像的哈希编码 $H(q)$,其中 q 为查询图像 d 维的GIST特征;再根据查询数据编码 $H(q)$ 与待检索图像库中数据编码 $H(y_i)$ 之间的汉明距离 $D_h(H(q), H(y_i))$,返回待检索图像库中与查询数据汉明距离最小的 L 个数据作为检索结果,其中 N 为待检索图像库的大小, $1 \leq i \leq N$, L 为由用户设定的正整数。

2. 根据权利要求1所述的基于自适应位分配哈希算法的大规模图像库检索方法,其中所述训练步骤S1包括:

步骤S11: 从待检索图像库 D 中选取部分图像 T 作为训练集, n 为训练集的大小, $n \leq N$;

步骤S12: 对训练集 T 中的图像提取 d 维的GIST特征,其中图像 T_i 的GIST特征记为 x_i ,将训练集中所有图像的GIST特征组成矩阵 X , X 的第 i 行由 x_i 构成;

步骤S13: 对训练集中的图像特征做PCA旋转,以实现投影预处理,计算均值 μ 和投影矩阵 P ;

步骤S14: 计算数据投影后每一维度上的离散度 λ , X 第 i 维的离散度计算公式为:

$$\lambda_i = \frac{1}{n} \sum_{j=1}^n (X_j^i - \frac{1}{n} \sum_{m=1}^n X_m^i)^2$$

其中, X_j^i 为 X 中第 j 行第 i 列(维)的数据, $1 \leq i \leq d$;

步骤S15: 根据每一维度的离散度自适应地为每一维分配不同的位数,第 i 维的编码长度 c_i 计算公式如下:

$$c_i = \begin{cases} \left[k \cdot \frac{\lambda_i}{\sum_{n=i}^d \lambda_n} + 0.5 \right] & i=1 \\ \left[(k - \sum_{i=1}^{i-1} c_i) \cdot \frac{\lambda_i}{\sum_{n=i}^d \lambda_n} + 0.5 \right] & i \geq 2 \end{cases}$$

其中, k 是总的编码长度, $1 \leq i \leq d$;

步骤S16: 对于每一维, 根据该维分配的位数计算该维度的阈值; 对于第 i 维, 其编码长度为 c_i , 计算第 i 维的编码阈值 t_i^j , 其中 $1 \leq j \leq (c_i+1)$ 。

3. 根据权利要求2所述的基于自适应位分配哈希算法的大规模图像库检索方法, 其中所述待检索图像库编码步骤S2包括:

步骤S21: 对待检索图像库中的每一幅图像提取 d 维的GIST特征, 假设提取的第 r 幅待检索图像库中图像的GIST特征为 y_r ;

步骤S22: 利用步骤S13中训练得到的PCA投影参数均值 μ 和投影矩阵 P 对 y_r 进行PCA预处理, 具体操作为: 先将 y_r 减去均值 μ , 然后利用投影矩阵进行旋转, 即 $(y_r - \mu)P$;

步骤S23: 由步骤S15和步骤S16中训练得到的每一维的编码长度 c_i 和每一维的阈值 t_i^j , 分别对 y_r 的每一维进行编码, 得到第 i 维的子编码 s_i ;

步骤S24: 由每一维度的子编码 s_i 拼接得到最后完整的对图像库特征数据 y_r 的自适应位分配哈希编码 $H(y_r) = (s_1, s_2, \dots, s_d)$, 其中 $1 \leq i \leq d$ 。

4. 根据权利要求2所述的基于自适应位分配哈希算法的大规模图像库检索方法, 其中所述查询步骤S3包括:

步骤S31: 对查询图像提取 d 维的GIST特征 q ;

步骤S32: 利用步骤S13中训练得到的PCA投影参数均值 μ 和投影矩阵 P 对 q 进行PCA预处理, 具体操作为: 先将 q 减去均值 μ , 然后利用投影矩阵进行旋转, 即 $(q - \mu)P$;

步骤S33: 由步骤S15和步骤S16中训练得到的每一维的编码长度 c_i 和每一维的阈值 t_i^j , 分别对 q 的每一维进行编码, 得到第 i 维的子编码 s_i , 其中 $1 \leq j \leq (c_i+1)$;

步骤S34: 由每一维度的子编码 s_i 得到最后对完整查询特征数据 q 的编码, 其中 $1 \leq i \leq d$, 由此得到最终的对特征数据 q 的自适应位分配哈希编码 $H(q) = (s_1, s_2, \dots, s_d)$;

步骤S35: 计算查询数据编码 $H(q)$ 与待检索图像库中每一个特征数据的编码 $H(y_i)$ 之间的汉明距离 $D_h(H(q), H(y_i))$

$$D_h(H(q), H(y_i)) = \sum_{j=1}^k H(q)[j] \oplus H(y_i)[j]$$

$H(q)[j]$ 和 $H(y_i)[j]$ 分别表示 $H(q)$ 和 $H(y_i)$ 的第 j 位, \oplus 表示异或操作, k 为总的编码长度, $1 \leq i \leq N$;

步骤S36: 根据查询数据编码 $H(q)$ 与待检索图像库中数据编码 $H(y_i)$ 之间的汉明距离 $D_h(H(q), H(y_i))$, 返回待检索图像库中与查询数据汉明距离最小的 L 个数据作为检索结果, 其中 $1 \leq i \leq N$ 。

