INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

A Survey On: CBIR And Mobile Agent For Image Retrieval Over Networks

Gaurav G. Rakhonde
Dept. of Computer Science, Research Fellow, G.H.R.I.E.M., Jalgaon, India

Mr. Prashant P. Rewagad
Dept. of Computer Science, Asst. Prof, G.H.R.I.E.M., Jalgaon, India

Abstract:
As we know that the various images are stored over different locations on network. CBIR System retrieves the images according to the contents or features such as color, texture and shape of image. So for retrieving the images over network we reviews the mobile agent technology. Mobile agents are having the capability of migrating freely over the network to collect the desired information and returns back to the user with desired results. So in this paper we are surveying the CBIR and mobile agent technology for retrieving images through the combination of these both technologies.

Key words: *Mobile agents, content based image retrieval, distributed CBIR*

1.Introduction

As multimedia devices such as mobile phones are becoming very usual, huge collections of digital images are available today. Finding images belonging to a specific category in these ever growing collections is a difficult task since searching within by hand has become impossible. Content Based Image Retrieval (CBIR) has been successfully proposed to answer this problem. In a CBIR system, low-level visual image features (e.g., color, texture, and shape) are automatically extracted for image descriptions. To search for desirable images, a user presents an image as an example of similarity, and the system returns a set of similar images based on the extracted features. The problem of such techniques is the well known semantic gap between the numerical values attached to images and the semantical concepts they belong to. In order to reduce the gap, machine learning techniques have been successfully adapted to train a similarity function in interaction with the user (using her labeling of the results) leading to the so called "relevance feedback". The main idea is to build a representation of the image based on its content, and then to find a relation between this representation and the semantic we associate to the image. Machine learning techniques such as active learning have been successfully adapted order to deal with image retrieval distributed over a network. The best improvement was done with the introduction of relevance feedback [1], [2] into the process.

With the expansion of networks such as the Internet, peer to- peer networks or even personal networks, image retrieval has become a difficult task. As images are split into many collections over the web, the problem of CBIR is not only to find the most relevant images, but also to find the localization of relevant collections. The major part of CBIR computation being dedicated to the processing of the image descriptors, the fact that images are distributed over many sources should be more an advantage than a drawback since it means a possible paralleling. In their system, the links between peers of the network are optimized in order to propagate the query to relevant hosts. Here the smart cooperation is taken between the interactive CBIR and a localization learning based on mobile agents.

2.Problem Definition

The Internet or p2p networks provides huge volumes of Information and to search these information search engines have been developed in order to find the best localizations of data matching a query. When it comes for mining multimedia documents then these search engines usually gives poor results as they search on contextual web pages or Meta information attached with multimedia objects. The results of web search engines are far from expected regarding the semantics of the documents. Also the user is unable to crawl the network by hand, in that sense, the work on search engines is highly valuable for today's applications.

The system adapts machine learning techniques such as active learning in order to deal with image retrieval, based on Content Based Image Retrieval technique, distributed over a network. The system is a Two-step learning scheme which keeps the track of the path leading to the collection containing the relevant images and the similarity between images. This learning scheme efficiently implements this two-step learning combination by using an ant-like behavior algorithm.

3.Distributed Content Based Retrieval

3.1.Content Based Image Retrieval

Motivated by the lack of an efficient image retrieval technique, content-based image retrieval was introduced. "Content-based" means that the technology makes direct use of content of the image rather than relying on human annotation of metadata with keywords.

Current content-based image retrieval systems make use of low-level features to retrieve desired images from image collections. It is based on matching of the features of the query image with that of image database through some image These image features are normally basic image information like shapes, color and texture.

3.2.Relevance Feedback For CBIR

Relevance feedback is more often used with content-based image retrieval than text based image retrieval. Content-based image retrieval systems often support relevance feedback. Relevance feedback for content-based image retrieval means that the user can mark the results of the query as "relevant", "not relevant" images, which are then again fed back to the systems as a new, refined query for the next round of retrieval. This process is repeated until the user is satisfied with the query results. This is a way for the system to learn and to personalize the answers. The query with the new information is resubmitted and improved results are returned.

3.3.CBIR In Distributed Collections

In the distributed image retrieval scheme, images are spread into several well known collections. This is advantage since the processing of every image could be naturally paralleled. In the classical distributed information retrieval scheme, documents are spread into several well known collections. The problem is at first to build a description of each collection, then to select where to retrieve the documents, and finally to merge the results into a single ranked list [3].

CBIR systems require an interaction with the user to be efficient, which is not taken into account in the classical distributed information retrieval scheme. Finally, In peer-to-peer networks, it is not possible to identify the few well known collections anymore. Instead, each peer must index its own images and queries must be propagated from one peer to another. In DISCOVIR [4], King proposes an algorithm for selecting links between peers based on the content of their shared images. The queries are more likely to be propagated to peers which are known to host similar images. With this method, they achieve to improve the retrieval and reduce the network load.

