International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

An Overview of Content Based Image Retrieval System using Mobile Agent

Yogita B. Verulkar[1], Bharat K. Chaudhari[2]

[1, 2]Computer Engineering, Padm. V. B. Kolte College of Engineering, Malkapur, Maharashtra, India

Abstract: ***As we know that various images are stored over different locations on network. Content Based Image Retrieval (CBIR) System retrieves the images according to the contents or features of the images such as color, texture and shape of image. So, for retrieving the images over network we review the mobile agent technology. Mobile agents are the software programs having the ability of migrating freely over the network. They collect the desired information by crawling over the network and returns back to the user with desired results. So, in this paper we are presenting these two technology i.e. CBIR and mobile agents for retrieving images over network.***

Keywords: Content based image retrieval, distributed CBIR, mobile agents, peer-to-peer network

1. Introduction

As we know multimedia devices such as mobile phones are becoming very usual so large collections of digital images are offered today. Finding images belonging to a specific category in these growing collections has become a difficult task since searching by hand has become impossible. Content Based Image Retrieval has been successfully proposed to solve this problem. In a CBIR system, low-level visual image features such as color, texture, and shape are extracted automatically for image descriptions. To search for desirable images, the user gives an image as an example of similarity, and the system returns back with a set of similar images based on the extracted features of those images. The problem of such techniques is the renowned semantic gap between the numerical values attached to images and the semantical concepts to which they belongs. In order to reduce the differences, machine learning techniques have been successfully accepted to train a similarity function in interaction with the user (using her labeling of the results) leading to the technique called “relevance feedback”. Machine learning techniques such as active learning have been successfully adapted in order to deal with image retrieval distributed over a network. The main goal is to build a representation of the image based on its content, and then to find a relation among their representation and the semantic we associate to the image. The best improvement was done with the starter of relevance feedback [1], [5] into the process. With the growth of networks such as the Internet and peer-to- peer networks or even personal networks, image retrieval has become a very difficult task. The major part of content based image retrieval system is the computation being dedicated to the processing of the image descriptors. As images are divided into many collections over the network, the goal of CBIR is not only to find the most relevant images, but also to find the localization of relevant collection. The reality that images are distributed over many hosts must be more an advantage than a drawback since it means a possible paralleling. In their system, the links between hosts of the network are optimized in order to travels the query to relevant hosts. Here the smart cooperation is taken among the interactive CBIR and a localization learning based on mobile agents.

2. Problem Definition

The Internet or peer-to-peer networks provides huge volumes of data and to search these data or information search engines have been developed in order to find the best localizations of data matching a query. When it deals with mining multimedia documents these search engines usually gives poor results as their search usually depends on contextual web pages or Meta information which are attached with multimedia objects. Regarding the semantics of the documents the results of web search engines are far from expected. As it is not possible to the user to crawl the network by hand, in that sense, the task on search engines is highly valuable for today’s applications [5].

The Content Based Image Retrieval technique, adapts machine learning techniques such as active learning so as to deal with image retrieval, based on system 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 similarity between images and the relevant images [11].

3. Distributed Content Based Retrieval

3.1 Content Based Image Retrieval

Content-based image retrieval systems uses of low-level features to retrieve the desired images from image database. The term “Content-based” means directly using content of the image instead relying on human annotation of metadata with keywords such as name or texts related with the images. It is based on matching of the features of the query image with that of image present in the database. These image features are generally the basic image information like shapes, color and texture. In content-based image retrieval systems initially the visual contents of the images in the database are extracted from the given query image and described by multi-dimensional feature vectors [2]. These feature vectors of the images in the database then form a feature database. To retrieve similar images from the database, users provide the retrieval system with example images or any sketched figures. The system then changes these images into its internal representation of feature vectors considering various terms such as its color, shape and texture. The similarities or the distances between the feature vectors of the query example (given input image) or sketch and those of the images in the database are then calculated and retrieval is performed with the use of an indexing scheme.

