International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 06 | May -2017 www.irjet.net p-ISSN: 2395-0072

EFFECTIVE APPROACH FOR CONTENT BASED IMAGE RETRIEVAL IN PEER-PEER TO NETWORKS

DATTATRAYA.T[1], S.S. DESAI[2],

[1]Student, CSE dept., BLDEA College, Karnataka, India
[2]Assistant professor, CSE dept., BLDEA College, Karnataka, India

---

Abstract *Peer-to-peer networking offers a scalable solution for sharing multimedia data across the network. With a large amount of visual data distributed among different nodes, it is an important but challenging issue to perform content-based retrieval in peer-to-peer networks. While most of the existing methods focus on indexing high dimensional visual features and have limitations of scalability, in this paper we propose a scalable approach for content-based image retrieval in peer-to-peer networks by employing the bag-of-visual words model. Compared with centralized 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, we 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 codewords. In order to further improve retrieval performance and reduce network cost, indexing pruning techniques are developed. Our comprehensive experimental results indicate that the proposed approach is scalable in evolving and distributed peer-to-peer networks, while achieving improved retrieval accuracy.*

***Key Words*: P2P, CBIR,BOVW.**

1. 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 and tens of millions of files. Unlike webpage's which mainly consist of textual documents such as news, blog articles or forum posts, multimedia files play a dominant role in most P2P networks. The ever-growing amount of multimedia data and computational power on P2P networks exposes both the need and potential for large scale multimedia retrieval applications such as content-based image sharing, and copyright infringement 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 contrast 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, structured overlay networks such as Distributed Hash Tables (DHTs) are often implemented on top of a physical network. By organizing the nodes in a structured way, messages can be efficiently 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. Usually, the feature vectors are indexed by a distributed high-dimensional index or Locality Sensitive Hashing (LSH) over the DHT overlay. However, due to the limitation known as "curse of dimensionality", the majority of these solutions have high network costs or serious workload balance issue among nodes when the dimensionality of feature vectors is high.

2.1 Summary of the project

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**.**

- Usually, the feature vectors are indexed by a distributed high-dimensional index or Locality Sensitive Hashing **(**LSH**)** over the DHT overlay In contrast 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**.**

- 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**.**
- Dexing and Locality-Sensitive Hashing**.** The high-dimensional indexing based approaches store the feature vectors in a data structure, usually a tree or a graph, to achieve effective search space pruning during retrieval**.** In structured P2P networks, the high-dimensional index is defined in a distributed way over the P2P overlay, dexing and Locality-Sensitive Hashing**.**
- The high-dimensional indexing based approaches store the feature vectors in a data structure, usually a tree or a graph, to achieve effective search space pruning during retrieval**.** In structured P2P networks, the high-dimensional index is defined in a distributed way over the P2P overlay**.**

2.2 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 [16], [17], [18], [19]. 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 [31], [32], [33]. 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.

A query will only be sent to the nodes managing corresponding terms, and answered by combining the inverted list returned from them. Therefore, document partition typically has a higher network cost than term partition, especially when the index has a good term sparsity [32]. 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 [16], [17], [18], [19]. 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 [34] proposes a codebook re sampling mechanism to split the overloaded codewords and merge the under loaded codewords to maintain a balanced workload among different codewords. However, it does not take the relevance information into account. In addition, the split/merge is based on random re-sampling, which is heuristic.