Figure 1.1 shows the architecture for the overall distributed retrieval system. It can be noticed that each archive is linked to the network through a proxy module. Each proxy stores all relevant information about the resource it gives access to. This information includes summaries of resource content (resource descriptions), specification of supported media types, database schema and query predicates for each schema attribute, rules to format a query so that it can be processed by the resource, and rules to format retrieval results so that they conform to a common, predefined syntax. A new proxy is built each time a new resource wants to join the network.

At that time, the resource is sampled either following a cooperative protocol or an uncooperative one (only for text libraries). In the former case, it is assumed that the digital library adheres to the project, joining the network of federated libraries and providing content descriptors for all its documents.

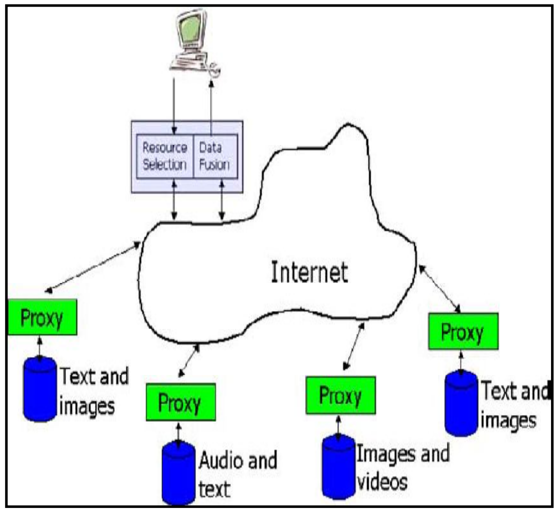

Figure 1: Architecture of Distributed CBIR

The latter case occurs when the digital library does not explicitly join the federated network, and its content can be only estimated through some form of sampling. Sampling the resource aims at the extraction of the resource description, that is a summary of resource content.

Resource descriptions play a central role in the operational cycle of the distributed retrieval system, and are used both for resource selection and for data fusion. Resource selection is the process that, given a user query, selects which resources is best candidates to contain relevant items. Of course, selection is not performed by actually issuing the query to the resource. Rather, estimation of the relevant items that are contained in a generic resource is accomplished by comparing the query against the resource description.

Resource descriptions are also used for data fusion, as described in the next Sections. In general, being resource descriptions a summary of resource content, they are used to help the definition of a model representing how items are distributed in the collection (resource).

However, there are two challenges for distributed CBIR [5, 6]; the first challenge is image source selection when searching large numbers of distributed image sources. It is likely that only a relatively small number of image sources may contain image relevant to the query. It is obvious (but wasteful) to query all image sources for a list of similar images. The second challenges in distributed CBIR is result merging, where a query Q to find top k best results is sent to n image sources which returns the k best image results.

Thus, a total of k n results will be returned. The results are merged from all image sources and the top k results are return to the user [7].

3.4.Mobile Agents

The term software agents refer to programs that perform certain tasks on behalf of the user. Software agents can be classified as static agents and mobile agents. Static agents achieve the goal by executing on a single machine. On the other hand, mobile agents migrate from one computer to another and executes on several machines. Mobility increases the functionality of the mobile agent. To perform the needed parallelization of feature vector processing, we have chosen to use mobile agents.

Actually the working of mobile agents is given here. A mobile agent consists of the program code and the program execution state. Initially a mobile agent resides on a computer called the home machine. The agent is then dispatched to execute on a remote computer called a mobile agent host. When a mobile agent is dispatched the entire code of the mobile agent and the execution state of the mobile agent is transferred to the host. The host provides a suitable execution environment for the mobile agent to execute. Another feature of mobile agent is that it can be cloned to execute on several hosts. Upon completion, the mobile agent delivers the results to the sending client or to another server [8]

A mobile agent can be thought of as a software program which travels from one platform to another in order to get its work done, during this process it carries its state and data with itself and resume its execution from the state it had left on the previous platform [9]. The reason for using mobility is the improved performance which can be achieved by moving the agent closer to the new host, where it can use the services locally. We can take an example where agent needs information from several host situated on different platforms. It can use remote procedure call (RPC) where it can request the desired information and obtain the results by invoking the remote methods. This RPC follows the client-server paradigm. But if the volume of data is large it can create bandwidth and network traffic problem. In such cases the mobile agent can migrate to those remote hosts and perform the functions locally and come back with the desired results. It would be a more efficient way to process the data.