5. 根据权利要求2所述的基于自适应位分配哈希算法的大规模图像库检索方法, 其中对于每一维, 根据该维分配的位数计算该维度的阈值的步骤S16, 进一步包括步骤:

步骤S161: 对于第 i 维, 其编码长度是 c_i , 利用 k 均值聚类算法将 X 的第 i 列的 n 个数据聚成 (c_i+1) 个簇;

步骤S162: 将聚类中心从小到大排序, 得到第 i 维的编码阈值 t_i^j , 其中 $1 \leq j \leq (c_i+1)$ 。

6. 根据权利要求3所述的基于自适应位分配哈希算法的大规模图像库检索方法, 其中

利用训练得到的每一维的编码长度 c_i 和阈值 t_i^j 分别对图像库特征数据的每一维进行编码的步骤S23,进一步包括步骤:

步骤S231:对于第 i 维,其编码长度是 c_i ,编码阈值为 t_i^j ,将待检索图像数据库GIST特征 y 的第 i 维按照最近邻规则分配到 t_i^j 形成的簇中,其中 $1 \leq j \leq (c_i+1)$;

步骤S232:若 y 的第 i 维的最近邻簇为 t_i^m 所形成的簇,则 y 的第 i 维的子编码 s_i 由连续的 (c_i+1-m) 个0和紧接着连续的 $(m-1)$ 个1组成,即

$$s_i = (\overbrace{0, \dots, 0}^{c_i+1-m}, \overbrace{1, \dots, 1}^{m-1})。$$

7.根据权利要求6所述的基于自适应位分配哈希算法的大规模图像库检索方法,其中利用训练得到的每一维的编码长度 c_i 和阈值 t_i^j 分别对查询数据 q 的每一维进行编码的步骤S33进一步包括步骤:

步骤S331:对于第 i 维,其编码长度是 c_i ,编码阈值为 t_i^j ,将查询图像GIST特征 q 的第 i 维按照最近邻规则分配到 t_i^j 形成的簇中,其中 $1 \leq j \leq (c_i+1)$;

步骤S332:若 q 的第 i 维的最近邻簇为 t_i^m 所形成的簇,则 q 的第 i 维的子编码 s_i 由连续的 (c_i+1-m) 个0和紧接着连续的 $(m-1)$ 个1组成,即

$$s_i = (\overbrace{0, \dots, 0}^{c_i+1-m}, \overbrace{1, \dots, 1}^{m-1})。$$

8.根据权利要求1-7任一所述的基于自适应位分配哈希算法的大规模图像库检索方法,其中维度 $d=512$ 。

一种基于自适应位分配哈希算法的大规模图像库检索方法

技术领域

[0001] 本发明属于图像检索技术领域,涉及一种基于内容的图像检索方法,尤其涉及一种基于自适应位分配哈希算法的大规模图像库检索方法。

背景技术

[0002] 随着互联网上图片数据的日益增多,如何快速、准确地为用户提供所需要的图片资源显得越来越重要。基于内容的图像检索(content-based image retrieval,CBIR)可以比较好地解决这个问题,因此受到了许多研究者的关注。现有的检索方法通过提取图像的底层特征对图像内容进行描述,然后利用特征比对判断是否为相似图像。因此,CBIR主要包括两部分的核心研究内容,一是有效的图像特征表述,二是高效检索算法。本发明主要解决高效检索算法问题。

[0003] 高效的检索算法主要包括:基于树的检索算法和基于哈希的检索算法。基于树的检索算法利用超平面递归划分整个数据空间,在数据比较低维的情况下效果很好,但是当数据维度比较高的时候,基于树的检索算法会退化成穷尽搜索。基于哈希的检索算法的主要思想是将原始数据映射成汉明空间中的二值串(binary string),数据之间的相似度可以利用它们在汉明空间的二值串之间的汉明距离来度量。基于哈希的高效检索算法有两个主要优点:一是可以减少数据存储空间;二是可以提高检索效率。

[0004] 局部敏感哈希(locality-sensitive hashing,LSH) [Mayur Datar,Nicole Immorlica,Piotr Indyk and Vahab S.Mirrokhani.Locality-sensitive hashing scheme based on p-stable distributions.In Proceedings of the twentieth annual symposium on computational geometry,ACM,2004] 利用c个投影函数来对原始数据分别进行投影,再把投影后的数据阈值化为0和1,这样就得到了原始数据的c位的编码。但是由于LSH的投影函数是数据无关的(data-independent),随机产生的,并且产生的投影函数可能彼此是相关的,因此LSH编码的效果不是很理想。

