Université Libanaise
École Doctorale
Sciences et Technologies

Univeristy of
Cagliari
Italy

Co-supervision thesis
For the degree of Doctor
University of Cagliari - Italy
and
Lebanese University - Lebanon
(The Doctoral School of Sciences and Technology)

Speciality : Computer Sciences

Presented and defended by

Abbas Akram Khorsheed

19 December 2014

Automatic Image Annotation using Image Clustering in Multi – Agent Society

Jury Members

Mr. Hussein CHIBLE, Professor - Lebanese University	Director
Mr. Luigi Raffo, Professor - University of Cagliari	Director
Mr. Giorgio Giacinto, Professor - University of Cagliari	Co-Director
Mr. Abbass Hijazi, Professor - Lebanese University	Examiner
Mr. Luca Didaci, Professor - University of Cagliari	Reviewer
Mr. Kifah TooT, Professor - Lebanese University	Reviewer

Acknowledgment

First of all, I would like to say Alhamdulillah, for giving me the strength and health to do this project work until it was done. I am heartily thankful to my supervisors, Prof. Giorgio Giacinto, Prof. Hussein Chible and Prof. Luigi Raffo, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

Dedication

I would like to express my gratitude for all the support, encouragement and help that I have received from: SPIRIT OF MY FATHER, My mother, friends, colleagues at the PhD school (Lebanese university and university of Cagliari). Without whom I could not have completed this thesis.

Author’s Declaration

I hereby declare that I am the sole author of this Thesis. I authorize the Lebanese University and the University of Cagliari to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize both universities to reproduce this thesis or dissertation by photocopying or by other means, in total or in part.

Signature:

January 10, 2015

Abstract

The rapid growth of the internet provides tremendous resource for information in different domains (text, image, voice, and many others). This growth introduces new challenge to hit an exact match due to huge number of document returned by search engines where millions of items can be returned for certain subject. Images have been important resources for information, and billions of images are searched to fulfill user demands, which face the mentioned challenge. Automatic image annotation is a promising methodology for image retrieval. However most current annotation models are not yet sophisticated enough to produce high quality annotations. This thesis presents online intelligent indexing for image repositories based on their contents, although content based indexing and retrieving systems have been introduced, this thesis is adding an intelligent technique to re-index images upon better understanding for its composed concepts. Collaborative Agent scheme has been developed to promote objects of an image to concepts and re-index it according to domain specifications. Also this thesis presents automatic annotation system based on the interaction between intelligent agents. Agent interaction is synonym to socialization behavior dominating Agent society. The presented system is exploiting knowledge evolution revenue due to the socialization to charge up the annotation process.

Key Words: Image annotation, image retrieval system, multi agent system, agent society.

Contents

Chapter One: Introduction

1.1. Background

Recent advances of the technology in digital imaging, broadband networking and digital storage devices make it possible to easily generate, transmit, manipulate and store large numbers of digital images and documents. As a result, image databases have become widespread in many areas such as art gallery and museum management, architectural and engineering design, interior design, remote sensing and management systems, weather forecasting, fabric and fashion design, trademark and copyright database management, law enforcement, criminal investigation picture archiving and communication systems. Furthermore, the rapid growth of the World Wide Web has led to the formation of a very large but disorganized, publicly available image collection. Recent studies show that there are 180 million digital images on publicly indexable Web and millions of new images being produced every day.

Retrieving specific digital images from large resources, such as the image repositories spread all over the internet, has become an area of wide interest nowadays. Among image retrieval approaches, text based retrieval is widely used as it has been commercialized already. But it is not effective as it involves time consuming text annotation process. Also there is difference in understanding of image content which affects image labeling process. Content based image retrieval (CBIR) is another method of retrieving images from large image resources, which has been found to be very effective. CBIR involves the use of low-level image features, like, color, texture, shape, and spatial location, etc. to represent images in terms of their features. To improve existing CBIR performance, it is very important to find effective and efficient feature extraction mechanisms. This research aims to improve the performance of CBIR using texture features.

Texture is one of the most important and prominent properties of an image. It is the surface pattern of objects in the image or the whole image. Texture features effectively describe the distinguishing characteristics between images. After extensively studying existing texture feature descriptors, we have proposed a wrapping based discrete curvelet texture descriptor for future use in CBIR. Discrete curvelet transform is one of the most powerful approaches in capturing edge curves in an image. Related works on curvelet features are also investigated. In this research, we generate a texture features descriptor using wrapping based discrete curvelet transform. This descriptor is used to represent images in a large database in terms of their features and to measure the similarity between images.

1.2. Problem Statement

1- User experience does have valuable information that can be utilized in categorizing images; this due to the tendency to select images that share common visual properties with the query rather than relying on the textual matching; this due having huge redundancy in the results returned by search engines due to the overlapped in image description vector assigned to these images. Google search engine having this type of problem when submitting queries looking for certain image.

Here user experience can add another axis for promoting search engines in retrieving images.

2- Images can be retrieved by topic rather than by matching natural language used to describe the image. Topic is semantic in the hidden space (i.e., latent semantic). Images can be interpreted in different views; this depends tremendously on the culture of the requestor (i.e., users requesting images), and this presents a great potential to annotate images using latent semantic.

1.3. Objective of this thesis

1- Capture user experience when working on images returned by submitting queries to search engines, and utilize captured experience in better annotation process.

2- Build multi-agent platform bound to special search engine (i.e., which is to be designed by this thesis) or to the existing search engines (i.e., Google search engine).

3- Develop knowledge over the Mutli-agent platform in the behalf of image retrieval engine, where experience of users are integrated

4- Enhance the indexing methodologies by presenting meta-indexing descriptors that encapsulate users' experience. Users' experience provides non-semantic relationships between the submitted queries and retrieved images.

5- This thesis also aims to de-noise annotations associated with images; this is to be accomplished by removing redundant concepts (i.e., concepts are visual described by words or interpretation for visual materials within the image) through using LSA (Latent Semantic Analysis).

1.4. Related Work and Literature Review

1- In [1] a model has been proposed to formalize the growth dynamics in social networks; in this model a great attention has been presented to the effect of node behavior and how it affects the behavior of other nodes, and this eventually will affect the growth of the network. In term of knowledge evolution due to socialization; this model has a lot in common with our approach, though it has nothing to do with image retrieval system.

   The key similarity is:

   - The behavior-awareness where the interaction of node (i.e., the co-author s) with certain events (e.g., papers) is to be realized as a potential relationship among those nodes. In fact this approach develops knowledge at the network level which helps increasing growth factor of social network and eventually, the productivity of such a network.

2- In [2] an ontological approach was presented to accomplish computing model aimed to annotate images on two levels: *Image Annotation* and *Annotation of Annotation*; this model is the focus presented on query for annotations using National Cancer Institute's Cancer Biomedical Informatics Grid's (CaBIG) Annotation and Image Markup (AIM) project.

AIM project defines ontology of annotation and image markup, a UML information model and provides the extensible markup language (XML) artifacts for creating them. A long term vision of AIM project is for large collections of annotations to be created in conjunction with the already large collections of clinical and research medical images. This will allow query of annotation, not only for retrieval of relevant images, but for correlation of image observations and their characteristics with biomedical data including genomic expression. In this paper many concepts are coherent with what we presented in our work in the area of retrieving images based on accompanied annotation, but this approach does not introduce autonomous annotation in any context and it does not consider behavior of image requesters; this lack developing knowledge and leaves no room for innovations come up be using autonomous ontology.

The presented model exploits annotations to build semantic network among images while our work provides autonomous annotation schema based on the behavioral interpretation of the user. AIM project can be integrated with what we are presenting to provide consistent ontological environment for image retrieving and annotations.

3- The same annotation context is presented by [3], where it depends on the retrieval and extraction of knowledge from the resources available on the global net. Global resources are inferred for knowledge regarding certain concepts through the use of collaborative system. This work has much in common with what we are presenting in this thesis in term of automatic annotation based on collaboration of multiple sources, but it does not consider the experience of the user which it holds crucial information to annotate certain images.

4- In [4] a novel system is presented to exploit the format of multimedia sharing web sites in order to discover the underlying structure; this has been used later for more sophisticated mining for these sites to infer knowledge about certain images. Again, we have many attributes in common with these approaches but still the effect of the behavioral responses of the users absent.

5- In [5] a study for establishing stable architecture for socialization is conducted and the outcome was a conclusion that in a society of agents; there are three main parameters that enforce the stabilization of the architecture; these are: take on roles, play roles and locate in some society organization at all time. In our proposal, the society composed by agents is maintained stable by strict discipline through which roles are fairly distributed and all agents are capable of playing these roles by accurate interpretation of client behavior, furthermore, we adopt fixed organizational distribution of the agents which sustain the stability. In our proposal, the specification of the problem domain has different characterization due to the potential tendency toward clusterization on two different levels: host level and network level.

6- In [6] a study had submitted to address the fault assumption of regarding multi-agent systems as single learning system which is wrong assumption due to intuitive tendency to introduce social activity with neighbors other than communicating other far agents; this dual capabilities of an agent’s referencing: self-referential and social-referential has presented in this work as a bi-referential model, in which each referencing capability is implemented by an evolutional computation method of classifier system.

In our referential model the evaluation function is global and updated on the fly by delivering knowledge to central repository which holds the annotation for images. The annotations are revealed and referenced based on confidence degree assigned to that annotation. In our referential model, the behavior of the evaluation function is dynamic due to the continues changing of confidence degree of annotation; this is due to activities produced by the client clusterization behavior (i.e., self-referential model).

7- In [7], an interactive query for images' content by semantic descriptors is presented; this effort introduced a distributed content-based image query system (DCBIQ) based on the WWW. A model was proposed to integrate knowledge from image processing, semantic descriptor, multi-agent, and WWW navigation. Again in this model the image content plays the essential role in describing the image, thus low level extraction methodologies are more important than the opinion of the social communities which are using it.

In our proposal, the knowledge obtained by social interaction is more important than low level features like colors, textures or spatial relationships, and even semantic interpretation of image contents is not important as the social opinion about the image and its relation to other images or domains.

8- In [8] an attractive model is presented where a web-based image digital library is proposed; in this library agent system was used to traverse part of the web page looking for images that fit certain criteria. The methodology used by the agent is by detecting URL within web pages that refers to images, and when such URL is encountered then the text accompanied that image is inferred for correlation with other features such as topic name, domain that this image falls in or any other matching criteria. In our proposal the same ontology for allocating text accompanied the image is used as the following matched methodologies:

$$\forall image\, \exists tag\, \exists txt\ \big(\big(Presenting(image, tag)\ AND\ Asso(tag, txt)\big) \\ \rightarrow Asso(image, txt)\ AND\ Select(txt)\big)$$

$$\exists paragraph\, \exists hyper\ \big(\big(hyper\ \in paragraph\big) \\ \rightarrow Select(paragraph)\big)$$

$$\exists page\, \exists image\ \ \exists!\, title\ \big(\big(image\ \in page\big)\ \ AND\ has(title, page\big) \\ \rightarrow Associate(image, title) AND\ Select(title)\big)$$

Where

Asso: Association

hyper: hyper link reference

tag: HTML tag

txt: text accompanied the image (i.e., basically is the annotation).

The key difference of our approach is we don’t design mining agent which is responsible on inferring web pages, but we exploit Google search APIs which are published over the web. The only web page we are after is the results of the Google search APIs and don’t investigate individual pages.

1.5. Required Platform

The automatic annotation system proposed by this thesis is an enterprise application that needs an http server which has been selected to be Linux based server with Apache/TomCat ver 7.0.41 installed. TomCat is the web server needed to host the proposed automatic annotation system; it has to support HTML5 due to massive inclusion of tools and facilities provided by HTML5. After all, client side agents are built using java script, and this is a real challenge on different aspects for example JavaScript is a single threaded programming language while agent programming demand the support of multi-threading environment; here HTML5 plays the main role to overcome such challenges due to the technologies it provides such as the WebWorker which enhance the computation power of the environment.

A registration for Google API is required to access Google image repositories to grant our designed system the ability to retrieve images automatically, Google search API is needed to deliver search results as JSON (Java Script Object Notation) objects; JSON objects are easy to be interpreted due to the fact it is an XML based tagged objects.

This proposal needs also to install latest version of JADE (Java Agent Development Environment); this is crucial to create agent platform and manage it due to the wide range of tools and utilities provided by JADE such as sniffers and remote management.

Chapter Two: Image & Annotation

2.1. *Introduction*

This chapter is dedicated to present essential terminologies and tools deployed to implement multi-agent based automatic annotation system; the presented topics are constructing the platform over which the Automatic Image Annotation system is implemented.

After this introduction a brief reviews of different approaches introduced so far in the domain of content-based image retrieval. Image contents can be conceptualized in different approaches based on the tools used to reveal image concepts, for example images can be transferred into isomorphic low level domain and introduce new conceptualization domain; this is where wavelet, Fourier transformation, spatial analysis and other analogous mathematical analysis tools, are deployed.

Higher level of abstraction is also introduced in this chapter; this is the semantic level where low level features do not provide much information about visual objects as the semantic level.

2.2. *Content Based Image Retrieval approaches*

Annotation stands for the process of describing images, and retrieval stands for the process of finding images. The two major approaches to image retrieval are content-based image retrieval that analyzes the actual image data, and metadata-based approach that retrieves images based on human-annotated metadata. Also relevance feedback has been used in image retrieval complementing text-based systems [9].

The metadata that describes images can be roughly divided in two parts. One part concerns the concepts that give information about the creator of the image, tools used in the process of creating the image, art style of the image and the artist, price, and other explicit properties of the image. the other part describes what is actually in the image, the implicit properties that can be understood by perception the image itself [9].

A number of low-level image features can be extracted from an image. Detailed studies on image features are presented in [8]. Some commonly used low-level image features in recent literature include the application of color, texture, shape, spatial location, etc.

Some CBIR approaches use a combination of more than one low-level feature to improve retrieval performance. In this section we briefly describe the features used in recent CBIR researches and their impacts.

Color is one of the most prominent visible properties of an image and each pixel of image contains a different value for color. As human vision perception can easily distinguish different colors, application of color features has widely been accepted in numerous CBIR applications. Before generation of a color descriptor, it is necessary to define a suitable color space. From the recent literature, we find HSV or HSL or HSB, YCrCb, CIE-L*u*v*, CIE-L*a*b* are popularly used in CBIR [10].

Various color spaces have already been developed and used for different purposes in image processing. In some retrieval approaches, color features are combined with texture features to obtain a better performance. For convenience in color feature extraction process, color space conversion processes have been introduced. The transformation from RGB to HSV, HSB or HSL space is described in many efforts. Among the color spaces, HSV is more useful in measuring perceptual similarity [10].

Commonly used color descriptors include the use of the color histogram, color moments, the color coherence vector, and the color correlogram. Sometimes more than one color descriptors is used for image [9].

2.3. Image Query Paradigms

Image retrieval system is generally composed of indexing, searching and query builder, where users enter their query that represents a conceptualization of the required image; this is the initiation of the retrieving process. The crucial point is unification of the representation of images in the indexing and query phases; this due to the fact that queries are keys used to search the database and it should be correlated to the keys used to index images within the repository [9][11].

Anyway, based on what concepts used to index images, different retrieving methodologies are deployed, and it is generally categorized into the following categories:

2.3.1. Query by Text

The text-based approaches are based on the idea of storing a key-word, a set of keywords, or textual description of the image content, created and entered by a human annotator, in addition to a pointer to the location of the raw image data. Image retrieval is then shifted to standard database management capabilities combined with information retrieval techniques. Some commercial image search engines, such as Google's image search and Lycos Multimedia search can be categorized as text-based engines, despite the fact that Google is developing new image retrieval schemes. These systems extract textual annotations of images from their file names and surrounding text in web pages. Usually, it is easier to implement an image search engine base on keywords or full-text descriptions than on the image content provided that image annotations can be obtained. The query processing of such search engines is typically very fast due to the existing efficient database management technologies [10][11].

2.3.2. Query by Image

Query by image allows the user to provide an example image as a representation of their query. The example can be either an image selected externally or internally from the system, the characteristics of this query is convoluted by external pictorial example and query by internal pictorial example. Query by external pictorial example permits the user to submit their own image to the system and is generally perceived as the simplest approach to query formulation. However, it is based on the assumption that the user has a suitable representative image to use as the bases of their query [11]. Query by internal pictorial example is query by browsing where the user selects an image from the system database. All the images contained in the database are presented or a selection is generated randomly in a n-dimensional matrix. Similar to the problems associated with query by browsing, the main disadvantages of this method are providing suitable access mechanisms to retrieve the internal example and the size of the image collection the user is willing to search through in order to find a suitable image [10],[11].

2.3.3. Query by Painting

Query by painting allows the user to manually specify the percentage or the distribution of color values. For example, the user is able to specify the percentages of color within a composition, such as 50% green, 25% yellow, and 25% red. Similarly, the user is able to specify the coordinates of each color; this is done in the query canvas [9].

2.3.4. Query by Sketch

Query by sketch allows the user to draw a sketch of the desired image by combining several features commonly found in computer graphic applications. The sketch represents a template of either a completed object or scene.

Queries formulated by this approach are simplistic, relatively crude sketches of the desired query image and that the tool has a limited functionality for expressing more complex image queries. This approach stressed that drawing a shape query is inherently time consuming and requires a certain modicum of artistic ability. Similarly, the effectiveness of shape matching features are highly sensitive to noise and pixel arrangement in the query image [9],[11].

2.4. Image Annotation

Image annotation is the process of associating metadata with a digital image. The annotations might provide data regarding where, how, and when the image was collected, or the annotations could provide semantic information about what the image data actually means. Performing this semantic annotation by applying informative terms or tags to an image or image region provides information that is difficult to infer from the image data itself. The problem is that while images contain large amounts of data, the meaning of this data not explicit. Semantic annotation provides context for image data and allows meaning to be easily accessed. This, in turn, allows large image datasets to be more efficiently stored, queried, and analyzed [11],[12],[13].

Image retrieval has been widely studied from two paradigms: content-based and annotation-based image retrieval [12].The former requires users to formulate a query using an example image. The retrieval system then returns the set of images that best match the given example based on visual content, i.e., low-level features like color and texture. Annotation-based image retrieval, on the other hand, enables users to naturally formulate semantic queries using textual keywords. In order to support this retrieval paradigm, many automatic image annotation techniques have been proposed which assign a few relevant keywords to an un-annotated image to describe its visual content for image indexing and retrieval [10],[12].

The keywords are often derived from a well-annotated image collection and the number of keywords is often limited to few hundreds.

2.5. Automatic Image Annotation System

Automatic image annotation refers to the task of assigning a few relevant keywords to an un-annotated image to describe its visual content; the keywords are then indexed and used to retrieve images [12].These keywords are often derived from a well-annotated image collection, and the latter serves as training examples for automatic image annotation. Regions in an image are assumed to be described using a small vocabulary of blobs. Blobs are generated from low level image features through clustering. The joint probability distribution of textual keywords and blobs is learned from the annotated image collection to compute the probabilities of keywords associating with a test image. A family of image annotation methods, built on nearest-neighbor hypothesis (*i.e.,* visually similar images likely share keywords), are proposed and evaluated in [11]. Given a query image, the *k*-nearest neighbors are retrieved and their associated keywords are transferred to the query image. The accuracy of image annotation can be evaluated based on the correctness of the assigned keywords or through image retrieval by using the assigned annotations. Although image retrieval is often used to evaluate image annotation methods, the key focus of image annotation is to assign images with keywords. The dimensions in matching a textual query with the keyword-annotated images have not been systematically evaluated. In this work, our focus is to evaluate the tag-based image retrieval methods, where annotations in the form of *user assigned tags* are provided. Moreover, in image annotation research, the keywords are carefully selected and the number of keywords is often very small. For instance, the number of keywords selected in the commonly-used image annotation datasets, such as

Corel5K, iarp tc12, and esp game datasets, ranges from 100 to 500 [11],[12],[14].

The two larger datasets Corel30K and psu, containing 31K and 60K images, are annotated with 5,587 and 442 keywords, respectively [12]. On the other hand, social tags are keywords assigned by users not from any controlled vocabulary. For the nus-wide dataset used in this work, consisting of 269K tagged images, there are more than 420K distinct tags.

2.6. Image Annotation Approaches

Many approaches have been proposed to address the annotation task. Three main groups are identified: generative models, discriminative models and nearest neighbor based methods.

- **Generative models** can be further categorized as topic models and mixture models. Topic models annotate images as samples from a specific mixture of topics. Each topic is a distribution over image features and annotation words. Examples of topic models include latent Dirichlet allocation, probabilistic latent semantic analysis, hierarchical Dirichlet processes, and machine translation methods. Mixture models define a joint distribution over image features and annotation keywords. Given a new image, these models compute the conditional probability over keywords given the visual features by normalizing the joint likelihood. A fixed number of mixture components over visual features per keyword can be used, or a mixture model can be defined by using the training images as components over visual features and keywords **[**15],[14],[11].

- **Discriminative models** for keyword prediction have also been proposed. These methods learn a separate classifier for each keyword, and use them to predict whether the test image belongs to the class of images that are annotated with each particular keyword. However, both generative and discriminative models preselect features and do not analyze the differences within features. Feature selection is not a concern either [10],[12],[15].
- **Nearest neighbor** based methods have become more attractive recently since the amount of training data is rapidly increasing, such as using label diffusion over a similarity graph of labeled and unlabeled images, and learning discriminative models in neighborhoods of test images. A nearest-neighbor keyword transfer mechanism was recently introduced. In this method, image annotation is solved as a retrieval problem. Nearest neighbors are determined by the average of several distances (called Joint Equal Contribution, JEC) computed from different visual features. Keywords are then transferred from neighbors to the given image. Elementary color and texture features are tested and compared. Regularization based feature selection is also considered by using keyword similarity. Weights are computed in the "feature level", which means that all histograms within the same feature share one [15].

2.7. Agent Based Image Annotation Scheme

Retrieving an image of huge image repositories depends heavily on the quality of the textual annotation of images, where retrieving an image is a matching process of the query with the annotation. Images are described using meta-data corresponding to its content and the better description for the content leads to better performance in image retrieval. Image annotation is a laborious task that requires consistent domain knowledge; this due to the huge number of images and the wide spectrum of categories in which those images are classified, thus special software is required to overcome this challenge [16].

Software agent is software that is capable on perceiving the environment by conceptualizing events occurred within environment, the conceptualization schemes are defined by domain specific ontologies (software Agent technology will be presented in details in the next chapter).

In Semantic web, a knowledge representation framework has been proposed to enable software agents to share domain knowledge on the web in terms of XML-based (Extensible Markup Language) ontology languages such as RDF/RDFS (Resource Description Framework/Schema) and OWL (Ontology Web Language). The ontology languages provide a well-defined set of relational terms that essential for building domain concepts. They also provide the semantic interoperability at different platforms that allow knowledge exchange in machine-readable format. RDF/RDFS each semantic relation as an information resource in terms of a triple of subject, predicates, and objects [16],[17].

Ontology is regarded as the specification of conceptualization that enables formal definitions about things and states by using terms and relationships between them, thus, in Agent based image retrieval which it is a web application, web pages are converted into concepts by referring to domain-specific ontologies which employ a hierarchical concept structure [16].

2.8. Enterprise Application Architecture

Enterprise Applications (EAs) are generally understood to be on-demand, user-interaction based applications that are meant to be accessed by multiple users, usually from the same organization. Web-based Enterprise Applications (WEAs) imply EAs made available through the Internet. These applications (EAs and WEAs) generally use databases for persistent storage. E-commerce sites (such as Amazon [reached at www.amazon.com] and eBay [reached at www.ebay.com]), banking sites, webmail, online casinos and search engines are some of the many examples of WEAs [18],[19].

WEAs are client-server applications. Making a WEA means implementing the server side of the application as well as what will run on client machines (usually in a client's web browser). This separation does not coincide cleanly with the boundaries of the three layers .A common assumption is that all presentation components should exist on the client side, but this does not take into account server-side decisions about presentation or security [18].

Web applications have become an inseparable part of the Internet. They offer rich interactivity and functionality which would be unthinkable for older style, static web sites.

More and more companies are finding a way to offer their products and services online. For this, they need feature-rich web applications which allow them to register their clients, accept visitors' feedback, accept online payments, ensure that products can be easily found in their database and are properly presented to visitors and easily purchased [18].

More and more sophisticated web applications will be required by businesses over time, hence the requirement for web developers to make their work more productive and to make the web applications they produce more reliable, maintainable and easily scalable.

As a result, a significant number of Java web frameworks have appeared over time. Frameworks shift the focus of development to a higher level by bringing most low-level solutions "out of the box", thoroughly tested and ready for reuse. Many frameworks also bring with them some sort of architectural solution, based on the best practices of web development.

2.9. Java Tools for Enterprise Applications

The main tendency in World Wide Web development has always been towards more functionality, more interactivity and riche user experience. Java, being platform-independent, looked perfect for WWW which spread across continents and computer networks. Java starts supporting enterprise applications by first introducing 'Java applets', which is a small programs downloaded to the client browser and executed there; this has added more interactivity to previously static web pages, this approach has faced degrading due to the need of the privilege required by applet code in accessing the machine. Figure 1 illustrate the process of downloading the applet to client side machine, it is client's responsibility to provide the environment enough to execute the applet [18].

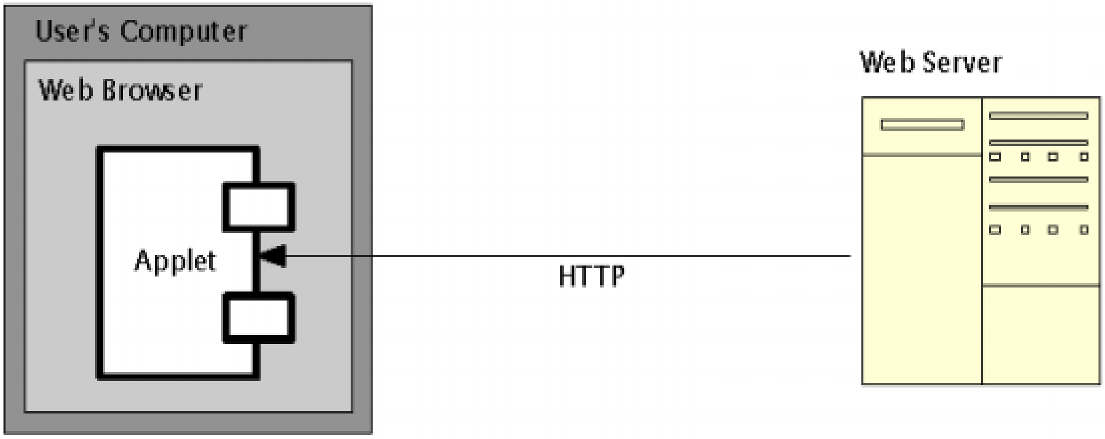


Figure 1: Applet executed in client side machine with support of client machine.

The real success came to Java technologies when they began to work on the server side. Instead of downloading Java programs and running them in a client's browser, Java code could work on the server side and send only results to the client side; this is done by creating HTML pages "on-the-fly" and sending them to the browser [18]. This was exactly the idea implemented in Servlet technology, as it is shown in Figure 2:

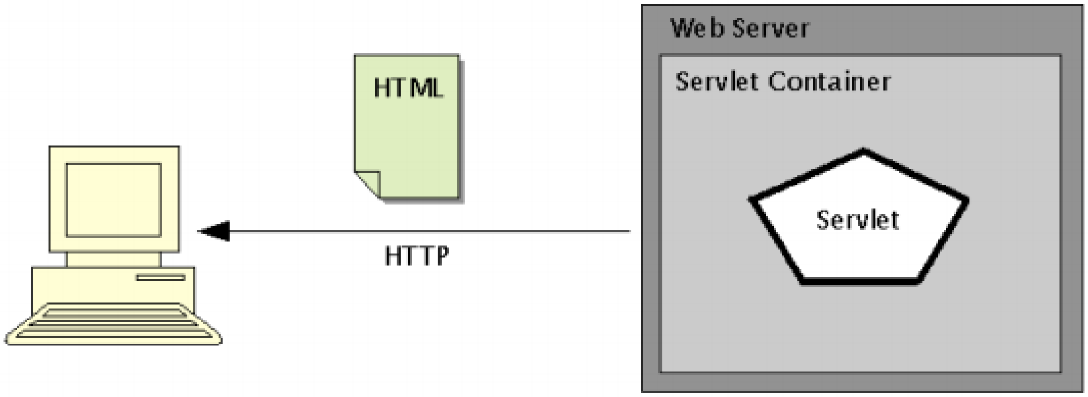


Figure 2: Java servlets are executing at server side and results transferred to client side.

The idea of executing components of the enterprise application on the server side is not new where CGI programs written in languages like Perl or C had already been creating HTML pages dynamically for years. However, Servlets had significant benefits: higher scalability due to multithreading, functionality, readily available for developers in Servlet API and provided by Servlet container, inherent security and power of the Java language as well as all the services of J2EE platform which was beginning to emerge [18],[19].

By using Servlet API, developers could concentrate on the functionality they were creating and leave many trivial tasks to the Servlet container. For example, all the request parameters were ready to use in the ServletRequest object and could be easily retrieved from it, and session management was very easy with all the burden of setting session cookies and most of the URL rewriting carried by the Servlet container [19].

The weak side of Servlets was that all the HTML output had to be created inside of Java code. All the design was tightly embedded into the code-hence the necessity to recompile Servlet with every little change in design. HTML pages created by designers would somehow have to be processed and converted into chunks legible arguments [18].

As web application grew and developed, the whole team had plenty of work to do. In other words, there was low maintainability due to the fact that presentation was embedded into the code. It was natural in this situation to invent an approach which would at least automate the conversion of HTML to be outputted into Servlet code. Indeed, many development teams created their own solutions for this problem, but they all became unnecessary after the emergence of JSP technology.