The ability of an agent to migrate from one environment to another is not a requirement for agent hood. Still mobility is an important property for many agent-based systems and necessary for a certain class of application. The basic architecture of the mobile agent can be thought of as a client sends out an agent who travels the network visiting servers in order to perform some required action.

Fig 1.2 shows the Mobile agent which provides a new design model for applications as compared to the traditional client server model. First and foremost, the mobile agent blows apart the very notion of client and server. With mobile agents, the flow of control actually moves across the network, instead of using the request/response architecture of client-server. In effect, every node is a server in the agent network, and the agent moves to the location where it may find the services it needs to run at each point in its execution [8]. The various components of the architecture are: -

Agent Manager: The Agent Manager provides the communication infrastructure using the TCP/IP stack for agent transmission. It abstracts the network interface in order that agent programmers need not know any network specifics nor need to program any network interfaces.

3.5.Security Manager

The Security Manager is responsible for identifying users, authenticating their agents, protecting server resources and ensuring the security and integrity of agents.

Persistence Manager: The Persistence Manager is completely transparent and maintains the state of agents in transit around the network. As a side benefit, it allows for the checkpoint and restart of agents in the event of system failure.

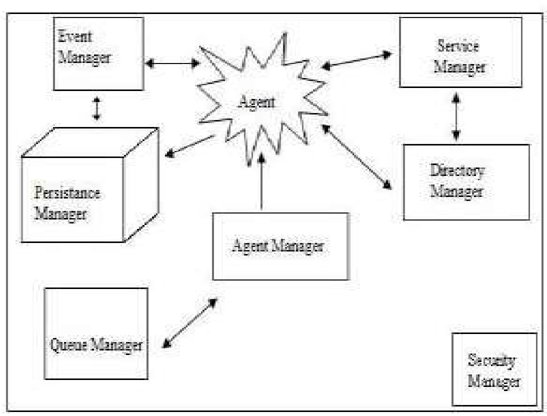

Figure 2: Architecture of Mobile Agent

Event Manager: The Event Manager handles the registration, posting and notification of events to and from agents. The Event Manager can pass event notification to agents on any node in the network thus supporting agent collaboration.

3.6.Queue Manage

The Queue Manager is responsible for the scheduling and possibly retrying the movement of agents between systems which include maintenance of agent and persistence of agent state.

3.7.Directory Manager

The Directory manager provides naming service in the network. The Directory Manager may consult a local name service or may be set up to pass requests to other, existing name servers.

3.8.Service Manager

The Service manager provides the interface from agents to the services available at the various machines in the network. It comprises a set of programming extensions to provide access the native API"s and interfacing.

3.9.*Ant-Like Agents*

In the case of distributed retrieval, ant-agents crawl the network to find the relevant documents. They move from one peer to another and mark the visited hosts (by changing a numerical value locally stored on these hosts, called marker). Software agents acting as ants moving on the web, starting from the user's computer (niche) and looking for "food" (information). The main questions are: how to specify the searched information? How the "ant" agent could deposit "pheromone-like" markers? These markers can be viewed as a collective memory of paths leading to the relevant sites. This behavior-based mapping of the network is well adapted to inconsistent networks such as peer-to-peer networks, since the marked paths evolve with the global trend of the agent movements [11], [12]. In our distributed CBIR context, we have to do several travels between the user's computer and the information sources.

The principle of "ant" strategies [13] is to optimize routes towards a given resource by reinforcing markers (called pheromones) on sites situated along the pathway from the source to the destination. Given a set of agents launched from site *s1* (source) and which should reach site *sN* (destination), the optimization is the result of the emergence of a dynamical attractor coming from the interaction between all the agents and their environment.

4.Active Learning

Active learning algorithm is used for conducting effective relevance feedback for image retrieval. The algorithm selects the most informative images to query a user and quickly learns a boundary that separates the images that satisfy the user's query concept from the rest of the dataset. The main issue with active learning is finding a way to choose informative images within the pool to ask the user to label. We call such a request for the label of an image a pool-query. Most machine learning algorithms are passive in the sense that they are generally applied using a randomly selected training set. The key idea with active learning is that it should choose its next pool-query based upon the past answers to previous pool-queries.

4.1.Local Images Selection

Each time an agent gets to a site containing a collection; it has to choose some examples to add to the training set. As many agents reach the same host with the same relevance function, the active strategy should not answer in a deterministic way; otherwise all these mobile agents will get the same answers, and thus act as one single agent. We rank images given their distance to the boundary. We divide the selection of images into selections of a single image. Each of these selections are done over a set of images with ranking between 1 and using an uniform distribution. The selected image is then removed from the set [6].