Besides P2P networks, a BoVW based CBIR system in distributed servers is proposed in [33], which seems most relevant to our work. It builds an inverted index among distributed servers with term partition. Learning processes are used to first filter the terms to reduce network cost, then distribute the terms into different servers to improve workload balance. Although the objectives are similar, their method is not directly applicable in P2P networks, as their learning method is designed to be an off-line process, which cannot deal with data under constant churn. It will incur high network cost if their learning method is performed in an on-line manner to keep up with changing data, as it requires co-occurrence information among all the terms to be collected and analyzed. Our proposed method achieves this in a very different way: we keep the term distribution unchanged, but update the codebook (the way each term is defined) to maintain the performance when data is changed. In this way, nodes managing different terms can adjust the workload individually with a much lower network cost Until now, very few researchers considered the problem of resolving conflicts in multi-party privacy management for Social Media. Wish art et al. Proposed a method to define privacy policies collaboratively. In their approach all of the parties involved can define strong and weak privacy preferences. However, this approach does not involve any automated method to solve conflicts, only some suggestions that the users might want to consider when they try to solve the conflicts manually.

2.3 BOVW CODEBOOK GENERATION

Unlike the BoW model, which has a natural vocabulary, the visual words of the BoVW model are obtained by quantizing the features using the codebook. Unsupervised methods such as k-means and sparse coding [35] aim to minimize the distortion between the original features and the quantized codewords. On the other hand, in [36], a supervised learning process is used to generate a discriminative codebook, where the loss of information provided by the codebook about the training samples is minimized. In [37], the codebook compactness, along with the discriminability is optimized. Alternatively, LSH methods [28] are also exploited for quantization. However, to the best of our knowledge, none of the existing methods is designed for P2P environments. Besides assigning features to codewords, alternative feature encoding approaches have been proposed. For example, VLAD [38] and Fisher Vector [39] represent features with the deviation from the codeword's (or a

generative model in Fisher Vector), which showed improved performance on many retrieval and classification datasets.
Most of the methods need to process the entire data collection in a centralized manner, which is infeasible in P2P networks. In addition, the specific issues for a distributed codebook such as network cost, workload balance and data churn are not well investigated. Besides the quantization method, the performance of a codebook is also affected by its size

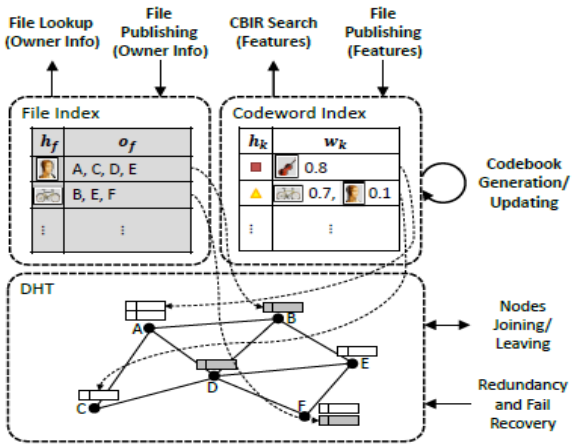

Fig. 1. Illustration of the network overlay structure. The proposed system builds a file index and a codeword index over the DHT overlay. The file index consists of file IDs (h_f) and its corresponding owners (o_f), while the codeword index consists of codeword IDs (h_k) and its corresponding postings (w_k). For example, in the codeword index, the second codeword has two postings: the bike image with a feature posting of 0.7, and the face image 0.1. The entries are distributed to the nodes of the network according to their keys, which are depicted by the dashed arrows from file/codeword index to DHT.

It is reported by many papers that a sophisticated method can be easily outperformed by a larger codebook [8], [30], [36], as more codewords generally leads to finer-grain quantization.

However, while a larger codebook yields better performance, it requires more computational resources. In centralized servers or clusters, the size of the codebook is usually predetermined, as available computational resources are fixed. However, in P2P networks, the available resources are under constant change, as peers join/leave the network. Predetermined codebook size is unlikely to produce optimal performance. Therefore, our proposed codebook learning method takes both codebook discriminability and workload balance into consideration.
The discriminability is measured by the mutual information provided by the codebook about user feedback, which is partially inspired by [36]. However, since our target application is CBIR and [36] targets classification, the objective function we derived is significantly different. The workload balance is measured by the difference between the current and "ideal" workload for each codeword. To make our codebook adaptive to dynamic P2P environments, the codebook partitioning is optimized by splitting/ merging codewords, thereby allowing the codebook to grow/shrink in accordance to the data distribution and available resources. In summary, our algorithm is tailored to deal with distributed and highly dynamic P2P environments.