Java Server Pages: In their essence, JavaServer pages are the same Servlets with the only difference being that developers were allowed not to worry about how to process large amounts of HTML and insert them into Java code. This was done automatically in the process of conversion of a JSP page into a Servlet. Instead of inserting HTML into Java code, the JSP developer was inserting Java code into HTML, using directives [18],[19].

JSP was an invaluable solution for exactly those problems which were most difficult for Servlets – when the page to be sent as a response contained mostly static HTML, with just a small amount of dynamic content. However, when a large amount of Java code was embedded into the page, the mixture of Java and HTML was becoming very difficult to manage.

It is often implied that a web application can be considered as an enterprise level web application if it is distributed over a network and was built using Enterprise Java Beans (EJB) technology.

Figure 3 presents the architecture of Web application which used Java technologies:

2.10. Custom Google Search APIs

Google allows users to embed elements for conducting Web search, local search, image search and others, into their own web pages and applications using JSON (Java Script Object Notation) / Atom Custom Search API (Application Programming Interface) ; this API is used by web application developer to query Google servers and get the results of the search as series of JSON objects; these JSON objects are composed of XML tags which facilitates the interpretation by java script programs [18].

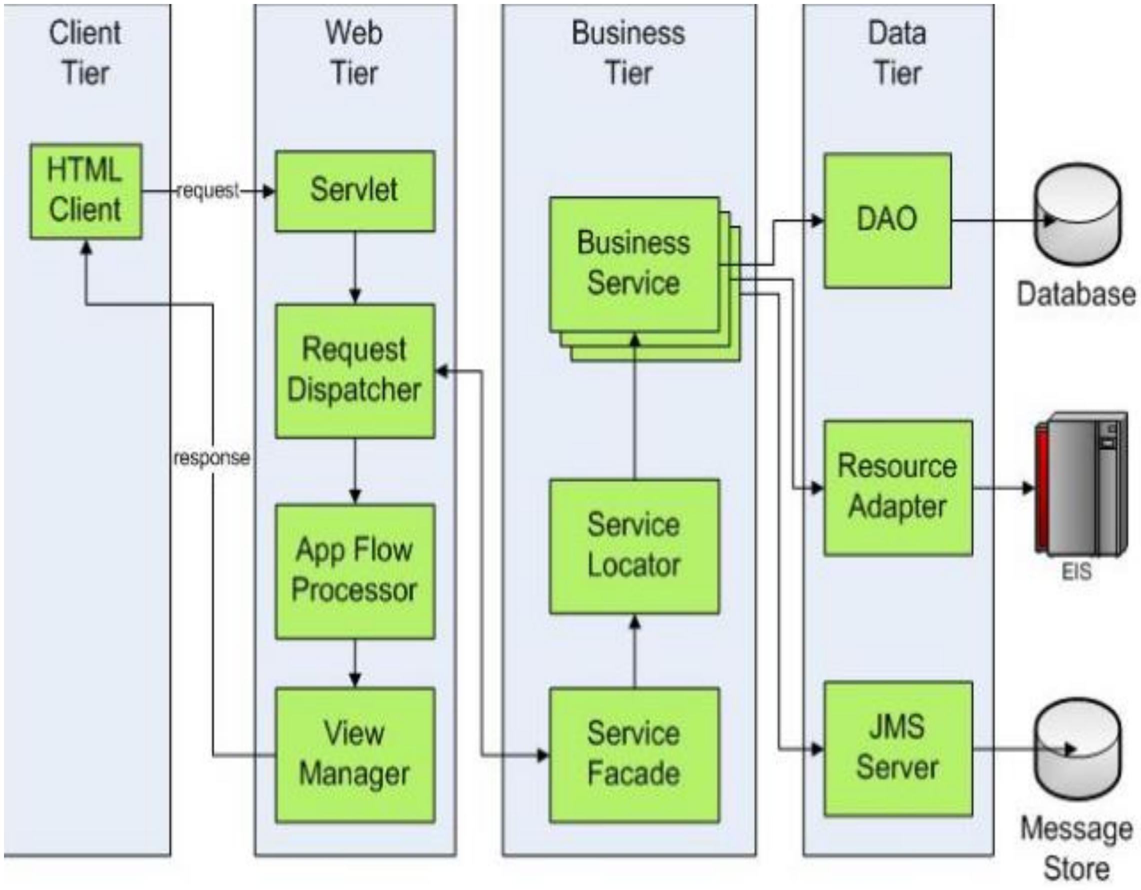


Figure 3: Java based Web Application Architecture.

The template that is used by java script code to call Google API is:

https://www.googleapis.com/customsearch/v1?parameters

Google Search APIs are not free to be used by web application, they have pricing plan in using these APIs, and thus, a Google Account is needed to sign into Google control panel and get a registration key. Web applications have to use registration key before being able to dispatch this service.

2.11. SINGULAR VALUE DECOMPOSITION

Images can be represented be series of features or concepts that describe the contents of each image; here an analysis tool is needed to determine the importance (i.e., weight) of individual and structural concept in identifying each image; this is due to the fact that images contain dominant concepts (i.e., visual objects within the image) and miner concepts (i.e., visual objects that do not compose the meaning of the image). Anyway, images can be represented by matrix of features or concepts each column with this matrix is the image and each row represent certain feature or concept; an analysis tool for this matrix is a crucial factor in determining indexing scheme for those images, moreover, the selected analysis tool should capable of performing in the natural language domain; this is due to the fact that image annotation is a description for images in natural language.

Singular Value Decomposition is a mathematical model in linear algebra that decomposes a matrix into three factor matrices; this is to reduce the complexity in manipulating systems described by large matrices. SVD is a much more complex approach than other decomposition methodologies, yet it is more worthy in term of semantic gained by the decomposed components and how fast is the calculation to turn over, where, once the original matrix has been decomposed, operations on the matrix are rather quick [20].

SVD is an extension of Latent Semantic Analysis (LSA). LSA is a method of analyzing a group of terms and documents to find relationships between the terms (i.e., represented by vectors) and the documents hold these terms;

this is beside the need for efficient weighting function to reflect the importance degree of terms vectors in related to the documents [20].

SVD takes as input a matrix of size m x n. This matrix is decomposed into three different matrices: $U\sum V_T$. The output of these three matrices is a relationship to the original matrix.

SVD has designed to reduce a dataset containing a large number of values to a dataset containing significantly fewer values, but which still contains a large fraction of the variability present in the original data [21],[22],[20].

$$A = U\Sigma V^T \qquad \text{---eq.1}$$

Where

- $EigenVector(AA^T) \rightarrow Columns(U)$
- $EigenVector(A^TA) \rightarrow Columns(V)$
- $EigenValue(A^TA)\ OR\ EigenValue(AA^T) \ \rightarrow \Sigma$

the first structure is the single pattern that represent the most variance in the data, after all, SVD is an orthogonal analysis for dataset, U is composed of eigenvectors of the variance-covariance matrix of the data, where the first eigenvector points to the direction which holds the most variability produced by all other vectors jointly. U is an orthogonal matrix where all its structures are mutually uncorrelated. Eignevalues are representing scalar variance of corresponding eigenvectors; this way total variation exhibited by the data is the sum of all eigenvalues and singular values are the square root of the eigenvalues [22].

U and V_T are both orthogonal matrices and ∑ is a diagonal matrix. These three matrices are further identified as:

- U is the right singular vectors
    - Sized: *m* x *r*
- V_T is the left singular vectors
    - Sized: *r* x *r*
- ∑ is the singular values
    - Sized: *n* x *r*

For SVD, "*r*" is considered to be the rank of the matrix, which is the minimum of the original matrix dimensions. In general, all matrices must be full rank, meaning *r* is equal to either *m* or *n*. In this case, we can exactly reconstruct the original matrix given the three decomposed matrices.

SVD has an interesting property that allows for less than full rank of the matrices to approximate the original matrix. For the purposes of this project and LSA, we don't want the original data back (perfect reconstruction of the original matrix), but rather we want underlying relationships in the movie data [21].

Figure 4 below shows the breakdown of matrix reduction. Instead of all three matrices having a full rank of *r*, we can reduce all three matrices based on a common factor *k*. This is called rank reduction. The arrows in the figure show which direction each matrix reduces. Since ∑ is ordered from largest value in the first cell, to smallest value in the last cell, rank reduction on the matrices will remove those components that are contributing the least to the overall model.

Once the matrices are reduced, we recompose the new matrix A_k, which then gives us information about the underlying relations of the cells in matrix A, instead of the original data back [22].

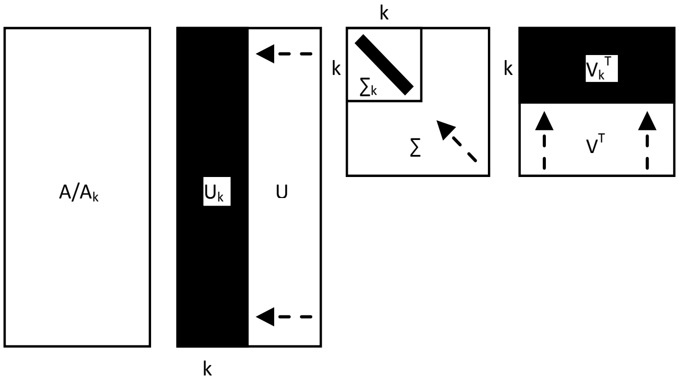


Figure 4: SVD Matrix reduction *k* .

2.12. Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the meaning of words. Meaning is estimated using statistical computations applied to a large corpus of text [22]. In text mining model a document is represented as a vector where each dimension corresponds to a separate feature from the document. A feature could be a term or any other unit that is a representative attribute of the documents in the given corpus. If a feature occurs in the document, its value in the vector is non-zero. An important step in LSA is to transform the term-document vector space into a concept-document and document-concept vector space. By reducing the number of concepts, the documents and their terms are projected into a lower-dimension concept space. As a consequence, new and previously latent relations will arise between documents and terms. In order to apply LSA, term-document matrix A is generating from the given corpus. Then, the singular-value decomposition (SVD) is applied to the resultant are, as it has been said in previous sections, are three matrices that represent the latent semantic in words (i.e., that represents the rows of the matrix) and documents (i.e., vectors that represent the columns) [21],[22],[23].

The corpus embodies a set of mutual constraints that largely determine the semantic similarity of words and sets of words. These constraints can be solved using linear algebra methods, in particular, singular value decomposition [22],[23].

LSA has been shown to reflect human knowledge in a variety of ways. For example, LSA measures correlate highly with humans' scores on standard vocabulary and subject matter tests; it mimics human word sorting and category judgments; it simulates word-word and passage-word lexical priming data; and it accurately estimates passage coherence [22], [15].

The core processing in LSA is to decompose A using SVD (Singular Value Decomposition);

In LSA data is subjected to two-part transformation [20]:

1- The word frequency (+1) in each cell is converted to its log.
2- The information-theoretic measure, entropy, of each word is computed as ($P \log P$) over all entries in its row and each cell entry then divided by the row entropy value.

The mentioned two parts transformation is crucial to build the semantic space of the system modeled by the matrix, where, words or features are weighted as an estimate of its importance in the passage [22],[23].

The geometrical interpretation of LSA introduces an excellent understanding scheme, where U_k and V_k matrices respectively are taken as coordinates of points representing the documents and terms in a k-dimensional space. With appropriate rescaling of the axes, by quantities related to the associated diagonal values of Σ, dot products between points (i.e., vectors) in the space can be used to compare the corresponding objects. In this decomposition, two terms can compared, two documents or a document with a terms also are measured [22],[23].

One important feature of LSA is the generalization to unseen objects, i.e. one can compute the representation of objects that did not appear in the original analysis. For example if there is a query expression composed of terms from the vocabulary. Using linear algebra, it is easy to show that the query can be represented as the centroid of its corresponding term points.

The main advantages of LSA are [20]:

- Synonymy: Synonymy refers to the fact that two or more different words have the same or similar meaning, such as movie and film. A traditional vector space model based Information Retrieval (IR) system cannot retrieve documents discussing the topic of a given query unless they have common terms (due to the limitation of exact matching) however mapping the query and the document to the concept space, they are both likely to be represented by a similar weighted combination of the SVD variables, hence the sine of the two vectors can be small.
- Polysemy: Polysemy refers to the fact that one word has multiple meaning, such as the word bank. The precision of the retrieval can be

reduced significantly, if the queries have a large number of polysemous words. Applying LSA to the query the rare and less important usages of certain terms can be filtered out, thereby increasing the precision of the search.

- Term dependence: the vector space model relies on the bag-of-words concept, i.e. the terms constituting the documents are completely independent from each other (they are orthogonal basis of the vector space), and however it is well known that there are strong correlations between terms. Term associations, for example can be exploited by adding phrases composed of two or more words to the vocabulary. LSA offers a more intuitive solution through the embedding of word-word, document-document and word-document correlations into the reduced LSA factor based representation.

2.13. LSA as A theory of Learning, Data mining, Memory and Knowledge

Basically, the input matrix to LSA is consisting of rows representing unitary event types and columns representing contexts in which instances of the event types appear; this matrix is subjected to analysis by statistical technique which is the SVD, as it has been mentioned earlier in this chapter. The output of the analysis is a re-representation of both the event and individual context as points or vectors in high dimensional abstract space; this allows measuring similarity between all pairs consisting of either event types or contexts.

The data that LSA starts with are raw, first-order local associations between a stimulus and other temporally contiguous stimuli, or, equivalently, as associations between stimuli and the contexts or episodes in which they occur. The stimuli or event types may be thought of as unitary chunks or perception or memory.

Table 1 presents the matrix initiated by LSA before starting the SVD to analyze it in term of variance, in other words finding the basis of this matrix.

Table 1: LSA – General Matrix Scheme.

Word	Doc_1	Doc_2	Doc_3	…………………………	Doc_n
$Word_1$	$Occur_1$	$Occur_1$	$Occur_1$	…………………………	$Occur_1$
$Word_2$	$Occur_2$	$Occur_2$	$Occur_2$	…………………………	$Occur_2$
$Word_3$	$Occur_3$	$Occur_3$	$Occur_3$	…………………………	$Occur_3$
⋮	⋮	⋮	⋮	⋮	⋮
$Word_m$	$Occur_m$	$Occur_m$	$Occur_m$		$Occur_m$

$$A = U\Sigma V^T = \sum_{i=1}^{N} \sigma_i u_i v_i^T$$

$$\xrightarrow{yields} A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \sigma_3 u_3 v_3^T + \ldots + \sigma_N u_N v_N^T$$

And

σ_i *is the vairance and it is equal to* $\sqrt{\lambda}$

Σ is a diagonal matrix of σ_i and reflects the variance of the latent semantic in attributes domain (i.e., words) and the semantic in the documents domain. Despite the fact that SVD comes out to dimensionality reduction, there is another beneficial outcome which is knowledge condensing vectors; this is the vector that results on maximum knowledge about the document.

In data mining application, the initial matrix is an array of objects and attributes. The number of rows, n, of the matrix is typically very large, in the range $10^3 - 10^9$. The number of columns, m, is also large $10 - 10^4$. However, this is large enough for many of the difficulties of working in high dimension to play a significant role.

Singular value decomposition is an efficient tools used to reduce high dimensionality to lower degree; this results in lowering the computation power in doing the calculations. Basically SVD power introduces new conceptualization to problem domain, where all concepts are transformed to the space as vectors and system dynamic is obviously captured, the weighting of attributes that urge system dynamic can be revealed [22].

Chapter Three: Intelligent Software Java Agent System

3.1 Definition of Agent

The term ‘agent’, or software agent, has found its way into a number of technologies and has been widely used, for example, in artificial intelligence, databases, operating systems and computer networks literature. Although there is no single definition of an agent, all definitions agree that an agent is essentially a special software component that has autonomy that provides an interoperable interface to an arbitrary system and/or behaves like a human agent, working for some clients in pursuit of its own agenda. Even if an agent system can be based on a solitary agent working within an environment and if necessary interacting with its users, usually they consist of multiple agents. These multi-agent systems (MAS) can model complex systems and introduce the possibility of agents having common or conflict goals. These agents may interact with each other both indirectly (by acting on the environment) or directly (via communication and negotiation). Agents may decide to cooperate for mutual benefit or may compete to serve their own interests [24],[25].

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. Human agent has eyes, ears, and other organs for sensors and hands, legs, mouth, and other body parts for effectors. A robotic agent substitutes cameras and infrared range finders for sensors and various motors for effectors [25]. Figure 5 illustrates this simple idea.

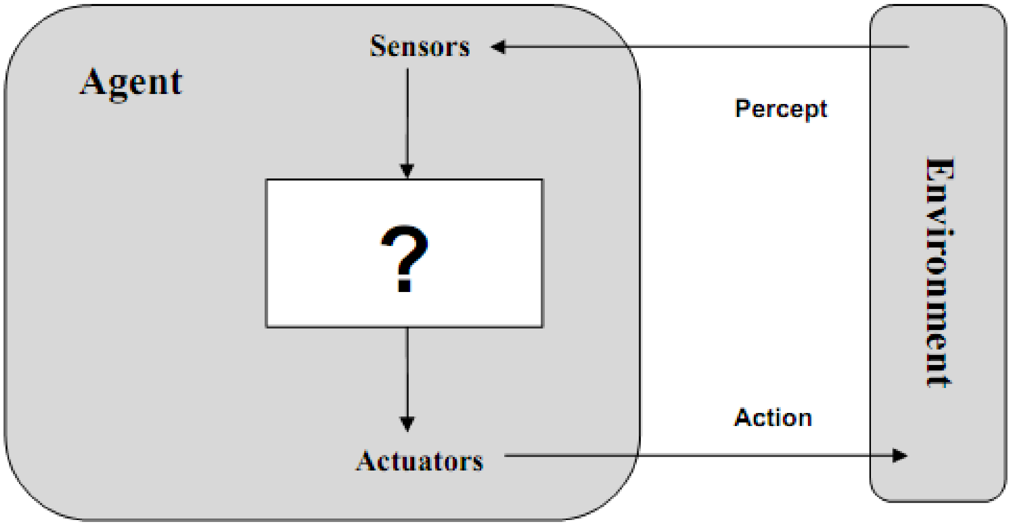
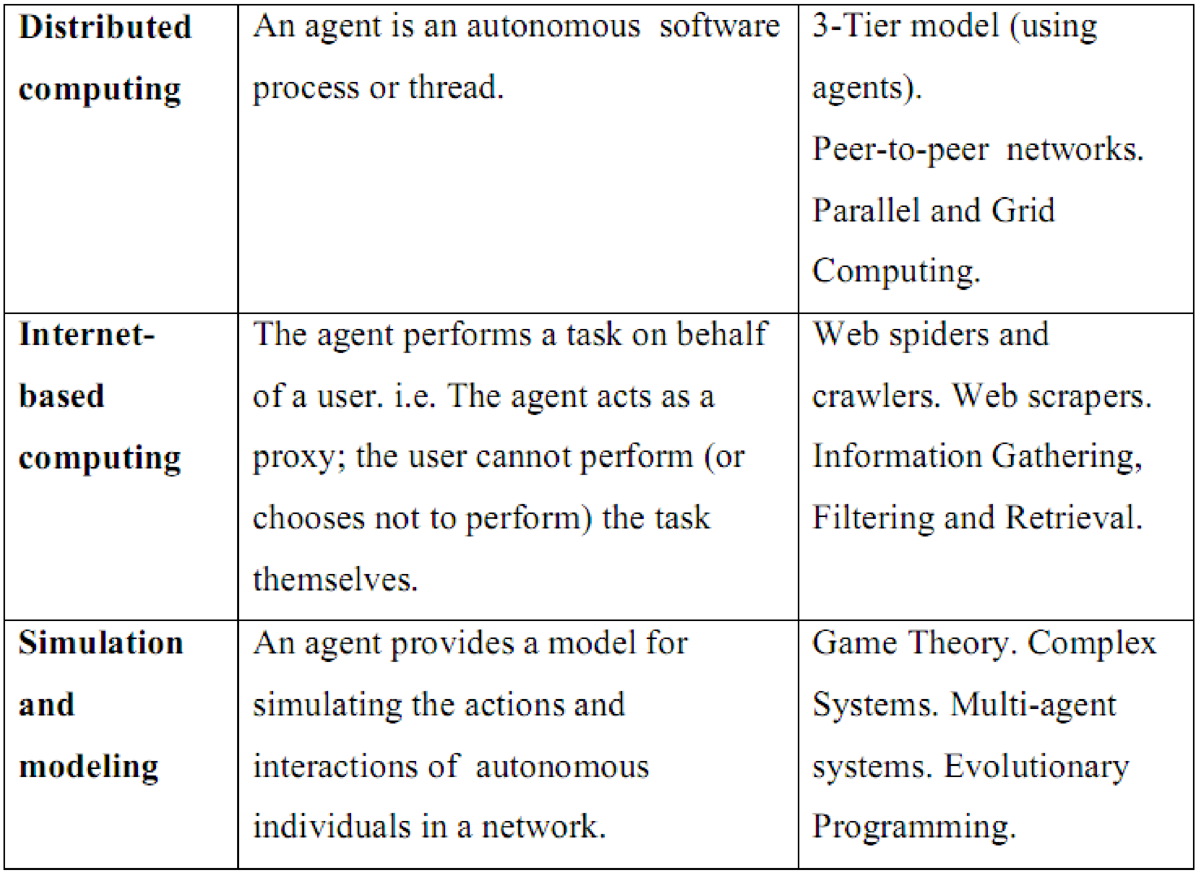


Figure 5: Agents Interact with Environments through Sensors and Actuators.

There is much confusion over what people mean by an "agent " Table 2 lists several perspectives for the meaning of the term "agent"[27].

Table 2: Various Perspectives on the Meaning of the Term "Agent".

Perspective	Key Ideas	Some Application Areas
Artificial intelligence	An agent is embodied (i.e. situated) in an environment and makes it own decisions. It perceives the environment through sensors and acts on the environment through actuators.	Intelligent Agents . Intelligent Systems. Robotics.



Distributed computing	An agent is an autonomous software process or thread.	3-Tier model (using agents). Peer-to-peer networks. Parallel and Grid Computing.
Internet-based computing	The agent performs a task on behalf of a user. i.e. The agent acts as a proxy; the user cannot perform (or chooses not to perform) the task themselves.	Web spiders and crawlers. Web scrapers. Information Gathering, Filtering and Retrieval.
Simulation and modeling	An agent provides a model for simulating the actions and interactions of autonomous individuals in a network.	Game Theory. Complex Systems. Multi-agent systems. Evolutionary Programming.

3.3Properties of Agents

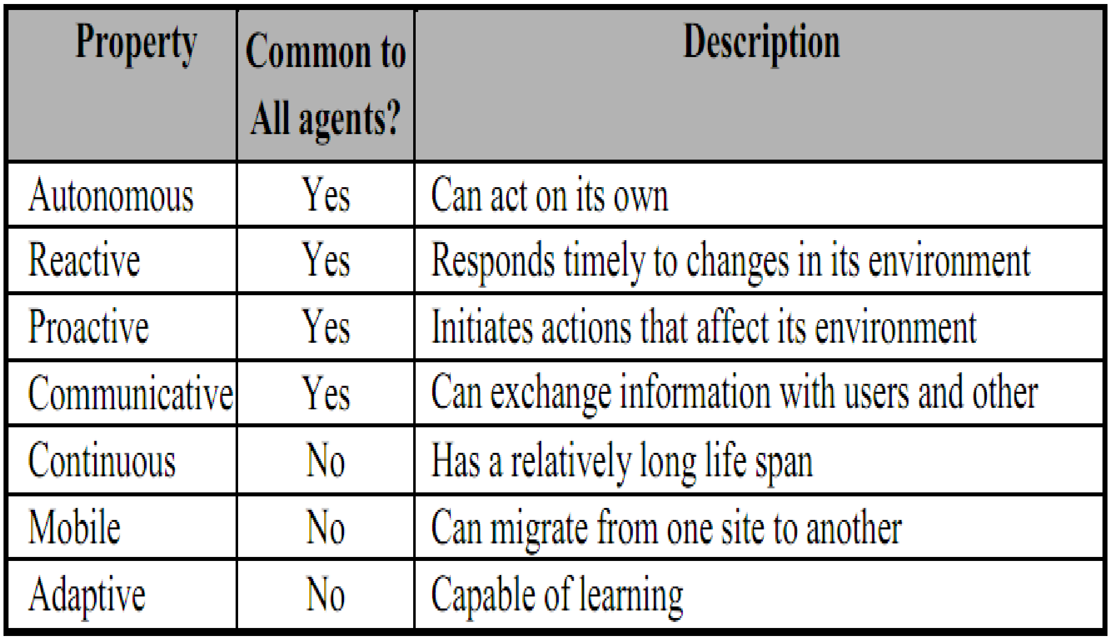
The main properties of Agent are [26] [27]:-

- An agent is autonomous, because it operates without the direct intervention of humans or others and has control over its actions and internal state.
- An agent is social, because it cooperates with humans or other agents in order to achieve its tasks. An agent is reactive, because it perceives its environment and responds in a timely fashion to changes that occur in the environment.

- An agent is proactive, because it does not simply act in response to its environment but is able to exhibit goal-directed behavior by taking initiative.
- An agent can be mobile, with the ability to travel between different nodes in a computer network.
- It can be truthful, providing the certainty that it will not deliberately communicate false information.
- It can be benevolent; always trying to perform what is asked of it.
- It can be rational, always acting in order to achieve its goals and never to prevent its goals being achieved, and it can learn, adapting itself to fit its environment and to the desires of its users.

Anyway, Table 3 summarizes the most crucial attributes of software Agent:

Table 3: Agent Properties.



Property	Common to All agents?	Description
Autonomous	Yes	Can act on its own
Reactive	Yes	Responds timely to changes in its environment
Proactive	Yes	Initiates actions that affect its environment
Communicative	Yes	Can exchange information with users and other
Continuous	No	Has a relatively long life span
Mobile	No	Can migrate from one site to another
Adaptive	No	Capable of learning

3.4 Agent Architectures

Agent architectures are the fundamental mechanisms underlying the autonomous components that support effective behavior in real-world, dynamic and open environments[24].

There are three classes of architectures:

- Deliberative Architectures (or Classical Approach);
- Reactive Architectures (or Alternative Approach);
- Hybrid Architectures.

a. ***Deliberative Architectures*** contain an explicitly represented, symbolic model of the world in which decisions (such as what actions to perform) are made via (logical) reasoning, based on pattern matching and symbolic manipulation. This approach builds agents as a type of knowledge-based system. The main problem with this type of architecture is performance.

   The architecture needs to translate the real world into a correct symbolic description and the agents need to reason with this information quickly enough for the results to be useful. This problem leads to *work on vision, speech understanding, learning, knowledge representation, automated reasoning, planning,* etc. Examples of deliberative agent architectures include Intelligent.

b. ***Reactive Architectures*** are based on the assumption that intelligent agent's behavior can be generated without an explicit representation and abstract reasoning of the kind that symbolic Artificial Intelligent (AI) proposes and is an emergent property of certain complex systems. One Can be identifies two key ideas:

1. Situatedness and embodiment: “Real” intelligence is situated in the world, not in disembodied systems such as theorem proves or expert systems.

2. Intelligence and emergence: “Intelligent” behavior arises as a result of an agent’s interaction with its environment.

c. ***Hybrid Architectures*** are designed by the product of the marriage of the two approaches discussed so far. Both the classical and the alternative approach to agent architecture have their own disadvantages, hence neither a completely deliberative nor completely reactive approach is suitable for building agents.

Obviously, a hybrid agent is built out of two (or more) subsystems: ***a deliberative one***, containing a symbolic world model, which develops plans and makes decisions in the way proposed by mainstream symbolic AI and A reactive one, which is capable of reacting to events that occur in the environment without engaging in complex reasoning. Often, the reactive component is given some kind of precedence over the deliberative one, so that it can provide a rapid response to important environmental events [27][28].

3.5 Basic Structure of Agent

The internal structure of agent is mainly to describe the module and how these modules work together, in the collaborative design and manufacturing system based agent, Agent’s features include: network center-based, interactive, semi-autonomy, responsiveness, consultative, collaborative, proactive, predictive, adaptive, flexible, durable and movable. Though, in the specific design and manufacture of agent-based system may only need a subset of these features, to achieve these characteristics, the appropriate module is necessary. The simple agent may only need a small number of modules (such as perception, inference, and execution), while the complex agent need more. Based on the previous model, the subject put forward the basic structure of agent.

1- Communication rules: to send the mission, to transmit the information and express the attitude of the target tasks on various agents.
2- Application programming interface, database interface and the tool interface: the interface is provided between agent and the application, database and connection of some tools.
3- Security modules: supply the security services for the interact information between the agent and the outside world, such as encryption or decryption, digital signature and signature verification.
4- Perception module: it is responsible for the sensation of the outside information, understanding of the situation as complete as possible. At the same time, it filter the received information, assist reasoning module identify and translate the information.
5- The reasoning module: recognize, translate, and decompensate the received information. The module make agent with a higher intelligence, it is the key to the agent with the complex decision-making and knowledge processing.

6- Decision-making modules: according to the information received and the agent’s goal make decisions based on the existing knowledge.
7- Planning module: according to the overall objective of agent plan their behavior.
8- The implementation of the modules: are mainly used for implementation planning.
9- Knowledge library: it contains two types of knowledge, one is the rule and the other is the knowledge block. Agents accomplish tasks conveniently and independently according to this knowledge.

3.6 The Agent Behavior

A particular behavior of an embodied, situated agent is a series of actions it performs when interacting with an environment. The specific order or manner in which the actions’ movements are made and the overall outcome that occurs as a result of the actions defines the type of behavior. We can define an action as a series of movements performed by an agent in relation to a specific outcome, either by volition (for cognitive-based actions) or by instinct (for reactive-based actions). With this definition, movement is being treated as a fundamental part of the components that characterize each type of behavior .The distinction between a movement and an action is that an action comprises one or more movements performed by an agent, and also that there is a specific outcome that occurs as a result of the action. For example, a human agent outcome of the action is that the light gets switched on. This action requires a series of movements to be performed such as raising the hand up to the light switch, moving a specific finger up out of the hand, then sing that finger to touch the top of the switch, then applying pressure downwards until the switch moves. The distinction between an action and a particular behavior is that a behavior comprises one or more actions performed by an agent in a particular order or manner. For example, an agent may prefer an energy saving type of behavior by only switching lights on when necessary (this is an example of a cognitive type of behavior as it involves a conscious choice). Another agent may always switch on the light through habit as it enters a room (this is an example of a mostly reactive type of behavior). Behavior is the way an agent acts in a given situation or set of situations [27],[29],[30],[31].