4.2.Collection Selection

The relevant category is very little in front of the available data. Thus, a relevant image might often be considered as more informative than an irrelevant one. a good collection selection strategy the one that selects the collections containing mainly relevant images. Our collections selection strategy is performed by the ant algorithm. At the very beginning, all collections have an equal chance of being visited by an agent (all markers are set to 1). After a few rounds the highest probability of selecting images will be obtained for the collection that returned the largest set of positively labeled images [6].

In case of a collection with only few positive images, this strategy will at first reinforce the selection on this collection. But as soon as the vein is exhausted, no more positive labels will reinforce the marker, and the probability will decrease quickly. This shows how the strategy adapts through the dynamics of active learning.

5.Active Learning Methods

Two different strategies are usually considered for active learning: the uncertainty-based sampling, that selects the images for which the relevance function is the most uncertain and the error reduction strategy, that aims at minimizing the generalization error of the classifier [14].

5.1.Uncertainty-Based Sampling

In our context of binary classification, the learner of the relevance function has to classify data as relevant or irrelevant. Any data in the pool of unlabeled samples may be evaluated by the learner. Some are definitively relevant, others irrelevant, but some may be more difficult to classify. Uncertainty-based sampling strategy aims at selecting unlabeled samples that the learner is the most uncertain about.

The efficiency of these methods depends on the accuracy of the relevance function estimation close to the boundary between relevant and irrelevant classes.

5.2.Error Reduction-Based Strategy

Active learning strategies based on error reduction [15] aim at selecting the sample that, once added to the training set, minimizes the error of generalization of the new classifier.

6.Conclusion

The purpose of this survey is to provide an overview of the Distributed Content based Image Retrieval System by using mobile agent technology. Also a new active learning strategy for searching images over networks is discussed.

7.Reference

1. M. Wood, N. Campbell, and B. Thomas, "Iterative refinement by relevance feedback in content-based digital image retrieval," in ACM Multimedia 98, Bristol, UK, Sep. 1998, pp. 13–20.
2. T. Huang and X. Zhou, "Image retrieval with relevance feedback: From heuristic weight adjustment to optimal learning methods," in Int. Conf. Image Processing (ICIP'01), Thessaloniki, Greece, Oct. 2001, vol. 3, pp. 2–5.
3. J. Callan, Distributed Information Retrieval. Norwell, MA: Kluwer, 2000, pp. 127–150, ir 5.
4. King, C. H. Ng, and K. C. Sia, "Distributed content-based visual information retrieval system on peer-to-peer networks," ACM Trans. Inform. Syst., vol. 22, no. 3, pp. 477–501, 2004.
5. Picard, M. Cord, and A. Revel,"Image Retrieval Over Networks: Active Learning Using Ant Algorithm" IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 10, NO. 7, NOVEMBER 2008.
6. D. Picard, A. Revel , and M. Cord, "Long term learning for image retrieval over networks". Image Processing, 2008. ICIP 2008 15th IEEE International Conference on , vol., no., pp.929-932, 12-15 Oct. 2008
7. Alaa M. Riad, Ahmed Atwan, Hazem M. El-Bakry, Sameh Abd EL-Ghany, "An Intelligent Distributed Algorithm for Efficient Web Image Retrieval", International journal of comm. Issue 3, vol 3, 2009.
8. Sabu .M Thampi1, Dr. K. Chandra Sekaran2," Mobile Agents For Content-Based Www Distributed Image Retrieval", International Conference on Human Machine Interfaces 20-23, Dec. 2004
9. Danny B.Lange and Mitsuru Oshima, "Programming and Developing Java Mobile Agents with Aglets"
10. Rakesh Kumar, Dr. Mayank Dave," Mobile Agent as an Approach to Improve QoS in Vehicular Ad Hoc Network", IJCA Special Issue on "Mobile Ad-hoc Networks" MANETs, 2010
11. A. Revel,"Web-agents inspired by ethology:A population of "ant"-like agents to help finding user-oriented information," in IEEE WIC'2003: Int. Conf. Web Intelligence, Halifax, NS, Canada, Oct. 2003, pp. 482–485.

12. A. Revel, “From robots to web-agents: Building cognitive software agents for web-information retrieval by taking inspiration from experience in robotics,” in ACM Int. Conf. Web Intelligence, Compiègne, France, Sep. 2005.
13. M.Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1(26):29–41, 1996.
14. D. Cohn, “Active learning with statistical models,” Journal of Artificial Intelligence Research, vol. 4, pp. 129–145, 1996.
15. N. Roy and A. McCallum, “Toward optimal active learning through sampling estimation of error reduction,” in International Conference on Machine Learning, 2001.
16. Danny B. Lange and Mitsuru Oshima, “Seven Good Reasons for Mobile Agents ", Communications of ACM, vol. 42, no. 3, March 1999.