[0005] 为了克服LSH的缺点,谱哈希(spectral hashing,SH) [Yair Weiss,Antonio Torralba,and Rob Fergus.Spectral Hashing.In NIPS,2008] 根据原始数据,利用机器学习的方法寻找合适的投影函数,建立哈希构造机制。主成分哈希(PCA hashing,PCAH) [Bin Wang,Zhiwei Li,Mingjing Li and Wei-Ying Ma.Efficient duplicate image detection algorithm for web images and large-scale database.In ICME,2006.] 首先利用PCA对数据进行投影,然后利用每一维度的均值将数据进行阈值化为0,1来对数据进行编码。但是数据经过PCA投影之后,每个维度的方差非常不均匀,差别很大,因此每一维度同等对待地利用1-bit来进行编码是不合理的,并且实验也验证了PCAH的这个缺点。各向同性哈希(Isotropic hashing,IsoH) [Weihao Kong and Wu-Jun Li.Isotropic hashing.In NIPS,2012.] 的提出就是为了解决这个问题,在IsoH中,数据被PCA投影之后,会被一个学习到的正交各向同性矩阵重新投影,经过两次投影之后,数据在每一维度的方差都是相等的,之后再使用1-bit来分别编码每一维。但IsoH存在不同维度具有不同信息,利用同样的位数来

编码并不合理的问题。

发明内容

[0006] 针对上述问题,本发明提出了一种自适应位分配哈希算法(Adaptive bit allocation hashing, ABAH),根据不同维度的离散度,自适应地分配不同的位数来编码相应的维度。本发明的特点在于,对于投影后的数据,离散度比较大的维度会被更多的位数来编码,离散度比较小的维度会被比较少的位数来编码。经过ABAH编码之后,数据之间的相似度可以利用它们在汉明空间的编码之间的汉明距离来度量,而汉明空间的ABAH编码可以很好地保持原始数据的近邻结构。

[0007] 由此,本发明可以解决针对海量图像检索存在的图像特征库存储空间大,检索速度慢的问题,克服了LSH、SH、PCAH方法存在的不足。

[0008] 本发明提出的一种基于自适应位分配哈希算法的大规模图像库检索方法的技术方案是:对于图像库中的图像,选取一部分图像作为训练图像集;对于图像库和训练集,都采用特征描述符提取特征向量,作为检索特征;然后,通过训练集的特征向量,来训练构造出哈希函数;根据训练得到的哈希函数对特征库中的每一个特征向量进行哈希编码,将特征向量映射到汉明空间中;对于查询图像,首先提取其图像表述的特征向量,然后将特征向量映射到汉明空间,之后计算查询特征向量编码与图像库中特征向量编码之间的汉明距离,利用距离大小来衡量查询图像与图像库中待检索图像之间的相似性,返回相似度高的图像。

[0009] 本发明的一种基于自适应位分配哈希算法的大规模图像库检索方法的具体实现步骤包括:

[0010] 步骤S1:训练步骤,从待检索图像库中选取部分图像作为训练集,计算训练集的编码长度 c_i 和编码阈值 t_i^j ,其中 $1 \leq i \leq d, 1 \leq j \leq (c_i+1)$, d 为维度;

[0011] 步骤S2:待检索图像库编码步骤,计算待检索图像的自适应位分配的哈希编码 $H(y_r)$,其中 y_r 为待检索图像库中第 r 幅图像的GIST特征;

[0012] 步骤S3:查询步骤,先计算查询图像的自适应位分配的哈希编码 $H(q)$,为查询图像 d 维的GIST特征;再根据查询数据编码 $H(q)$ 与待检索图像库中数据编码 $H(y_i)$ 之间的汉明距离 $D_h(H(q), H(y_i))$,返回待检索图像库中与查询数据汉明距离最小的 L 个数据作为检索结果,其中 N 为待检索图像库的大小, $1 \leq i \leq N, L$ 为由用户设定的正整数。

[0013] 优选地,所述训练步骤S1中根据每一维度的离散度自适应地为每一维分配不同的位数,第 i 维的编码长度 c_i 计算公式如下:

$$[0014] \quad c_i = \begin{cases} \left\lceil k \cdot \frac{\lambda_i}{\sum_{n=1}^d \lambda_n} + 0.5 \right\rceil & i=1 \\ \left\lceil (k - \sum_{t=1}^{i-1} c_t) \cdot \frac{\lambda_i}{\sum_{n=1}^d \lambda_n} + 0.5 \right\rceil & i \geq 2 \end{cases}$$

[0015] 其中, k 是总的编码长度, λ_i 为每一维的离散度。

[0016] 优选地,所述训练步骤S1包括:

[0017] 步骤S11:从待检索图像库D中选取部分图像T作为训练集, n 为训练集的大小, $n \leq N$;

[0018] 步骤S12:对训练集T中的图像提取 d 维的GIST特征,其中图像 T_i 的GIST特征记为 x_i ,将训练集中所有图像的GIST特征组成矩阵 X , X 的第 i 行由 x_i 构成;

[0019] 步骤S13:对训练集中的图像特征做PCA旋转,以实现投影预处理,计算均值 μ 和投影矩阵 P ;

[0020] 步骤S14:计算数据投影后每一维度上的离散度 λ , X 第 i 维的离散度计算公式为:

$$[0021] \quad \lambda_i = \frac{1}{n} \sum_{j=1}^n (X_j^i - \frac{1}{n} \sum_{m=1}^n X_m^i)^2$$

[0022] 其中, X_j^i 为 X 中第 j 行第 i 列(维)的数据, $1 \leq i \leq d$;

[0023] 步骤S15:根据每一维度的离散度自适应地为每一维分配不同的位数,第 i 维的编码长度 c_i 计算公式如下:

$$[0024] \quad c_i = \begin{cases} \left[k \cdot \frac{\lambda_i}{\sum_{n=1}^d \lambda_n} + 0.5 \right] & i=1 \\ \left[(k - \sum_{t=1}^{i-1} c_t) \cdot \frac{\lambda_i}{\sum_{n=1}^d \lambda_n} + 0.5 \right] & i \geq 2 \end{cases}$$