3.7 Agent and Environments

The environment that influences an agent's behavior can itself be influenced by the agent. They tend to think of the environment as what influences an agent but in this case the influence is bidirectional. An environment is everything in the world that surrounds the agent that is not part of the agent itself. They can think of the environment as being everything that surrounds the agent, but which is distinct from the agent and its behavior. This is where the agent 'lives' or operates, and provides the agent with something to sense and somewhere for it to move around. The comparison of an environment being like the world we live in is often implicitly used when the term 'environment' is used in computer science and AI in particular. An agent can explore, get lost in, and map a virtual computer environment just the same as a human in the real world environment. The ability to observe /sense and move around the environment are key properties of both. The environment may also be a simulation of a real environment, where the goal is to simulate specifically chosen real physical properties as closely as possible. A problem with simulated environments, however, is that it is often difficult to achieve realism in the simulation, as the simulation may diverge from reality in unpredictable ways [27],[29].

3.7.1 The environment properties

- Accessible versus inaccessible: An accessible environment is one in which the agent can obtain complete, accurate, up-to-date information about the environment's state. Most real-world environments are not accessible in this sense.
- Deterministic versus non-deterministic: A deterministic environment is one in which any action has a single guaranteed effect there is no uncertainty about the state that will result from performing an action, otherwise it is non-deterministic.
- Static versus dynamic. A static environment is one that can be assumed to remain unchanged except by the performance of actions by the agent. In contrast, a dynamic environment is one that has other processes operating on it, and which hence changes in ways beyond the agent's control. The physical world is a highly dynamic environment, as is the Internet.
- Discrete versus continuous. An environment is discrete if there are a fixed, finite number of actions and percepts in it; otherwise it is continuous [27].

3.8 Ontology

Ontology is a formal explicit specification of a shared conceptualization. A conceptualization, in this context, refers to an abstract model of how people think about things in the world, usually restricted to a particular subject area. An explicit specification means the concepts and relationships of the abstract model are given explicit terms and definitions [26].

Ontology is defined as a hierarchical representation of the objects from the application domain. It includes the following:

1-descriptions of the different types of objects (called concepts)

2- Descriptions of individual objects (called instances)

3- Properties of each object and the relationships between objects [26].

The subject ontology plays a crucial role in Disciple and in cognitive assistants, in general, being at the basis of knowledge representation, user-agent communication, problem solving and learning. Ontology has several functions:

1- The object ontology provides the basic representational constituents for all the elements of the knowledge base, such as the problems, the problem reduction rules.
2- The agent's ontology enables the agent to communicate with the user and with other agents by declaring the terms that the agent understands. Consequently, the ontology enables knowledge sharing and reuse among agents that share a common vocabulary which they understand.

3- The problem solving methods of the agent are applied by matching them against the current state of the agent's world which is represented in the ontology [26].

3.8.1 Ontology Basic Terms

A concept (or class) is a general representation of what is common to a set of instances (or individuals). Therefore, it may be regarded as a representation of that set of instances [26],[27].

An instance (individual) is a representation of a particular entity in the application domain. The objects in an application domain may be described in terms of their properties and their relationships with each other [26],[29].

An object feature is itself characterized by several features which have to be specified when defining a new feature. They include its domain, range, super features, sub features, and documentation. The domain of a feature is the concept that represents the set of objects that could have that feature. The range is the set of possible values of the feature [26].

3.7.2 Steps in Ontology Development

The ontology development steps can be summarized as follows [26].

1. Define basic concepts (types of objects) and their organization into a hierarchical structure.

2. Define generic object features by using the previously defined concepts to specify their domains and ranges.

3. Define instances (specific objects) by using the previously defined concepts and features.

4. Extend the object ontology with new concepts, features, and instances.

5. Repeat the above steps until the ontology is judged to be complete enough.

3.9 Multi-agent systems (MASs) and Societies of Agents

Multi-agent systems (MASs) are computational systems in which a collection of loosely-coupled autonomous agents interact in order to solve a given problem. As this problem is usually beyond the agents' individual capabilities, agents exploit their ability to communicate, cooperate, coordinate and negotiate with one another. Apparently, these complex social interactions depend on the circumstances and may vary from altruistic cooperation through to open conflict [24]. An MAS can be defined as a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver[24],[28],[30].

The main different between Multi Agent and single agent systems is that in MAS several agents exits, and they are aware of each other's goals and actions besides being aware of each other's intentions and behavior, in a fully general multi-agent system, agents also communicate with one another, either to help an individual agent achieve its goal, or in a rare case, prevent it.

Multi-agent systems are composed of several autonomous entities, which have the following general characteristics [29]:

1- Each agent has incomplete capabilities to solve the problem.
2- There is no global control.
3- Data is decentralized.

3.10 JADE

JADE (Java Agent Development environment): is a software framework to facilitate the development of interoperable intelligent multi-agent systems that is used by a heterogeneous community of users as a tool for both supporting research activities and building real applications [26].

It simplifies the implementation of multi-agent systems through a middle-ware that claims to comply with the FIPA (Foundation for Intelligent, Physical Agents) specifications and through a set of tools that supports the debugging and deployment phase. The agent platform can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by creating new agents and moving agents from one machine to another one, as and when required [26]. The goal of JADE is to simplify development while ensuring standard compliance through a comprehensive set of system services and agents. To achieve such a goal, JADE offers the following list of features to the agent programmer [26],[28],[29].

• FIPA-compliant Agent Platform, which includes the AMS (Agent Management System), the default DF (Directory Facilitator), and the ACC

(Agent Communication Channel). All these three agents are automatically activated at the agent platform start-up.

• Distributed agent platform. The agent platform can be split among several hosts. Only one Java application, and therefore only one Java Virtual Machine, is executed on each host.

• Java API to send/receive messages to/from other agents; ACL messages are represented as ordinary Java objects.

• Library of FIPA (Foundation for Intelligent, Physical Agents) interaction protocols ready to be used.

• Support for agent mobility within a JADE agent platform.

• Graphical user interface to manage several agents and agent platforms from the same agent. The activity of each platform can be monitored and logged.

- Automatic registration of agents with the AMS.
- FIPA-compliant naming service: at start-up agents obtain their GUID (Globally Unique Identifier) from the platform [26],[27].

3.10.1 JADE Architecture Overview

Figure (6) represents the main JADE architectural elements. An application based on JADE is made of a set of components called Agents each one having a unique name. Agents execute tasks and interact by exchanging messages [26].

Agents live on top of a Platform that provides them with basic services such as message delivery. A platform is composed of one or more Containers. Containers can be executed on different hosts thus achieving a distributed platform. Each container can contain zero or more agents. For instance, with reference to figure (6), container "Container 1" in host3 contains agents A2 and A3. Even if in some particular scenarios this is not always the case, we can think of a Container as a JVM (so, 1 JVM ==> 1 container ==> 0 or many agents). A special container called Main Container exists in the platform. The main container is itself a container and can therefore contain agents, but differs from other containers in that [26],[27].

1. It must be the first container to start in the platform and all other containers register to it at bootstrap time

2. It includes two special agents: the AMS (Agent Management System) that represents the authority in the platform and is the only agent able to perform platform management actions such as starting and killing agents or shutting down the whole platform (normal agents can request such actions from the AMS). The other agent is the DF (Directory Facilitator) that provides the Yellow Pages service where agents can publish the services they provide and find other agents providing the services they need. It should be noticed that if another main container is started, as in Host 4 in Figure 6, this constitutes a new platform [26].

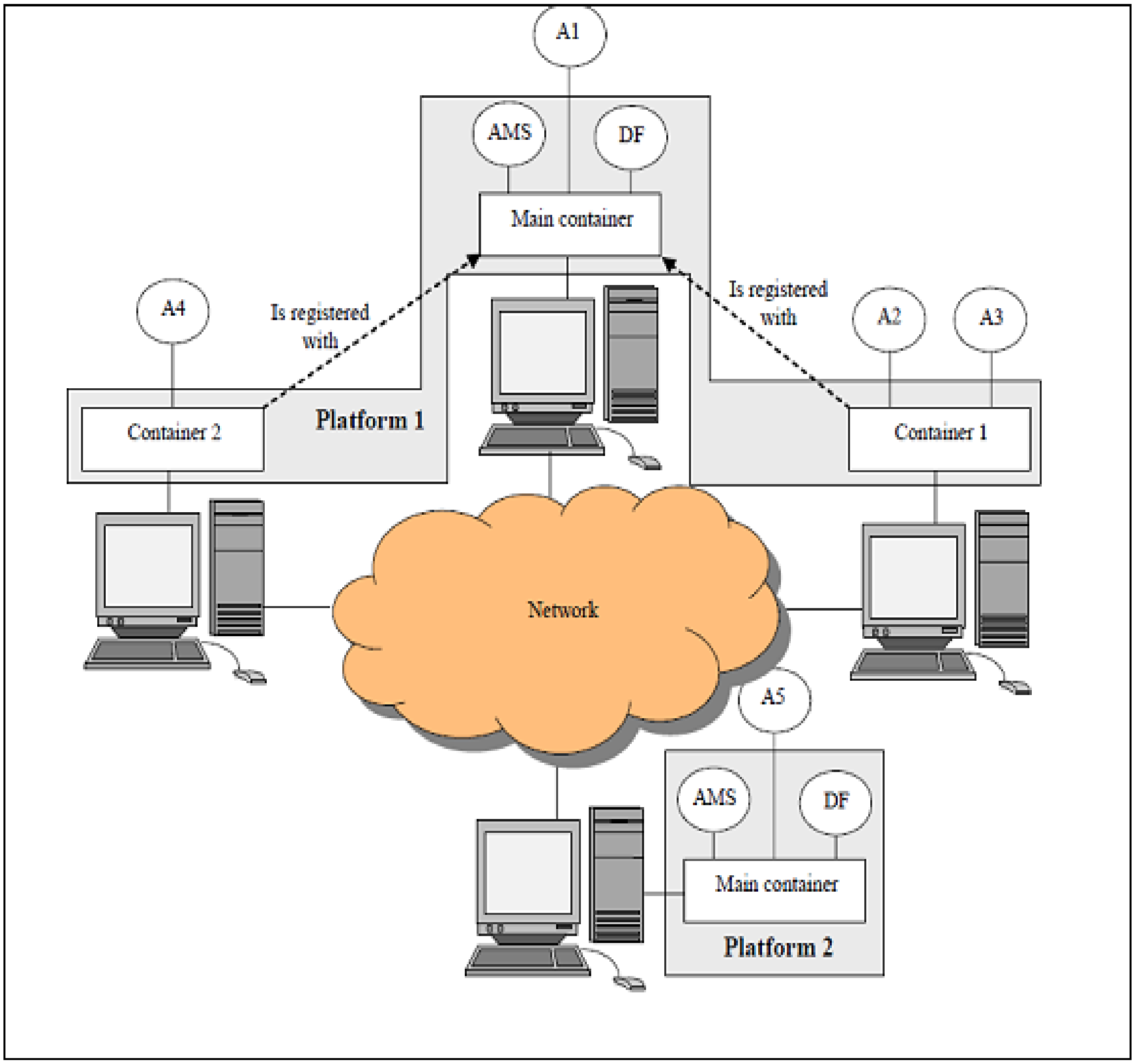


Figure 6: The JADE Architecture [26].

3.11 Agent Communications

Agent communication is one of the most important areas for standardization where Agents can communicate transparently regardless of whether they live in the same container (e.g. A2 and A3), in different containers (with same or in different hosts) belonging to the same platform (e.g. A and A2) or in different platforms (e.g. A1 and A5) as shown in Figure 6.

Communication is based on an asynchronous message passing paradigm. Message format is defined by the ACL (Agent Communication Language) language defined by FIPA, an international organization that issues a set of specifications for agent interoperability. An ACL Message contains a number of fields including [26],[30]:

- The sender
- The receiver(s)
- The communicative act (also called per formative) that represents the intention of the sender of the message.
- The content i.e. the actual information conveyed by the message [26].

3.12 FIPA ACL Message Structure

A FIPA ACL message contains a set of one or more message parameters. Precisely which parameters are needed for effective agent communication will vary according to the situation; the only parameter that is mandatory in all ACL messages is the per formative, although it is expected that most ACL messages will also contain sender, receiver and content parameters [26],[26],[27].

If an agent does not recognize or is unable to process one or more of the parameters or parameter values, it can reply with the appropriate not-understood message [26].

Specific implementations are free to include user-defined message parameters other than the FIPA ACL message parameters specified in Table 4. The semantics of these user-defined parameters is not defined by FIPA, and FIPA compliance does not require any particular interpretation of these parameters. Some parameters of the message might be omitted when their value can be deduced by the context of the conversation. However, FIPA does not specify any mechanism to handle such conditions, therefore those implementations that omit some message parameters are not guaranteed to interoperate with each other[27],[29].

The full set of FIPA ACL message parameters is shown in Table 4 without regard to their specific encodings in an implementation. FIPA-approved encodings and parameter orderings for ACL messages are given in other specifications. Each ACL message representation specification contains precise syntax descriptions for ACL message encodings based on XML, text strings and several other schemes.

Table 4: ACL Message Parameters [26].

Parameter	**Category of Parameters**
Performative	Type of communicative acts
Sender	Participant in communication
Receiver	Participant in communication
reply-to	Participant in communication
Content	Content of message
Language	Description of Content
Encoding	Description of Content
Ontology	Description of Content
Protocol	Control of conversation
conversation-id	Control of conversation

reply-with	Control of conversation
in-reply-to	Control of conversation
reply-by	Control of conversation

3.13 FIPA STANDARDS

Between different MAS implementations, In particular, the FIPA agent management reference model shown in

Figure 7: The FIPA Agent Management Reference Model [26].defines “the normative framework within which FIPA agents exist and operate. It establishes the logical reference model for the creation, registration, location, communication, migration and retirement of agents” .The normative framework includes a set of entities that FIPA-compliant MAS must contain, according to Figure 8.These include [26],[27],[29]:

- An agent runtime environment for defining the FIPA notion of agency;
- An Agent Platform (AP) for deploying agents in a physical infrastructure;
- A Directory Facilitator (DF) which provides a yellow pages service for the agents registered on the platform.
- An Agent Management System (AMS) acting as a white pages service for supervisory control over access to the agent platform.
- A Message Transport Service (MTS) for communication between the agents registered on different platforms [26].

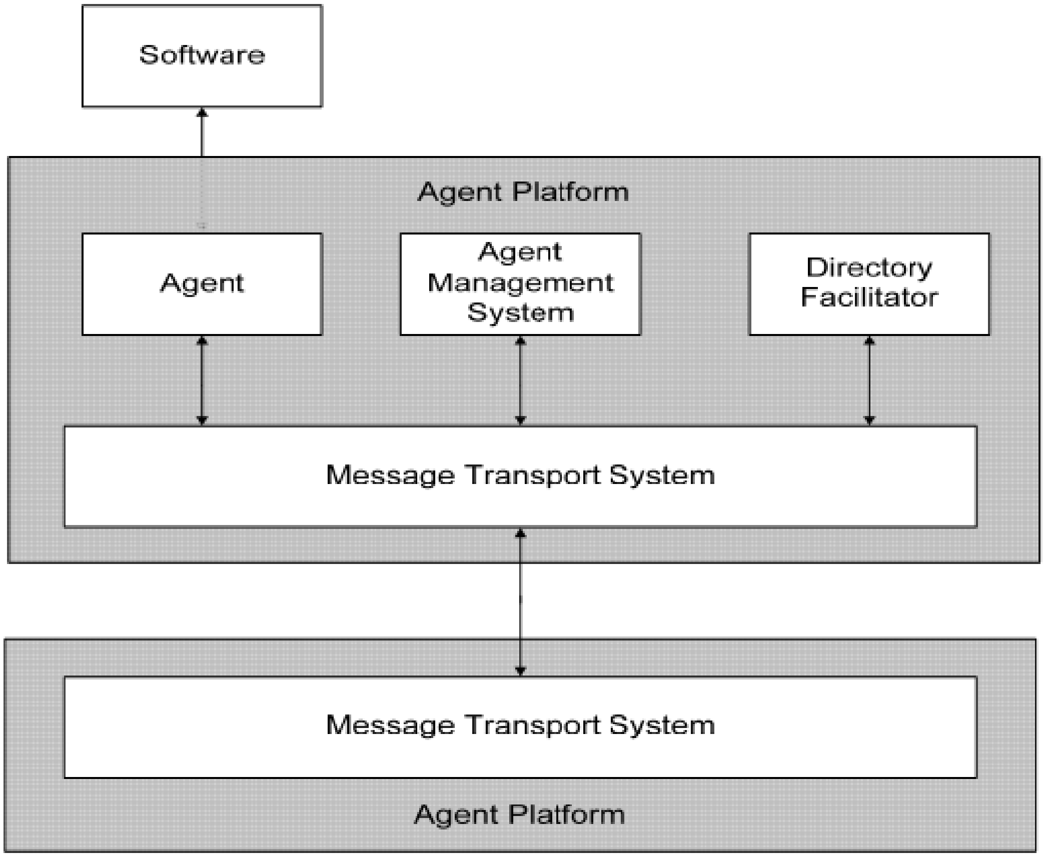


Figure 7: The FIPA Agent Management Reference Model [26].

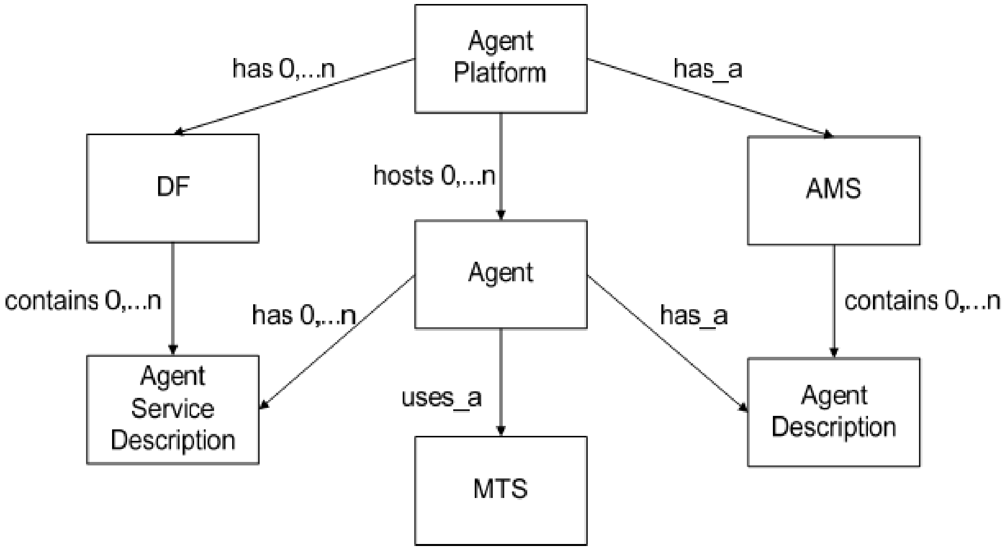


Figure 8: The entities that comprise a FIPA-compliant AP [26].

Chapter Four: The Proposed Systems "AIAMAS"

My approach is to use Socialization Feature of Multi – agent platform in order to develop annotation accompanied Images based on Usage.

4.1 Introduction

This chapter is dedicated to present Automatic Image Annotation using Multi-Agent System (AIAMAS), where multiple agents are propagated to client side over the cloud to transfer user experience in working with retrieved images due to posting certain queries. The kernel idea of using intelligent software agent is to provide deliverable environment in which knowledge is developed due to the acquisition of user interaction with the retrieved images against; user interaction reflects his/her experience in matching images to the query. Developed knowledge is used later on to annotate those images for better description of its content.

In this chapter, the presentation of AIAMAS is categorized into three areas: first area is the development platform and environment based on the implementation of AIAMAS is accomplished. In this area tools and software technologies (i.e., Servlet/JSP, Servlet Filters, Agent under Tomcat and HTML5) are presented.

The Second area is the core specification of AIAMAS is presented which is the knowledge development.

The third area is focused to present the approach implemented by this thesis to de-noise image annotations; this is done by processing annotations and the query as natural language paragraphs, then a semantic space is built up as a matrix and analyzed after that using LSA (Latent Semantic Analysis) to retrieve semantic correlation and similarities; this is to be used to reduce dimensions of the space and the results, as it will show later in this chapter, are a filtered annotation with less redundancy.

4.2 System General Scheme

Essentially, the proposed scheme is a java web application (i.e., Servlet/JSP) hosted by a Linux server. Its main functionality is to proxy queries posted by clients; those queries are processed and forwarded to Google search engine.

The results returned from the Google search engine are delivered to user's side over the cloud.

Dedicated agents are injected in web pages through which user is posting queries; those agents are responsible primarily on revealing user experience in inferring returned results (i.e., list of images returned from Google image search engine) related to certain queries.

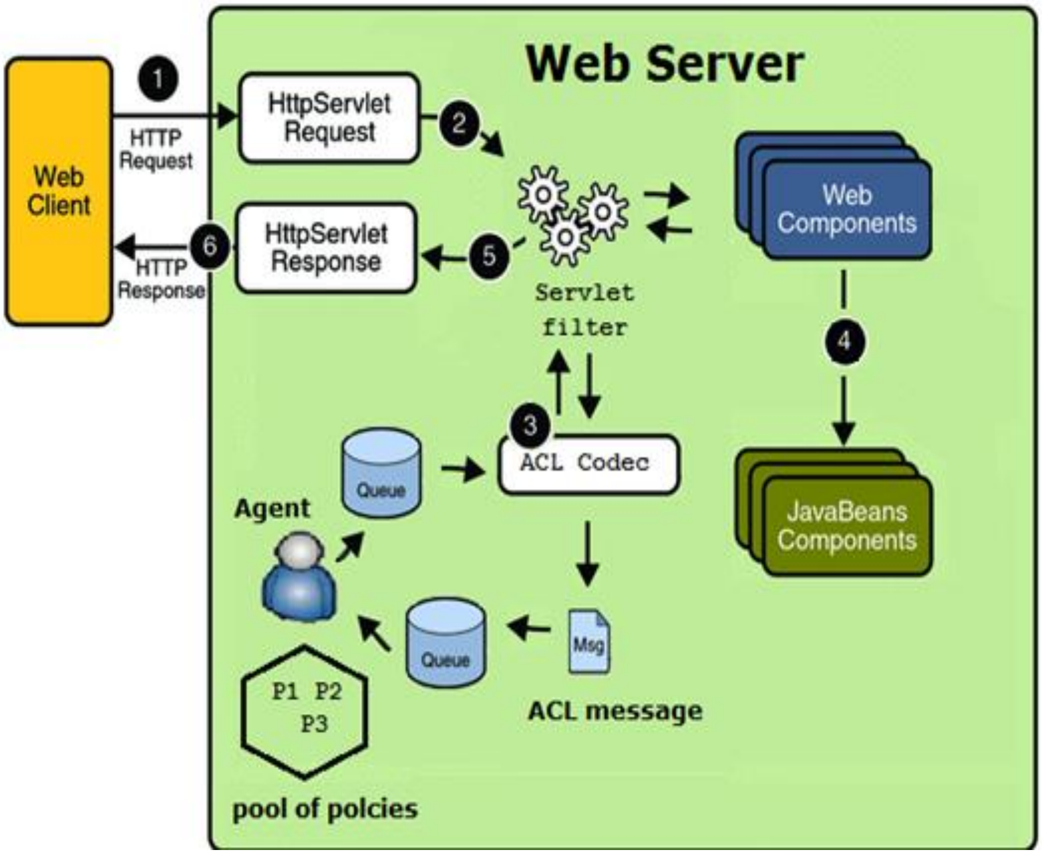


Figure 9: The essential software components constructing the proposal model.

As Figure 9) we have a java agent listening for users' queries and these are routed to agent module through Servlet filter module. Distributed agents over the cloud (i.e., the internet environment) are able to communicate 'server agent' through posting HTTP requests (i.e., POST or GET) to the URL of the web application implemented by this thesis. In the thesis, web based agents are able to use WebSocket protocol, which is one of the essential initiatives of HTML5, to establish a full-duplex communication channel over TCP connection.

Anyway, the JADE platform is launched at server side by invoking the ‘jade.Boot’ class and, after that, java agents are allowed to join that platform. Basically, java agents are not part of web application, thus, users’ HTTP based traffic is routed automatically using Servlet filter to java agent modules.

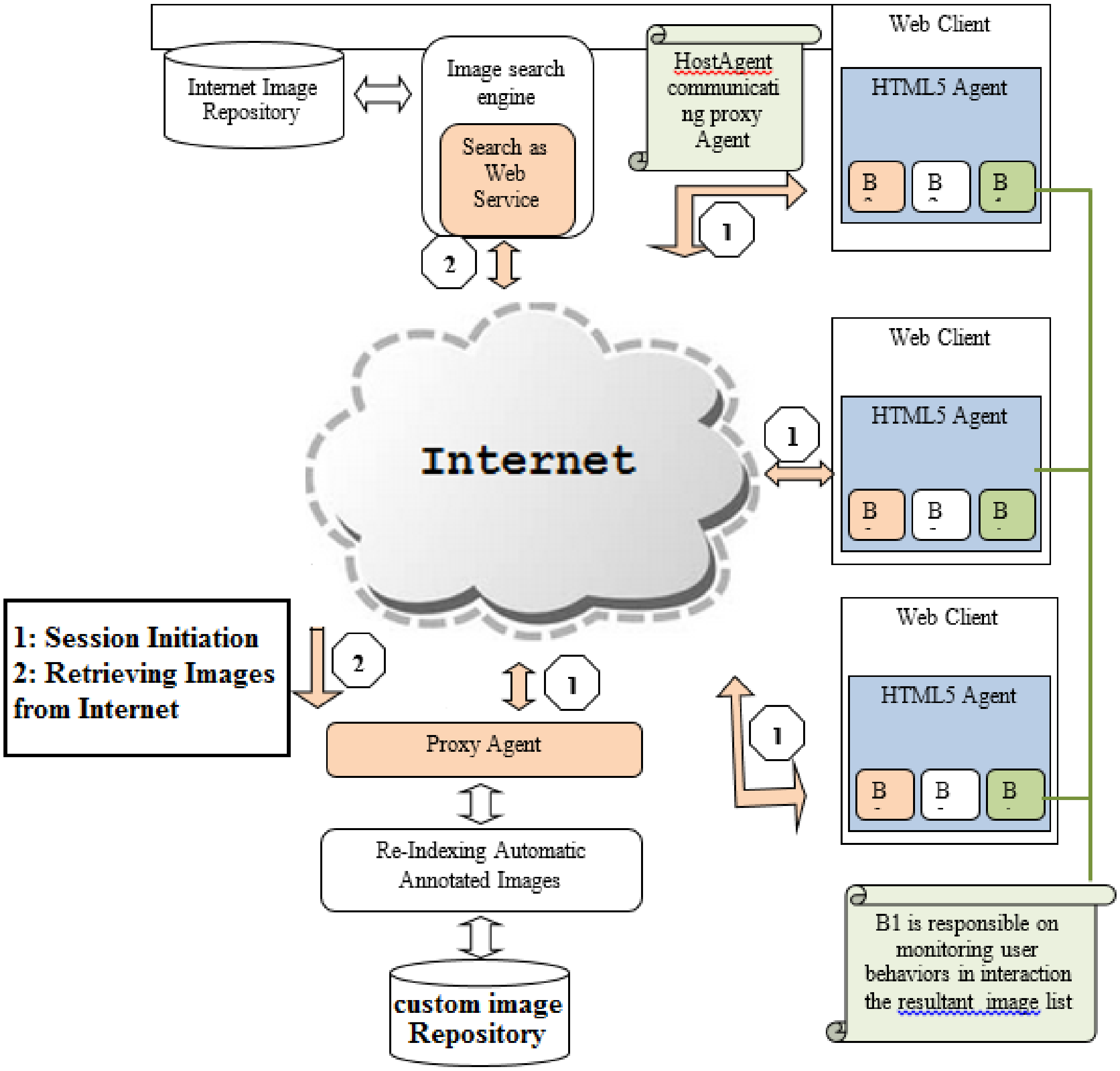

Internet Image Repository
Image search engine
Search as Web Service
HostAgent communicating proxy Agent
Web Client
HTML5 Agent
B
B
B
2
1
Internet
Web Client
HTML5 Agent
B
B
B
1
Web Client
HTML5 Agent
B
B
B
1: Session Initiation
2: Retrieving Images from Internet
2
1
1
Proxy Agent
Re-Indexing Automatic Annotated Images
custom image Repository
B1 is responsible on monitoring user behaviors in interaction the resultant image list

Figure 10: General scheme of social basic automatic annotation system.

Figure 10) presents a detailed version of the system proposed by this thesis. The most significant part is where indexing is accomplished; this is done after acquiring user experience as following section will presents. Figure **(**11**)** illustrates sequential diagram of the entire process in general, where specific knowledge is developed about certain query.

From figure (10), agents, either residing at server side or the one transferred to user side, share the following behaviors:

1- **Web Service Integration (B3):** this behavior encapsulates web service integration functionalities by implementing SOAP based invocation to google web service.

