GRAPH-DENSITY-BASED VISUAL WORD VOCABULARY FOR IMAGE RETRIEVAL

Lingyang Chu[1], *Shuhui Wang*[1], *Yanyan Zhang*[2], *Shuqiang Jiang*[1], *Qingming Huang*[1,2]

[1]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
{lingyang.chu, yanyan.zhang, shuqiang.jiang}@vipl.ict.ac.cn, wangshuhui@ict.ac.cn, qmhuang@jdl.ac.cn

ABSTRACT

Descriptive visual word vocabulary serves as the foundation of large scale image retrieval systems. However, the visual word descriptive power is limited by the construction mechanisms based on either cluster center or partitioned feature space, since such mechanisms may merge the sparsely distributed features and split the densely distributed features. Besides, there are a large number of outlier features that are not similar with any visual word. Quantizing such features into visual words inevitably decreases the visual word descriptive power. In this paper, we propose a novel Graph-Density-based visual word Vocabulary (GDV), which constructs the visual word by dense feature subgraph and directly measures the *intra-word similarity* by the corresponding graph density. Our method remarkably enhances the visual word descriptive power from the following three aspects: 1) GDV guarantees the high *intra-word similarity* by constructing visual words under the criterion of large graph density; 2) GDV improves the *inter-word dissimilarity* by alleviating the unexpected effect of subgraph splitting; 3) GDV suppresses the influence of outlier features by *selectively* quantizing only the features that are similar enough with the visual words. Extensive experiments demonstrate GDV's advanced descriptive power over traditional visual word vocabularies in enhancing both the retrieval accuracy and efficiency, which provides a higher level starting point for most image retrieval systems.

Index Terms— image retrieval, vocabulary, clustering

1. INTRODUCTION

Large scale image retrieval have been deeply investigated for years. Most image retrieval approaches represent images by a vocabulary of visual words obtained by clustering a set of SIFT features [1]. The image retrieval performance is directly affected by the visual word descriptive power, which mainly consists of the following three factors: 1) The *intra-word similarity* expects all SIFTs of the same visual word to be similar with each other. 2) The *inter-word dissimilarity* requires the SIFTs of different visual words to be as dissimilar as possible. 3) The *selectivity* requires the outlier SIFTs, which are not similar with any visual word, to be filtered out.

The *intra-word similarity* and *inter-word dissimilarity* are the primary objectives of most visual word vocabulary construction approaches [2, 3], such factors determine the basic descriptive power of the visual word. The *selectivity* focuses on filtering out the outlier SIFTs, which are mostly produced by the non-duplicate cluttered image background and will reduce the visual word descriptive power if they are quantized into visual words [4]. In all, these three factors directly affects the visual word descriptive power, which is the foundation to build reliable image retrieval systems. Great efforts [4, 5, 6] have been made in improving the *intra-word similarity* and *inter-word dissimilarity*, which, however, remains to be a challenging task due to the high dimensions of SIFT and the large amount of outlier SIFTs. The *selectivity* is generally overlooked by most of the current solutions [2, 7, 8, 9], which quantize all the SIFTs (including the outliers) into visual words. Besides, high scalability is a standard requirement for most visual word vocabulary construction approaches, since they generally deal with millions of SIFTs.

In this paper, we propose a novel Graph-Density-based visual word Vocabulary (GDV) to remarkably improve the *intra-word similarity*, the *inter-word dissimilarity* and the *selectivity*. With a SIFT affinity graph regarding each SIFT as a vertex, GDV defines each visual word as a dense SIFT subgraph and naturally measures the *intra-word similarity* by the corresponding graph density [10]. Each new SIFT is quantized into a visual word only if it is strongly connected to most of the graph vertexes in the dense SIFT subgraph. To efficiently construct GDV from millions of SIFTs, we propose a Scalable Maximization Estimation (SME) iteration to find the dense SIFT subgraphs in linear time complexity.