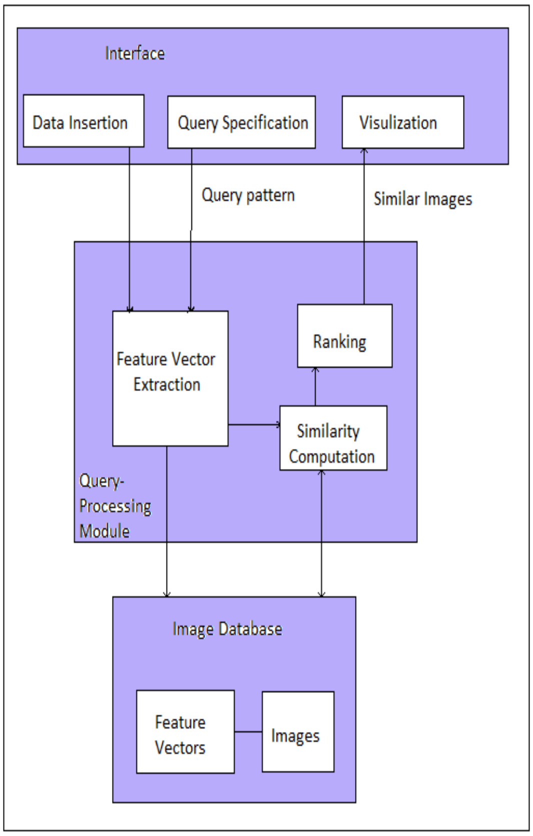

Figure 1: Typical Architecture of Content Based Image Retrieval System

The figure1 shows a typical architecture of a content-based image retrieval system. The two main functionalities of this system are: data insertion and query processing. The data insertion subsystem is generally works offline which extract appropriate features for mining and loading them into the image database. The query processing, in turn, is organized as follows: the interface allows a user to specify a query by means of a query pattern and to visualize the retrieved similar images. The query-processing module initially extracts a feature vector from a query pattern and then applies a metric to evaluate the similarity between the query image and the database images. Later, it ranks the database images in a decreasing order of similarity to the query image and forwards the most similar images to the interface module.

3.2 Features of Content Based Retrieval System:

This section introduces three features: texture, shape, and color, which are used most often to extract the features of an image. The features are described in more detail as follows [7].

3.2.1 Color Features:

Color is linked to the chromatic part of an image. A color histogram provides allotment of colors which is achieved by damaging image color and plus how many numbers of pixels fit into every color.

3.2.2 Texture Features:

By dissimilarity in brightness with high frequencies in the image spectrum, textures are characterized. While making a distinction between areas of the images with same color, these features are very useful. Measures of image texture such as the degree of contrast, coarseness, directionality, regularity and randomness can be calculated using second-order statistics.

3.2.3 Shape Features

By either the global form of the shape or local elements of its boundary, shape features can be differentiated. Global form of the shapes is like the area, the extension and the major axis orientation. Local elements of its boundary are like corners, characteristic points or curvature elements

3.3 CBIR in Distributed Collections

Place In the distributed image retrieval scheme, images are spread into several collections of database. This is advantage since the processing of every image could be naturally paralleled. The problem is at first to build a description of each collection, then to choose where to retrieve the documents, and finally to combine the results into a sole ranked list [3]. In the classical distributed information retrieval scheme, documents are spread into numerous collections. CBIR systems needs an interaction with the user to be efficient. This is not taken into account in the classical distributed information retrieval scheme. Finally, in p2p networks, it is not possible to classify the few well-known collections anymore. Instead, each peer must index its own images and queries must be transferred from one peer to another. In DISCOVIR [3], King proposes an algorithm for selecting links between peers based on the content of their shared images. The queries are more likely to be transferred to peers which are known to host similar images. By this, they achieve to advance the retrieval and lessen the network load [2], [6].

3.4 Tables *Relevance Feedback for CBIR:*

Relevance feedback has been often supported by Content-based image retrieval system. 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. Relevance feedback is more often used with content-based image retrieval than text based image 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 improved results are returned by resubmitting the query with the new information.

4. Mobile Agents

The term software agents refer to programs that perform certain tasks as given by the user. Software agents can be classified as static agents and mobile agents. Static agents generally execute on a single machine. On the other hand, mobile agents crawl from one computer to another and executes on several machines. Mobility increases the functionality of the mobile agent. Mobile agents have the ability to move from host to host, executing at each place and then keeping the results before moving to the next server. A simple picture of mobile agents in a network [12]. 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, Mobile agents are processes transmitted from a source computer to accomplish a task given by the client host. A mobile agent comprises of the program code and the program execution state [4]. Mobility increases the functionality of the mobile agent. Initially a mobile agent exists on a computer called the home engine. The agent is then posted to accomplish an execution 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 from one host to another. A suitable execution environment is provided by the host to the mobile agent for their execution [10]. Another feature of mobile agent is that it can be cloned to execute on numerous hosts. Upon completion, the mobile agent carries the results to the sending client or to another server.

Aglet Technology is used as a framework for programming mobile network agents in Java developed by the IBM™ Japan research group. An aglet is a lightweight Java object which can be dispatched to any remote host that supports the Java Virtual Machine.The IBM's mobile agent is known 'Aglet This requires that the remote host must have a pre-install Tahiti, a tiny aglet server program implemented in Java along with it must be provided by the Aglet Framework. The IBM Aglet team provided the so-called "FijiApplet". FijiApplet preserves some kind of an aglet context. From within this context, aglets can be formed, dispatched from and draw back to the FijiApplet. an abstract applet class that is portion of a Java package known as "Fiji Kit" to allow aglets to be fired from within applets.

We can take an example where agent requests information from various host situated on different platforms. It can use remote procedure call (RPC) where it can request the desired information and get the results by appealing the remote methods. This RPC follows the client-server paradigm. But if the volume of information is large then it can create bandwidth and network traffic problem [8]. 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 a significant stuff for many agent-based systems and essential for a certain class of application. The mobile agent setbacks apart the very notion of client and server. With mobile agents, the control flow actually transfers across the network, as an alternative of using the request/response architecture of client-server. In effect, every node is a server in the agent network, and the agent transfers to the place where it may find the facilities it wants to run at each fact in its execution [9].