2- **Socialization Behavior(B2):** this behavior is responsible on socializing other agents within the platform to determine dominant tags for an image

3- **User behavior monitoring (B1):** this behavior is responsible on monitoring selections made by the user after querying Google search Web service. Highlighted images are grouped in clusters and socialization behavior is signaled. The following behaviors are considered: *Highlighted images, selected images, revisited images* and *saved images*

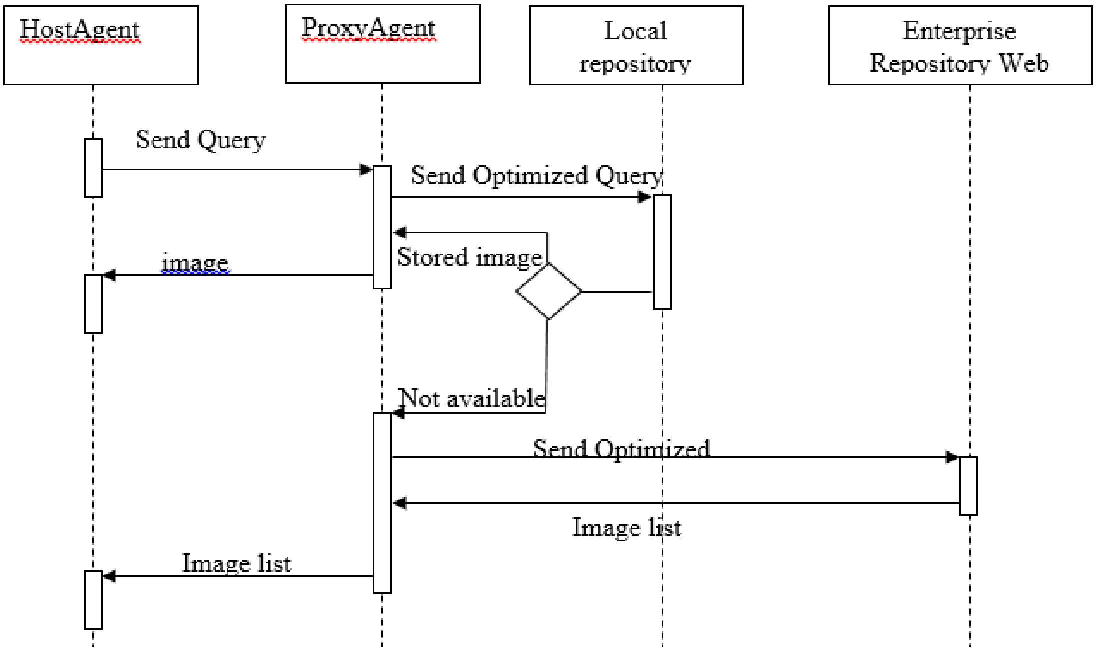


Figure 11: **Sequence diagram of complete 2-tier image annotation session.**

Figure 12 presents the general skeleton of the proposed system.

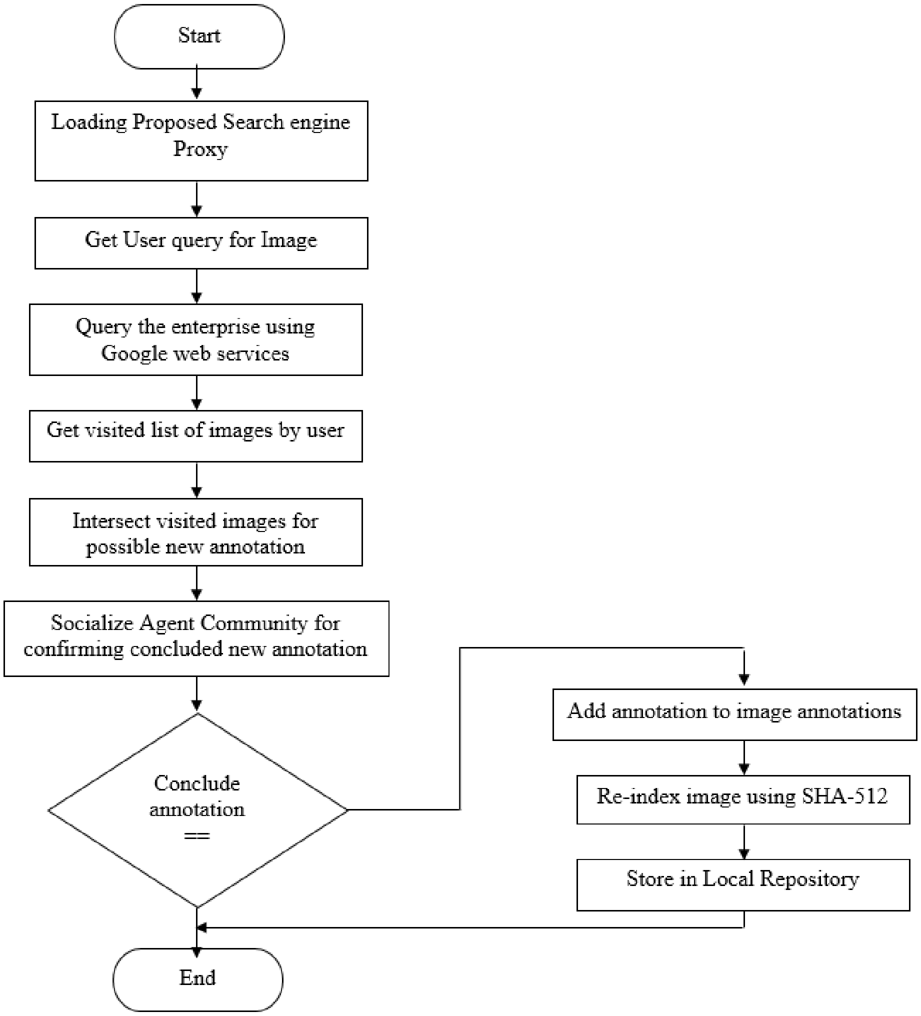

Start
Loading Proposed Search engine Proxy
Get User query for Image
Query the enterprise using Google web services
Get visited list of images by user
Intersect visited images for possible new annotation
Socialize Agent Community for confirming concluded new annotation
Conclude annotation ==
Add annotation to image annotations
Re-index image using SHA-512
Store in Local Repository
End

Figure 12: the general skeleton of the proposed system.

Agents are autonomous software components that have the ability to perceive events occurred with the environment. In this thesis, Agents are designed to conceptualize image annotation, thus we had to equip them with the specification for image annotation semantic, as it is shown in Figure 13).

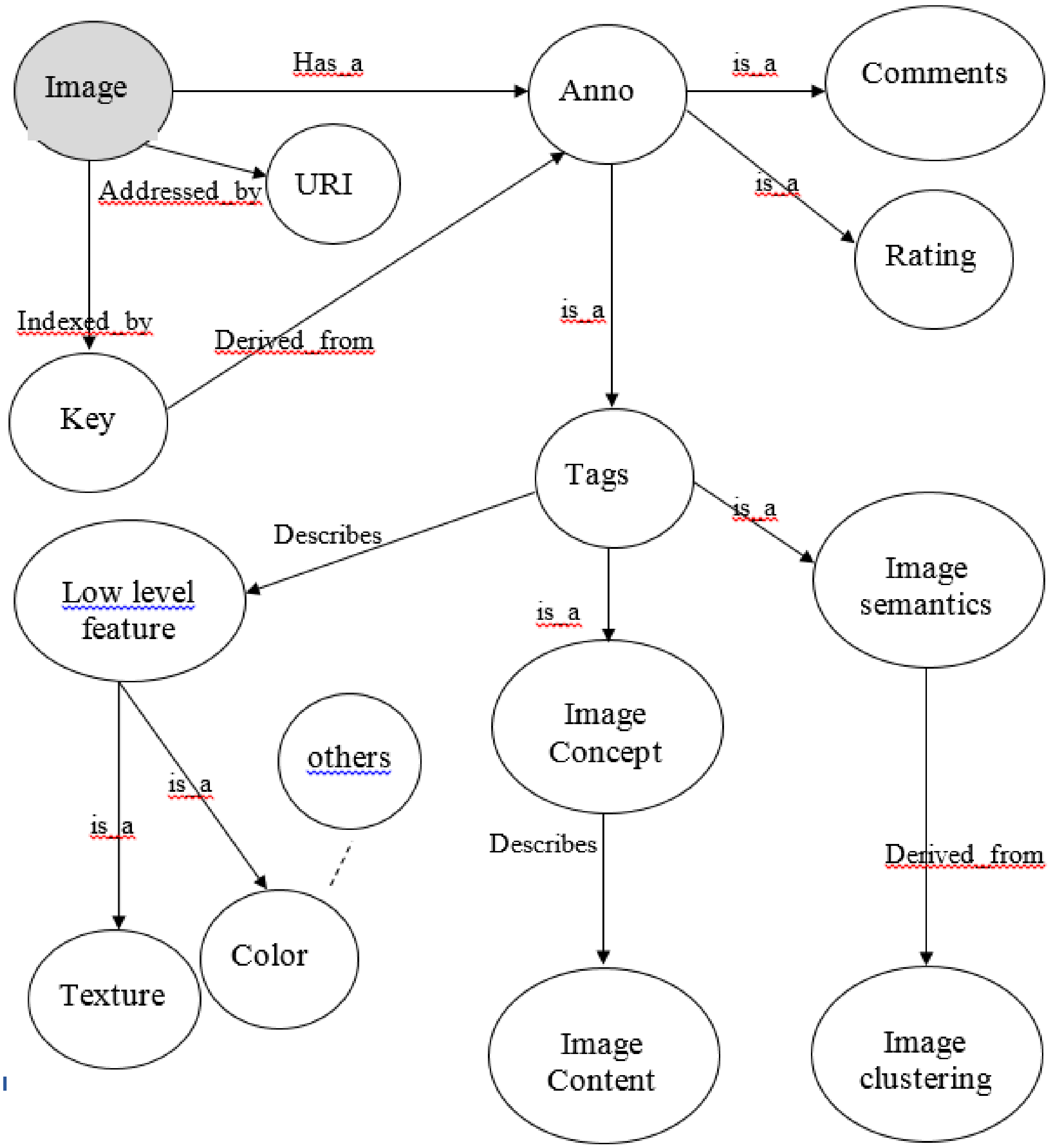


Figure 13: **Ontology to conceptualize Image Annotation.**

4.3 Mathematical Model of the Proposed Annotation Scheme

Let R_T^I be a Repository of images accessed over the Internet, and Images are indexed according to annotations coupled to these images, thus we can express this Repository as the following equation:

$$R_T^I = \sum_{i=1}^{k} (I_i \, , Anno_i)$$

Where:

$\forall \, image \;\in R_T^I \;\; \exists \, Anno(image) \;\; and \; Anno(image) = \; \sum_1^m c_i \; , \;\; c_i$ is a textual semantic concept (keyword) referring to the image and m is number of keywords.

For a successful image indexing the following condition has to be satisfied

$$\forall \, indexed(imgae_i) \, \exists \;\; c \;\; \in Anno(image_i) \;\; so \; that \;\; c \\ \notin Anno\left(image_j\right) for \; all \; j \; \neq i$$

Search engines, like Google search engine, are sending queries in two types:

First: textual queries where user is expressing his inquiry using natural language, thus the Optimized Search Function (OSF) is defined as following mapping:

OSF: Query $\xrightarrow{\hspace{6em}}$ c while non-optimized search function would produce many indexes that satisfy the following:

Let Query = S_i = {k_1 k_2 ... K} , where k_1 k_2 ... K are keywords used by the requestor for $image_i$

SearchResult(Query) = $\sum_{j=1}^{M} images_j \;\; iff$

$$S_i \cap Anno\left(image_j\right) \neq \emptyset \text{ and OSF(Query)} \neq c$$

OR

$$\text{SearchResult(Query) =} image_i \; iff \; \text{OSF(Query) = c}$$

4.4 Agent Social Effect

Let requestor behavior be U_{ij} where i=1.. M, is number of behaviors and j=1..N, is number of Agents And impact(U_{ij}, Agent-Socio) is the impact of behavior U_{ij} on the multi-Agent system. It will be evaluated along the implementation of the proposal, for the time being it can represented by the non-linear formula:

$$impact\left(U_{ij}, Agent - Socio\right) \propto^{\mathrm{nL}} Query_j$$

$$impact\left(U_{ij}, Agent - Socio\right) \propto^{nL} Anno(image_j)$$

The social impact (influence) of an agent A on another agent B is expressed by the following formula:

$IMP(A,B) = \sum_U P_A(U)\, impact(U, d(A,B))$ Where d (A,B) is Euclidean distance between agent A and agent B.

Along the implementation impact () formula will be defined in term of updating annotation based on social effect of requestor's behaviors U.

4.5 Intelligent Agent Implementation in Client Side

Intelligent agent is designed and implemented to fulfill crucial roles in this proposal, and as it has been clarified in previous chapters that software agent has pool of running behaviors, where Agents are encapsulating their functionalities in these behaviors (i.e., behavior is a term in Agent programming terminology, and it refers to actions conducted by Agents and when these actions are fired).

Behaviors run in different schemes of synchronization but they share the same triggering event, which is the arrival of ACL Message to the Message queue of the platform, in the proposed system multiple threads are required to tackle behaviors' functionality; this was a challenge to be accomplished in a client side web application (i.e., web page) due to the limitation of scripting language (i.e., jQuery and JavaScript). Javascript is a single threaded scripting language, thus, it does not fulfill the requirement.

HTML5 introduces many facilities to enforce multi-threading programming environment; Web Worker is the essential element in this issue. Web worker spawn piece of code, which is JavaScript code in this proposal, to work in different thread and does not affect the overall latency of the program.

Web worker is designed to carry out heavy weight process in the background but does not have access to the foreground components (i.e., DOM - Document Object Model); this due to the sophisticated synchronization model needed to manage accessing DOM objects from multiple Web Workers.

Agent behaviors are implemented in separate JavaScript files and dispatched when needed to run from within the page executing context in the client machine, as Figure 14 presents.

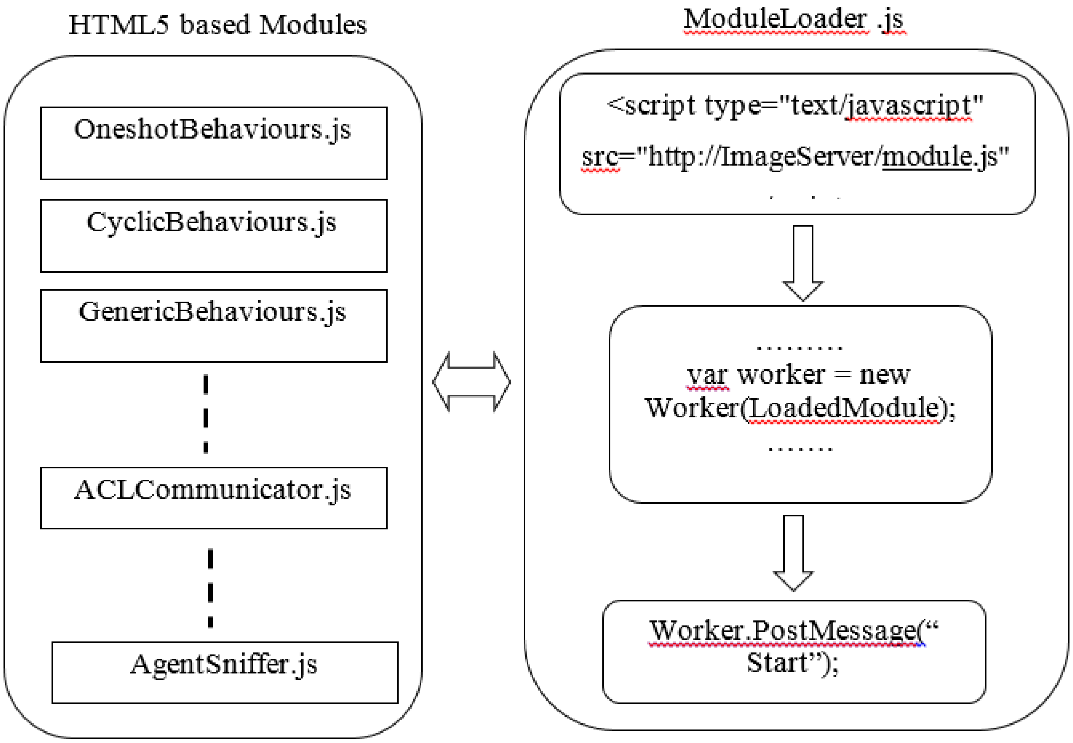
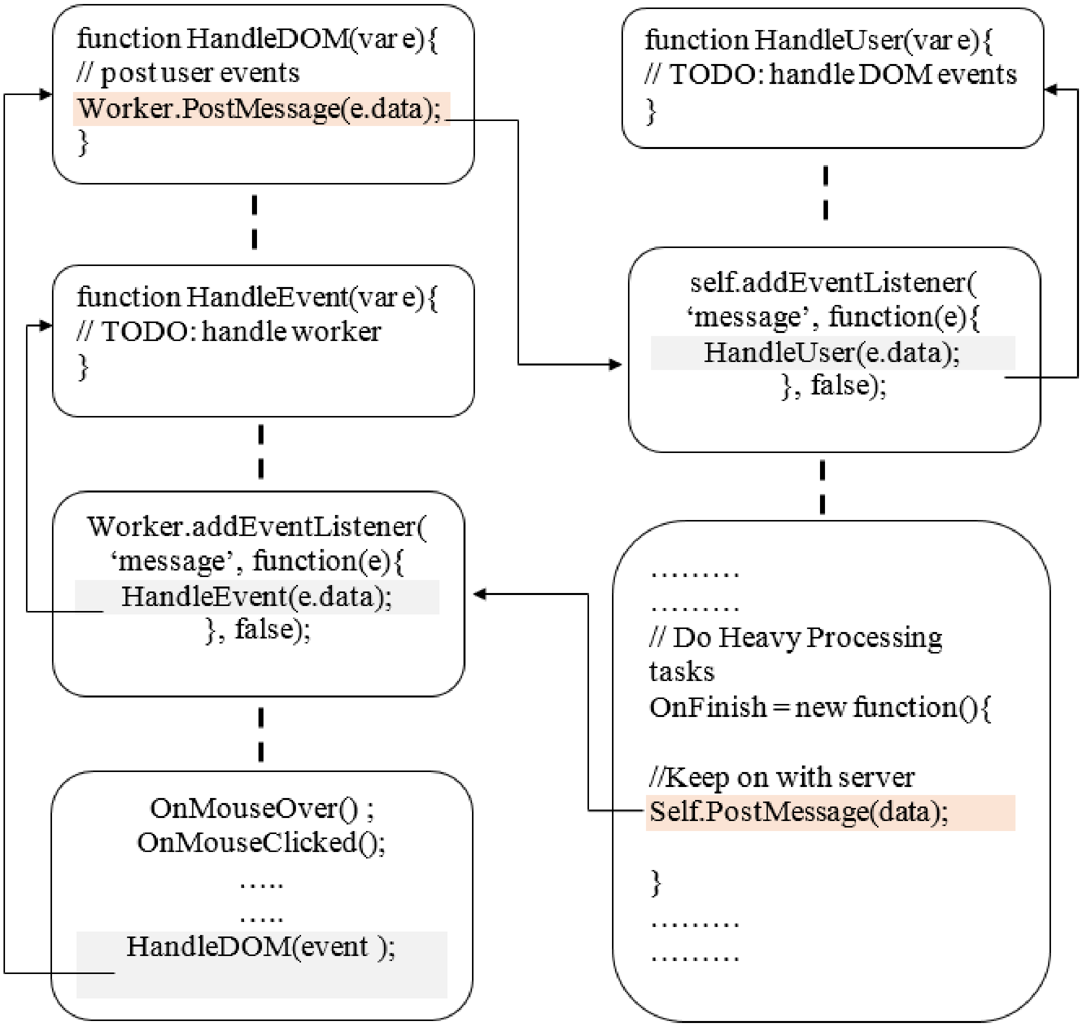


Figure 14: HTML5 Based Intelligent Agent Architecture.

Web Worker communicates with the main web page using an event driven scheme where events generated by user interaction with the DOMs, are captured and sent to the background worker module through notification messages as Figure 15 depicts:



```
function HandleDOM(var e){
// post user events
Worker.PostMessage(e.data);
}
```

```
function HandleEvent(var e){
// TODO: handle worker
}
```

```
Worker.addEventListener(
'message', function(e){
HandleEvent(e.data);
}, false);
```

```
OnMouseOver() ;
OnMouseClicked();
.....
.....
HandleDOM(event );
```

```
function HandleUser(var e){
// TODO: handle DOM events
}
```

```
self.addEventListener(
'message', function(e){
HandleUser(e.data);
}, false);
```

```
.........
.........
// Do Heavy Processing
tasks
OnFinish = new function(){

//Keep on with server
Self.PostMessage(data);

}
.........
.........
```

Figure 15: Agent Behavior (on the left) communicating Image Annotation Main Page.

4.6 Agent Communication Scheme

In this proposal, agents need to communicate over the cloud; this is a challenge due to the design limitation of JADE environment with no support of establishing agent platform over the cloud. Intelligent Agent built up using JADE are abstracting networking into higher level where Agent developer do need to worry about networking, but this does not include the cloud, where new encapsulation is required to packaging ACL Message in

HTTP envelop and send it over the cloud to the destination. Special hidden parameters are injected in HTTP requests to indicate that ACL Message is carried with the request; Figure 16) presents the specifications.

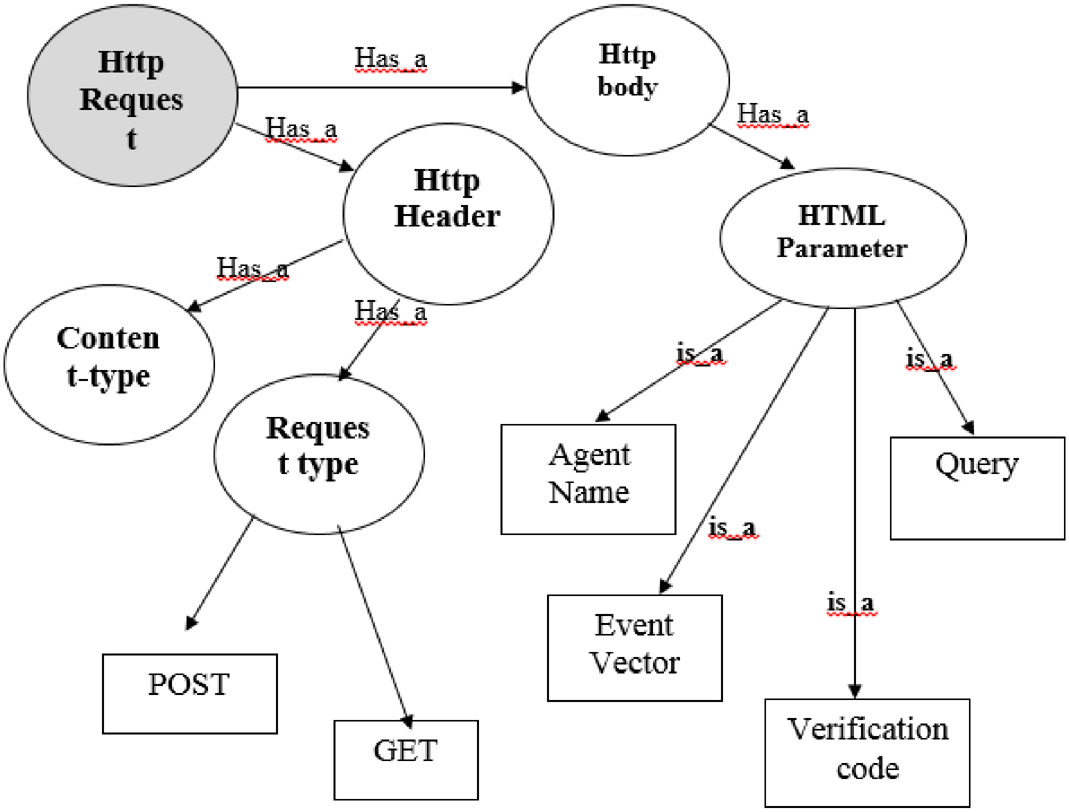


Figure 16: Ontology used to interpret HTTP request packet.

4.7 De-noise Image Annotation

In annotation based image retrieval, the redundancy of images returned by the image retrieval engine is due to redundancy in the natural language words composing the annotation.

In this proposal, images are represented by the concepts they hold. Image concepts are the projection of human interpretation of the visual structures within an image, hence:

$$I = \sum_{i=1}^{N} C_i \,.\, \vec{v}_i \qquad \text{--- eq.2}$$

Where I is any image and C_i is the i^{th} concept recognized with that image

$$q = \sum_{i=1}^{K} w_i . \vec{u}_i \qquad \text{---eq.3}$$

Where q is the query entered by the user, w_i is the i^{th} word within the query and $\vec{u}_i$ is the semantic unit vector. Semantic meaning for image's concept should correlate human's interpretation for that concept; hence, eq.3 is a prerequisite

$$\vec{v}_i \,.\, \vec{w}_i = 1 \qquad \text{---- eq.4}$$

The semantic space is represented by a $A_{m\,x\,n}$ matrix and this matrix is decomposed into its principal components as the following equation:

$$A = U\Sigma V^T = \sum_{i=1}^{N} \sigma_i u_i v_i^T \qquad \text{---eq.5}$$

Where σ_i is the i^{th} singular value of the matrix, σ_1 and v_1^T are the most effective direction, in other words, we can decompose the matrix A into its individual components as the following:

$$A = U\Sigma V^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \sigma_3 u_3 v_3^T + \cdots + \sigma_n u_n v_n^T \quad \text{---eq.6}$$

And we have singular values aligned in descending as the following:

$$\sigma_1 > \sigma_2 > \cdots > \sigma_n$$

Thus, values of σ are utilized as indicators to reduce matrix dimensions (i.e., the matrix that represent the annotations and queries).

Block similarity is measured by the following formula:

$$\theta = \cos^{-1}\frac{v_i\, v_j}{|v_i\,|\,|v_j|} \qquad \text{--- } \mathbf{(5)}$$

Where

$$Min(\theta) < Threshold \rightarrow v_i \equiv v_j \quad \text{--- } \mathbf{(6)}$$

The priorities of using a specific word to index and retrieve a certain image correspond to the singular values calculated by the SVD algorithm, this way words with less singular values can be omitted from the annotation.

The Σ matrix can be used as a noise filter where queries are treated as vectors within the semantic space and those who are on the same direction toward the most singular value; those queries would be composed of the most affected words.

Figure(10) presents a flow chart of the de-noising process, where query and annotations accompanied the resultant list of images are used to build the semantic space; this semantic space is analyzed using LSA to evaluate the most affecting variance of the query and omitting less important component of the query, in other words remove the redundancy from the annotations.

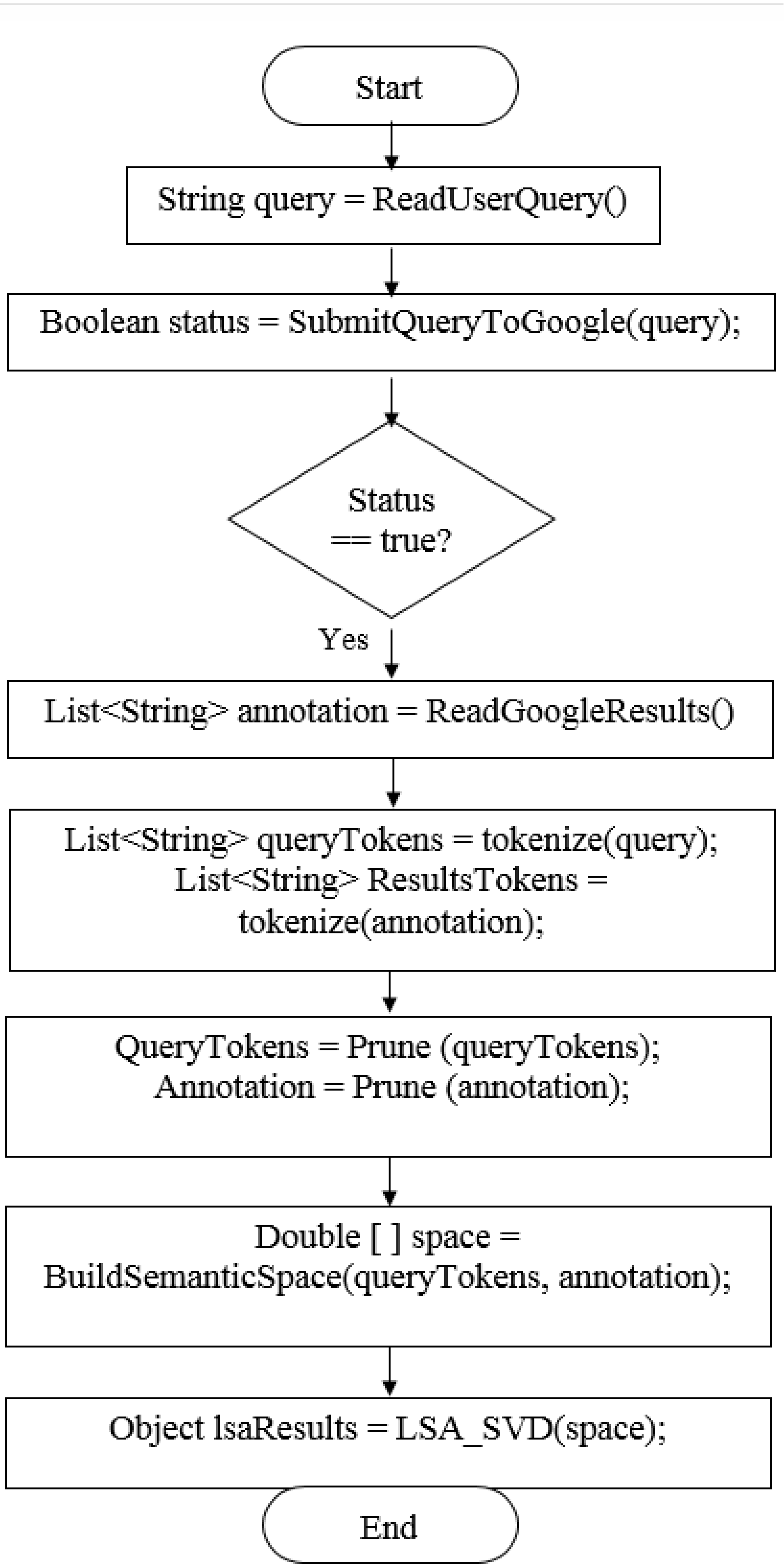

Start
String query = ReadUserQuery()
Boolean status = SubmitQueryToGoogle(query);
Status == true?
Yes
List<String> annotation = ReadGoogleResults()
List<String> queryTokens = tokenize(query);
List<String> ResultsTokens = tokenize(annotation);
QueryTokens = Prune (queryTokens);
Annotation = Prune (annotation);
Double [ ] space = BuildSemanticSpace(queryTokens, annotation);
Object lsaResults = LSA_SVD(space);
End

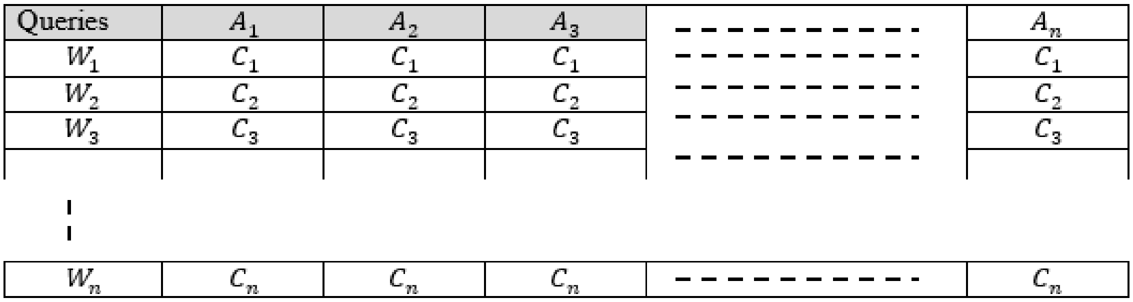
Figure 17: De-noising Image Annotation Using LSA.