GDV outperforms the traditional visual word vocabulary construction approaches [2, 4, 7, 9] in the following aspects: 1) GDV effectively improves the *intra-word similarity*, since only the dense SIFT subgraphs with local maximum graph density are regarded as visual words. 2) The high *inter-word dissimilarity* is also guaranteed, since GDV regards the entire dense SIFT subgraph as a single visual word and rarely splits

This work was supported in part by National Basic Research Program of China (973 Program): 2012CB316400, in part by National Natural Science Foundation of China: 61025011, 61332016, 61322212 and 61303160, and in part by 863 program of China: 2014AA015202.

it into multiple visual words. 3) GDV significantly improves the *selectivity* by constructing the visual word vocabulary in a selective manner, which only selects the dense SIFT subgraphs with large graph densities as visual words and filter out all outlier SIFTs that are not similar with any visual word.

2. RELATED WORK

Traditional clustering methods for visual word vocabulary construction mainly consists of two categories: the center-based methods [2, 3, 11, 12] and the partition-based methods [9, 13, 14, 15, 16]. One typical center-based method is *k*-means [3], which initializes *k* cluster centers and shift them to the stable feature modes. The hierarchical *k*-means [2] embeds *k*-means into a tree structure to achieve good scalability. Other methods [11, 12] belonging to the *k*-means family are also based on cluster center. Despite the good scalability of the center-based methods, their visual word descriptive power is intrinsically limited by the clustering criterion that every SIFT must be assigned to the nearest visual word. According to [4], such clustering criterion is ambiguous, thus cannot guarantee the high *intra-word similarity* and high *inter-word dissimilarity*. As claimed by R. Ji *et al.* [6], the center-based methods tend to over-split the densely distributed SIFTs, which reduces both the *intra-word similarity* and the *inter-word dissimilarity*. Moreover, the enforcement of quantizing all SIFTs into visual words ignores the *selectivity*, which further impairs the visual word descriptive power.

Typical partition-based methods utilize random hyper planes to partition the whole feature space into subspaces and regard each subspace as a visual word. The ERC-Forest [14] applies the random forest [13] to partially guide the generation of visual vocabulary. The randomized locality sensitive vocabulary (RLSV) [9] generates visual word vocabulary by partitioning the feature space with locality sensitive hashing [15]. Despite the fast vocabulary construction speed, the partition-based methods also have their limitations in improving the visual word descriptive power: First, the randomness of the hyper planes makes it difficult to achieve high *intra-word similarity* and high *inter-word dissimilarity*. Second, regarding all the partitioned subspaces as visual words totally ignores the *selectivity*, which impairs the visual word descriptive power in a similar way as the partition-based methods.

3. PRELIMINARY

The SIFT affinity graph is defined as $G = (V, W, A)$, where (V, W, A) are specifically illustrated as follows:

- $V = \{v_1, v_2, ..., v_n\}$ is the vertex set, where each vertex v_i represents a 128-Dim SIFT and n is the total number of vertexes.
- $W = \{w_{ij} \mid w_{ij} = e^{-\alpha\|v_i - v_j\|}\}$ is the set of edge weights, where $\|\cdot\|$ is the L_2-$norm$ and w_{ij} measures the similarity between two SIFTs. $\alpha \geq 0$ is the scale factor that controls the connection strongness between vertexes.
- $A = \{a_{ij} \mid a_{ij} = w_{ij}\}$ is the n-by-n affinity matrix, where the diagonal matrix elements are set to zero to avoid self-loop in the SIFT affinity graph.

A SIFT subgraph is uniquely represented by a *probabilistic coordinate* $x \in \triangle^n$, where $\triangle^n = \{x \mid x \in R^n, x_i \geq 0, \sum_i x_i = 1\}$ is the standard simplex. The i-th bin of x (i.e., x_i) is the probability that the corresponding SIFT subgraph contains vertex v_i. Any vertex v_i with $x_i = 0$ is not contained by the SIFT subgraph. For convenience, we refer to the probabilistic subgraph as x and denote the vertex v_i belonging to subgraph x as $v_i \in x$.

The graph density of x is defined as $g(x) = x^T Ax = \sum_i x_i (Ax)_i$. Particularly, $a(v_i, x) = (Ax)_i$ is the affinity value between vertex v_i and x, which measures the average connection strongness between v_i and all the vertexes in x. Therefore, $g(x) = x^T Ax = \sum_i x_i (Ax)_i$ is the average connection strongness of all the connections between the vertexes in x, which is also a natural and robust measurement of the *intra-word similarity*.

