Uninterrupted Whole Result Involving Queries over WSN

Arokia Subancy Vinitha.B, J.Prabula
arolesuban14@gmail.com
III MCA, Lord Jegannath College of Engineering & Technology
Assistant Professor, Department of Computer Applications,
Lord Jegannath College of Engineering & Technology

Abstract: Peer-to-peer networking offers a scalable solution for sharing multimedia data across the network. With a large amount of visual data spread among different nodes, it is an important but challenging issue to perform content-based retrieval in peer-to-peer networks. Compared with central environments, the key challenge is to efficiently obtain a global codebook, as images are distributed across the whole peer-to-peer network. In addition, a peer-to-peer network often evolves dynamically, which makes a static codebook less effective for retrieval tasks. Therefore, I propose a dynamic codebook updating method by optimizing the mutual information between the resultant codebook and relevance information, and the workload balance among nodes that manage different code words. The broad experimental results point to that the future approach is scalable in budding and distributed peer-to-peer networks, while achieving improved revival accuracy.

I. INTRODUCTION

Peer-To-Peer (P2P) networks, which are formed by equally privileged nodes connecting to each other in a self-organizing way, have been one of the most important architectures for data sharing. Popular P2P file-sharing networks such as eDonkey1 count millions of users [1] and tens of millions of files. Unlike WebPages which mainly consist of textual documents such as news, blog articles or forum posts, multimedia files play a dominant role in most P2P networks [2]. The ever-growing amount of multimedia data and computational power on P2P networks exposes both the need and potential for large scale multimedia recovery application such as content-based image contribution, and copyright contravention detection.

While P2P networks are well known for their efficiency, scalability and robustness on file sharing, providing extended search functionality such as content-based image retrieval (CBIR) faces the following challenges: 1) in dissimilarity to centralized environments, data in P2P networks is distributed among different nodes, thus a CBIR algorithm needs to index and search for images in a distributed manner; 2) unlike distributed servers/clouds, nodes in P2P networks have limited network bandwidth and computational power, thus the algorithm should keep the network cost low and the workload among nodes balanced; and 3) as P2P networks are under constant churn, where nodes join/leave and files publish to/remove from the network, the index needs to be updated dynamically to adapt to such changes.

To support content indexing and avoid message flooding, prearranged overlay networks such as Distributed Hash Tables (DHTs) [3], [4] are often implemented on top of a physical network. By organizing the nodes in a structured way, messages can be cost-effectively routed between any pair of nodes, and the index integrity can be maintained

during network churn. For the CBIR functionality, most of the existing systems adopt a global feature approach: an image is represented as a high-dimensional feature vector (e.g., color histogram), and the similarity between files is measured using the distance between two feature vectors [5], [6], [7]. Usually, the feature vectors are indexed by a distributed high-dimensional index or Locality Sensitive Hashing (LSH) over the DHT overlay.

On the other hand, the bag-of-visual-words (BoVW) model has been successfully utilized for large scale image retrieval [8]. In the BoVW model, each image is represented with a bag of local features, which mimics the bag-of words(BoW) model where each document is a collection of unordered words.2 Generally, to employ the BoVW model, the following three steps are required [9], [10]: Firstly, a number of local regions (through image segmentation or uniform image partitioning) or key points (through key point detection algorithms such as Hessian-Affine detector [11]) will be identified from an image and each region or key point will be represented with a high dimensional descriptor. In our experiments, the widely used Scale-Invariant Feature Transform (SIFT) descriptor [12] is employed.

Secondly, since the features extracted are in a continuous space, a codebook is generated to quantize the feature vectors into discrete codeword's, thus an image can be interpreted as a set of feature codeword's. One of the most commonly used quantization schemes is nearest-neighbor quantization (e.g., k-means [9], [10]), where each feature vector is represented by its nearest codeword centroid, and the codebook forms a Voronoi partitioning of the feature space.

Lastly, similar to the BoW model, statistical distributions of the code words in a given image is utilized to represent the image. In this paper we utilize the well-studied tfidf weighting scheme and cosine distance as the similarity measurement.