U: represents query words vectors corresponding to the annotations of resultant images in the hidden word space

V: represents the words vectors corresponding to the queries in the hidden word space.

Thus, to study the relationship of i^{th} word within the query to the j^{th} annotation (i.e., certain image), all we need to do is to conduct the Dot Product between i^{th} vector of U and the j^{th} vector of V.

Figure (18) illustrate how the table is constructed to build semantic space in which words in queries are weighted in a document (i.e., annotation text accompanied the image) in term of structural occurrence.



Queries	A_1	A_2	A_3	-----------	A_n
W_1	C_1	C_1	C_1	-----------	C_1
W_2	C_2	C_2	C_2	-----------	C_2
W_3	C_3	C_3	C_3	-----------	C_3

⁞					
W_n	C_n	C_n	C_n	-----------	C_n

Figure 18: Word’s space revealed from set of queries and resultant images.

Anyway, LSA studies the relationship among words in term of its association in documents. It is true that SVD has the capability to infer connections among words in queries even they don’t appear together in a single query or event they don’t appear in any single annotation.

Chapter 5: Practical Results

Introduction

This chapter is dedicated to present the implementation of proposed system where it starts by introducing the steps taken to setup and configure the proposed system, needed requirements are declared and software technologies are specified.

After introducing the setup and configuration of the system, this chapter moves to introduce the first stage in the proposed system which is to build the conventional text based image retrieval layer. At this stage, custom search engine is implemented to be the front layer of Google image search engine through which queries for images are conducted and executed. The goal of this layer is to generate special indexing scheme over images retrieved in established sessions; this indexing scheme is due to the experience gained from monitoring user behaviors in their interactions with retrieved images against certain queries.

One important step of the setup for the proposed system components is the integration of Java intelligent agent with the web server (i.e., Apache/TomCat); this integration is presented in separate section in this chapter due to the importance and role of Java agent in this proposal, basically this would not by hard due to the fact that JADE which is the development environment is a Java library by itself, thus it would not be hard to integrate it with Apche/TomCat which is a Java programs.

Essentially, the proposed automatic image annotation system depends has a lifecycle of two stages: first stage is the acquiring knowledge and stage two is deploying of the acquired knowledge in the first stage.

System setup and configuration

As it has been presented in previous chapter, the proposed system is composed of multiple software modules that collaborate to achieve the objectivities of the proposed system. Essentially, two categories are divided into two categories:

Client Category: in this category, modules are implemented using JavaScript and loaded on demand by the system loader; these modules are implemented as run able components that can be executed by internet explorer (i.e., the proposed system is after all a web application). JavaScript, WebSocket and HTML5 are the main software technologies used to implement client side.

The major challenge faced building the proposed automatic annotation system is the implementation of client side Intelligent Agent, where no implementation for Java agent (i.e., that is embeddable in internet explorer) is found, thus web applications that introduce agent based functionality are, so far, rely on the server side to implement those functionalities. In our automatic annotation system this was not acceptable due to crucial needs for intensive monitoring for; this is from a side and mining these collections of behaviors to extract knowledge as it regards certain retrieving domain; this is from the other side, therefore we started the process of implementing limited version of agent module based on JavaScript and HTML5. Many software classes have been designed and implemented using javascript with the help of its awesome new tags and technologies such as WebWorker and Web Socket. Anyway, essential classes built along this implementation are:

- **Cyclic Behavior**: this is a javascript class that coding Agent behavior; it is implemented as a web worker that is executed in the background to carry out heavy load processing. In matter of fact, this module caries agent capabilities to monitor user interaction with the retrieved images.
- **BootLoader.js:** this JavaScript module is responsible on loading client side agent. The booting process including joining the main platform which is far in the application server.
- **Agent Communicator:** this class is the most crucial module in the client side where it is responsible on integrating client side to the server side; this integration is done by establishing web socket connection and deploying a special communication protocol designed and implemented by this work.

Server Category: in this category, modules are implemented using Java language with added libraries such as JADE (Jave Agent DEvelopment) library. Minimum TomCat version required for the implementation of this proposal is 7.0.27; this is due to the support of WebSocket technology.
A VPS (Virtual Private Server) has been exploited to host PersonalClassPro of tomcat which grants the possibility to assign ports and coduct manual configuration.

Setup JADE on Server Side

Automatic image annotation scheme presented by this dissertation employs intelligent software agent which is a special Java program build using JADE development environment. Since we are after creating web application, Java based intelligent agent has to cope with the environment imposed by enterprise platforms. JADE has presented a special agent that is embeddable in the web applications hosted TomCat; this agent is the AgentGateway which is interoperable with tomcat and can interpret Http Requests and Http Respnses.

Before starting AgentGateway, JADE boot core has to started to provide shell environment for the Java Agents; this is done by using Java module at the server side as the following code snippet:

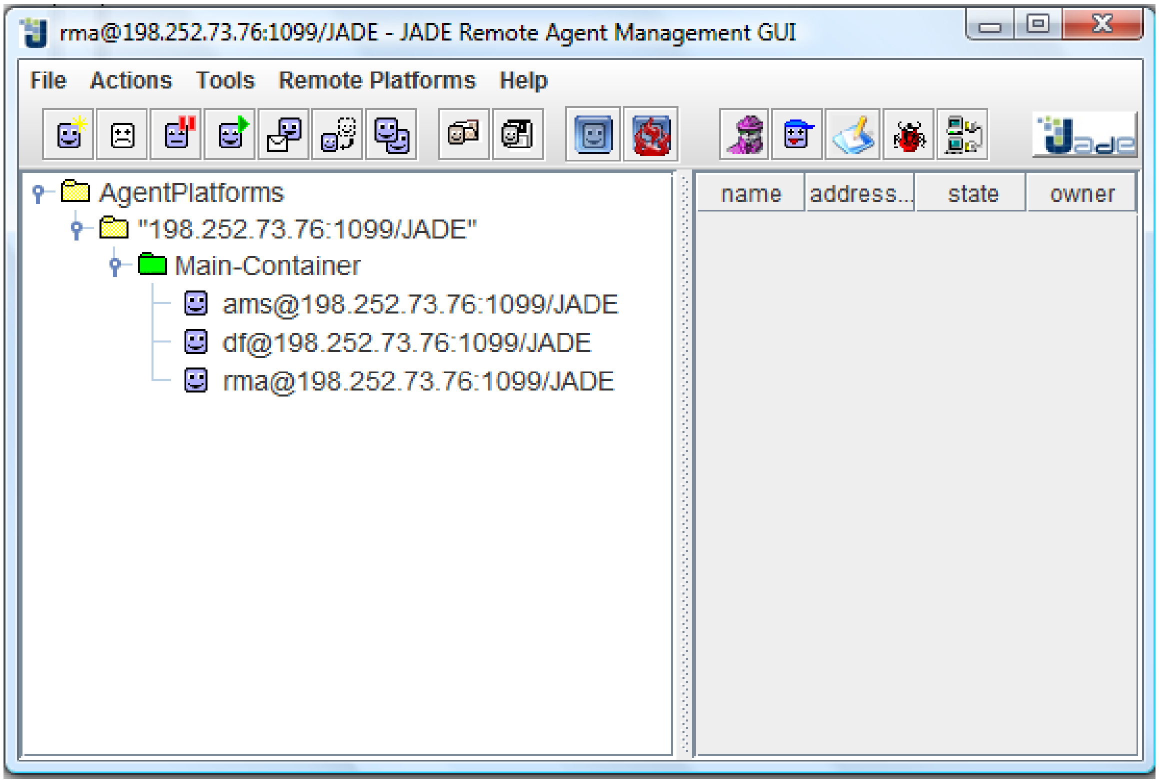
```
try {
        Process JadeProcess=Runtime.getRuntime ().exec ("java
-cp /bin/jade.jar Boot -gui");

        InputStream is = p.getInputStream();

        BufferedReader br = new BufferedReader (new
InputStreamReader (is));

        String errorMSG = br.readLine();

        while (errorMSG!=null) {
            SendThroughSocket(errorMSG);
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
```

Figure 19: Starting Instance on JADE environment programmatically.

As it is shown in figure (19), error messages are manipulated through the function Send Through Socket; error messages are processed by this function and a notification is sent to the client side; this was crucial in order to sustain confidence that the agent platform is running and behaviors are subjected to mining, where without starting JADE environment; Agents would not start and eventually the entire proposed system is stopped.

Anyway, if the code presented in figure (19) is executed properly, then JADE shell is booted and kernel agents are started as it is presented in figure (20); in this figure basic agents (DF, RMA, and AMS) are launched to represent basic functionalities requested to manage platforms of Agents. In this implementation a new challenge faced communicating kernel agents where client Agent presented in this proposal is not a native agent but it is only a front end and the back end (i.e., native java code) is residing at the server side.

In this implementation we needed a gateway agent (i.e., JADE platform gateway); this agent is responsible on receiving request to instantiate new agent instance or to terminate existing one, anyway, all client side agents (i.e., front agent which is coded using javascript) should forward their traffic to the gateway agent which rout it corresponding back end agents.

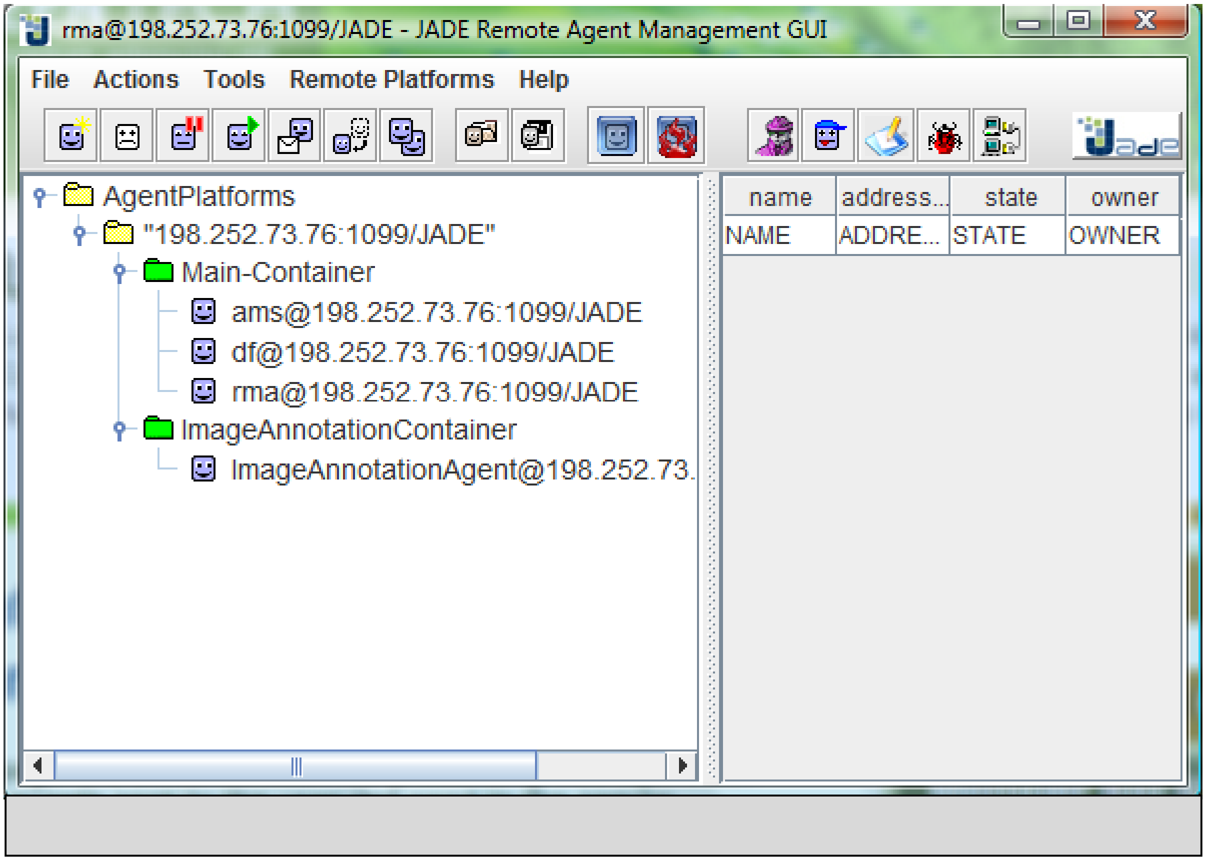

rma@198.252.73.76:1099/JADE - JADE Remote Agent Management GUI
File
Actions
Tools
Remote Platforms
Help
name
address...
state
owner

```
AgentPlatforms
  "198.252.73.76:1099/JADE"
    Main-Container
      ams@198.252.73.76:1099/JADE
      df@198.252.73.76:1099/JADE
      rma@198.252.73.76:1099/JADE
```

Figure 20: Booting Up the Jade environment at the Server Side.

The front side agent (i.e., JavaScript module) is designed to receive notification messages from its corresponding back end and when it is shutdown it sends a termination message to the gateway agent.

The proposed automatic image annotation is built as a web application thus agents are a hidden layer resides behind the web application skeletal. As it has been presented, this work needs two types of agents, one is running at the server side will the other is the client side agent which is implemented as JavaScript module. Anyway, at the loading for the search engine; these two agents are started and joined the platform as it has shown in figure (21).


rma@198.252.73.76:1099/JADE - JADE Remote Agent Management GUI
File
Actions
Tools
Remote Platforms
Help

AgentPlatforms
"198.252.73.76:1099/JADE"
Main-Container
ams@198.252.73.76:1099/JADE
df@198.252.73.76:1099/JADE
rma@198.252.73.76:1099/JADE
ImageAnnotationContainer
ImageAnnotationAgent@198.252.73.

name	address...	state	owner
NAME	ADDRE...	STATE	OWNER

Figure 21: New Agent is created and joined the platform at web application invocation.

Client side agent is responsible of monitoring user behavior in responding to the retrieved images, and grades this responding according to the grading policy presented in previous chapter. Client agent updates server agent with user selections and queries submitted in the first place; this is presented in next sections, anyway, to monitor the messages exchanged between client agent and server agent an instance of sniffer agent is invoked (sniffer agent is part of JADE package). Figure (22) presents sniffer agent started at the server side to monitor underlying traffic; this monitoring is only for validating the communication protocol among agents (i.e., back end and front end agents).

Front end agent is a custom implementation for intelligent agent functionalities and communication language, in other words the front agent is not a complete implementation for JADE-compliment agents. In this proposal and ACL-Communication packet is used to transfer informative from the front agent to the back through the gateway agent; this packet holds most of the conceptualization imposed by standard ACL-Message introduced by JADE environment.

Gateway agent is responsible of generating standard ACL-Message that is perceived by the agents' modules, gateway agent's responsibility to build a valid message with the information send by the front agent and forward built message to the designated agent.

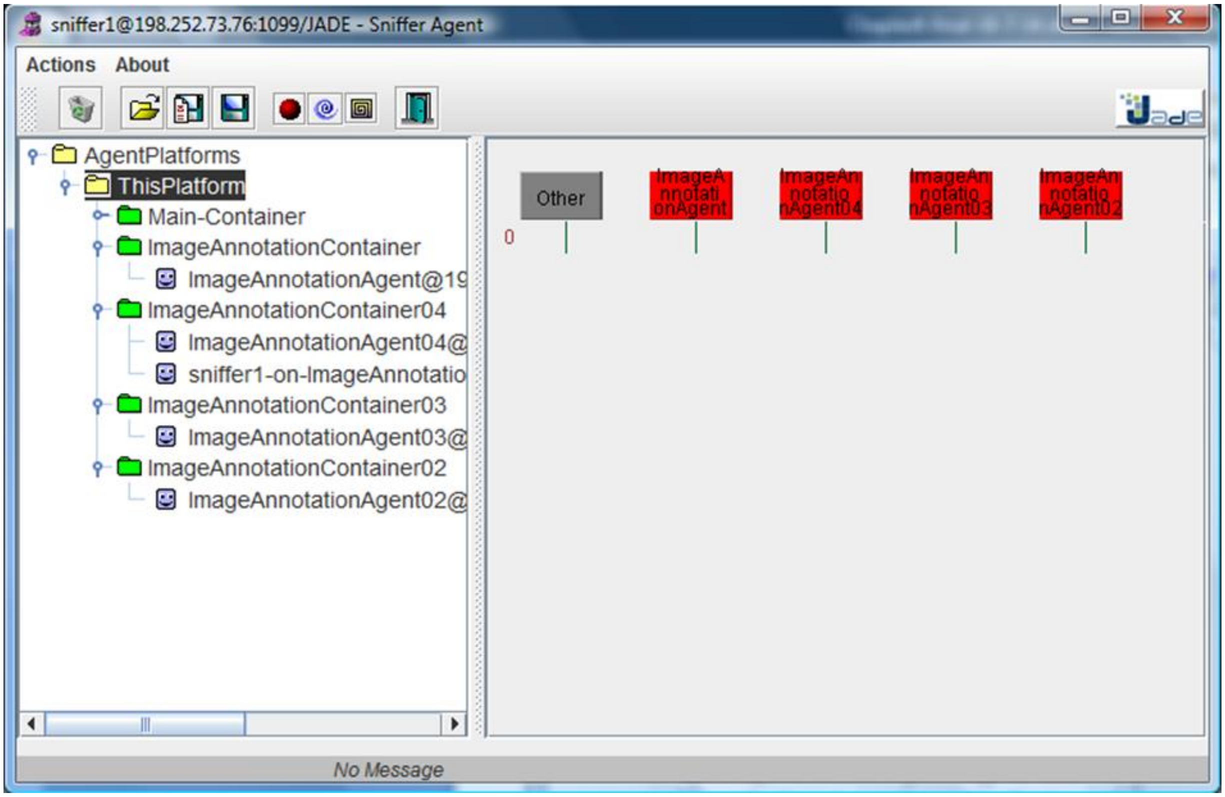
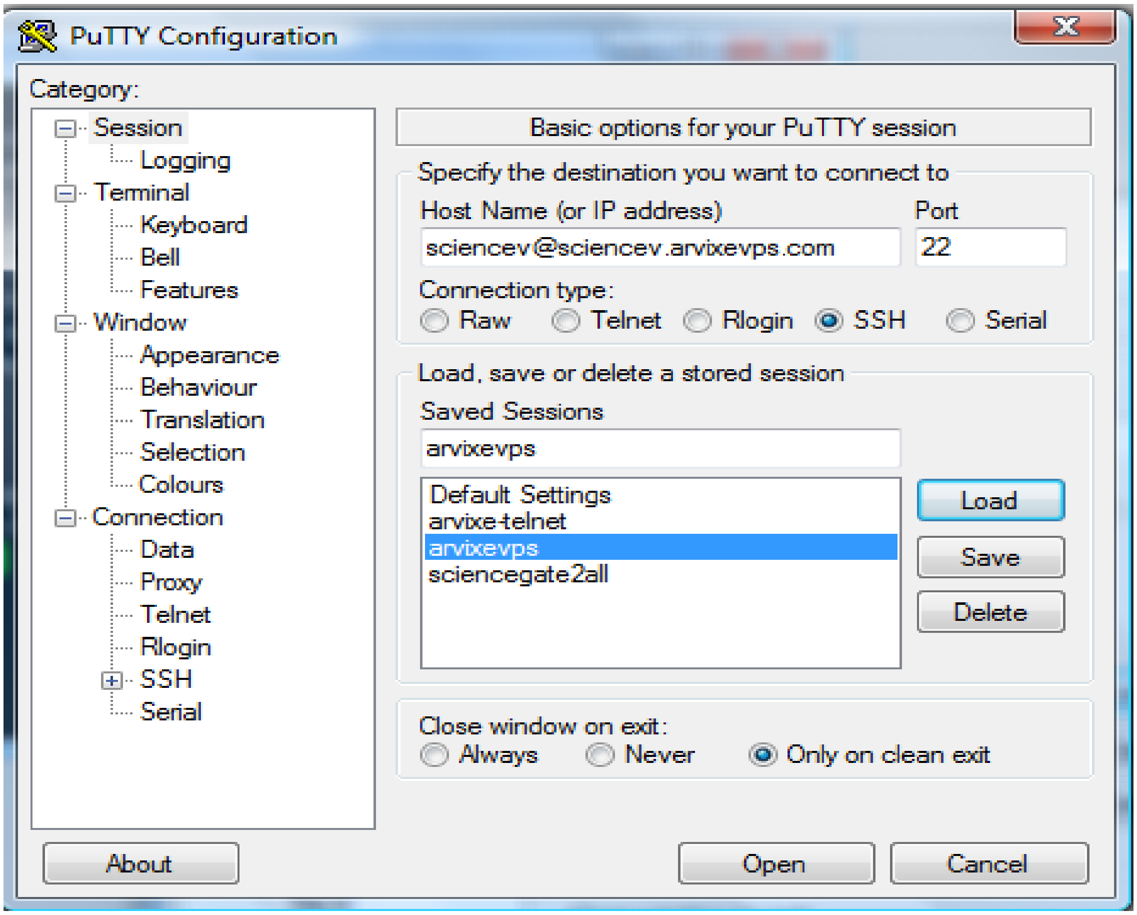


Figure 22: Sniffer Agent Started With Monitoring GUI at Server Side.

Anyway, figure (22) does not provide any information or tracing the packet sent by the front agent but it monitor only the standard ACL-Messages exchanged among standard JADE agents.

In this work, only one instance on JADE is needed thus the administrator has to start JADE manually by logging into the server and invoke the code presented in figure (19). Since we are working on a remote server where our web application is loaded, we need a terminal emulator to connect to remote server; we used PuTTY which is a free and open-source terminal emulator, and serial console and network file transfer application.

Figure (23) present the GUI window of the terminal emulator used to login into the remote server in order to start JADE environment. Error messages are handled manually by the administrator while error messages occurred along the runtime is sent to the front side agent for notification purposes; user has to reload the system once more for generating new instances of agents at server and client side.



Specify the destination you want to connect to

Host Name (or IP address): sciencev@sciencev.arvixevps.com

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

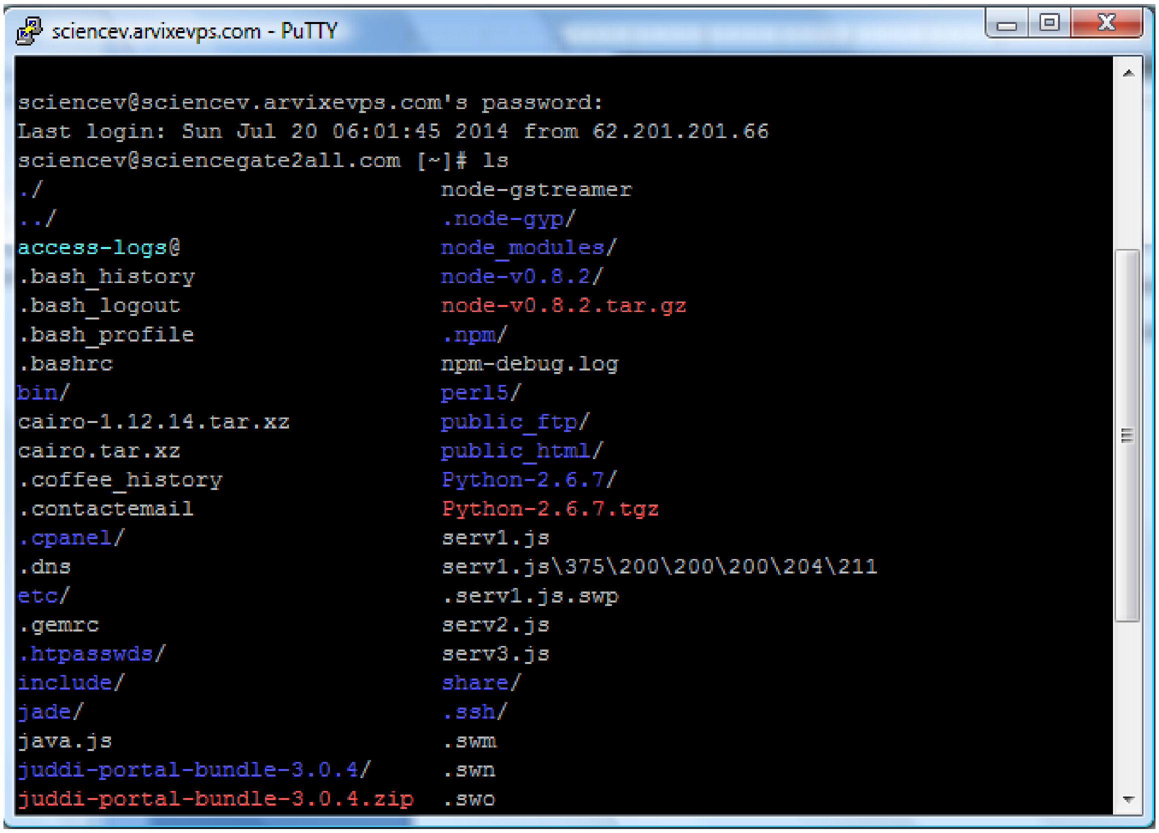
Saved Sessions: arvixevps

Default Settings
arvixe-telnet
arvixevps
sciencegate2all

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 23: Configuring Terminal Emulator to Login into the Application Server.

One important issue is making sure that JADE binary and source packages are exists as the same as the installation of proper JDK version. Figure (24) presents a screen shot of the server directory where jade is downloaded from its official site and configured to listen on its dedicated port (i.e., 1099); this was the main issue in getting a private server where shared servers don't permit subscribers to configure ports.

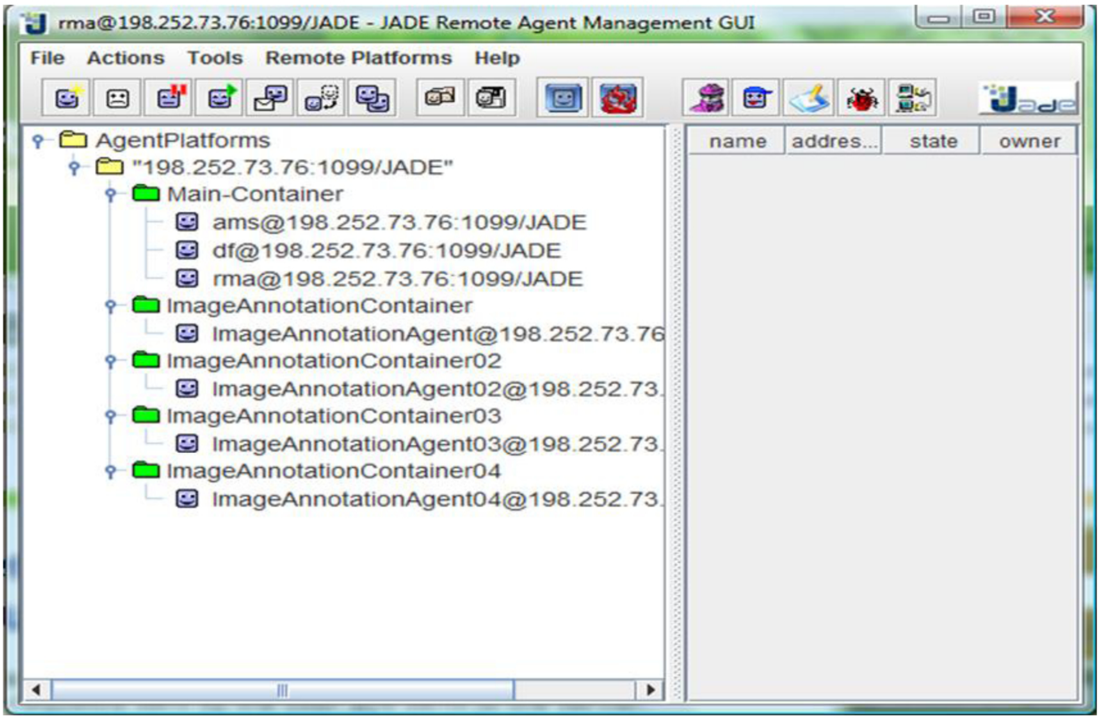


```
sciencev@sciencev.arvixevps.com's password:
Last login: Sun Jul 20 06:01:45 2014 from 62.201.201.66
sciencev@sciencegate2all.com [~]# ls
./                             node-gstreamer
../                            .node-gyp/
access-logs@                   node_modules/
.bash_history                  node-v0.8.2/
.bash_logout                   node-v0.8.2.tar.gz
.bash_profile                  .npm/
.bashrc                        npm-debug.log
bin/                           perl5/
cairo-1.12.14.tar.xz           public_ftp/
cairo.tar.xz                   public_html/
.coffee_history                Python-2.6.7/
.contactemail                  Python-2.6.7.tgz
.cpanel/                       serv1.js
.dns                           serv1.js\375\200\200\200\204\211
etc/                           .serv1.js.swp
.gemrc                         serv2.js
.htpasswds/                    serv3.js
include/                       share/
jade/                          .ssh/
java.js                        .swm
juddi-portal-bundle-3.0.4/     .swn
juddi-portal-bundle-3.0.4.zip  .swo
```

Figure 24: Console Screen shot showing that jade directory is exist.