Each visual word of GDV is defined as a dense SIFT subgraph with local maximum graph density (i.e., *intra-word similarity*), which uniquely corresponds to a local maximum point x^* of the following standard quadratic optimization problem (StQP):

$$\begin{cases} Maximize & g(x) = x^T Ax \\ s.t. & x \in \triangle^n \end{cases} \tag{1}$$

4. THE GRAPH-DENSITY-BASED VISUAL WORD VOCABULARY

In this section, we introduce a novel approach to efficiently detect the dense SIFT subgraphs by finding the local maximum points of the StQP problem. We first illustrate the proposed Maximization Estimation (ME) iteration to detect the dense SIFT subgraphs. Then we introduce the Scalable ME (SME) iteration, which embeds the ME iteration into a carefully designed *K*-ary tree to process millions of SIFT in linear time complexity. In the end, a visual word selection approach is proposed to build GDV by selecting dense SIFT subgraphs with large graph density as visual words, where new SIFTs are selectively quantized using the *K*-ary tree structure.

4.1. The Maximization Estimation Iteration

The ME iteration finds the dense subgraph of G by two steps: The *maximization step* finds the local dense subgraph x^* with maximal graph density $g(x^*)$ on a small subgraph $S \subset G$. The *estimation step* updates the subgraph S with newly discovered candidate vertexes on G to further increase $g(x^*)$. This iteration process stops when the graph density $g(x^*)$ can no longer be increased by any graph vertex in G, thus a dense subgraph of G is detected.

Maximization step. Given a subgraph $S \subset G$ with m vertexes (i.e., SIFTs), the probabilistic coordinate is initialized as $x(0) = \{x_i(0) = \frac{1}{m} \mid \forall v_i \in S, x(0) \in \triangle^n\}$ and the local maximum point of the corresponding StQP problem on S can be solved by the replicator dynamics [17]:

$$x_i(t+1) = x_i(t)\frac{(Ax(t))_i}{x(t)^T Ax(t)}, i = 1, ..., n \tag{2}$$

The computational overhead of the *maximization step* is small, since we only compute a m-by-m affinity sub-matrix A_S for S, where m is limited by a fixed upper bound $M \ll n$.

Estimation step. As proved by H. Liu *et al.* [10], if $\exists v_i \in G, a(v_i, x^*) > g(x^*)$, then the dense subgraph x^* of S is not a dense subgraph of G and the graph density $g(x^*)$ can be further increased by adding v_i into subgraph x^*. Therefore, we propose the *estimation step* to find the candidate vertexes v_i that can further increase the graph density $g(x^*)$. First, a hypersphere $H(c, r)$ is estimated from x^* to include all the vertexes satisfying $a(v_i, x^*) > g(x^*)$. Then, the vertexes inside the hypersphere are regrouped as a new subgraph S for the next round of *maximization step* to further increase the graph density. The center c and radius r of the hypersphere $H(c, r)$ are defined as follows:

$$c = \sum_i v_i x_i^* \tag{3}$$

$$r = \frac{1}{\alpha}\ln\frac{U}{g(x^*)},\ U = \sum_j x_j^* e^{\alpha\|v_j - c\|} \tag{4}$$

where c is the sum of all v_i weighted by x_i^* and α is the same scale factor as in $w_{ij} = e^{-\alpha\|v_i - v_j\|}$.

Theorem 1. *$\forall v_i \in V$, if $\|v_i - c\| > r$, then $a(v_i, x^*) < g(x^*)$.*

Proof. Define $f(v_i) = Ue^{-\alpha\|v_i - c\|}$ and substitute U (in Eqn.4) into $f(v_i)$, we get:

$$f(v_i) = \sum_j x_j^* e^{\alpha[\|v_j - c\| - \|v_i - c\|]} \tag{5}$$

By applying the *triangle inequality*:

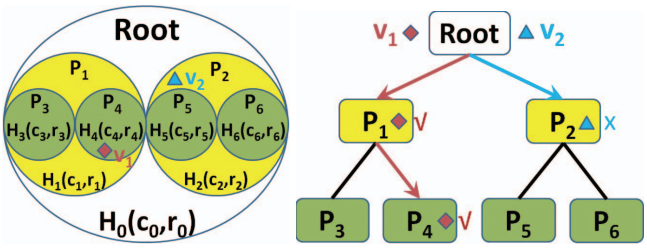
$$\|v_j - c\| - \|v_i - c\| \geq -\|v_i - v_j\| \tag{6}$$

we obtain the following inequality:

$$f(v_i) \geq a(v_i, x^*) = \sum_j x_j^* e^{-\alpha\|v_i - v_j\|} \tag{7}$$

Set $\|v_i - c\| = r$ and substitute Eqn.4 into $f(v_i)$ (Eqn.5), we have $f(v_i) = Ue^{-\alpha r} = g(x^*)$. Considering the monotonicity of $f(v_i)$ and Eqn.7, we can prove: if $\|v_i - c\| > r$, then $g(x^*) > f(v_i) \geq a(v_i, x^*)$. Thus, *Theorem 1* is proved. □

Theorem 1 guarantees that any vertex v_i outside the hypersphere $H(c, r)$ satisfies $a(v_i, x^*) < g(x^*)$. This means only the vertexes inside the hypersphere may further increase the graph density. Hence, such vertexes are regrouped as a new subgraph $S \subset G$ for the next round of *maximization step*. This iteration repeats until no v_i satisfying $a(v_i, x^*) > g(x^*)$ exists, where the last found x^* is a dense subgraph of G according to H. Liu *et al.* [10].



(a) hyperspheres (b) K-ary tree

Fig. 1. The K-ary tree with children limit $K = 2$ and depth limit $d = 3$. (a) shows a naive hypersphere structure of the K-ary tree, where different hyperspheres may overlap in real cases. (b) shows that the SIFT v_1 is pushed down the tree to the leaf node p_4; while the SIFT v_2 is discarded at node p_2, since no child of p_2 contains v_2.

4.2. The Scalable Maximization Estimation Iteration

The scalability of the ME iteration is limited by the exact search for all candidate vertexes within the hypersphere. To tackle this problem, we further propose the Scalable ME (SME) iteration, which embeds the ME iteration into a carefully designed K-ary tree to efficiently detect most dense subgraphs of G on millions of SIFTs.

Fig.1 shows the K-ary tree, where the i-th tree node p_i in Fig.1 (b) corresponds to a hypersphere $H_i(c_i, r_i)$ in Fig.1 (a). For a node p_i with at most K children nodes $\{p_j\}_K$, the corresponding hypersphere $H_i(c_i, r_i)$ is defined as the smallest hypersphere that covers all the children hyperspheres:

$$\begin{cases} c_i = \frac{1}{K}\sum_{j=1}^{K} c_j \\ r_i = \min_j (r_j + \|c_j - c_i\|) \end{cases} \tag{8}$$

where the centers $\{c_j\}_K$ and radiuses $\{r_j\}_K$ correspond to the hyperspheres of the K children nodes $\{p_j\}_K$.

The K-ary tree is initialized by the following steps: 1) Assigned all SIFTs to the root node. 2) For each node p_i, grow its K children by randomly choosing K SIFTs that p_i contains as the corresponding hypersphere centers. 3) assign each SIFT contained by p_i to the nearest children of p_i. This iteration stops when a node contains less than K SIFTs or the tree depth limit d is reached. The radius r of all the nodes are initially set to $r = \infty$. Once the K-ary tree is initialized, we embed the ME iteration as follows:

Embedded maximization. We apply the *maximization step* of the ME iteration to every leaf node of the K-ary tree. For each leaf node p_i, the SIFTs belonging to it are grouped as a subgraph S_i and the replicator dynamics (Eqn.2) is applied to find the dense subgraph x_i^* of S_i. Recall that x_i^* is not guaranteed to be the dense subgraph of G, we propose the *embedded estimation* to further increase the graph density.

Embedded estimation. This step first estimates the hypersphere $H_i(c_i, r_i)$ of each leaf node p_i by Eqn.3 and Eqn.4, then update the hyperspheres of all the tree nodes in a bottom-up manner by Eqn.8. Thereafter, all SIFTs are reassigned

through (top-down) the *K*-ary tree to the leaf nodes, so that the next round of *embedded maximization* can be started. When the hyperspheres of multiple tree nodes overlap with each other, we first push the SIFT down through every tree node (i.e., hypersphere) that contains it until the SIFT reaches multiple leaf nodes; then we assign the SIFT to the single leaf node with the largest graph density. The SME iteration continues until all leaf nodes converges to a dense subgraph of G.