2.4 File Publishing

File lookup

Looking up the owners of an exact file is performed with a DHT lookup operation: given a file ID hf, a list of nodes that has a copy of the file is returned by GET (hf).

File Publishing/Removing

Publishing a new file is performed by a DHT store operation: the file ID hf and the list of owner nodes of are stored by PUT (hf, of). For performance and fault tolerance considerations, such information needs to be reposted periodically; otherwise it would be removed from the owner list of. Therefore, removing an entry is achieved by stopping reposting.

3. SYSTEM ARCHITECTURE

To facilitate the BoVW retrieval process, our system builds inverted indices over the hash table interface of DHT. Before any further discussion, we briefly review DHT. DHT is a class of structured P2P overlay networks that provides GET (k) and PUT (k, v) operations similar to a hash table, where k, v are the key and value of a table entry, respectively. While different DHT implementations organize node connections with different topologies, most of them guarantee that a message from any node can reach the corresponding node in O(log n) hops, where n is the number of nodes. Additionally, DHT handles most issues in node management, including redundancy and failure recovery mechanisms in cases of nodes joining/leaving/failing, and caching and content mirroring for hot spots.

Therefore, DHT forms an infrastructure that can be used to build more complex applications. In order to support various operations of our CBIR system, we build a file index and a codeword index over DHT, as illustrated in Fig.2. The file index stores (hf, of) entries with file ID hf as key, and the file ownership information of as value. The codeword index, which stores the postings of each codeword, is added to support the storage and retrieval of BoVW features. It is essentially an inverted index which stores (hk, wk) entries with codeword ID hk as DHT key, and the corresponding postings wk as value.

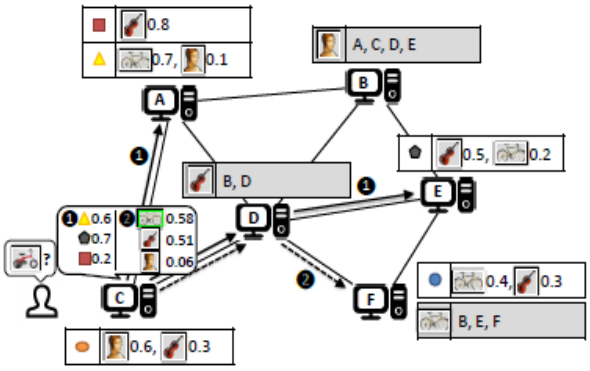

Fig. 2. Illustration of the CBIR process of a query over the DHT overlay network. The network has six nodes $(A, B, \ldots, F)$, and three images (face, yellow bike, and violin). The entries of the file index (marked with gray background) and codeword index (marked with white background) are stored distributedly among different nodes. A CBIR query is answered by first extracting the BoVW representation of the query image locally (list 1 of node C), looking up corresponding postings (solid arrows), computing the similarities and produce the rank list (list 2 of node c), and finally looking up the owners of the relevant image (dashed arrow).

Fig 2 CBIR process of a query over the DTH network.

3.1 PHASES

1) Admin

In this module, users have to register with peer network like (peer1, peer2, .peer5) after registration he has login by select peer and valid user name and password. After login successful he can do some operations such as view all user and their details, view images search requests and generate secret key for request, Add images by selecting category like (birds, animals, human beings) and its details, view all images belongs to same peer network with rank , comments for dislikes, view all images uploaded by him and perform operations like (edit or delete), view all images based on category wise, View all images search history and search ratio and finally generate chart for all images based on rank and search ratio.