Figure (25) presents a case study with four agents started when four users invoked our search engine. The probability of having an efficient automatic annotation is proportional to the number of users that invoked the search engine within the same domain; this is due to the sustaining of valuable annotation concepts which are revealed from the intersection of surfed images.

For presentation purposes, all four users are interested in a topic containing car but with no further details; this case study passes through two stages: first one is the retrieving stage where images are retrieved against certain query for all users and second stage which is the annotation stage where user behavior is mined in their interaction with the retrieved images.


rma@198.252.73.76:1099/JADE - JADE Remote Agent Management GUI
File
Actions
Tools
Remote Platforms
Help
name
addres...
state
owner

AgentPlatforms
"198.252.73.76:1099/JADE"
Main-Container
ams@198.252.73.76:1099/JADE
df@198.252.73.76:1099/JADE
rma@198.252.73.76:1099/JADE
ImageAnnotationContainer
ImageAnnotationAgent@198.252.73.76
ImageAnnotationContainer02
ImageAnnotationAgent02@198.252.73.
ImageAnnotationContainer03
ImageAnnotationAgent03@198.252.73.
ImageAnnotationContainer04
ImageAnnotationAgent04@198.252.73.

Figure 25: Four Back end Agents Joined the Platform corresponding to four users.

By starting the environment for JADE and launching sniffer agent, the system is ready to collect data and develop knowledge, as it will be presented in next section.

Collecting data and Mining User Behaviors

Automatic image annotation is primarily an image retrieval engine combined with a behavioral and data mining unit that inference user interaction, thus the first step in the implementation is to build the image retrieving unit. As it has been presented earlier, we count on Google image repository and Google API to request images against textual queries. Google API is a tool used to get JSON objects that represents a complete description to the retrieved images.

in this stage, the system is started to collect data (i.e., queries, retrieved images) and observe behavior of the requestors (i.e., selecting matched images); this a mining phase at two levels: data level where queries are analyzed against annotations accompanied retrieved images using LSA (Latent Semantic Analysis) and behavior level where knowledge is collected by conceptualizing user behavior in interacting resultant images. Figure (26) presents user submitted a query (car) to the system and the images retrieved from Google repository using Google custom search API.

At this stage nothing significant occurred where using Google search engine retrieve the same results but here all retrieved image URLs are pushed into a database and their notations are also embedded into corresponding records. User behavior with retrieved images is the key difference that discriminate proposed system than the traditional search engines where front side agent is starting a cyclic behavior to monitor and conceptualize user interactions.

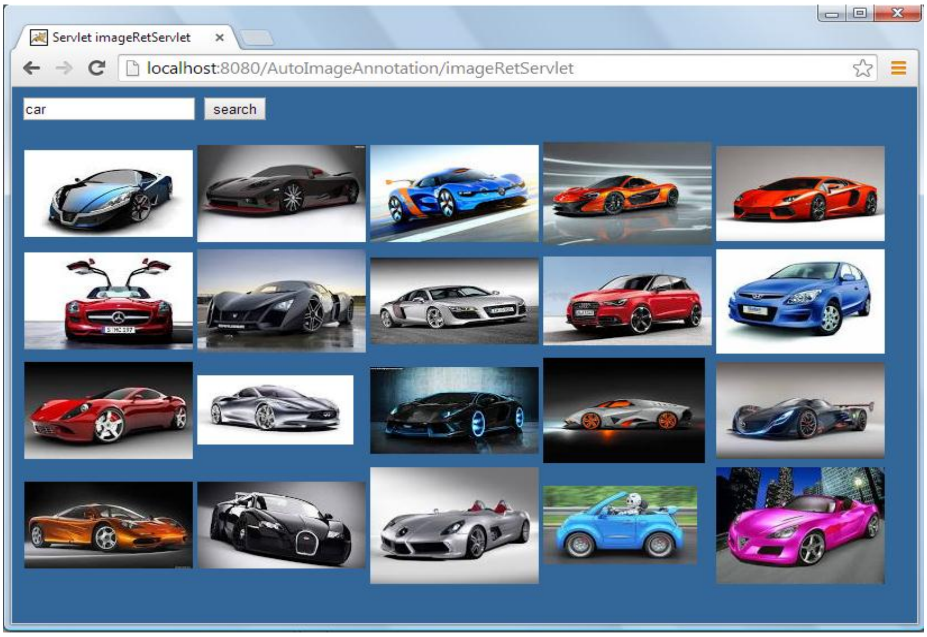


Figure26: Images Retrieved Against User Simple Query

The following are samples of the records generated due the submitting of the query and retrieving the images; it is shown from these records that image URLs are retrieved and the annotation accompanied images is extracted.

http://news.bbcimg.co.uk/media/images/72628000/jpg/72628696car1.jpg
Car reverses into Royston bungalow kitchen. Car crashed into house in Royston The driver reversed across the road, through a brick wall and into the house

href="http://www.google.com/imgres?imgurl=http://openclipart.org/image/2400px/svg_to_png/188903/cyberscooty-cartoon-car.png
Clipart - Cartoon Car

http://i.telegraph.co.uk/multimedia/archive/01249/car_ultimate_aero__1249846c.jpg
Ultimate Aero EV car: Fastest electric car in the world unveiled. Shelby Supercars will reach an extraordinary 208mph. The car with a house, which is due to go on sale and anti crash

http://www.hdwallpaperscool.com/wp-content/uploads/2013/11/mac-classic-cars-vintage-hd-wallpapers-fresh-new-desktop-widescreen-cars-images.jpg
high resolution old classic cars vintage hd wallpapers widescreen old cars images desktop

As it is shown in the generated records that annotations hold redundant words that introduce a confusion in conceptualizing images by their annotations, thus, a stemming process is started to remove redundant words (i.e., stop words, plurals,…,etc). After stemming the above annotation we have the following annotation concepts which are presented in Table 5:

Table 5: Stemming Image Annotations.

Doc No.	Annotation After Stemming
1	Car, reverse, Royston, bungalow, kitchen, crash, house, drive, across, road, brick, wall
2	Car, cartoon, clipart
3	Ultimate aero, car, fast, house, electric, world, unveil, shelpy, supercar, reach, extraordinary, sale, crash
4	High, resolution, old, classic, car, vintage, wallpaper, widescreen, image, desktop

Users are interacted with these retrieved images as it fulfill their intention, thus we collected weighted interaction of the four users as it is presented in figure (27).

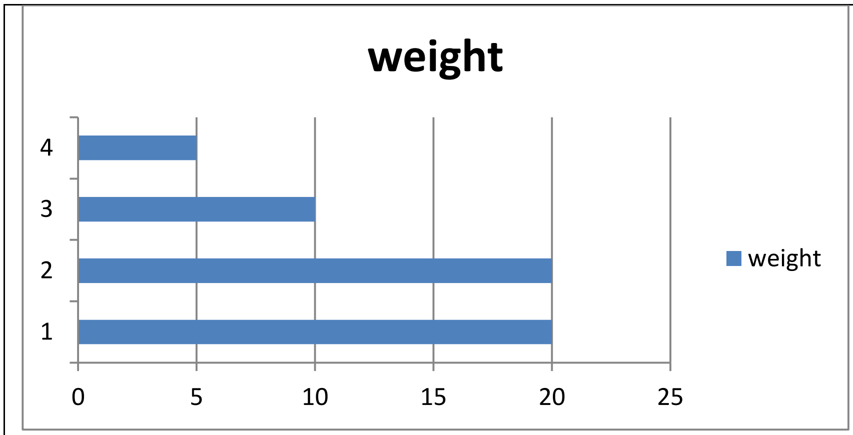


Figure 27: Evaluating User Interaction with Retrieved Images.

Doc	Weight
1	**20**
3	**20**
2	**10**
4	**4**

From figure (27), annotation of image 1 and 3 are selected and saved by the four users thus it weights higher than the others. By intersecting annotations of these images we have the probability for candidate new annotation concepts as it is presented in **Table 6**:

Table 6: Candidate Probability for New Annotation Concepts.

item	concept	Repetition	Probability
1	**Car**	**2**	**0.09**
2	**House**	**2**	**0.09**
3	**Reverse**	**1**	**0.045**
4	**Roystone**	**1**	**0.045**
5	**Crash**	**2**	**0.09**

6	**Road**	**1**	**0.045**
7	**Brick**	**1**	**0.045**
8	**Drive**	**1**	**0.045**
9	**Kitchen**	**1**	**0.045**
10	**Bungalow**	**1**	**0.045**
11	**Wall**	**1**	**0.045**
12	**Across**	**1**	**0.045**
13	**Ultimate**	**1**	**0.045**
14	**Fast**	**1**	**0.045**
15	**Electric**	**1**	**0.045**
16	**World**	**1**	**0.045**
17	**Unveil**	**1**	**0.045**
18	**Shelpy**	**1**	**0.045**
19	**Supercar**	**1**	**0.045**
20	**Reach**	**1**	**0.045**
21	**Sale**	**1**	**0.045**
22	**Extraordinary**	**1**	**0.045**

From the **Table 6**, the most nominated concepts are (house, crash, car). Next time when user submitted the query (car), more concepts are added to the query automatically due to the customization obtained by mining user behaviors on retrieved images the new annotations are added to the selected images and query is optimized to contain the new concepts, thus next time 'car' query is submitted, it will turn into a query contains the new added concepts as it is shown in figure (28).

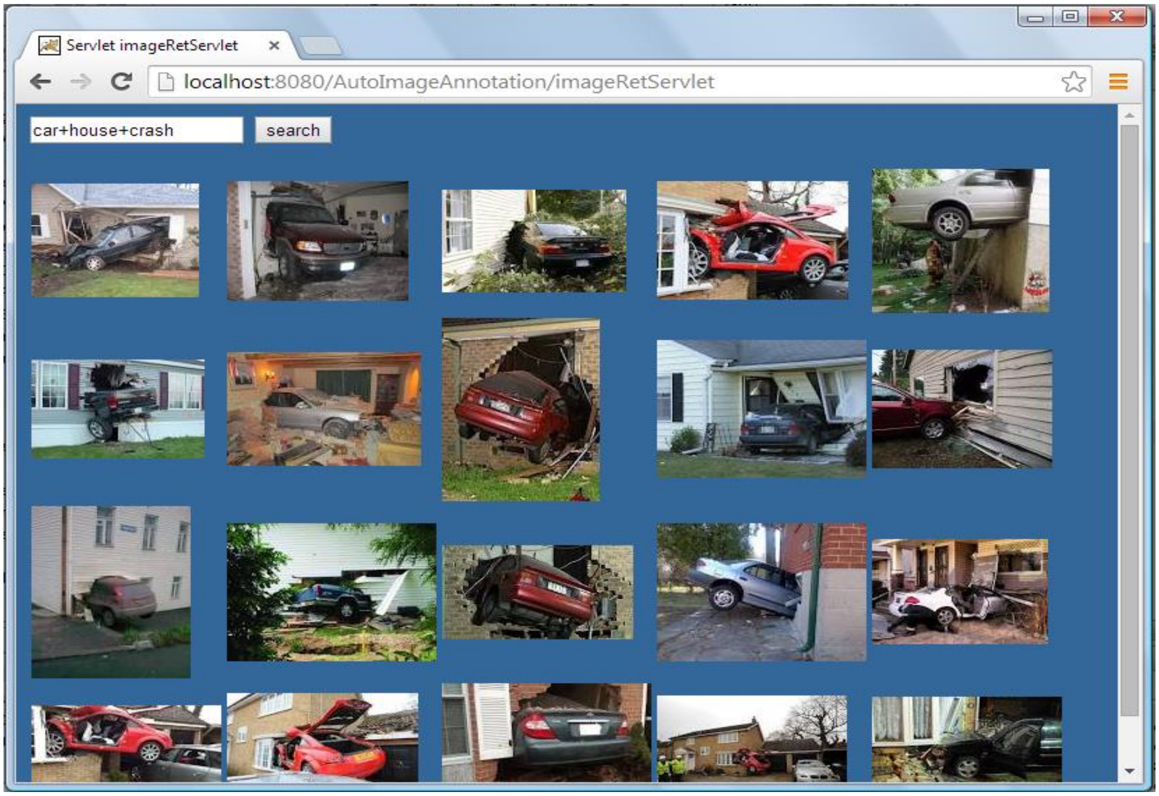


Figure 28: Images Retrieved for more Complex Query.

In figure (29), new images are retrieved that it cops with the user intend; this in the cost of losing generalization where by submitting 'car' query the proposed system will tend to retrieve images for car crash accidents. Losing generalization is handled by listing images resultant of the mining of user interactions and redundant images resultant to submitting the original query; this will maintain the generalization due to the next level interaction will change the type of concepts generated and the probability of these concepts.

The following are samples of the images retrieved for the query 'car+house+crash' where two concepts are added to the original query 'car'.

<table>
<tr><td>"http://www.salem-news.com/stimg/january012010/mva.helvetia_crash_1350.jpg</td><td> </td></tr>
<tr><td colspan="2">Car crash into house 1-1-09. Photo: Washington County Sheriff Car Crashes Into House Causing Extensive Damage – Salem</td></tr>
</table>

<table>
<tr><td>http://7bluec4-2012.wikispaces.com/file/view/car-crash-into-house.jpg/292052019/car-crash-into-house.jpg</td><td> </td></tr>
<tr><td colspan="2">Even story car crash into house blue</td></tr>
</table>

http://www.edp24.co.uk/polopoly_fs/1.1990005.1364130278!/image/118057481.jpg_gen/derivatives/landscape_630/118057481.jpg	
An Audi TT which left the road and crashed through a hedge, over two cars parked in a driveway and into a house on Long Meadow Walk in Carlton Colville	

http://rdb.maptacklocal.netdna-cdn.com/wp-content/uploads/2013/05/truck-crash-into-house.jpg	
truck-crash-into-house. Emergency Construction San Antonio, Texas. Was your home or business involved in an accident? If so, you could be losing business	

http://img.thesun.co.uk/aidemitlum/archive/01438/car-crash-house1 6 1438534a.jpg	
lucky escape as car crashes into her lounge. House about that ... amazingly nobody was hurt when the Mondeo crashed into Andrea Bar	

To study the feasibility of the approach introduced in this proposal, we deployed our system on a public gateway 3 months and the resultant statistics are shown in Table 7.

Table (7) represents user behavior when query is submitted; this is to retrieve images within certain domain. Users might just ignore resultant images or pay different level of attentions to these images; this is up to the correlation degree with his/her expected images. As it has been shown in the proposal of this dissertation, user behavior is an indication to the matching degree, anyway, user behavior can be weighted as high when images are clicked for enlargement and saved eventually (i.e., this the visited column in table 7), and it is weighted low when user give some attention to the image by moving mouse over certain images (i.e., this is the focused attribute in table 7). Selected images are the images that clicked for enlargement but never saved.

Table 7: The resultant statistics of a public gateway 3.

Domain	Images	Period (days)	Visited	Selected	Focused
Car and race	25717	47	17%	1.31%	5.3%
Celebrities	56015	30	46%	34%	70%
Accessories (smart phone)	48238	30	21%	11%	27%
Nature	33912	30	17%	9%	13%
Computer	61943	42	3%	11%	19%
Misc.	111728	47	35%	55%	31%

Table (7) represents the domains for user queries; images were collected according to the domain that the user searches in. according the results presented in table (7) it is obvious that most of the retrieved images do not match user intention when submitting the query; these results also introduce

another concept which is the topic through which the query is submitted, for example ‘Car and race’ domain only 20 % of the images gain attention at different levels and others are ignored; this is a poor outcome in term of fulfilling user request with optimized network load and user effort.
The statistics presented in Table (7) is an outcome of sampling users’ entries by installing the implemented proposed system on their gateway, the people, who are involved the survey, were randomly acquired for their input queries, the only determination was for the domain of the subjected queries.

People who are sampled for their input, had been monitored for their behavior in responding to results come up due to posting their queries, selected images are grouped and weighted according to the schemes implemented by the proposed system. All queries falling out determined categories are labeled with ‘Misc.’ class name. figure (29) represents the statistics of user behavior on individual nodes:

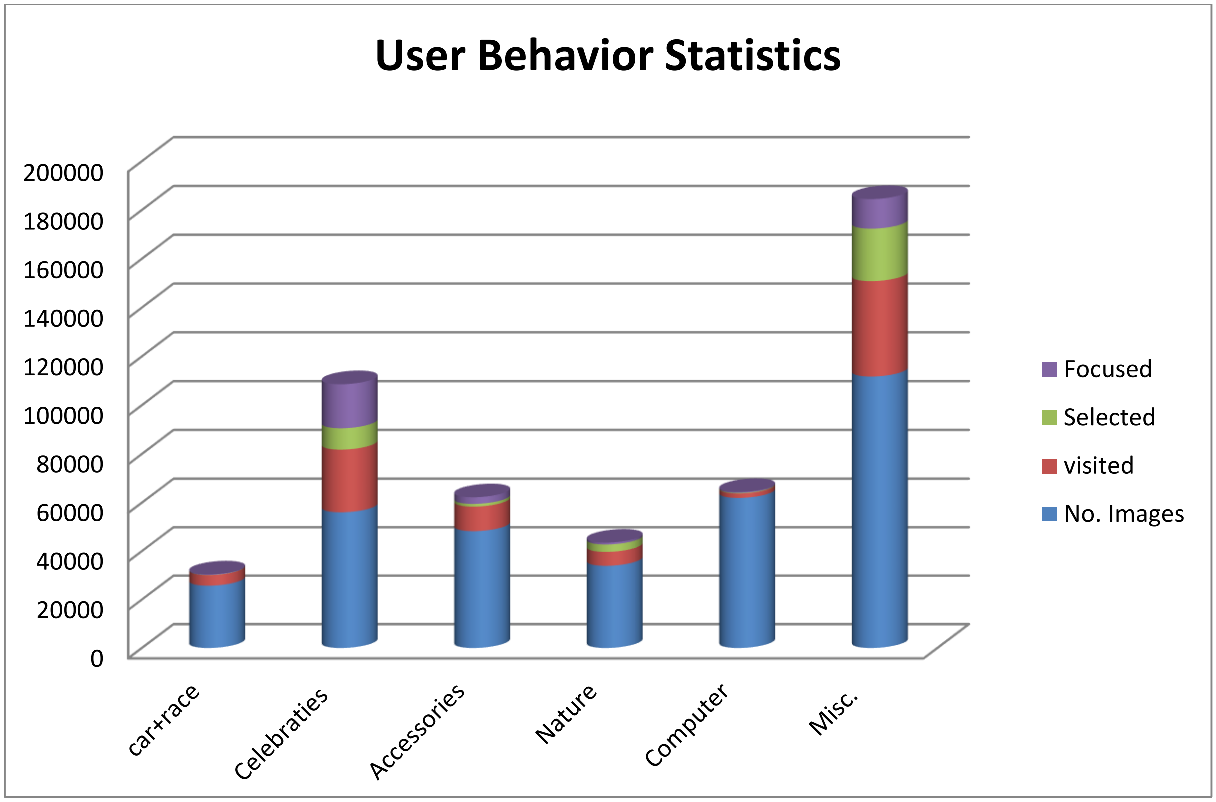


Figure 29: user behavior statistics.

Deploying knowledge to optimize Image query

In this proposal, LSA (Latent Semantic Analysis) has been deployed to optimize image query by preserving the most affected attributes in the image annotation. Table 8 presents the semantic space built for the attributes in the selected images.

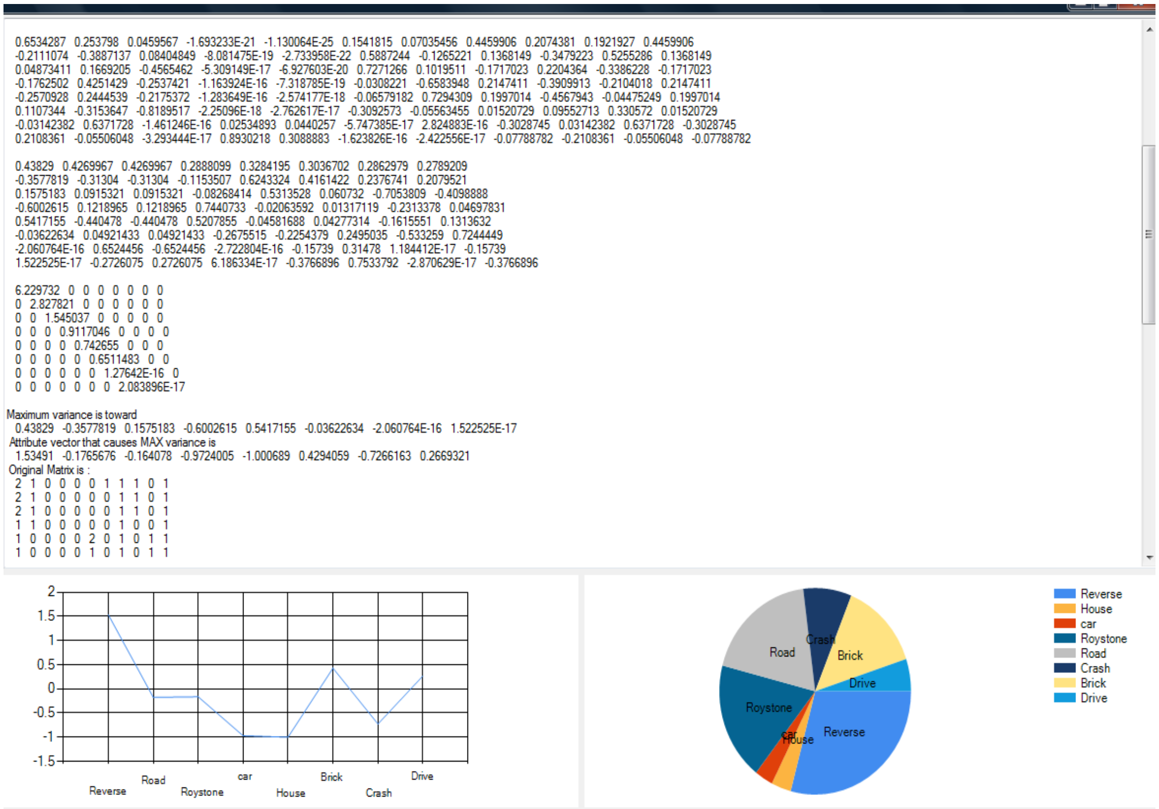
Table 8: Semantic Matrix for Image Annotation.

item	concept	Image1	Image2	Image3	Image4
1	Car	2	1	2	1
2	House	2	0	1	0
3	Reverse	2	0	0	0
4	Roystone	1	0	0	0
5	Crash	1	0	1	0
6	Road	1	0	0	0
7	Brick	1	0	0	0
8	Drive	1	0	0	0
9	Kitchen	1	0	0	0
10	Bungalow	1	0	0	0
11	Wall	1	0	0	1
12	Across	1	0	0	1
13	Ultimate	0	0	1	0
14	Fast	0	0	1	0
15	Electric	0	0	1	0
16	World	0	0	1	0
17	Unveil	0	0	1	0
18	Shelpy	0	0	1	0
19	Supercar	0	0	1	0
20	Reach	0	0	1	0
21	Sale	0	0	1	0

22	**Extraordinary**	**0**	**0**	**1**	**0**

Table (8) has been constructed as a list out for concepts available in the annotations accompanied the selected images and the queries posted to retrieve these images. To build table (8), users are monitored for their posted queries and their selected images, where cells in table (8) represent number of occurrence for each query's concept within the image annotation, for example 'house' concept which is posted in query is occurred twice within the annotation of 'image1' and does not occurred in 'image3' and 'image4'.

The fact is: the most variance captured for attribute in image annotation is the fewer candidates for annotating selected images. Figure (4-13) is a screen shot for the output of the program designed to implement the LSA; from this figure it is obvious that the less variance in the selected four images is captured from the following attributes (car, house, crash), thus these attributes are added to images where it is absent.



```
0.6534287  0.253798  0.0459567  -1.693233E-21  -1.130064E-25  0.1541815  0.07035456  0.4459906  0.2074381  0.1921927  0.4459906
-0.2111074  -0.3887137  0.08404849  -8.081475E-19  -2.733958E-22  0.5887244  -0.1265221  0.1368149  -0.3479223  0.5255286  0.1368149
0.04873411  0.1669205  -0.4565462  -5.309149E-17  -6.927603E-20  0.7271266  0.1019511  -0.1717023  0.2204364  -0.3386228  -0.1717023
-0.1762502  0.4251429  -0.2537421  -1.163924E-16  -7.318785E-19  -0.0308221  -0.6583948  0.2147411  -0.3909913  -0.2104018  0.2147411
-0.2570928  0.2444539  -0.2175372  -1.283649E-16  -2.574177E-18  -0.06579182  0.7294309  0.1997014  -0.4567943  -0.04475249  0.1997014
0.1107344  -0.3153647  -0.8189517  -2.25096E-18  -2.762617E-17  -0.3092573  -0.05563455  0.01520729  0.09552713  0.330572  0.01520729
-0.03142382  0.6371728  -1.461246E-16  0.02534893  0.0440257  -5.747385E-17  2.824883E-16  -0.3028745  0.03142382  0.6371728  -0.3028745
0.2108361  -0.05506048  -3.293444E-17  0.8930218  0.3088883  -1.623826E-16  -2.422556E-17  -0.07788782  -0.2108361  -0.05506048  -0.07788782

0.43829  0.4269967  0.4269967  0.2888099  0.3284195  0.3036702  0.2862979  0.2789209
-0.3577819  -0.31304  -0.31304  -0.1153507  0.6243324  0.4161422  0.2376741  0.2079521
0.1575183  0.0915321  0.0915321  -0.08268414  0.5313528  0.060732  -0.7053809  -0.4098888
-0.6002615  0.1218965  0.1218965  0.7440733  -0.02063592  0.01317119  -0.2313378  0.04697831
0.5417155  -0.440478  -0.440478  0.5207855  -0.04581688  0.04277314  -0.1615551  0.1313632
-0.03622634  0.04921433  0.04921433  -0.2675515  -0.2254379  0.2495035  -0.533259  0.7244449
-2.060764E-16  0.6524456  -0.6524456  -2.722804E-16  -0.15739  0.31478  1.184412E-17  -0.15739
1.522525E-17  -0.2726075  0.2726075  6.186334E-17  -0.3766896  0.7533792  -2.870629E-17  -0.3766896

6.229732  0  0  0  0  0  0  0
0  2.827821  0  0  0  0  0  0
0  0  1.545037  0  0  0  0  0
0  0  0  0.9117046  0  0  0  0
0  0  0  0  0.742655  0  0  0
0  0  0  0  0  0.6511483  0  0
0  0  0  0  0  0  1.27642E-16  0
0  0  0  0  0  0  0  2.083896E-17

Maximum variance is toward
 0.43829  -0.3577819  0.1575183  -0.6002615  0.5417155  -0.03622634  -2.060764E-16  1.522525E-17
 Attribute vector that causes MAX variance is
 1.53491  -0.1765676  -0.164078  -0.9724005  -1.000689  0.4294059  -0.7266163  0.2669321
 Original Matrix is :
 2  1  0  0  0  0  1  1  1  0  1
 2  1  0  0  0  0  0  1  1  0  1
 2  1  0  0  0  0  0  1  1  0  1
 1  1  0  0  0  0  0  1  0  0  1
 1  0  0  0  0  2  0  1  0  1  1
 1  0  0  0  0  1  0  1  0  1  1
```

Figure 30: LSA is used to Study the Effect of every Attribute over Image Selection.

The most important vector within the LSA analysis is the orthogonal matrix V which captures the relation as image to image while U matrix captures the relation attribute to attribute in the latent semantic space. Figure (30) shows the vector where the most variance occur and the variance degree for the affected attribute.

Chapter 6: Conclusions and Future Work

Conclusions

Based on the literature review and after identifying the problem to be solved. I proposed a methodology to proceed in my thesis. I used the AIAMAS to improve the images annotation Recovered from GOOGLE search engine and I used and built multi software like Agent, Jade, HTML5, LSA and SVD, SQL database for index and re-index images.....etc as shown in chapter 4. I have obtained finally the required results and I published them in international journal and conferences.

By implementing the proposed automatic annotation system and obtained results many conclusions have been introduced:

1- Multi-Agent systems are an excellent backbone for knowledge development; this due the availability of developing environment for software agent (e.g., JADE environment as an example). Along the implementation, client side software agent was an obstacle due to the absent of JavaScript code that implements software agent; this is in term of functionality and standardization. Thus we had to implement a JavaScript module loaded by the invocation of the web application; this module carry out the task of encapsulating data in ACL Message and send it to the platform host. Anyway, multi-agent system is having a promising role in developing web based mining tools where the mining backbone is the monitoring of user behaviors in his/her interaction with the web applications. Agent tendency to be social and autonomous software provides a consistent environment for knowledge consolidation and integration.

2- Image Annotation has many confusing concepts which introduce a big challenge for image retrieval engines to match user query to the proper list of images. is generated autonomously and a confidence value is assigned to each annotation; this value represents the acceptance of society for this annotation as a key index for associated image.