II. RELATED WORK

A). Skyline Query Processing

Skyline calculate was first introduced to the record field in [13] with *Block-Nested-Loops (BNL)*, *Divide-and - Conquer (D&C)* and B-tree algorithms. Later work [14] better *BNL* by pre-sorting. [15] Proposed progressive *Bitmap* and *Index* algorithms. Afterwards, there are studies employing R-tree index to compute skyline [16, 17] faster and using external algorithm to calculate the maximal vectors [20]. Some recent works investigate the semantics of skyline by subspace analysis [18, 19], processing of skyline over data streams [21] and in high dimensional paces [22]. Distributed skyline query was recently addressed in [23] for web information systems and in [24] against mobile devices. By visiting all nodes to retrieve skyline answers, the above approaches dedicated for small-scale distributed systems are not efficient to be applied on P2P networks. Recent work of skyline query processing on P2P [25] parallelizes search by enforcing a partial order on query propagation based on CAN.

B). Multi-dimensional Indexing on P2P

Multi-dimensional indexing for supporting efficient search on structured P2P networks has been a hot research topic. MAAN [26] uses locality preserving hashing to map data values onto Chord identifier pace. Mercury [27] attempts to support multi-attribute range queries by placing values of each attribute contiguously on a separate routing hub, while performing explicit load balancing for non-uniform data distribution. However, the divide attributes arrangement in MAAN and Mercury are not effective for dealing out skyline queries that absolutely denote the constraints in the middle of the attributes. The works closer to ours are Murk [30], Skip Index [28] and ZNet [29]. Murk indexes multi-dimensional data partition using the kd-tree. Like Murk, Skip Index stores partition in sequence in a binary tree, while ZNet splits data partition in a quad-tree manner. Though Skip Index and ZNet sense of balance data loads during partition, they do not deal with the query load balancing problem present in skyline query processing.

C). Global Feature Model

The global characteristic model represents each image with one high-dimensional feature vector, and measures the similarity between images with the distance between their feature vectors. This model is adopted by many existing P2P CBIR systems [5], [6], [31], [32], [33], [34], [35]. To store and retrieve the trait vectors powerfully, generally two categories of methods can be useful: distributed high-dimensional indexing and Locality-Sensitive Hashing (LSH). The high-dimensional indexing based approaches store the feature vectors in a data structure, usually a tree or retrieval. In structured P2P networks, the high-dimensional index is defined in a distributed way over the P2P overlay, where each node manage a part of the index. For example, in M-Chord [30], data structures based on distance to reference points are built to facilitate metric-based similarity search in Chord [3] networks. In [6], a novel naming mechanism and a tree summarization strategy are employed to build a tree structure over a DHT overlay. However, even in a centralized environment, the performance of high dimensional indexing suffers from the well-known "curse of dimensionality". That is, as the dimensionality of a feature space increases, the performance of the index decreases rapidly, and a search needs to traverse a larger part of the index [35], [36], [37]. In P2P networks, the situation becomes worse by the fact that the index structure is distributed, as we need to take into consideration the cost of data transfer between nodes.

On the other hand, the Locality-Sensitive Hashing (LSH) based approaches use special hash functions that output the same value for similar objects. In P2P networks, these hash functions are usually combined with the hash table interface of DHTs [31], [32], [33], [34], thus similar feature vectors are stored on the same/neighboring nodes to enable efficient similarity searches. To improve the locality of the hash functions, most works compromise the even distribution of hash buckets [38], [39], which translates to an imbalanced workload among nodes in a P2P network. Some works [33], [38], [39] do tackle this issue by learning more independent hash functions, but all of them perform one-time learning without considering the changes of data distribution that is common in P2P networks due to network churn. As the data is stored among nodes of corresponding hash ID, a one-bit change of the hash function output will result in large portion of (if not all) data being assigned to a different node, causing heavy network traffic.

D). BoVW Model

The bag-of-visual-words (BoVW) model represents each image with a bag of quantized codeword's derived from local features, and measures the similarity between images with the BoVW histogram analogous to a bag-of-words (BoW) model of text retrieval [10]. The retrieval process is typically supported by an inverted index. Though we are not aware of any BoVW based P2P CBIR systems, many existing P2P text retrieval systems build a distributed inverted index in a highly efficient manner over DHT, using term ID as key and document ID as value [40], [41], [42], [43].