[0025] 其中, k 是总的编码长度, $1 \leq i \leq d$;

[0026] 步骤S16:对于每一维,根据该维分配的位数计算该维度的阈值;对于第 i 维,其编码长度为 c_i ,计算第 i 维的编码阈值 t_i^j ,其中 $1 \leq j \leq (c_i+1)$ 。

[0027] 优选地,所述待检索图像库编码步骤S2包括:

[0028] 步骤S21:对待检索图像库中的每一幅图像提取 d 维的GIST特征,假设提取的第 r 幅待检索图像库中图像的GIST特征为 y_r ;

[0029] 步骤S22:利用步骤S13中训练得到的PCA投影参数均值 μ 和投影矩阵 P 对 y_r 进行PCA预处理,具体操作为:先将 y_r 减去均值 μ ,然后利用投影矩阵进行旋转,即 $(y_r - \mu)P$;

[0030] 步骤S23:由步骤S15和步骤S16中训练得到的每一维的编码长度 c_i 和每一维的阈值 t_i^j ,分别对 y_r 的每一维进行编码,得到第 i 维的子编码 s_i ;

[0031] 步骤S24:由每一维度的子编码 s_i 拼接得到最后完整的对图像库特征数据 y_r 的自适应位分配哈希编码 $H(y_r) = (s_1, s_2, \dots, s_d)$,其中 $1 \leq i \leq d$ 。

[0032] 优选地,所述查询步骤S3包括:

[0033] 步骤S31:对查询图像提取 d 维的GIST特征 $q \in R^d$;

[0034] 步骤S32:利用步骤S13中训练得到的PCA投影参数均值 μ 和投影矩阵 P 对 q 进行PCA预处理,具体操作为:先将 q 减去均值 μ ,然后利用投影矩阵进行旋转,即 $(q - \mu)P$;

[0035] 步骤S33:由步骤S15和步骤S16中训练得到的每一维的编码长度 c_i 和每一维的阈值 t_i^j ,分别对 q 的每一维进行编码,得到第 i 维的子编码 s_i ,其中 $1 \leq j \leq (c_i+1)$;

[0036] 步骤S34:由每一维度的子编码 s_i 得到最后对完整查询特征数据 q 的编码,其中 $1 \leq i \leq d$,由此得到最终的对特征数据 q 的自适应位分配哈希编码 $H(q) = (s_1, s_2, \dots, s_d)$;

[0037] 步骤S35:计算查询数据编码 $H(q)$ 与待检索图像库中每一个特征数据的编码 $H(y_i)$ 之间的汉明距离 $D_h(H(q), H(y_i))$

$$[0038] \quad D_h(H(q), H(y_i)) = \sum_{j=1}^k H(q)[j] \oplus H(y_i)[j]$$

[0039] $H(q)[j]$ 和 $H(y_i)[j]$ 分别表示 $H(q)$ 和 $H(y_i)$ 的第 j 位, \oplus 表示异或操作, k 为总的编码长度, $1 \leq i \leq N$;

[0040] 步骤S36:根据查询数据编码 $H(q)$ 与待检索图像库中数据编码 $H(y_i)$ 之间的汉明距离 $D_h(H(q), H(y_i))$,返回待检索图像库中与查询数据汉明距离最小的 L 个数据作为检索结果,其中 $1 \leq i \leq N$ 。

[0041] 优选地,对于每一维,根据该维分配的位数计算该维度的阈值的步骤S16,进一步包括步骤:

[0042] 步骤S161:对于第 i 维,其编码长度是 c_i ,利用 k 均值聚类算法将 X 的第 i 列的 n 个数据聚成 (c_i+1) 个簇;

[0043] 步骤S162:将聚类中心从小到大排序,得到第 i 维的编码阈值 t_i^j ,其中 $1 \leq j \leq (c_i+1)$ 。

[0044] 优选地,利用训练得到的每一维的编码长度 c_i 和阈值 t_i^j 分别对图像库特征数据的每一维进行编码的步骤S23,进一步包括步骤:

[0045] 步骤S231:对于第 i 维,其编码长度是 c_i ,编码阈值为 t_i^j ,将待检索图像数据库GIST特征 y 的第 i 维按照最近邻规则分配到 t_i^j 形成的簇中,其中 $1 \leq j \leq (c_i+1)$;

[0046] 步骤S232:若 y 的第 i 维的最近邻簇为 t_i^m 所形成的簇,则 y 的第 i 维的子编码 s_i 由连续的 (c_i+1-m) 个0和紧接着连续的 $(m-1)$ 个1组成,即

$$[0047] \quad s_i = (\underbrace{0, \dots, 0}_{c_i+1-m}, \underbrace{1, \dots, 1}_{m-1})。$$

[0048] 优选地,利用训练得到的每一维的编码长度 c_i 和阈值 t_i^j 分别对查询数据 q 的每一维进行编码的步骤S33,进一步包括步骤:

[0049] 步骤S331:对于第 i 维,其编码长度是 c_i ,编码阈值为 t_i^j ,将查询图像GIST特征 q 的第 i 维按照最近邻规则分配到 t_i^j 形成的簇中,其中 $1 \leq j \leq (c_i+1)$;