4.3. Visual Word Selection And New SIFT Quantization

Since the detected dense SIFT subgraphs with small graph densities possess low descriptive power due to the low *intra-word similarity*, we filter them out by a density threshold ρ:

$$\rho = e^{-\alpha\epsilon} \tag{9}$$

where ϵ is the average SIFT distance limit between all the SIFTs in one leaf node. The remaining dense SIFT subgraphs form the final visual words of GDV. We use the *K*-ary tree to quickly quantize new SIFT by finding the leaf node that contains it, where the SIFTs that are not contained by any leaf node are discarded (see v_2 in Fig.1).

4.4. Complexity Analysis

The time complexity of the SME iteration is evaluated by the SIFT distance computation ($\|v_i - v_j\|$), which is the most time consuming part of the entire iteration. Recall that d is the depth limit of the *K*-ary tree and M is the SIFT number limit of every leaf node, the time complexity of the embedded maximization and embedded estimation are $\mathcal{O}(Mn)$ and $\mathcal{O}((Kd+1)n)$, respectively. This results in a linear overall time complexity of $\mathcal{O}((Kd+M+1)n)$, which enables the SME iteration to efficiently process massive SIFTs. As demonstrated by the experimental results in Table 1, the Vocabulary Construction Time (VCT) of GDV grows linearly w.r.t the increasing amount of SIFTs.

Table 1. The VCT of GDV. ($M = 400$, $K = 30$, $d = 4$)

Training SIFT size (n)	20k	30k	40k	50k
VCT (hour)	3.61	6.45	9.25	12.1

5. EXPERIMENTS

In this section, we compare the descriptive power of GDV with four different traditional visual word vocabularies [2, 4, 7, 9] on two published ground truth datasets: the IPDID dataset [18] and the TinEyeDemo (TED) dataset [19]. Each of the ground truth datasets is split by half into a training part and evaluation part. To simulate the situations of practical image search engine, we mix the training part with 50 thousand random Web images to generate the training dataset for visual word vocabulary construction and mix the evaluation part with 1 million random Web images to generate the evaluation dataset for performance evaluation. The Web images are randomly crawled from different image websites and all images in both the training and evaluation datasets are uniformly resized within a bounding box of 400×400 pixels. The descriptive power of visual word vocabularies are evaluated by the mean Average Precision (mAP) and the Average Retrieval Time (ART) of the image retrieval system built on them. All the experiments are performed on a PC with Core2 Quad CPU (2.67GHz) and 8 GB physical memory.

5.1. Parameter Analysis

The influences of parameters on the descriptive power of GDV are analyzed on the IPDID evaluation dataset with the standard bag-of-word image retrieval system [20].

Influence of the scale factor α. The scale factor α controls the graph structure of G and affects the graph density distribution. A larger α reduces the connection range of every vertex, which increases the number of dense SIFT subgraphs (i.e., visual words) and reduces the number of SIFTs in each visual word. As shown in Table 2, the vocabulary size monotonously grows with the increase of α, where retrieval system reaches the optimal mAP at $\alpha = 0.0025$. Such mAP performance is rational, since the descriptive power of GDV first increases with the increasing amount of visual words, then decreases when the average number of SIFTs in each visual word is over reduced. GDV controls the connection range of each SIFT by α and finds the visual words by the sole criterion of large graph density (i.e., *intra-word similarity*). This effectively alleviates the over-splitting and over-merging of the dense SIFT subgraphs, hence increases both the *intra-word similarity* and the *inter-word dissimilarity*.

Table 2. The impact of the scale factor α.

α	$\alpha = 0.002$	$\alpha = 0.0025$	$\alpha = 0.003$
Vocabulary size	3275	6347	8927
mAP	0.474	0.488	0.462

Influence of the density threshold. The density threshold ρ (Eqn.9) is the lower bound for the *intra-word similarity* of each visual word in GDV. A large ρ guarantees the high *intra-word similarity* of the selected visual words and reduces the vocabulary size of GDV. A small ρ increases the vocabulary size, but may select some less descriptive visual words with low *intra-word similarity*. As shown in Table 3, the vocabulary size of GDV monotonously grows with the decrease of ρ, where the retrieval system reaches the optimal mAP at $\rho = 0.67$ ($\epsilon = 200$). GDV selects descriptive visual words with high *intra-word similarity* and only quantizes new SIFTs that are similar enough to the visual word. This effectively filters out the outlier SIFTs, hence improves the *selectivity*.