Generally, there are two strategies to distribute index tuples: document partition (or local indexing), and term partition (or global indexing), both are well exploited in the literature [44], [45], [46]. With document partition, each node manages an index for a subset of documents. A query will be sent to all index nodes, and be answered by combining the lists of candidate documents returned from them. With term partition, each node manages an index for a subset of terms. This is not a very big issue in shared-memory or distributed servers, but does pose a challenge in P2P networks, as the nodes in P2P networks are loosely coupled and have much lower bandwidth. As a result, term partition is a more popular choice in P2P networks [40], [41], [42], [43]. To further reduce the network cost and tackle the issue of workload balance with term partition, different techniques have been proposed. For BoVW based CBIR in P2P networks, our previous work [47] proposes a codebook resembling mechanism to split the overloaded codeword's and

merge the under loaded codeword's to maintain a balanced workload among different codeword's. However, it does not take the relevance information into account. In addition, the split/merge is based on random re-sampling, which is heuristic.

III. PROPOSED METHODOLOGY

A scalable approach for content-based image retrieval in peer-to-peer networks by employing the bag-of-visual-words model. A dynamic codebook updating method by optimizing the mutual information between the resultant codebook and relevance information, and the workload balance among nodes that manage different codeword's. Improve retrieval performance and reduce network cost, indexing pruning techniques are developed. The experimental results indicate that the proposed approach is scalable in evolving and distributed peer-to-peer networks, while achieving improved retrieval accuracy.

IV. MODULES

1). FILE INDEX

An exact file is performed with a DHT lookup operation. Publishing a new file is performed by a DHT store operation. Performance and fault tolerance considerations, such information needs to be reposted periodically, otherwise it would be removed from the owner list.

2). CODEWORD INDEX

The CBIR search is essentially an inverted index lookup in the codeword index. Codeword's will be extracted on node C locally. The codeword's will be looked up in the codeword index. Similarity measurement to produce the ranked results for the user. The owner information of relevant images can be obtained by the file lookup process described.

3). COMPLEXITY ANALYSIS

The P2P network consists of nodes sharing a total of images utilized for retrieval, and each image has an average of code words. Our system completes a query in the following steps: feature extraction, quantization, and Sending posting lookup message, receiving postings, aggregating postings and producing the rank list.

4). WORK LOAD BALANCING

To partition the feature space evenly and accommodate the computational capacity of each nodes, so that no nodes would be overloaded or under loaded. While achieving good discriminability is important, a codebook

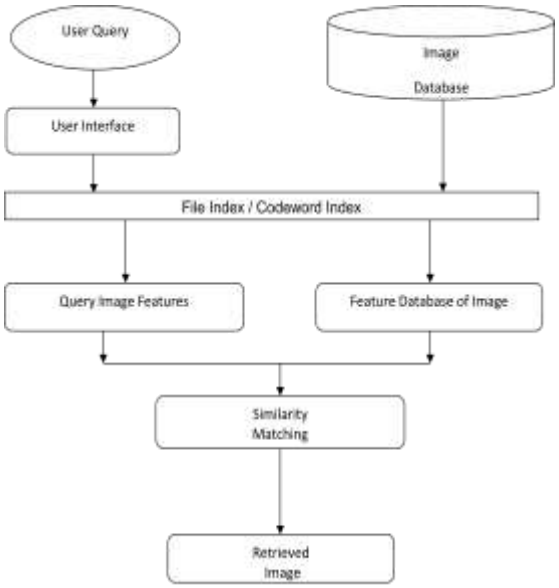

Fig 1. Architecture Diagram

used in P2P networks must also produce fair workload for each node. The frequency of update iterations is determined by the degree of churn. As shown by the experiments in a very low update frequency (once a few hour) is enough to maintain the performance.

V. PERFORMANCE AND EVALUATION

The codebook is ready, for a given query, the retrieval process essentially consists of three steps: extracting visual features and obtaining BoVW based representation for the query, retrieving the postings via DHT lookup, and measuring the similarity between the query and candidate images. In large scale BoW based retrieval systems, index pruning has been used to reduce the retrieval cost. Its basic idea is to identify

and discard the postings which are not likely to contribute to top results. In the experiments, two threshold based pruning techniques similar to [47] are implemented:

Reducing the query terms: Here, apply a threshold Θ_Q on the posting $W_{Q,k}$ of query image Q, so that only the terms satisfying $W_{Q,k} > \Theta_Q$ will be sent.

Reducing the answer terms: Here, apply a threshold Θ_A on the similarity scores of the candidate postings, so that only the postings satisfying WQ, kWA, $k > \Theta_A$ will be sent, where $W_{A,k}$ is the posting of a candidate image A.