[0050] 步骤S332:若 q 的第 i 维的最近邻簇为 t_i^m 所形成的簇,则 q 的第 i 维的子编码 s_i 由连续的 (c_i+1-m) 个0和紧接着连续的 $(m-1)$ 个1组成,即

$$[0051] \quad s_i = (\underbrace{0, \dots, 0}_{c_i+1-m}, \underbrace{1, \dots, 1}_{m-1})。$$

[0052] 优选地,维度 $d=512$ 。

[0053] 由此可见,本发明解决了数据投影后各个维度信息不均衡利用同样位数编码每一维不合理的问题,提出了一种基于自适应位分配哈希算法的大规模图像检索方法。该发明中,原始数据被映射到汉明空间,同时在原始欧氏空间中重要的维度在汉明空间中依然是重要的,较好地保持了原始数据的近邻结构。图像特征向量经过本发明的方法编码,能够减少存储空间,同时也能够极大地提高检索效率。

附图说明

[0054] 图1是本发明的基于自适应位分配哈希算法的大规模图像检索方法的流程示意图;

[0055] 图2是本发明用于建立训练集的部分样本图像;

[0056] 图3是4幅查询图像在 $k=512$ 位编码时的检索结果,其中最左边的是查询图像,图3中只显示了汉明距离最小的10幅图像;

[0057] 图4是本发明与其他哈希方法在不同编码长度对应的检索效果平均准确率(average mean precision, MAP)的曲线图,其中取 $L=1000$ 作为检索结果计算MAP;

[0058] 图5是本发明与其他哈希方法在不同编码长度对应的检索召回率(Recall)曲线图,其中取 $L=1000$ 作为检索结果计算召回率。

具体实施方式

[0059] 为使本发明的目的、技术方案和优点更加清楚明白,以下结合技术方案和附图详细叙述本发明的具体实施方式。以FLICKR1M [Mark J. Huiskes, Michael S. Lew. The MIR Flickr retrieval evaluation. In Proceedings of ACM International Conference on Multimedia Information Retrieval, 2008] 数据集为例进行说明。FLICKR1M包含100万张图片,都是从Flickr网站下载的,图片的内容千差万别,尺寸也是多种多样。

[0060] 图1的左边部分表示本发明实施例的训练过程的主要流程图,如图所示,

[0061] 步骤S11:将FLICKR1M数据集分为两部分:待检索图像库D(99万5千张)和查询图像集Q(5千张)。并从待检索图像库中随机选取10万张图片作为训练集T。部分训练图像样本如图2所示。

[0062] 步骤S12:对训练集T中的图像提取512维的GIST特征,将训练集中所有图像的GIST特征组成矩阵 X , $X \in R^{n \times d}$, X 的第 i ($1 \leq i \leq 100000$)行由 x_i 构成,其中 x_i 表示训练集中第 i 幅图像的GIST特征, $x_i \in R^d$, $n=100000$, $d=512$ 。

[0063] 步骤S13:对训练集图像的GIST做PCA旋转,以实现投影预处理,保存均值 μ 和投影矩阵 P 。均值的计算公式如下:

$$[0064] \quad \mu = \frac{1}{n} \sum_{i=1}^n x_i$$

[0065] 将每一个 x_i 减去均值 μ ,得到 $\bar{x}_i = x_i - \mu$,此时由 \bar{x}_i 组成的特征矩阵记为 \bar{X} 。计算协方差矩阵 $C = \bar{X}^T \bar{X}$,其中 \bar{X}^T 为 \bar{X} 的转置。对矩阵 C 进行特征分解得到 $C = P \Lambda P^T$,即得到投影矩阵 P 。

[0066] 步骤S14:计算投影后的 X 的每一维的离散度,本发明中利用方差信息来度量离散度。 X 第 i ($1 \leq i \leq d$)维的离散度计算公式为:

$$[0067] \quad \lambda_i = \frac{1}{n} \sum_{j=1}^n (X_j^i - \frac{1}{n} \sum_{m=1}^n X_m^i)^2$$

[0068] 其中, X_j^i 为 X 中第 j 行第 i 列(维)的数据。

[0069] 步骤S15:由步骤S14得到的每一维的离散度 λ_i ,利用下式计算每一维的编码长度。

第 i ($1 \leq i \leq d$) 维的编码长度 c_i 计算公式如下:

$$[0070] \quad c_i = \begin{cases} \left[k \cdot \frac{\lambda_i}{\sum_{n=i}^d \lambda_n} + 0.5 \right] & i=1 \\ \left[\left(k - \sum_{i=1}^{i-1} c_i \right) \cdot \frac{\lambda_i}{\sum_{n=i}^d \lambda_n} + 0.5 \right] & i \geq 2 \end{cases}$$

[0071] 其中, k 是总的编码长度。本实施例中 k 取值512。

[0072] 步骤S16:利用训练得到的每一维的编码长度,计算每一维的阈值。对于第 i 维,其编码长度是 c_i ,利用K均值聚类算法将 X 的第 i 列的 n 个数据聚成 (c_i+1) 个簇。之后,将聚类中心从小到大排序(由于每一维都是1维数据,因此聚类中心也是1维数据,故可以排序),得到第 i 维的编码阈值 t_i^j ($1 \leq j \leq (c_i+1)$)。