Table 3. The impact of the distance threshold ϵ. ($\alpha = 0.002$)

ρ	$\rho = 0.74$	$\rho = 0.67$	$\rho = 0.61$
ϵ	$\epsilon = 150$	$\epsilon = 200$	$\epsilon = 250$
Vocabulary size	2453	3253	3875
mAP	0.449	0.475	0.463

Table 4. Retrieval performance comparison on the IPDID evaluation dataset.

Data set/Size	IPDID/1 million				IPDID/300k		
Framework	BOW		SC		BOW		
Methods	GDV	HKM	GDV	HKM	GDV	RLSV	AMB
mAP	**0.4863**	0.4023	**0.2704**	0.2289	**0.5780**	0.2568	0.3197
ART (sec)	**0.379**	0.442	**0.411**	0.724	**0.213**	8.72	3.51

Table 5. Retrieval performance comparison on the TED evaluation dataset.

Data set/Size	TinEye/1 million					TinEye/300k		
Framework	BOW			SC		BOW		
Methods	GDV	HKM	HE	GDV	HKM	GDV	RLSV	AMB
mAP	**0.8678**	0.7235	0.58	**0.8792**	0.7471	**0.8839**	0.4801	0.6021
ART (sec)	**0.321**	0.482	0.574	**0.378**	0.581	**0.186**	8.45	3.42

Fig. 2. Sampled images of the IPDID dataset.

Fig. 3. Sampled images of the TED dataset.

5.2. Performance Evaluation

In this section, we compare the descriptive power of GDV with four traditional visual word vocabulary construction approaches: 1) Randomized locality sensitive vocabulary (RLSV) [9]; 2) Hamming embedding (HE) [7]; 3) Hierarchical k-means (HKM) [2]; 4) Ambiguity (AMB) [4]. All experiments are performed on the evaluation datasets of IPDID and TED. The descriptive power of each vocabulary is evaluated by the retrieval performances of two retrieval systems based on the bag-of-word framework (BOW) [20] and the spatial coding framework (SC) [19], respectively.

As shown in Fig.2, the IPDID dataset [18] contains 10 classes of images with rotation, illumination change, occlusion and zooming. Every compared method is optimally tuned. For HKM and AMB, we use the optimal vocabulary size of 1 million. For RLSV, the optimal settings of 12 hash functions and 20 hash tables are applied [9]. For GDV, we use $\alpha = 0.0025, \epsilon = 200$ with a K-ary tree of $K = 30, d = 4$, where the resulting vocabulary size is 8021. All vocabularies are constructed on the training dataset and evaluated on the corresponding evaluation dataset.

According to the mAP performances on the IPDID evaluation dataset (Table 4), the proposed GDV significantly outperforms HKM under both the BOW and SC retrieval frameworks. Since RLSV and AMB are designed for the BOW framework and can only index 300 thousand images on 8GB physical memory, we compare with them on the IPDID/300k evaluation dataset, which is generated by mixing the evaluation part of IPDID with 300 thousand Web images. Again, the mAP of GDV remarkably outperforms RLSV and AMB on IPDID/300k. Such outstanding mAP performances demonstrate the strong descriptive power of GDV.

We can see from Table 4 that the ART of GDV significantly outperforms all compared methods as well. Such fast retrieval speed of GDV is attributed to its significant *selectivity*, which effectively reduces the invert list size of the BOW framework by filtering out large amount of useless outlier SIFTs. In contrast, both RLSV and AMB are slow due to their large redundant invert lists, where AMB quantizes each SIFT into multiple visual words and RLSV uses 20 independent vocabularies to quantize every SIFT 20 times.

The TED dataset (Fig.3) was developed by Zhou *et al.* [19] with the Cool-Searches [21] of the commercial image search engine "TinEye". TED contains 23 classes of near duplicate images with illumination change, zooming and poor resolution. The experimental settings for all compared methods are the same as the IPDID dataset. Table 5 shows the evaluation results on TED, where GDV significantly outperforms the other compared methods in both mAP and ART performances.