2) User

In this module, there are n numbers of users are present. User should register before doing some operations. After registration, successful he can login by using valid user name and password. Login successful he will do some operations like view profile details, send request (or) view response of secret key for searching images, entering secret key and verify key move to search page and search images and their details and give (like or dislike) and also shows search ratio (2:10=0.2%),view his image search details. Finally view top keyword used for search based on category.

How to Resolve Conflict

1. As soon as the user migrate to any group then automatically the shared images or messages has to send to corresponding user.
2. As soon as the other user accepts the mutual friendship and then the shared images or messages has to send to corresponding user.

4. IMPLEMENTATION

When 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. Most existing index pruning techniques discard terms based on tf-idf postings [43]. In the experiments, two threshold based pruning techniques similar to [34] are implemented:

Reducing the query terms: we apply a threshold θQ on the posting wQ;k of query image Q, so that only the terms satisfying wQ,k > θQ will be sent.

Reducing the answer terms: we apply a threshold θA on the similarity scores of the candidate postings, so that only the postings satisfying wQ, k, wA, k > θA will be sent, where wA,k is the posting of a candidate image A.

4. Implementation Details

Bovw Codebook Generation Algorithm:

We optimize EK by finding a suitable partition granularity and good centroid positions. The learning algorithm adjusts the partitioning by splitting/merging the partitions iteratively. Since the codeword index and relevance information is managed by codeword nodes pk, the decision to split/merge a codeword k is made by pk individually based on its own data. Generally, for a partition k, one of three cases applies:

1) SPLIT: The size of k is large enough for sub partitioning, and it is possible to get a good sub partitioning based on available data. In this case, a SPLIT operation is performed: the new centroids of the sub-partitions of k are published to the codebook, thus splitting k into a few smaller partitions.

2) MERGE: The current partition performs badly (has low Ek), but the size of k is not large enough for a good sub-partitioning. In this case, a MERGE operation is

performed: k is removed from the codebook, and its data is taken over by its neighbors.

3) **UNCHANGED:** The current partition performs well, and one cannot get a better partitioning with either SPLIT or MERGE. In this case, k remains unchanged.

We first try to SPLIT k to see if a sub-partitioning increases Ek. To get a discriminative yet compact sub partitioning, we first generate an over-complete set of N candidate centroids Cn, and then select the subset from Cn which maximizes Ek. The selection process is similar to feature selection [42], and generally there are two greedy selection schemes: backward elimination and forward selection. Starting from Cn, the backward elimination tries to remove one centroid at a time from the current candidate set Ci. Every centroid in Ci is considered for removal, and the configuration with maximum value of Ek is selected as Ci-1. Finally, from the sets C1; C2; Cn, the set with maximum value of Ek is selected as the final partitioning.

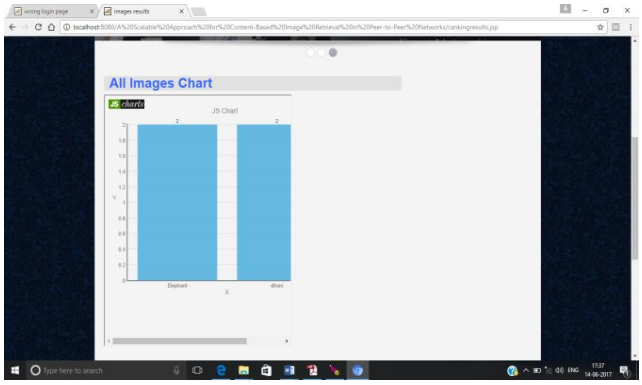

Figure 2: Bovw Codebook Generation Algorithm:

Issue resolution algorithm

The conflicted user is given as input to the set of rules. System locate outs consumer's willingness to alternate their preferred movement (furnish/deny) for particular centered person. Based on that system fashions concession rules and sooner or later consumer gets the answer as a battle resolved policy.