[0073] 图1的中间部分S2表示本发明实施例的待检索图像库编码过程的主要流程图,如图所示,

[0074] 步骤S21:对 D 中的每一幅图像,提取512维的GIST特征。假设第 r 幅图像库中的图像的GIST特征为 $y_r \in R^d$,其中 $d=512$, $1 \leq r \leq 995000$ 。

[0075] 步骤S22:利用训练得到的PCA投影参数均值 μ 和投影矩阵 P 对 y_r 进行PCA预处理,具体操作为:先将 y_r 减去均值 μ ,然后利用投影矩阵进行旋转,即 $(y_r - \mu)P$ 。

[0076] 步骤S23:由训练得到的每一维的编码长度 c_i 和每一维的阈值 t_i^j ($1 \leq j \leq (c_i+1)$),分别对 y_r 的每一维进行编码。对于 y_r 的第 i 维的数据 y_r^i ,找到离 y_r^i 最近的阈值,假设最近的阈值为 t_i^m ,则 y_r 的第 i 维的子编码 s_i 由连续的 (c_i+1-m) 个0和紧接着连续的 $(m-1)$ 个1组成。即,

$$[0077] \quad s_i = (\underbrace{0, \dots, 0}_{c_i+1-m}, \underbrace{1, \dots, 1}_{m-1})$$

[0078] 步骤S24:由每一维度的子编码 s_i ($1 \leq i \leq d$) 拼接得到最后完整的对图像库特征数据 y_r 的自适应位分配哈希编码 $H(y_r) = (s_1, s_2, \dots, s_d)$ 。

[0079] 图1的右边部分S3表示本发明实施例的查询过程的主要流程图,如图所示,

[0080] 步骤S31:对查询图像提取512维的GIST特征 $q \in R^d$,其中 $d=512$ 。

[0081] 步骤S32:利用训练得到的PCA投影参数均值 μ 和投影矩阵 P 对 q 进行PCA预处理,具体操作为:先将 q 减去均值 μ ,然后利用投影矩阵进行旋转,即 $(q - \mu)P$ 。

[0082] 步骤S33:由训练得到的每一维的编码长度 c_i 和每一维的阈值 t_i^j ($1 \leq j \leq (c_i+1)$),分别对 q 的每一维进行编码。对于 q 的第 i 维的数据 q^i ,找到离 q^i 最近的阈值,假设最近的阈值为 t_i^m ,则 q 的第 i 维的子编码 s_i 由连续的 (c_i+1-m) 个0和紧接着连续的 $(m-1)$ 个1组成。即,

$$[0083] \quad s_i = (\underbrace{0, \dots, 0}_{c_i+1-m}, \underbrace{1, \dots, 1}_{m-1})$$

[0084] 步骤S34:由每一维度的子编码 s_i ($1 \leq i \leq d$) 得到最后对完整查询特征数据 q 的编码。得到最终的对特征数据 q 的自适应位分配哈希编码 $H(q) = (s_1, s_2, \dots, s_d)$ 。

[0085] 步骤S35:计算查询数据编码 $H(q)$ 与待检索图像库中每一幅图像的编码 $H(y_i)$ ($1 \leq i \leq 995000$) 之间的汉明距离 $D_h(H(q), H(y_i))$ 。

$$[0086] \quad D_h(H(q), H(y_i)) = \sum_{j=1}^k H(q)[j] \oplus H(y_i)[j]$$

[0087] $H(q)[j]$ 和 $H(y_i)[j]$ 分别表示 $H(q)$ 和 $H(y_i)$ 的第 j 位, \oplus 表示异或操作, k 为总的编码长度。

[0088] 步骤S36: 根据查询数据编码 $H(q)$ 与待检索图像库中数据编码 $H(y_i)$ ($1 \leq i \leq 995000$) 之间的汉明距离 $D_h(H(q), H(y_i))$, 返回待检索图像库中与查询数据汉明距离最小的 $L=1000$ 个数据作为检索结果。限于篇幅, 图3中只给出了4幅查询图像, 并显示了排序前十的检索结果。此外, 为了与其他哈希方法对比, 我们还利用以上步骤重复实验了编码长度 $k=16, 32, 64, 128, 256$ 时的检索结果, 利用指标MAP和Recall衡量。对比结果见图4、图5, 图中显示的结果是对查询图像集 Q 中的所有查询图像的结果的平均。

[0089] 以上所述的具体实施例, 对本发明的目的、技术方案和有益效果进行了进一步详细说明, 应理解的是, 以上所述仅为本发明的具体实施例而已, 并不用于限制本发明, 凡在本发明的精神和原则之内, 所做的任何修改、等同替换、改进等, 均应包含在本发明的保护范围之内。