The experiments are conducted on two publicly available datasets:

UKBench [49]: A benchmark dataset for object recognition. It contains 10,200 images with 4 images per object in different conditions. In our experiments, an image is considered relevant to the query image if both of them come from the same object. The SIFT descriptors as used as local descriptors.

Holidays [49]: A benchmark dataset for image retrieval. It contains 1,491 images with 500 queries and 991 corresponding relevant images. The number of relevant images for each query varies from 1 to 11. The SIFT descriptors coming with the dataset are used as local descriptors.

Additional datasets are used to facilitate the experiments:

Flickr100K [49]: For the distracters for large scale experiments, we use the first 100,000 images from the Flickr1M, where images are downloaded from Flickr.

Flickr60K [49]: For the source of initial codebook, we use an independent dataset Flickr60K, which contains 67,714 images downloaded from Flickr. We compare the proposed P2P codebook learning method (PCL) with codebook re-sampling (RS) [34] and k- means (KM) clustering under different settings. For PCL, we use the ground truth relevance information from the top 10 results for training. The codebook re-sampling (RS) updates the codebook using split/merge operations similar to the proposed method. However, the decisions to split/merge are based on the size of codeword:

SPLIT(k) if $|k| > \Theta_M$;
MERGE(k) if $|k| < \Theta_m$;

Where Θ_M and Θ_m are the workload thresholds for split and merge operations, respectively.

The k-means clustering (KM) updates the codebook by moving the codeword centroids to the cluster means. In the experiments, we implement a distributed version of k-means algorithm. During iteration, each codeword computes the new centroid from the mean of its descriptors. The new centroids are synchronized across the network to form the codebook for the next iteration. In order to achieve fair comparisons, we try to set the parameters of different methods to produce codebooks with size $k \approx$ 20,000, which is a typical number in reported BoVW based CBIR systems [48]. For large scale Holidays + Flickr100K, the codebook sizes are set to $k \approx 100{,}000$. In other words, we aim at a target workload of $S_k = |X|/k$, where $|X|$ is the total number of descriptors of all candidate images. For PCL, we set the workload lower and upper bounds as $S_m = S_M = S_k$ when relevance information is available, and $S_m = 0.5S_k$, $S_M = 2S_k$ when relevance information is missing. For re-sampling, we set the workload thresholds as $\Theta_M = 1.5S_k$ and $\Theta_m = 0.5S_k$. For k-means, the codebook size is fixed on k. As discussed in Section 3, both the CBIR search and codebook generation/updating take place on the codeword index. For the large scale experiment (Holidays + Flickr100K, with 100,000 codeword nodes), it takes more than 1,000 hours of running time, and 380GB of memory. In correspondence to the challenges in P2P environments discussed in Section 1, we evaluate the following 4 properties of the codebooks in the experiment:

Retrieval accuracy: We follow the recommended evaluation protocols of the datasets to measure the retrieval accuracy. For both datasets, we report the Mean Average Precision (MAP), which is the

average area under the precision recall curve for all the queries; and R-Precision (RP), which is the average precision of top R results, where R is the number of relevant images. For the UKBench dataset, we also report the Kentucky Score (KS) (the average number of positive images in top 4 results, which is essentially RP * 4) used by the dataset authors.

Workload balance: The workload balance is measured by the Gini coefficient among sizes of codeword's, where a coefficient of 0 expresses perfect workload balance (all codeword's have the same number of descriptors), and a coefficient of 1 expresses maximum workload imbalance (one codeword has all the descriptors).

Updating cost: Based on the discussion in Section 3.3, we measure the updating cost by the codebook change ratio $p = N_C/|K|$, where NC is the number of codeword's changed (either split, merged, or moved) in this iteration, and $|K|$ is the codebook size (total number of codeword's).

Query cost: The query cost is measured by the average number of retrieved postings for all the queries. The number determines the data volume that a query node will receive upon a query, which contributes most to the retrieval time. We compare the performance of different codebooks in 3 scenarios: static environment, dynamic environment, and retrieval with index pruning. The detailed settings for different scenarios are discussed in their corresponding sections.