To sum up, the experimental results on two evaluation datasets solidly demonstrate the descriptive power advantage of GDV in improving both the mAP and ART performances of different image retrieval systems. The mAP advantage of GDV is due to the high *intra-word similarity*, high *inter-word dissimilarity* and the significant *selectivity*, which remarkably improve the visual word descriptive power and suppress the influence of outlier SIFTs. The impressive ART performance is attributed to the advanced *selectivity*, which effectively reduces the size of invert list by filtering out the outlier SIFTs.

6. CONCLUSION

In this paper, we propose the novel Graph-Density-based visual word Vocabulary (GDV), which significantly improves visual word descriptive power by constructing dense SIFT subgraphs into visual words. Such advanced descriptive power of GDV provides a higher level starting point for most image retrieval systems, making large scale image retrieval more reliable and faster. In our future work, we will focus on robustly weighting the visual words of GDV to adapt different contextual environments of SIFTs, so that the descriptive power of GDV can be further improved.

7. REFERENCES

[1] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[2] David Nister and Henrik Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of the Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 2161–2168.

[3] Stuart Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.

[4] Jan C. van Gemert, Cor J. Veenman, Arnold W.M. Smeulders, and Jan-Mark Geusebroek, "Visual word ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1271–1283, 2010.

[5] Xiaoyu Wang, Ming Yang, Timothee Cour, Shenghuo Zhu, Kai Yu, and Tony X Han, "Contextual weighting for vocabulary tree based image retrieval," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 209–216.

[6] Rongrong Ji, Hongxun Yao, Xing Xie, and Qi Tian, "Vocabulary hierarchy optimization and transfer for scalable image search," *IEEE MultiMedia*, vol. 18, pp. 66–77, 2011.

[7] Herve Jegou, Matthijs Douze, and Cordelia Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proceedings of the European Conference on Computer Vision*, 2008, pp. 304–317.

[8] Frank Moosmann, Bill Triggs, and Frederic Jurie, "Fast discriminative visual codebooks using randomized clustering forests," *Advances in Neural Information Processing Systems*, vol. 19, no. 3, pp. 985, 2007.

[9] Yadong Mu, Ju Sun, Han Tony X., Loong-Fah Cheng, and Shuicheng Yan, "Randomized locality sensitive vocabularies for bag-of-features model," in *Proceedings of the 11th European Conference on Computer Vision*, 2010, pp. 748–761.

[10] Hairong Liu and Shuicheng Yan, "Robust graph mode seeking by graph shift," in *Procedings of The 27th International Conference on Machine Learning*, 2010.

[11] Shindler Michael, Wong Alex, and Adam Meyerson, "Fast and accurate k-means for large datasets," *Advances in Neural Information Processing Systems*, 2011.

[12] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu, "A local search approximation algorithm for k-means clustering," in *Proceedings of the 18-th annual Symposium on Computational Geometry*, 2002, pp. 10–18.

[13] Leo Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[14] Frank Moosmann, Eric Nowak, and Frederic Jurie, "Randomized clustering forests for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1632–1646, 2008.

[15] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the 20-th annual Symposium on Computational Geometry*, 2004, pp. 253–262.

[16] Alexandr Andoni and Piotr Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM*, vol. 51, pp. 117, 2008.

[17] Jörgen W Weibull, *Evolutionary game theory*, MIT press, 1997.

[18] Zhipeng Wu, Qianqian Xu, Shuqiang Jiang, Qingming Huang, Peng Cui, and Liang Li, "Adding affine invariant geometric constraint for partial-duplicate image retrieval," in *Proceedings of the International Conference on Pattern Recognition*, 2010, vol. 0, pp. 842–845.

[19] Wengang Zhou, Yijuan Lu, Houqiang Li, Yibing Song, and Qi Tian, "Spatial coding for large scale partial-duplicate web image search," in *Proceedings of the international conference on Multimedia*, 2010, pp. 511–520.

[20] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of International Conference on Computer Vision*, 2003, vol. 2, p. 1470.

[21] TinEye, "`http://www.tineye.com/cool_searches/`," .