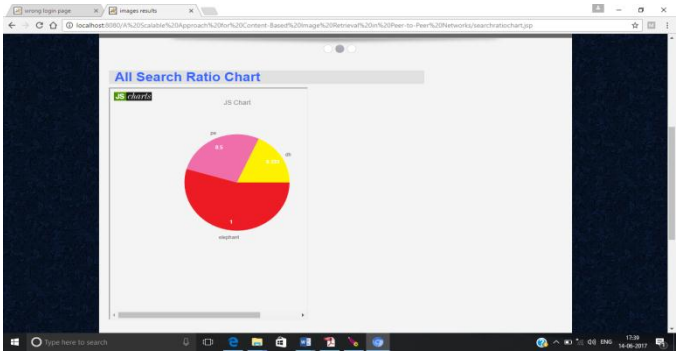

Figure 3: Conflict Detection Algorithm

5. CONCLUSIONS

This project aims at resolving the problem seen in content based image retrieval deployments in P2P networks, we present a bag-of-visual-words (BoVW) model based approach for content based image retrieval (CBIR) in peer-to-peer (P2P) networks. In order to overcome the difficulty in generating and maintaining a global codebook when the BoVW model is deployed in P2P networks, we formulate the problem of updating an existing codebook optimizing the retrieval accuracy and workload balance. As a result, the proposed approach is scalable to the number of images shared within a P2P network and the evolving nature of P2P networks. In order to further improve the retrieval performance of the proposed approach and reduce network cost, indexing pruning techniques are applied. We conduct comprehensive experiments to evaluate various aspects of the proposed approach while demonstrating its promising performance. In the future, we will 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. In particular, for the CAN network [4], we can embed the index into the CAN overlay. That is, we make 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 we can implement the SPLIT/MERGE operations as a CAN zone split/takeover, instead of adding and removing entries on DHT.

We proposed a new algorithm to prevent user's password from being stolen by adversaries. We introduced a new mechanism of authentication using a virtual password For Retrieving Document involving a small amount of human calculation to secure user's password in online environments and ATM's. We analyzed how the proposed scheme defends against phising, key-logger and shoulder surfing attacks.

6. AKNOWLDENENT

I would like to take this opportunity to express my thanks to my guide Prof.S.S.Desai for his esteemed guidance and encouragement. His guidance always helps me to succeed in this work. I am also very grateful for his guidance and comments while designing part of my research paper and learnt many things under his leadership.

REFERENCES

[1] M. Steiner, T. En-Najjary, and E. W. Biersack, "Long term studyof peer behavior in the KAD DHT," IEEE/ACM Transactions onNetworking, vol. 17, no. 5, pp. 1371–1384, Oct. 2009.

[2] H. Schulze and K. Mochalski, "Internet study 2008/2009," InternetStudies, ipoque, 2009.

[3] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internetapplications," 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 onApplications, Technologies, Architectures, and Protocols for ComputerCommunications, 2001, pp. 161–172.

[5] M. Mordacchini, L. Ricci, L. Ferrucci, M. Albano, and R. Baraglia, "Hivory: Range queries on hierarchical voronoi overlays," in IEEEInternational Conference on Peer-to-Peer Computing, Aug. 2010, pp. 1–10.

[6] Y. Tang, S. Zhou, and J. Xu, "LIGHT: A query-efficient yet lowmaintenanceindexing scheme over DHTs," IEEE Transactions onKnowledge and Data Engineering, vol. 22, no. 1, pp. 59–75, Jan. 2010.

[7] L. Zhang, Z. Wang, and D. Feng, "Efficient high-dimensionalretrieval in structured P2P networks," in IEEE International Conferenceon Multimedia and Expo Workshops, Jul. 2010, pp. 1439–1444.

[8] H. J´egou, M. Douze, and C. Schmid, "Improving bag-of-featuresfor large scale image search," International Journal of ComputerVision, 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 onComputer Vision, vol. 2, 2003, pp. 1470–1477.

[10] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluatingbag-of-visual-words representations in scene classification," inACM International Workshop on Multimedia Information Retrieval,2007, pp. 197–206.