VI. CONCLUSION

Our present a bag-of-visual-words (BoVW) model based approach for content based image retrieval (CBIR) in peer-to-peer (P2P) networks. I formulate the problem of updating an existing codebook as optimizing the retrieval accuracy and workload balance. Investigate DHT specific optimizations for cost reduction, more advanced matching refinement and multi-modal fusion techniques in P2P networks, and extensions of this approach to other distributed architectures. That is making the CAN address space corresponding to our feature space, and replace the CAN zones with codeword partitions. Such an embedding will eliminate the overhead of an additional DHT layer, as I can implement the SPLIT/MERGE operations as a CAN zone split/takeover, instead of adding and removing entries on DHT.

REFERENCES

1. M. Steiner, T. En-Najjary, and E. W. Biersack, "Long term study of peer behavior in the KAD DHT," IEEE/ACM Transactions on Networking, vol. 17, no. 5, pp. 1371–1384, Oct. 2009.
2. H. Schulze and K. Mochalski, "Internet study 2008/2009," Internet Studies, ipoque, 2009.
3. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2001, pp. 149–160.
4. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2001, pp. 161–172.
5. M. Mordacchini, L. Ricci, L. Ferrucci, M. Albano, and R. Baraglia, "Hivory: Range queries on hierarchical voronoi overlays," in IEEE International Conference on Peer-to-Peer Computing, Aug. 2010, pp. 1–10.
6. Y. Tang, S. Zhou, and J. Xu, "LIGHT: A query-efficient yet low maintenance indexing scheme over DHTs," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 1, pp. 59–75, Jan. 2010.
7. L. Zhang, Z. Wang, and D. Feng, "Efficient high-dimensional retrieval in structured P2P networks," in IEEE International Conference on Multimedia and Expo Workshops, Jul. 2010, pp. 1439–1444.
8. H. J´egou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," International Journal of Computer Vision, vol. 87, pp. 316–336, 2010.
9. J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in

IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 1470–1477.

10. J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in ACM International Workshop on Multimedia Information Retrieval, 2007, pp. 197–206.
11. K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in European Conference on Computer Vision, 2002, pp. 128– 142.
12. D. G. Lowe, "Object recognition from local scale-invariant features," in IEEE International Conference on Computer Vision, vol. 2, 1999, pp. 1150–1157.
13. S. B¨orzs¨onyi, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE*, pages 421–430, 2001.
14. J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with presorting. In *ICDE*, pages 717–816, 2003.
15. K. L. Tan, P. K. Eng, and B. C. Ooi. Efficient progressive skyline computation. In *VLDB*, pages 301–310, 2001.
16. D. Kossmann, F. Ramsak, and S. Rost. Shooting stars in the sky: An online algorithm for skyline queries. In *VLDB*, pages 275–286, 2002.
17. D. Papadias, Y. Tao, G. Fu, and B. Seeger. An optimal and progressive algorithm for skyline queries. In *SIGMOD Conference*, pages 467–478, 2003.
18. J. Pei, W. Jin, M. Ester, and Y. Tao. Catching the best views of skyline: A semantic approach based on decisive subspaces. In *VLDB*, pages 253–264, 2005.
19. Y. Yuan, X. Lin, Q. Liu, W. Wang, J. X. Yu, and Q. Zhang. Efficient computation of the skyline cube. In *VLDB*, pages 241–252, 2005.
20. P. Godfrey, R. Shipley, and J. Gryz. Maximal vector computation in large data sets. In *VLDB*, pages 229–240, 2005.
21. X. Lin, Y. Yuan, W. Wang, and H. Lu. Stabbing the sky: Efficient skyline computation over sliding windows. In *ICDE*, pages 502–513, 2005.
22. C. Y. Chan, H. V. Jagadish, K.-L. Tan, A. K. H. Tung, and Z. Zhang. On high dimensional skylines. In *EDBT*, pages 478–495, 2006.
23. W. T. Balke, U. G¨untzer, and J. X. Zheng. Efficient distributed skylining for web information systems. In *EDBT*, pages 256–273, 2004.
24. Z. Huang, C. S. Jensen, H. Lu, and B. C. Ooi. Skyline queries against mobile lightweight devices in manets. In *ICDE*, 2006.
25. .Wu, C. Zhang, Y. Feng, B. Y. Zhao, D. Agrawal, and A. E. Abbadi. Parallelizing skyline queries for scalable distribution. In *EDBT*, pages 112–130, 2006.
26. M. Cai, M. R. Frank, J. Chen, and P. A. Szekely. Maan: A multi-attribute addressable network for grid information services. In *GRID*, pages 184–191, 2003.
27. A. R. Bharambe, M. Agrawal, and S. Seshan. Mercury: supporting scalable multi-attribute range queries. In *SIGCOMM*, pages 353–366, 2004.
28. C. Zhang, A. Krishnamurthy, and R. Y. Wang. Skipindex: Towards a scalable peer-to-peer index service for high dimensional data. Technical report, Princeton University, 2004.
29. Y. Shu, K. L. Tan, and A. Zhou. Adapting the content native space for load balanced indexing. In *DBISP2P*, Pages 122– 135, 2004.
30. M. Batko, D. Novak, F. Falchi, and P. Zezula, "Scalability comparison of peer-to-peer similarity search structures," Future Generation Computer Systems, vol. 24, no. 8, pp. 834 – 848, 2008.
31. M. Bawa, T. Condie, and P. Ganesan, "LSH forest: Self-tuning indexes for similarity search," in International Conference on World Wide Web. New York, NY, USA: ACM, 2005, pp. 651–660.
32. D. Li, J. Cao, X. Lu, and K. C. Chan, "Efficient range query processing in peer-to-peer systems," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 1, pp. 78–91, Jan. 2009.
33. P. Haghani, S. Michel, and K. Aberer, "Distributed similarity search in high dimensions using locality sensitive hashing," in International Conference on Extending Database Technology: Advances in Database Technology. New York, NY, USA: ACM, 2009, pp. 744–755.
34. M. Zhou, H. T. Shen, X. Gong, W. Qian, and A. Zhou, "Personalized query evaluation in ring-based P2P networks," Information Sciences, vol. 220, no. 0, pp. 463–482, 2013.
35. C. B¨ohm, S. Berchtold, and D. A. Keim, "Searching in highdimensional spaces: Index structures for improving the performance of multimedia databases," ACM Computing Surveys, vol. 33, pp. 322–373, Sep. 2001.
36. R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in highdimensional spaces," in International Conference on Very Large Data Bases. San Francisco,

CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 194–205.

37. Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," Pattern Recognition, vol. 40, no. 1, pp. 262–282, 2007.
38. A. Joly and O. Buisson, "Random maximum margin hashing," in IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 873–880.
39. M. R. Trad, A. Joly, and N. Boujemaa, "Distributed KNN-graph approximation via hashing," in ACM International Conference on Multimedia Retrieval. New York, NY, USA: ACM, 2012, pp. 43:1– 43:8.
40. C. Tang, Z. Xu, and S. Dwarkadas, "Peer-to-peer information retrieval using self-organizing semantic overlay networks," in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 2003, pp. 175–186.
41. P. Reynolds and A. Vahdat, "Efficient peer-to-peer keyword searching," in ACM/IFIP/USENIX International Conference on Middleware, 2003, pp. 21–40.
42. Q. Xu, H. T. Shen, Y. Dai, B. Cui, and X. Zhou, "Achieving effective multi-term queries for fast DHT information retrieval," in International Conference on Web Information Systems Engineering (WISE), 2008, pp. 20–35.
43. H. Chen, J. Yan, H. Jin, Y. Liu, and L. M. Ni, "TSS: Efficient term set search in large peer-to-peer textual collections," IEEE Transactions on Computers, vol. 59, no. 7, pp. 969–980, 2010.
44. S. B¨ uttcher and C. L. A. Clarke, "A document-centric approach to static index pruning in text retrieval systems," in ACM Interna tional Conference on Information and Knowledge Management. New York, NY, USA: ACM, 2006, pp. 182–189.
45. A. Moffat, W. Webber, J. Zobel, and R. Baeza-Yates, "A pipelined architecture for distributed text query evaluation," Information Retrieval, vol. 10, no. 3, pp. 205–231, 2007.
46. R. Ji, L.-Y. Duan, J. Chen, L. Xie, H. Yao, and W. Gao, "Learning to distribute vocabulary indexing for scalable visual search," IEEE Transactions on Multimedia, vol. 15, no. 1, pp. 153–166, 2013.
47. L. Zhang, Z. Wang, and D. Feng, "Content-based image retrieval in P2P networks with bag-of-features," in IEEE International Conference on Multimedia and Expo Workshops, Jul. 2012, pp. 133–138.
48. D. Nist´er and H. Stew´enius, "Scalable recognition with a vocabulary tree," in IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2161–2168.
49. H. J´egou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in European Conference on Computer Vision. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 304–317.