3- Google API is reducing the cost and efforts needed to implement search engines over the internet; this is clearly obvious when it comes to image retrieval where Google search engine has indexed billion of images in Google repositories. Google has a very massive repository of images, thus it is more convenient to address this repository rather than inferring other repositories or web pages. Furthermore, Google search engine receives millions of request for images in multiple subjects, thus this will assist, statistically, revealing more reliable annotations.

4- Automatic image annotation that based on the monitoring of retrieved images against certain query added topic concepts to the annotation of an image rather than focusing on the content of that image. The presented automatic annotation for images has introduced an innovative approach to add concepts that reflect user conceptualization to the image rather than its components.

5- Low level mapping to high level of an image has been moved to next level due to the intervention of human been in this mapping implicitly; this has been accomplishing by equipping client side agent with enough knowledge to socialize other agents and integrate local knowledge about the image with external knowledge where users' selection to images against certain query is an implicit agreement of the user that part of the accompanying annotation is mapping image contents or in other words the visual objects of the image.

   An image is annotated, in our proposal, not on the basis of the graphical objects in the image or the low level features, but on the basis of its relation to the environment, for example an image could have some planets and this image can be interpreted using low level features and semantic contents as to relate to planet science, flowers, garden or some of the like. In our approach the planets image can be categorized into drug, medicine, health or so on; this is due to society opinion.

6- LSA has been used efficiently to filter annotation concepts (i.e., natural language words) where semantic similarities among annotations attached to certain images and the set of queries posted to search engines, is an effective approach to determine and omit redundant words. The accuracy of the results is corresponding to the distribution pattern of the natural language words over the query and the annotation at the same time, where semantic similarities among annotations and queries vectors should span the semantic space of a group of images that are to be de-noised. One of the most obstacles facing this approach is the intensive calculations required by the LSA when new image added to the group of images which have been de-noised against redundant words.

The proposed system provides a robust success measure in measuring the frequency of selecting nominated images for the user against certain query. As it has been explained along the presentation of the proposed system, after capturing knowledge from distributed agents about certain images and their indexing proposed annotation, new users who posted queries are fulfilled with nominated images' due to the proposed indexing scheme' and the resultant images due to Google respond.

Now, the success factor would be number of times in which users' have selected the nominated images rather than selected other images, where the bigger ration revealed the success factor for the proposed system; this factor starts small due to the few number of images available in the custom repository but it increases with time due to continue monitoring for user activities.

Future Work

All works that can be done to improve the work:

1- The presented agent based automatic image annotation is the first step toward annotating images based on the cognitive human interpretation for these images ; this approach can be extended to identify objects composing certain images where visual objects have to be described by the accompanied annotation (i.e., image annotation after all is a description to image contents). In this work images are de-assembled into its basic objects and triggering client side agents to socialize about selected images; the intersection of annotations for images containing matched visual objects will tend to describe the image; this is theoretically accepted but experimental results can derive new innovation in this issue.

2- A second candidate approach for future work is to extend the implementation of ontological socialization among agent society, as it has been presented in current work, to more sophisticated protocol. Agents' ontology reflects the conceptualization capability of the agent to interpret user interaction with the retrieved images. Socialization

3- Current work has implemented LSA as an offline approach to study the latent relationships among annotated images that have been retrieved and the queries submitted to the search engine, as an advance future work would be to implement LSA on the fly where distributed agents can share the computation power required for implementing LSA and conduct the analysis online. A new paradigm is required and an integration scheme is demanded.

Ph.D. Student Publications

The following table shows my publications titles and publication place:

#	Paper Title	Place of Publication
1	Automatic De-Noising For Image Annotation Using Latent Semantic Analysis	Journal of Electronics and Communication Engineering & Technology (IJECET), ISSN 0976 – 6464(Print), ISSN 0976 – 6472(Online), Volume 5, Issue 1, January (2014), pp. 113-118.
2	Automatic Image Annotation using Image clustering in Multi – Agent Society	International Journal of Computer Applications Technology and Research (IJCATR), (ISSN: 2319–8656 (Online)), Volume 3, Number 8, 2014, pp. 515 - 518. Available on line: http://www.ijcat.com/index.
3	Technique for Image Deblurring Using Least Square Filter With MLMVN Network	Iraqi journal of information technology,vol.4,no.1,Baghdad,Iraq,2011.
4	Conceptual Image Indexing and Searching Technique Based on Collaborative Multi-Agents platform	education faculty- al- mustansirya university, Baghdad, 2013
5	A. Khorshid & H. Chiblè. Neural Network Signal	Poster Presentation (PhD Thesis), Lebanese university, 2nd scientific days of PhD school of

	Processing for Handwritten Character Recognition	sciences and technologies, Beirut, Lebanon, December 8-10, 2010, pp. 181.
6	Neural Network Software For Handwritten Character Recognition	Abstracts of the LAAS 17th Int. Scientific Conference, National Council for Scientific Research & the Lebanese Association for the Advancement of Science, USEK University, Kaslik, Lebanon, November 12, 2010, pp. 71.
7	ARP Session Authentication Protocol Using Multi-Agent platform and kernel level filter driver.	LAAS 18th International Science Conference Abstracts, Lebanon,Beirut,2012.
8	Automatic Image Annotation using Image Clustering in Multi Agent Society,	Book of Abstracts of the 3rd Forum Doctoral – Ecole Doctoral des Sciences et des Technologies “EDST” – Lebanese University, June 25-26, 2013, pp. 101.
9	Automatic De-noising for Image Annotation Using Latent Semantic Analysis.	20th LAAS International Science Conference “Advanced Research for Better Tomorrow”; 27-29 March 2014; Hadath, Lebanon; pp: 199-200.

References

1-Baojun Qiu, Kristinka Ivanova, John Yen, and Peng Liu, “Behavior Evolution and Event-driven Growth Dynamics in Social Networks”

2-David S. Channin MD, Pattanasak Mongkolwat, Vladimir Kleper, and Daniel L. Rubin, “Computing Human Image Annotation”

3-Chen-Ue Lee, Von-Wun Soo, and Yi-Ting Fu, “How to Annotate an Image? The Need of an Image Annotation Guide Agent”

4-Nikolaos Papadakis, Klimis Ntalianis, Anastasios Doulamis, and George Stamoulis, “An Automatic Multi-Agent Web Image and Associated Keywords Retrieval System”

5-Jian Zhang, Guang-Zhou zeng, and Zhi-Feng Li, “ The Study of a Sociality Agent Architecture Based On Role”

6-Takayuki Shiose, Tetsuo Sawaragi, Osamu Katai, and Michio Okada, “Dynamics of Reciprocal Learning by Bi-Referential Model within Multiagent Systems”

7-Weng Zu Mao, and David A. Bell," Interactive Query for Image Contents by Semantic Descriptors and Multi-Agent"

8-Jesus Favela and Victoria Meza,"Image-retrieval agent: integrating image content and text".

9- Robert L.Grossman, Chandrika Kamath, Philip Kegelmeyer, Vipi Kumar and Raju R. Namburu, "Data Mining for Scientific and Engineering Applications", Kluwer Academic Publishers, Netherlands, 2001.

10- Mark Allen Christopher, "TRUTHMARKER: A TABLET-BASED Approach for Rapid Image Annotaion", Master thesis, University of Iowa, 2011.

11- G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. PAMI, 29(3):394–410, 2007.

12- Ge Bai, "Tag-Based Social Image Retrieval: An Empirical Evaluation",

13- Masashi Inoue, "On the Need for Annotation-Based Image Retrieval", National Institute of Informatics, Tokyo, Japan, 2006

14- O. Yakhnenko and V. Honavar. “Annotating images and image objects using a hierarchical dirichlet process model”. Proceeding MDM, pages 1–7, 2008.

15- Shaoting Zhang, Junzhou Huang, Yuchi Huang, Yang Yu, Hongsheng Li, Dimitris N.Metaxas, “Automatic Image Annotation Using Group Sparsity”,

16- Chen-Yu Lee, Von-Wun Soo, and Yi-Ting Fu,”An Image Annotation Guide Agent”, 7th Pacific Rim International Workshop on Multi-Agents, New Zealand, August 2004.

17- Jaeyoung Yang, Hyunsub Jung, and Joongmin Choi, “Building Web Navigation Agents Using Domain-Specific Ontologies”, 7th Pacific Rim International Workshop on Multi-Agents, New Zealand, August 2004.

18- Alexander Kolesnikov, “ Enterprise Application Development Using Jakarta Tapestry”, MSc. Thesis, Glasgow Caledonian, 2006.

19- Nash, M., “Java Frameworks and Components”. Cambridge University Press, 2003.

20- Panagiotis Symeonidis, Ivaylo Kehayov, and Yannis Manolopoulos, " Text Classification be Aggregation of SVD Eigenvectors", Springer, USA, 2012

21- Reinhard Koch and Fay Huang (Eds), " Computer Vision-ACCV 2010 Workshops: 'Two-Probabilistic Latent Semantic Model for Image Annotation and Retrieval' ", Springer, USA, 2011.

22- NursuriatiJamil and SitiAisyahSa'adan, Proceeding, "Visual Informatics: Bridging Research and Practice", visual informatics conference, IVIC 2009, Kuala Lumpur, Malaysia, 2009.

23- Thomas K. Landauer, Danielle S. McNamara, Simon Dennis and Walter Kintsch,"Handbook of Latent Semantic Analysis", Lawernce Erlbaum Associates, Inc, USA, 2011,ISBN:978-1.4106-1534-3

24- TRICHAKIS, PAVLOS (2009) Multi Agent Systems for the Active Management of Electrical Distribution Networks, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/57/

25- Willian John Teahan,”Artificial Intelligence-Agents and Environments” Ventus Publishing, USA, ISBN:978-87-7681-528-8, 2010

26- Fabio Bellifemine, Giovanni Caire and Dominic Greenwood,"Developing multi-agent systems with JADE", John Wiley & Sons, Ltd, 2007.

27- Fabio Bellifemine, Giovanni Caire, Tiziana Tucco and Giovanni Rimassa, " JADE Progammer's Guide", TILab S.P.A, 2010.

28- Rafael H. Bordini, "Multi-Agent Programming: Languages, Platforms and Applications", Springer, USA, 2005

29- Barabara Dunin-Keplicz & Rineke Verbrugge, “Teamwork in Multi-Agent Systems: A Formal Approach”, John Wiley, USA, 0-470-02875-0, 2010

30- Willian John Teahan,”Artificial Intelligence-Agent Behaviour”, Ventus Publishing ApS, ISBN:978-87-7681-559-2,2010

31- Gerhard Weiss, “Multi-Agent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intellignece”, MIT Press, Cambridge, England,1999

Table of Figures

Table of Tables

Abbreviation

ACC: Agent Communication Channel.

ACL: Agent Communication Language.

AIAMAS: Automatic Image Annotation Multi-agent System.

AIM: Image Markup.

AMS: Agent Management System.

AP: Agent Platform.

API: Application Programming Interface.

CaBIG: Cancer Biomedical Information Grids.

CBIR: Content Based Image Retrieval.

CGI: Common Gateway Interface.

DCBIQ: Distributed Content Based Image Query System.

DF: Directory Facilitator.

DOM: Document Object Model.

EAS: Enterprise Applications.

EJP: Enterprise Java Beans.

FIPA: Framework for Intelligent Physical Agent.

GUID: Globally Unique Identifier.

HTML: Hyper Text Markup Language.

HTTP: Hypertext Transfer Protocol.

IR: Information Retrieval.

JADE: Java Agent Development Environment.

JES: Joint Equal Contribution.

JSON: Java Script Object Notation.

JSP: Java Server Pages.

LSA: Latent Semantic Analysis.

MAS: Multi-agent Systems.

MTS: Message Transport Service.

OSF: Optimized Search Function.

OWL: Ontology Web Language.

RDF: Resource Description Framework.

RDFS: Resource Description Framework Schema.

RGB: Red Green Blue.

SOAP: Simple Object Access Protocol.

SVD: Singular Value Decomposition.

TCP: Transmission Communication protocol.

URL: Uniform Resource Locator.

WEAS: Web-Based Enterprise Applications.

XML: Extensible Markup Language.

Index

Appendix: Publication of the PhD Student

Journals

1. Abbas Khorsheed, dr. Haidar Kaddoum, Technique for Image Deblurring Using Least Square Filter With MLMVN Network, *Iraqi journal of information technology*,vol.4,no.1,Baghdad,Iraq,2011.

2. Abbass Khorsheed, Conceptual Image Indexing and Searching Technique Based on Collaborative Multi-Agents platform, education faculty journal; al- mustansirya university, Baghdad, pp: 357- 375, No. 3, 2012. ISSN 1812-0380.

3. Abbass.A.Khorsheed, Hussein Chible, Giorgio Giacinto, Ammar J. Fattah; Automatic De-Noising For Image Annotation Using Latent Semantic Analysis; *Journal of Electronics and Communication Engineering & Technology (IJECET),* ISSN 0976 – 6464(Print), ISSN 0976 – 6472(Online), Volume 5, Issue 1, January (2014), pp. 113-118.

4. Abbass.A.Khorsheed, Hussein Chible, Giorgio Giacinto; Automatic Image Annotation using Image clustering in Multi – Agent Society; *International Journal of Computer Applications Technology and Research (IJCATR),* (ISSN: 2319–8656 (Online)), Volume 3, Number 8, 2014, pp. 515 - 518. Available on line: http://www.ijcat.com/index.

Conferences

5. A. Khorshid & H. Chiblè. Neural Network Signal Processing for Handwritten Character Recognition, Poster Presentation (PhD Thesis),

Lebanese university, *2nd scientific days of PhD school of sciences and technologies*, Beirut, Lebanon, December 8-10, 2010, pp. 181.

6. A. Khorshid & H. Chiblè. Neural Network Software For Handwritten Character Recognition, Oral Communication and *Abstracts of the LAAS 17th Int. Scientific Conference,* National Council for Scientific Research & the Lebanese Association for the Advancement of Science, USEK University, Kaslik, Lebanon, November 12, 2010, pp. 71.

7. A. A. khorsheed, E. Salih, H. Chible; ARP Session Authentication Protocol Using Multi-Agent platform and kernel level filter driver, *LAAS 18th International Science Conference Abstracts*, Lebanon,Beirut,2012.

8. A. Khorshid, H. CHIBLÉ, G. Giacinto, L. Raffo. Automatic Image Annotation using Image Clustering in Multi Agent Society, Book of *Abstracts of the 3rd Forum Doctoral – Ecole Doctoral des Sciences et des Technologies "EDST"* – Lebanese University, June 25-26, 2013, pp. 101.

9. Abbass Khorsheed, Hussein Chible, Giorgio Giacinto, Luigi Raffo. Automatic De-noising for Image Annotation Using Latent Semantic Analysis; *20th LAAS International Science Conference "Advanced Research for Better Tomorrow"*; 27-29 March 2014; Hadath, Lebanon; pp: 199-200.

INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATION ENGINEERING & TECHNOLOGY (IJECET)

ISSN 0976 – 6464(Print)
ISSN 0976 – 6472(Online)
Volume 5, Issue 1, January (2014), pp. 113-118
© IAEME: www.iaeme.com/ijecet.asp
Journal Impact Factor (2013): 5.8896 (Calculated by GISI)
www.jifactor.com

IJECET
©IAEME

AUTOMATIC DE-NOISING FOR IMAGE ANNOTATION USING LATENT SEMANTIC ANALYSIS

Abbass.A.Khorsheed[1], Hussein Chible[1], Giorgio Giacinto[2], Ammar J. Fattah[3]

[1]Doctoral School of science and Technology, Lebanese University, Hadtah, Beirut, Lebanon
[2]Department of Electrical and Electronic Engineering, University of Cagliari, Italy
[3]CEO Science Gate Online Research Center.

ABSTRACT

Images are important material accessed through the internet by a huge number of applications such as medical, social, mining applications. The biggest challenge facing the usage of those billion of images is the retrieving challenge. Two approaches are available to retrieve images over the internet: first one is by using textual matching between user query and image annotation, and second one is by using image contents.

This paper introduces a novel approach to remove redundant words used to annotate images; this is done by using Latent Semantic Analysis (LSA) to build the semantic space that combines queries and annotations, and then use Singular Value Decomposition (SVD) to determine variance produced by annotation words. As a last step, words with less variance are omitted.

Keywords: Image Annotation, LSA, SVD, Automatic De-Noising, Semantic Space, Singular Values

1- INTRODUCTION

From the inspection of popular image search engines such as Google, Bing and Baidu, the retrieval paradigm employed by these search engines is still based on the keywords composing the query; this query is formulated by users to initiate image search process. Users use natural language words to describe requested image, or other multimedia contents, and the responsibility of a search engine is to scan databases for a proper match. The most crucial element is the search scenario is the indexing of images, or other multimedia contents, where natural language is demanded to achieve the labeling of available images with textual description; this process is called image annotation [1,2].

Content-based image retrieval, the problem of searching large image repositories according to their content, has been the subject of a significant amount of computer vision research in the recent past. While early retrieval architectures were based on the query-by-example paradigm, which formulates image retrieval as the search for the best database match to a user-provided query image, it was quickly realized that the design of fully functional retrieval systems would require support for semantic queries. These are systems where the database of images are annotated with semantic keywords, enabling the user to specify the query through a natural language description of the visual concepts of interest. This realization, combined with the cost of manual image labeling, generated significant interest in the problem of automatically extracting semantic descriptors from images [1,2,3].

Images are annotated using different methodologies, some are manually; this when clients comment on certain images and automatically such as mining the textual text in internet pages that hold that image. Crucial challenge in image annotation is the redundant words that increase false results such as the irrelevant images returned by Google search engine [3].

The earliest efforts in the area were directed to the reliable extraction of specific semantics, e.g. differentiating indoor from outdoor scenes, cities from landscapes, and detecting trees, horses, or buildings, among others. These efforts posed the problem of semantics extraction as one of supervised learning: a set of training images with and without the concept of interest was collected and a binary classifier trained to detect the concept of interest. The classifier was then applied to all database of images which were, in this way, annotated with respect to the presence or absence of the concept [2,3].

More recently, there has been an effort to solve the problem in its full generality, by resorting to unsupervised learning. The basic idea is to introduce a set of latent variables that encode hidden states of the world, where each state defines a joint distribution on the space of semantic keywords and image appearance descriptors (in the form of local features computed over image neighborhoods). After the annotation model is learned, an image is annotated by finding the most likely keywords given the features of the image [1, 2, 3].

2- LATENT SEMANTIC ANALYSIS (LSA)

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the meaning of words. Meaning is estimated using statistical computations applied to a large corpus of text [4].

The corpus embodies a set of mutual constraints that largely determine the semantic similarity of words and sets of words. These constraints can be solved using linear algebra methods, in particular, singular value decomposition [4].

LSA has been shown to reflect human knowledge in a variety of ways. For example, LSA measures correlate highly with humans' scores on standard vocabulary and subject matter tests; it mimics human word sorting and category judgments; it simulates word-word and passage-word lexical priming data; and it accurately estimates passage coherence [4, 5].

The core processing in LSA is to decompose A using SVD (Singular Value Decomposition); SVD has designed to reduce a dataset containing a large number of values to a dataset containing significantly fewer values, but which still contains a large fraction of the variability present in the original data [3, 4, 5].

$$A = U\Sigma V^T \qquad \text{---eq.1}$$

Where

1- $EigenVector(AA^T) \rightarrow Columns(U)$
2- $EigenVector(A^TA) \rightarrow Columns(V)$
3- $EigenValue(A^TA)\ OR\ EigenValue(AA^T) \ \rightarrow \Sigma$

the first structure is the single pattern that represent the most variance in the data, after all, SVD is an orthogonal analysis for dataset, U is composed of eigenvectors of the variance-covariance matrix of the data, where the first eigenvector points to the direction which holds the most variability produced by all other vectors jointly. U is an orthogonal matrix where all its structures are mutually uncorrelated. Eigen values are representing scalar variance of corresponding eigenvectors; this way total variation exhibited by the data is the sum of all eigenvalues and singular values are the square root of the eigenvalues [4, 6].

3- TEXTUAL IMAGE INDEXING AND RETRIEVAL

In 1970s, the conventional image retrieval system used keyword as descriptors to index an image however the content of an image is much richer than what any set of keywords can express [2].

Text-based image retrieval techniques employ text to describe the content of the image which often causes ambiguity and inadequacy in performing an image database search and query processing. This problem is due to the difficulty in specifying exact terms and phrases in describing the content of images as the content of an image is much richer than what any set of keywords can express. Since the textual annotations are based on language, variations in annotation will pose challenges to image retrieval [2, 5].

4- HYPOTHESIS

Hypothesis 1: Latent Semantic Analysis (LSA) reduces the redundant annotation of an image by truncating less variant key words of the annotation.

Hypothesis 2: variation in variance-covariance natural language semantic space is analogues to visual semantic space.

5- THE PROPOSED IMAGE INDEXING AND RETRIEVAL

In this proposal images are represented by concepts it hold. Image concept is the projection of human interpretation to the visual structures within an image, hence:

$$I = \sum_{i=1}^{N} C_i \,.\, \vec{v}_i \qquad \text{--- eq.2}$$

Where I is any image and C_i is the i^{th} concept recognized with that image

$$q = \sum_{i=1}^{K} w_i . \vec{u}_i \qquad \text{---eq.3}$$

Where q is the query entered by the user, w_i is the i^{th} word within the query and $\vec{u}_i$ is the semantic unit vector. Semantic meaning for image's concept should correlate human's interpretation for that concept; hence, eq.3 is a prerequisite

$$\vec{v}_i \,.\, \vec{w}_i = 1 \qquad \text{---- eq.4}$$

The semantic space is represented by a $A_{m\,x\,n}$ matrix and this matrix is decomposed into its principal components as the following equation:

$$A = U\Sigma V^T = \sum_{i=1}^{N} \sigma_i u_i v_i^T \quad \text{---eq.5}$$

Where σ_i is the i^{th} singular value of the matrix, σ_1 and v_1^T are the most effective direction

Block similarity is measured by the following formula:

$$\theta = \cos^{-1}\frac{v_i\, v_j}{|v_i\,|\,|v_j|} \qquad \text{--- } (\mathbf{5})$$

Where

$$Min(\theta) < Threshold \rightarrow v_i \equiv v_j \qquad \text{--- } (\mathbf{6})$$

The priorities of using specific word to index and retrieve certain image is corresponding to the singular values calculated by the SVD algorithm, this way words with less singular values can be omitted from the annotation.

Σ matrix can be used as a noise filter where queries are treated as vectors within the semantic space and those who are on the same direction toward the most singular value; those queries would composed of the most effected words.

Example:

To demonstrate the effect of the proposal hypothesis, real queries have been posted through Google search engine and textual annotations for some of the return images have been extracted. The extracted annotations and posted queries have been used to build the semantic space required by LSA, after that SVD algorithm has been applied to find out what direction holds the maximum variation, as the following presents:

S1: instead-of-mowing-grass-the-plains-man-wins-car
S2: Oregon_state_police_investigating_fatal_car_crash_west_of_valley
S3:pb_man_lying_on_grass
S4: free_ems_mini_plant_cut_hair_man_grass_doll
S5: vin_diesel_actor_man_car_wheel_serious_bald
S6: two_people_car_race_arrested_grass
Q1: car_man_grass
Q2: car_crash_race

LSA is applied to the annotations and the query to construct the semantic space matrix as it is presented in figure (3):

I	Query	S1	S2	S3	S4	S5	S6	Q1	Q2
1	Man	1	0	1	1	1	0	1	0
2	Car	1	1	0	0	1	1	1	1
3	Grass	1	0	1	1	0	0	1	0
4	Crash	0	1	0	0	0	0	0	1
5	Race	0	0	0	0	0	1	0	1

Figure 3: Semantic Space of LSA based on word repetition in Annotation

The analysis steps are shown below:

U =

-0.5891	0.4065	-0.0000	0.1621	0.6793	0.0000	0.0000	0.0000
-0.6135	-0.5192	0.0000	0.4895	-0.3382	0.0000	0.0000	0.0000
-0.4837	0.4274	-0.0000	-0.5317	-0.5483	0.0000	0.0000	0.0000
-0.1459	-0.4373	0.7071	-0.4751	0.2485	0.0000	0.0000	0.0000
-0.1459	-0.4373	-0.7071	-0.4751	0.2485	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Σ=

3.3777	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	2.3183	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.9691	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.5271	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

V=

-0.4993	0.1357	0.0000	0.1237	-0.3931	-0.7500	0.0000	0.0000
-0.2248	-0.4126	0.7071	0.0149	-0.1702	0.1667	0.3830	0.2748
-0.3176	0.3597	-0.0000	-0.3814	0.2486	0.0833	0.6038	-0.4371
-0.3176	0.3597	-0.0000	-0.3814	0.2486	0.0833	-0.2208	0.7119
-0.3561	-0.0486	0.0000	0.6724	0.6471	0.0000	0.0000	-0.0000
-0.2248	-0.4126	-0.7071	0.0149	-0.1702	0.1667	0.3830	0.2748
-0.4993	0.1357	0.0000	0.1237	-0.3931	0.5833	-0.3830	-0.2748
-0.2681	-0.6013	-0.0000	-0.4753	0.3012	-0.1667	-0.3830	-0.2748

Images indexing and retrieval, due to the above analysis, are described by the following weighted vector:

$$WeightedAnnotationVector = 3.3777\,\boldsymbol{Man} + 2.3183\,\boldsymbol{Car} + 1.0\,\boldsymbol{Grass} + 0.9691\,\boldsymbol{Crash} + 0.5271\,\boldsymbol{Race}$$

From the above vector, 'Race' can be omitted from the annotation of the processed group of images

6- CONCLUSION

LSA can be used efficiently to filter annotation concepts (i.e., natural language words) where semantic similarities among annotations attached to certain images and the set of queries posted to search engines, is an effective approach to determine and omit redundant words.

The accuracy of the results is corresponding to the distribution pattern of the natural language words over the query and the annotation at the same time, where semantic similarities among annotations and queries vectors should span the semantic space of a group of images that are to be de-noised.

The most obstacle facing this approach is the intensive calculations required by the LSA when new image added to the group of images which have been de-noised against redundant words.

7- REFERENCES

1- NursuriatiJamil and SitiAisyahSa'adan, Proceeding, "Visual Informatics: Bridging Research and Practice", visual informatics conference, IVIC 2009, Kuala Lumpur, Malaysia, 2009.

2- Masashi Inoue, "On the Need for Annotation-Based Image Retrieval", National Institute of Informatics, Tokyo, Japan, 2006.

3- Reinhard Koch and Fay Huang (Eds), "Computer Vision-ACCV 2010 Workshops: 'Two-Probabilistic Latent Semantic Model for Image Annotation and Retrieval' ", Springer, USA, 2011.

4- Thomas K. Landauer, Danielle S. McNamara, Simon Dennis and Walter Kintsch, "Handbook of Latent Semantic Analysis", Lawernce Erlbaum Associates, Inc, USA, 2011, ISBN:978-1.4106-1534-3.

5- NursuriatiJamil and SitiAisyahSa'adan, Proceeding, "Visual Informatics: Bridging Research and Practice", visual informatics conference, IVIC 2009, Kuala Lumpur, Malaysia, 2009.

6- Panagiotis Symeonidis, Ivaylo Kehayov, and Yannis Manolopoulos, "Text Classification be Aggregation of SVD Eigenvectors", Springer, USA, 2012.

7- Abhishek Choubey, Omprakash Firke and Bahgwan Swaroop Sharma, "Rotation and Illumination Invariant Image Retrieval using Texture Features", International Journal of Electronics and Communication Engineering & Technology (IJECET), Volume 3, Issue 2, 2012, pp. 48 - 55, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.

8- Madhubala Myneni and Dr.M.Seetha, "Feature Integration for Image Information Retrieval using Image Mining Techniques", International Journal of Computer Engineering & Technology (IJCET), Volume 3, Issue 3, 2012, pp. 273 - 281, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.

9- K.Ganapathi Babu, A.Komali, V.Satish Kumar and A.S.K.Ratnam, "An Overview of Content Based Image Retrieval Software Systems", International Journal of Computer Engineering & Technology (IJCET), Volume 3, Issue 2, 2012, pp. 424 - 432, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.

International Journal of Computer Applications Technology and Research
Volume 3– Issue 8, 515 - 518, 2014

Automatic Image Annotation using Image clustering in Multi – Agent Society

Abbas Akram Khorsheed
University of Mustansiriyah,
College of science, Computer
Science Department, Iraq

Hussein Chible
Doctoral School of science and
Technology, Lebanese
University, Beirut, Lebanon

Giorgio Giacinto
Cagliari Univesity,
Electronic Engineering
Department, Italy

Abstract: Image annotation is the natural language description for essential blobs within an Image. Many methodologies have recruited to reveal image semantics and represent it as annotation. Due to the exclusionary growth of number of images distributed over massive repositories, the task of manual annotation is tedious and over killing in term of times and efforts. This paper is presenting automatic annotation system based on the interaction between intelligent agents. Agent interaction is synonym to socialization behavior dominating Agent society. The presented system is exploiting knowledge evolution revenue due to the socialization to charge up the annotation process

Keywords: Social effect, Automatic annotation, intelligent agents

1. INTRODUCTION

Type Style and Fonts Image annotation is one of the most used methods to retrieve images from enterprise repositories by matching user text queries to these annotations. In general, annotation is represented as metadata or keywords assigned to digital images based on image contents [1][2].

More than 200 billion images are accessible online and the number is continuously growing [3] due to the numerous number of sources as digital cameras, mobile phones and other devices. This brings a great challenge in retrieving designated images which are identified by a unique number over the internet, i.e., the URI (Unique Resource Identifier), that is used to access each image over the web. Social decision theory [1] extends the theory of individual decisions to decisions made by the interaction of a group of agents.

Recent systems like Lable Me and Amazon mechanical turk distribute image annotation and evaluation tasks to Internet users. The volume of annotations generated from such crowd-sourcing techniques helps reduce the burden on experts without significantly sacrificing the quality of annotations. The annotators are provided with detailed instructions on how to best select labels that can be directly used for concept modeling. This ensures that relatively good quality annotations are generated for object detection, and relevance estimation tasks. It is shown that crowd-sourcing is a reasonable substitute for repetitive expert annotations, when there is high agreement among annotators.