4.1 Generic mobile agent system architecture

In order to make a mobile agent system work, it is not enough to build the agents themselves. A program at each site is also needed to handle the incoming agents and send out agents. This program is often called an agency. The agency can be built differently depending on which type of agent system is needed, but a general architecture can be seen in figure 2.

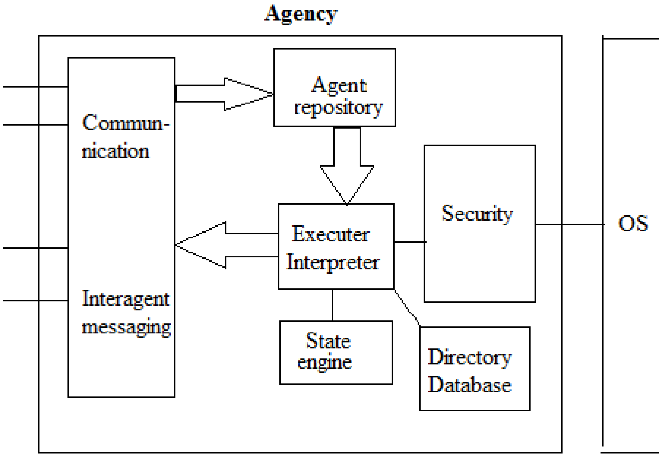

Figure 2: Generic mobile agent system architecture

The generic mobile agent system can have a range of varying components. It needs a communication module that handles incoming and outgoing agents, as well as the messaging between non-local agents. It has a repository that performs authentication, sets priorities and queues up agents for later execution. The executing module has an interpreter and can sometimes run agents written in different languages. The state engine contains the current state of the agency and can have some kind of rule or inference engine that decides what to do with the agents [4]. It also handles local inter-agent communication. There is also some kind of database or directory where data are stored or retrieved by the agents. The security module acts as a kind of sandbox that keeps track of what the agents are allowed to do. It also monitors the agency. There can, of course, be security functionality in the other modules too, such as encryption in the communication module.

4.2 Ant-Like Agents

In the case of distributed retrieval, ant-agents crawl the network to find the relevant documents. These ant-agents travel over the network by moving from one host to another host and mark the visited host by changing a numerical value which are locally stored on these hosts. These values are known as marker. Software agents travel over the network starting from the client host to next host in order to search the desired information [1]. Here the question arises that how to specify the information which has been searched? How the "ant" agent could credit "pheromone-like" markers? These markers can be viewed as a collective memory of paths leading to the relevant sites. Since the marked paths evolve with the global trend of the agent movements, this behavior-based mapping of the network is well adapted to inconsistent networks such as peer-to-peer networks [3]. We have to do several travels between the user's computer and the information sources in our distributed CBIR context. Ant-algorithm seems to be a good solution for learning the relevant paths through the dynamics of active learning.

5. Conclusion

In this paper a new approach for image retrieval is proposed It provides an overview of the Distributed Content based Image Retrieval System by using mobile agent technology, which gives us an efficient and faster image retrieval of images over a huge metadata network. Also, a new active learning strategy for searching images over networks is discussed.

References

[1] David Picard, Matthieu Cord, and Arnaud Revel, "Image Retrieval Over Networks: Active Learning Using Ant Algorithm" IEEE transactions on multimedia, vol. 10, no. 7, november 2008
[2] Stefano Berretti, Alberto Del Bimbo, Pietro Pala," Merging Results for Distributed Content Based Image Retrieval" Multimedia Tools Applicat., vol. 24, no. 3, pp. 215–232, 2004
[3] D. Picard, A. Revel , 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
[4] 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
[5] I. 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.
[6] V. Roth, U. Pinsdorf, and J. Peters, "A distributed content-based search engine based on mobile code," in SAC '05: Proc. 2005 ACM Symposiumon Applied Computing, New York, 2005, pp. 66–73.
[7] D. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J.Comput. Vis., vol. 20, pp. 91–110, 2003.
[8] J. Cho and S. Roy, "Impact of search engines on page popularity," in WWW'04:Proc. 13th Int. Conf. WorldWideWeb, New York, 2004, pp.20–29.
[9] V.G Kottawar "Uncertainty Based Sampling Approach for Relevence Feedback in Content Based Image Retrieval"
[10] S. Santini, A. Gupta, and R. Jain, "Emergent semantics through interaction in image databases," IEEE Trans. Knowl. Data Eng., vol. 13, no. 3, pp. 337–351, 2001.
[11] F. Sebastiani, "Machine learning in automated text categorization," ACMComput.Surv., vol. 34, no. 1, pp. 1–47, 2002.
[12] Parineeth M Reddy, "Mobile Agents Intelligent Assistants on the Internet", In Resonance July 2002