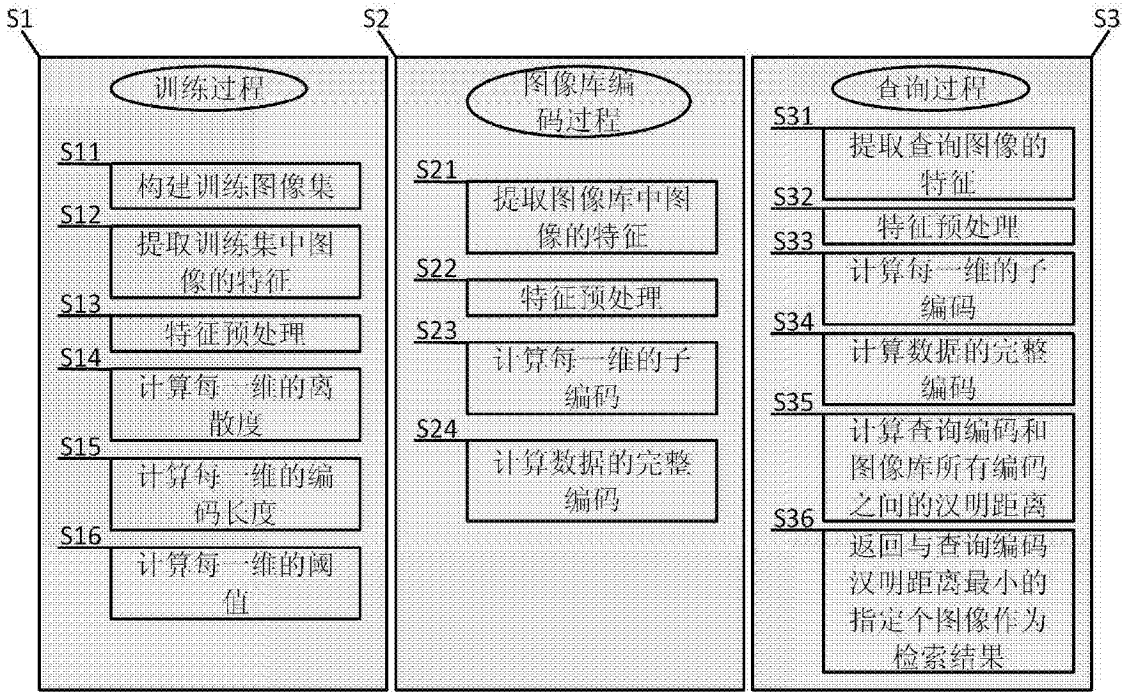


图1



图2

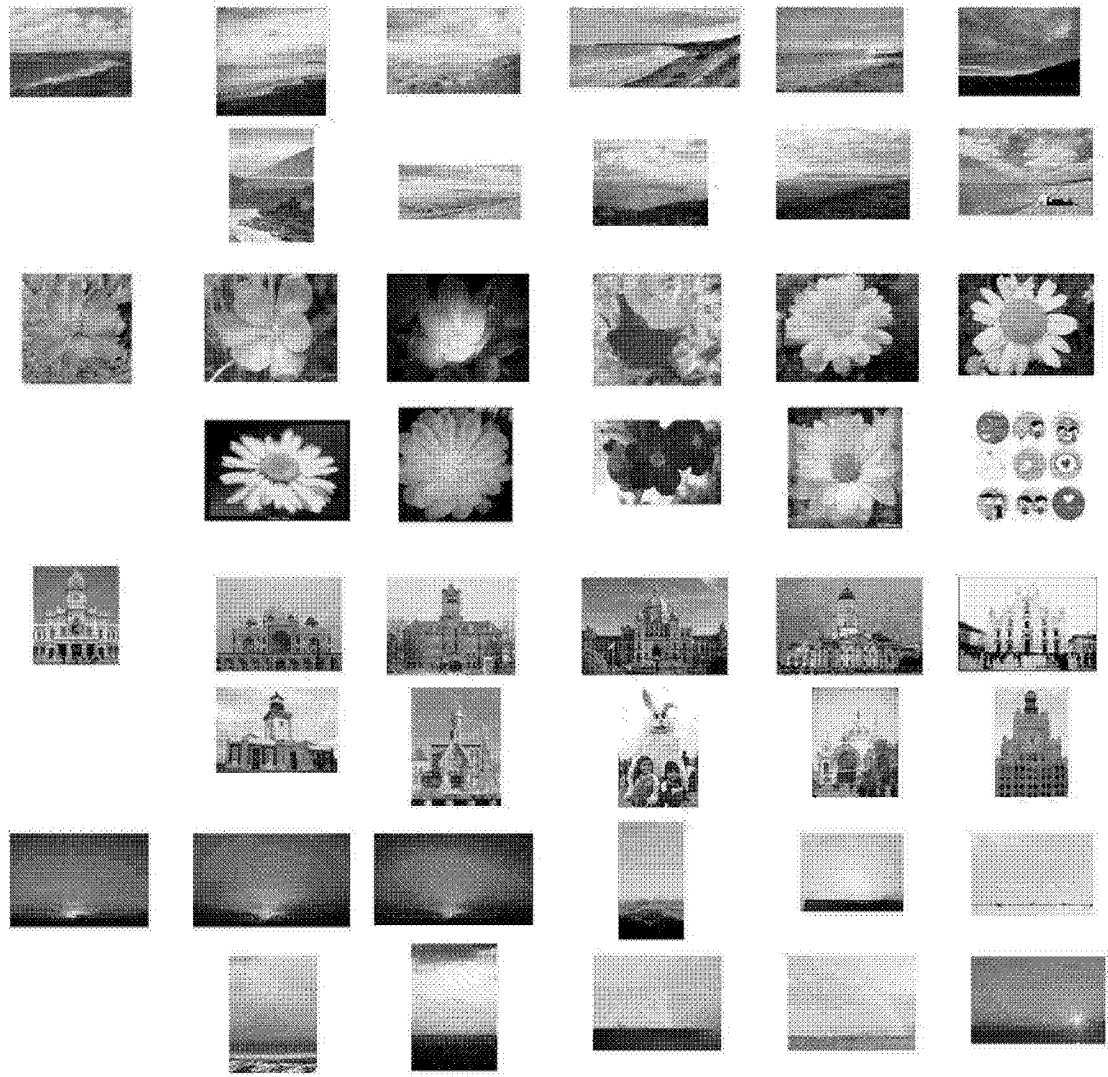


图3

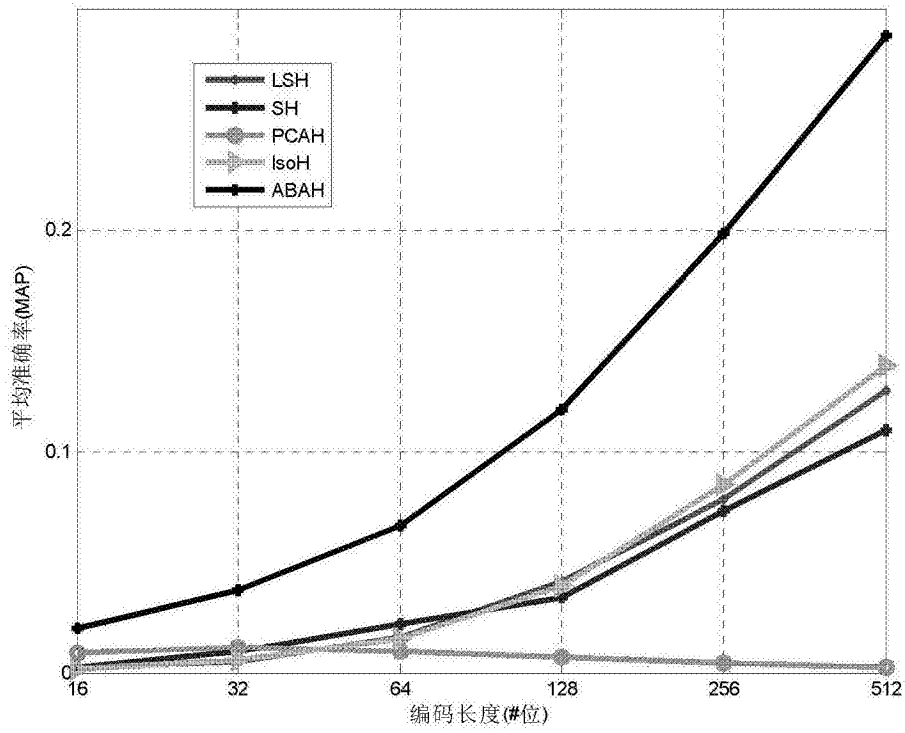


图4

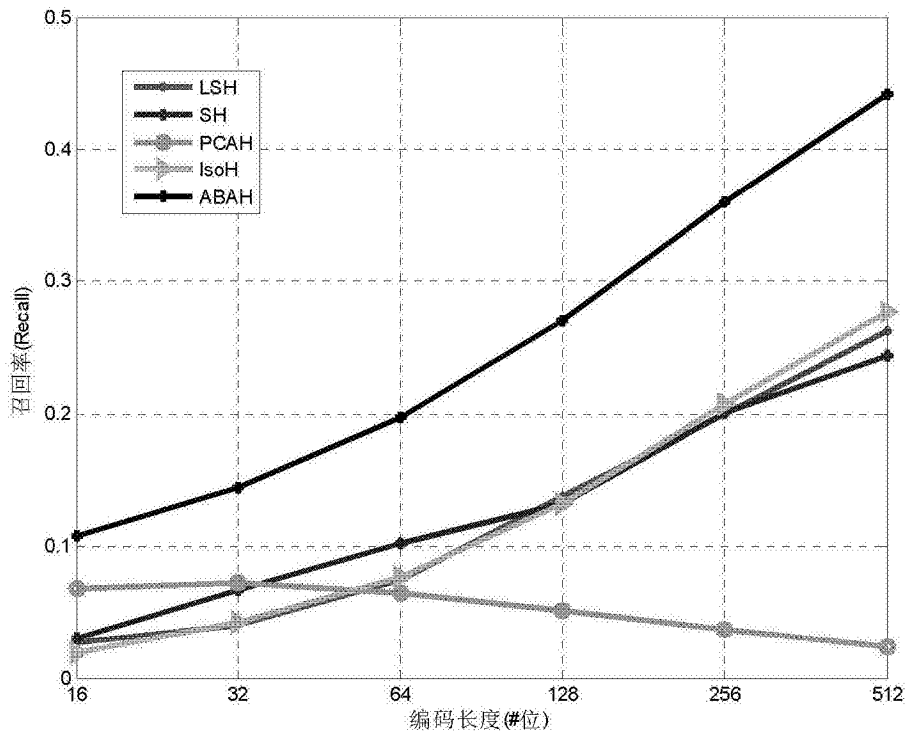


图5