Other sources of image annotations are collaborative games and social media sharing which undoubtedly represent the fastest growing labeled image collections in the world[3][4].

In this paper the multi Agent paradigm is proposed to simulate the social behavior of humans in developing knowledge regarding certain subjects. The JADE (Java Agent Development) environment has been used because it allows building multi agent platforms thanks to the utilities and wide spectrum classes provided by that environment.

2. RELATED WORKS

Many models were produced to characterize automatic image annotation frameworks. The exact specification of automatic image annotation is not yet established and researchers are investigating different approaches continually. Anyway, recent approaches can be categorized into two categories: one category is the semantic interpretation of image contents, while the other category is drawn from the epistemology field, where knowledge is revealed from the interaction among sources of knowledge. The society can be represented as a network of knowledge resources, and knowledge can be sustained or rejected upon the interaction among these resources. In this section we will focus upon previous efforts within the second category due to the orientation of this paper.

In [5] a model has been proposed to formulize the growth dynamics in social networks; in this model a great attention has been presented to the effect of node behavior, and how it affects the behavior of other nodes, and this eventually will affect the growth of the network. In term of knowledge evolution due to socialization; this model has a lot in common with our approach, though it has nothing to do with image retrieval system. The key similarities are:

1. The behavior-awareness where the interaction of node (i.e., the co-author s) with certain events (e.g., papers) is to be realized as a potential relationship among those nodes. In fact this approach develops knowledge at the network level, which helps increasing the growth factor of social network and, eventually, the productivity of such a network.

2. The clustering-coefficients where the tendency of grouping is related to the factor compose of these coefficients.

In [6] an ontological approach was presented to accomplish a computing model aimed to annotate images on two levels: Image Annotation and Annotation of Annotations; this model.

is focused on queries for annotations using the National Cancer Institute's Cancer Biomedical Informatics Grid's (caBIG) Annotation and Image Markup (AIM) project

The AIM project defines an ontology of annotations and image markup, a UML information model and provides the extensible markup language (XML) artifacts for creating them. A long-term vision of the AIM project is for large collections of annotations to be created in conjunction with the already large collections of clinical and research medical images. This will allow query of annotation, not only for the retrieval of relevant images, but also for the correlation of image observations and their characteristics with biomedical data including genomic expression.

In that paper many concepts are coherent with what we presented in our work in the area of retrieving images based on associated annotations, but this approach does not introduce autonomous annotation in any context, and it does not consider the behavior of image requesters; knowledge can't be developed to cluster images which is a crucial element in automatic image annotation strategy.

The model presented in [6] exploits annotations to build a semantic network among images, while our work provides autonomous annotation schema based on the behavioral interpretation of the user. The AIM project can be integrated with what we are presenting to provide consistent ontological environment for image retrieval and annotations. The same annotation context is presented by [7] and [4] but both depend on the retrieval and extraction of knowledge from the resources available on the global net.

In [8] a novel system is presented to exploit the format of multimedia sharing web sites in order to discover the underlying structure; this has been used to allow later, more sophisticated mining tasks for these sites to infer knowledge about certain images. Again, we have many features in common with these approaches, but still the effect of the behavioral responses of the users is absent.

In [9] a study for establishing a stable architecture for socialization is conducted and conclusion has been reached out along this study which is: in a society of agents there are three main parameters that enforce the stabilization of the architecture; these are: take on roles, play roles and locate in some society organization at all time. In our proposal, the society composed by agents is maintained stable by strict discipline through which roles are fairly distributed, and all agents are capable of playing these roles by accurate interpretation of client behavior. Furthermore, we adopt fixed organizational distribution of the agents which sustain the stability. In our proposal, the specification of the problem domain has different characterization due to the potential tendency toward clusterization on two different levels: the host level, and the network level. This approach has its roots back to [10] where a study addressed the fault assumption of regarding multi-agent systems as single learning system which is a wrong assumption due to the intuitive tendency to introduce social activity with neighbors rather than communicating with other far agents. This dual capabilities of an agent's referencing, i.e., self-referential, and social-referential, has been presented by [10] as a bi-referential model, in which each referencing capability is implemented by an evolutional computation method of classifier system.

In our referential model the evaluation function is global and updated on the fly by delivering knowledge to central a repository that holds the annotation for images. The annotations are revealed and referenced based on a confidence degree assigned to that annotation. In our referential model, the behavior of the evaluation function is dynamic due to the continuous change of confidence degree of annotation; this is due to the activities produced by the client clusterization behavior (i.e., the self-referential model).

Interactive query for images' content by semantic descriptors is an effort presented in [11]; this effort introduced a distributed content-based image query system (DCBIQ) based on the WWW. A model was proposed to integrate knowledge from image processing, semantic descriptor, multi-agent, and WWW navigation. Again in this model the image content plays the essential role in describing the image, thus low level extraction methodologies are more important than the opinion of the social communities which are using it.

In our proposal, the knowledge obtained by social interaction is more important than low level features like colors, textures or spatial relationships, and even semantic interpretation of image contents is not important as the social opinion about the image and its relation to other images or domains.

In [12] an attractive model is presented where a web-based image digital library is proposed; in this library agent system was used to traverse part of the web page looking for images that fit certain criteria. The methodology used by the agent is based on detecting URLs within web pages that refer to images, and when such URLs are encountered, then the text that is associated to that image is inferred for correlation with other features such as topic name, domain that this image falls in, or any other matching criteria. In our proposal the same ontology for allocating text accompanied the image is used as the following matched methodologies:

1-

∀Image∃tag∃txt((presenting(image,tag) AND Asso(tag,txt)) $\longrightarrow$ Asso(image,txt) AND Select(txt))

2-

∃paragraph ∃hyper ((hyper ∈ paragraph) $\longrightarrow$

Select(Paragraph))

3-

∃page∃ time ∃!title((image ∈ page) AND has(title,page) $\longrightarrow$ Associate(image,title)AND Select(title))

The main novelty of our approach is that we don't design a mining agent that is responsible on inferring web pages, but we exploit Google search APIs which are published over the web. The only web page we analyze is the results of the Google search APIs and don't investigate individual pages.

3. THE PROPOSED SYSTEM

This paper will focus on new category which is the dominant tags of the image as it is recognized by the society. Image repository (RTI) is a database holding labeled images (i.e. images tagged with annotation), thus it can be represented by eq.1

$$R_T^I = \sum_{i=1}^{k} (I_i , Anno_i) \qquad \text{eq.1}$$

Where $Anno_i = \sum_1^m c_j$ collection of concepts (i.e., these concepts are keywords, tagged or labels), and let the query made by the user in order to request images is representing by the following equation:

$$Query_i = \sum_1^n q_j$$ eq.2

∀ indexed(⟦imgae⟧_i) ∃ c ∈ Anno(⟦image⟧_i) so that c ∉ Anno(⟦image⟧_j) for all j ≠i

SearchResult($Query_i$) = $\sum_{j=1}^{M} images_j \;\; iff$

$Query_i \cap Anno\left(image_j\right) \neq \emptyset$

and OSF($Query_i$) $\neq c$

OR

SearchResult($Query_i$) =$image_i \; iff$

OSF(Query) = c

Behavior	Weight
Saved	5
Selected, saved	4
Revisited, saved	3
Revisited	2
Highlighted	1

User interaction behavior with the resultant list of images is weighted according to above table.

Hypothesis 1: Automatic annotation member is broadcasting candidate annotation for queried images.

Prove:

Let f(query) be a mapping function that maps images from the huge repositories spread over the internet to the desired image list requested by the user, such that

$$f{:}Re \longrightarrow De$$

Where Re is the huge repository over the internet and De is the desire domain where resultant of f(query) satisfies client request. The input to this function is the query entered by requestor and the output is a scalar value represents confidence degree

$f(query) =$ 0 Where 0: not desired

and 1: desired, thus

0= De

$$f(query) = \; v \left\{ < o < 1 \right.$$

0 ≠ De

Where

$v = T.W_{Agent}^{image}$ Which is the total weight produced by the interaction between the requestor client and the resultant list of images. If v ≥threshold then agent will broadcast a data structure composed of the following fields (Image URI , query, f(query)).

Definition: dominant annotation is the candidate new annotation for image being queried by society of agents, where f(query) for I want here sigmoid function to be the decision function to decide that certain annotation is to be added to the image annotation list .

Hypothesis 2: social group add new annotations to image

Let :

$$Query_A \cap Anno\left(image_j\right) = S_A$$

and $= \sum_{k=1}^{K} weight(behavior)$

$$Query_B \cap Anno\left(image_j\right) = S_B$$

and $= \sum_{l=1}^{L} weight(behavior)$

$$\vdots$$

$$Query_n \cap Anno\left(image_j\right) = S_n$$

and $T.W_{Agent\_n}^{image} = \sum_{m=1}^{M} weight(behavior)$

$T.W_{Agent_{ij}}^{image_j}$ is Total weight produced by $Agent_{ij}$ for $image_j$

Then

Added annotation set S = $S_A \cup S_B \dots \cup S_n$ is a set of new valuable annotations to be added

to $image_j$ with a binding value $T.W_{Agent_{ij}}^{image_j}$, hence the resultant set is only a candidate

annotation, it has to be dominant to get corresponding image get indexed with.

4. SOCIAL EFFECT

Social effect over Automatic Annotation Society will be treated in this section. In social environment, members are investigating propositions based on total weight granted by trusted members of the society. In this proposal we assume that all members are trustful and other members of the society are considering their weight evenly. Let us first define a new function that describes the acceptance of the society for the candidate annotation to be a dominant annotation, and the corresponding image can be indexed with. In this paper we propose the social effectiveness function to be a sigmoid function, due to the properties of this function especially the continuity and flexibility, hence $f(query)$ is defined as

$$SocialAcc(Image) = \frac{1}{1 + e^{-Su(query)}}$$

Where $su(query)$ is a total weight gained from all agents involved in the automatic annotation system and it is represented as the following:

$$Su(query) = v_1 + v_2 + \ldots + v_n$$

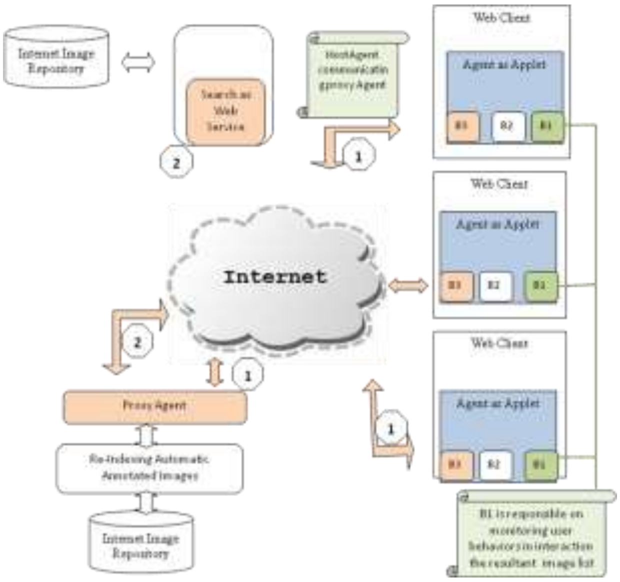

Internet Image Repository
Search as Web Service
2
HostAgent communicatin g proxy Agent
1
Web Client
Agent as Applet
B3
B2
B1
Web Client
Agent as Applet
B3
B2
B1
Internet
2
1
Proxy Agent
Re-Indexing Automatic Annotated Images
Internet Image Repository
1
Web Client
Agent as Applet
B3
B2
B1
B1 is responsible on monitoring user behaviors in interaction the resultant image list

Figure 1: General scheme of social basic automatic annotation system

The role of Agent in this proposal is characterized by three behaviors, as presented in figure(2)and are briefed as the following:

Web Service Integration (B3): this behavior encapsulates web service integration functionalities by implementing SOAP based invocation to Google web service.

Socialization Behavior(B2): this behavior is responsible on socializing other agents within the platform to determine dominant tags for an image.

User behavior monitoring (B1): this module is responsible for monitoring selections made by the user after querying the Google search Web service. Highlighted images are grouped in clusters and socialization behavior is signaled. The following behaviors are considered: Highlighted images, selected images, revisited images and saved images.

Proxy Agent: this agent is responsible of initiating the communication session over the internet. Proxy Agent is a crucial element in grant multi-agent system the ability to communicate over the internet. This Agent resides at the server side.

Host Agent: this Agent is an instance constructed at the client side and monitor his/her behavior and report back to Proxy Agent his observations.

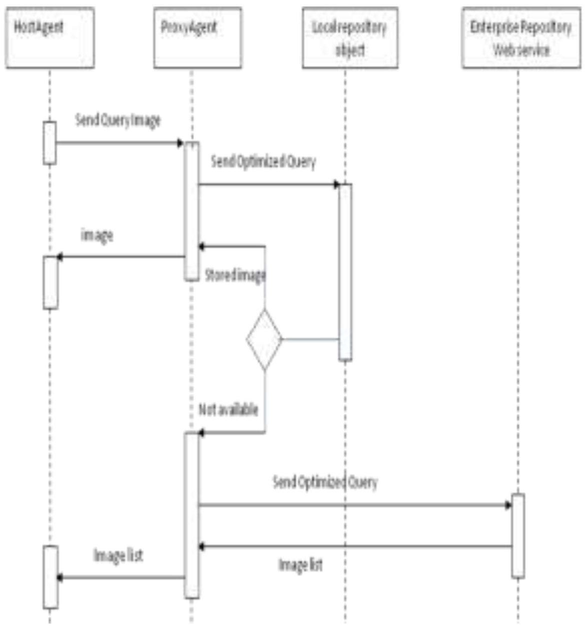


Figure 2: sequence diagram of complete 2-tier image annotation session

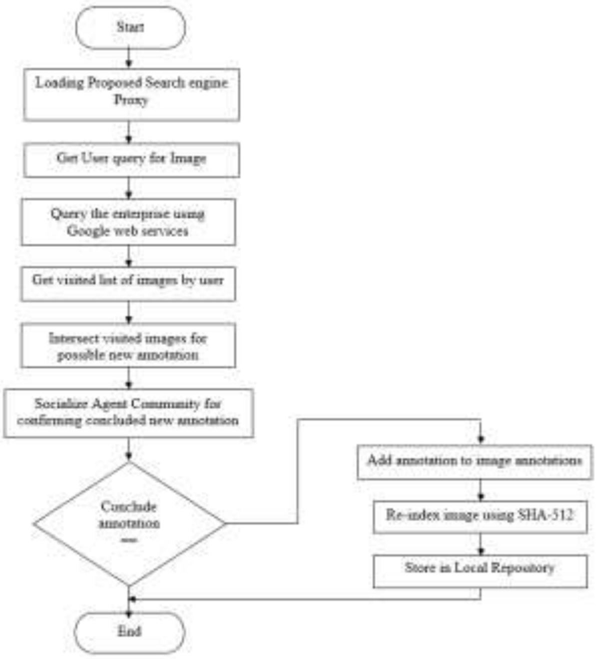

Start
Loading Proposed Search engine Proxy
Get User query for Image
Query the enterprise using Google web services
Get visited list of images by user
Intersect visited images for possible new annotation
Socialize Agent Community for confirming concluded new annotation
Conclude annotation
Add annotation to image annotations
Re-index image using SHA-512
Store in Local Repository
End

Figure 3: Automatic Image Annotation Procedure Using Multi-Agent socialization

.

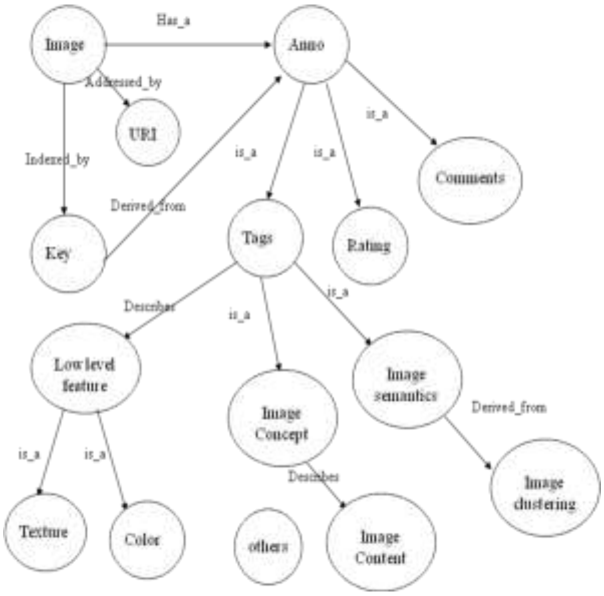


Figure 4: Ontology to conceptualize Image Annotation

5. RESULTS

Example:

Phase 1:

$client_1$ has posted the following query through the Chrome internet explorer

$$query_1 = \{ car, rental, company, race\}$$

After posting that query, 37,212 images have been listed in the internet explorer.

$client_1$ has selected and saved the following image

The session manager agent, which has HTTP listener, captured the URL or URI corresponding that image.

The following is the URL

Imgurl:http://www.koopman-racing.nl/images/sd2_1559.jpg

This image will be indexed using $query_1$.

Phase 2:

$client_2$ and $client_3$ have posted queries as the following

$$query_2 = \{ aspiring, driver, koopman, race\}$$

with max weight (5)

$$query_3 = \{ race, rental, koopman, car\}$$

with weight (4)

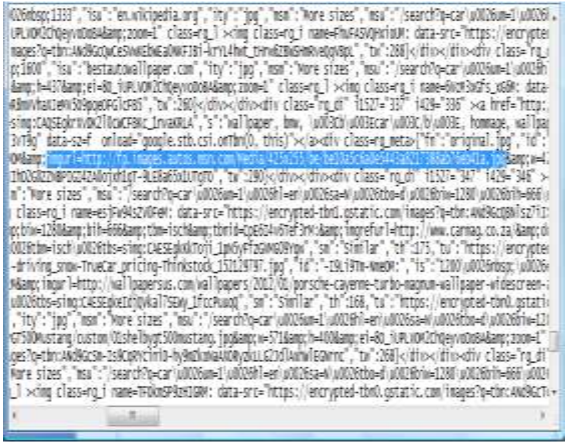
$$query_1 \cup \; query_2 = \{aspiring, driver, kooman\}$$

these annotations are to be more convenient to be used as indexing due to its weight factor and by socializing it to other clients like $client_3$ the highest effective annotations will be $\{ aspiring, driver\}$, This is for the same URL.

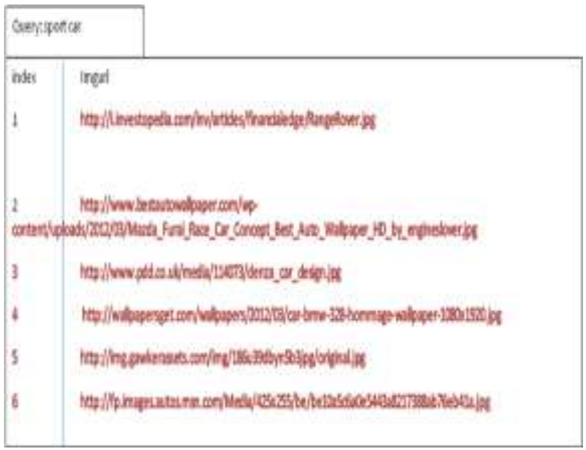
https://secure.booking.com/confirmation.fr.html?aid=350433;label=edr-xmlvswl-fr-users;sid=05686c51355c9e5ba1e2a8d843e2c461;dcid=2;bn=608419725;hostname=hotels.edreams.fr;pincode=6604#print

Algorithm1:
Automate Image Search Using Google

```
•   Procedure: Search Web
•   Input: query As String
•   Output: array of imgurls
•   Begin
•       Initialize user Query = query;
•     Initialize GoSearchConnection as URLConnection to Google URL + user Query;
•   Set GoSearchConnection Properties as
•           Method = 'GET';
•           Char-set = 'utf-8';
•           User-Agent = 'Mozila-4.0';
•    GoSearchConnection. Open;
•    Get input Stream from GoSearchConnection to stream Reader;
•      while stream Reader has imgurl do
•           add current imgurl to imgurl_list;
•   return imgurl_list;
End;
```

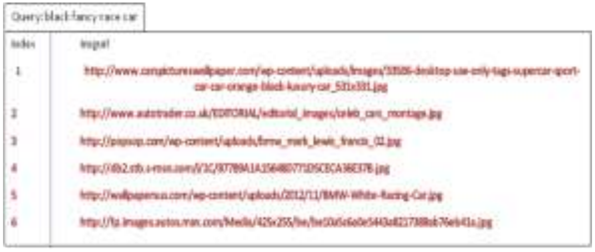


```
026nbsp;1333","isu":"en.wikipedia.org","ity":"jpg","msn":"More sizes","msu":"/search?q=car\u0026um=1\u0026
UPLX0M2ChQeyv0Q8&amp;zoom=1" class=rg_l ><img class=rg_i name=FhuF4SVQHxiQuM: data-src="https://encrypte
mages?q=tbn:ANd9GcQwCeSVwEbWEaDNKFI8i-krYL4hwt_tHrw8ZBkSHmRveQgVBpL","tw":288}</div></div><div class="rg_
p;1600","isu":"bestautowallpaper.com","ity":"jpg","msn":"More sizes","msu":"/search?q=car\u0026um=1\u0026h
&amp;h=437&amp;ei=80_iUPLX0M2ChQeyv0Q8A&amp;zoom=1" class=rg_l ><img class=rg_i name=0WM3a2fs_x6M: data-
A9mVhaCie4N509poeOFGlCF65","tw":280}</div></div><div class="rg_di" i1527="337" i429="336" ><a href="http:
simg:CAQSEgkrWOK2l0cwCF8Kc_1nvaKRLA","s":"wallpaper, bmw, \u003Cb\u003Ecar\u003C/b\u003E, hommage, wallpa
3vT9g" data-sz=f onload="google.stb.csi.onTbn(0, this)"></a><div class=rg_meta>{"fn":"original.jpg","id":
04&amp;imgurl=http://fp.images.autos.msn.com/Media/425b255/be/be10a5c6a0e5443a021[illegible]a.jpg&amp;w=4
1hD03G82DW8PDG242A0ojxd1qT-9LE8a65x1UTqT0","tw":290}</div></div><div class="rg_di" i1527="347" i429="346" >
n":"More sizes","msu":"/search?q=car\u0026um=1\u0026hl=en\u0026sa=N\u0026tbo=d\u0026biw=1280\u0026bih=666\
 class=rg_i name=esjFw94s2V0FeM: data-src="https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcQ8NIsz7I1
p;biw=1280&amp;bih=666&amp;tbm=isch&amp;tbnid=CpE6D4v6TeF3rM:&amp;imgrefurl=http://www.carmag.co.za/&amp;d
0026tbm=isch\u0026tbs=simg:CAESEgkdKToji_1pm6yFf2GAMG09Yqw","sm":"Similar","th":175,"tu":"https://encrypte
-driving_snow-TrueCar_pricing-Thinkstock_151129797.jpg","id":"-19L19Tm-NmeOM:","is":"1280\u0026nbsp;\u0026
M&amp;imgurl=http://wallpapersus.com/wallpapers/2012/01/porsche-cayenne-turbo-magnum-wallpaper-widescreen-
u0026tbs=simg:CAESEgkeIdQVkal7SEWy_1FccPuwqQ","sm":"Similar","th":168,"tu":"https://encrypted-tbn0.gstati
,"ity":"jpg","msn":"More sizes","msu":"/search?q=car\u0026um=1\u0026hl=en\u0026sa=N\u0026tbo=d\u0026biw=12
GT500Mustang/custom/01shelbygt500mustang.jpg&amp;w=571&amp;h=400&amp;ei=80_iUPLX0M2ChQeyv0Q8A&amp;zoom=1"
ges?q=tbn:ANd9GcSm-Is9CqRYCinl0-hy9mdkoNa4A0RyzkLL623dlAwnwlEGWnnc","tw":268}</div></div><div class="rg_di
More sizes","msu":"/search?q=car\u0026um=1\u0026hl=en\u0026sa=N\u0026tbo=d\u0026biw=1280\u0026bih=666\u002
_l ><img class=rg_i name=TF0kmSP9zHIGRM: data-src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT
```



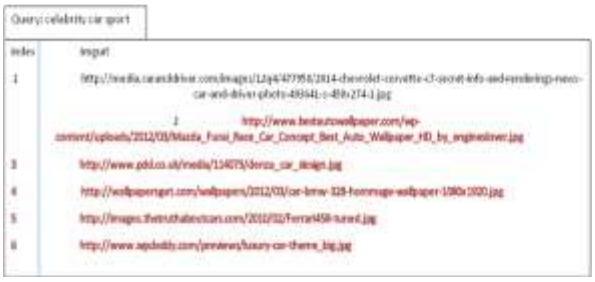
Query: sport car

index	Imgurl
1	http://i.investopedia.com/inv/articles/financialedge/RangeRover.jpg
2	http://www.bestautowallpaper.com/wp-content/uploads/2012/03/Mazda_Furai_Race_Car_Concept_Best_Auto_Wallpaper_HD_by_enginelover.jpg
3	http://www.pdd.co.uk/media/114073/denza_car_design.jpg
4	http://wallpapersget.com/wallpapers/2012/03/car-bmw-328-hommage-wallpaper-1080x1920.jpg
5	http://img.gawkerassets.com/img/186x39dbyn9b3jpg/original.jpg
6	http://fp.images.autos.msn.com/Media/425x255/be/be10a5d6a0e5443a8217388ab76eb41a.jpg



Query: black fancy race car

index	imgurl
1	http://www.coolpictureswallpaper.com/wp-content/uploads/Images/33506-desktop-size-only-tags-supercar-sport-car-car-orange-black-luxury-car_531x331.jpg
2	http://www.autotrader.co.uk/EDITORIAL/editorial_images/celeb_cars_montage.jpg
3	http://popsop.com/wp-content/uploads/forma_mark_lewis_francis_01.jpg
4	http://sb2.rdb.s-msn.com/i/1C/87789A1A156480771D5CECA36E37B.jpg
5	http://wallpapersus.com/wp-content/uploads/2012/11/BMW-White-Racing-Car.jpg
6	http://fp.images.autos.msn.com/Media/425x255/be/be10a5d6a0e5443a8217388ab76eb41a.jpg



Query: celebrity car sport

index	imgurl
1	http://media.caranddriver.com/images/12q4/477958/2014-chevrolet-corvette-c7-secret-info-and-renderings-news-car-and-driver-photo-493441-s-450x274-1.jpg
2	http://www.bestautowallpaper.com/wp-content/uploads/2012/03/Mazda_Furai_Race_Car_Concept_Best_Auto_Wallpaper_HD_by_enginelover.jpg
3	http://www.pdd.co.uk/media/114073/denza_car_design.jpg
4	http://wallpapersget.com/wallpapers/2012/03/car-bmw-328-hommage-wallpaper-1080x1920.jpg
5	http://images.thetruthaboutcars.com/2011/02/Ferrari458-tuned.jpg
6	http://www.appdaddy.com/previews/luxury-car-theme_big.jpg

Algorithm 2:

Reveal Local Knowledge

```
•   Input: Selected imageUrl_list
•   Output:  weighted imageUrl_list
•   Begin
•    For each image in image Url list Do
•          begin
•              Capture mouse and keyboard events
•                 Assign weight to image
•          end
•   End.
```

Algorithm 3

: Broadcast local knowledge

```
•   Input: weighted  imageUrl_list
•   Begin
•       Instantiate msg from ACLMessage;
•       Set msg.receiver   to be address of global
Agent
•       Set   msg.content   to   be   weighted
imageUrl_list
•   and the Query;
•     Send msg;
End
```

Algorithm 4:

Intersect broadcast knowledge

```
•   Input: msges[] as  Array of Agents' Messages
•   Begin
•      Initial  CommonVisited List  as String Array
•      Initial SumWeight as  integer Array
•     For  all messages in msges[]
•          tempMsg = nextMsg in msges[]
•             for all messages in msges[] and NOT
tempMsg do
•                 find shared imageUrl and add it to
•   CommonVisitedList.
  Sum total weight and add it to SumWeight in index
manner.
•   End.
```

After intersecting queries from different Agents, the following URI

- http://wallpapersget.com/wallpapers/2012/03/car-bmw-328-hommage-wallpaper-1080x1920.jpg

will be indexed using key {race, fancy, celebrity and sport .

6. CONCLUSIONS

After investigating a bunch of papers published within the same topic of our proposal, we found correlation in basic terminologies, but with distinct methodologies. Many models have been introduced to develop knowledge about retrieved images like what we introduced here and the significant features of Agent-based system are also exploited but the key differences between all these efforts and what we devised in our proposal can be summarized:

1-A Multi-Agent system has been deployed on two levels: host level and network level to develop knowledge regarding certain images, other approaches target mainly behavioral aspects of network interactions rather than host based.

2-Annotation is generated autonomously and a confidence value is assigned to each annotation; this value represents the acceptance of society for this annotation as a key index for associated image.

3-Third party web based tools has been included (i.e., the Google search engine APIs) while all other approaches tend to design custom search software modules. Google has a very massive repository of images, thus it is more convenient to address this repository rather than inferring other repositories or web pages. Furthermore, Google search engine receives millions of request for images in multiple subjects, thus this will assist, statistically, revealing more reliable annotations.

4-An image is annotated, in our proposal, not on the basis of the graphical objects in the image or the low level features, but on the basis of its relation to the environment, for example an image could have some planets and this image can be interpreted using low level features and semantic contents as to relate to planet science, flowers, garden or some of the like.

In our approach the planets image can be categorized into drug, medicine, health or so on; this is due to society opinion.

Other approaches Index this image based on its low level feature and its composed visual objects.

7. ACKNOWLEDGMENTS

My thanks to the University of Mustansiriyah – Iraq, that gives me the possibility to support and continue my PhD.

8. REFERENCES

[1] NursuriatiJamil and SitiAisyahSa'adan, Proceeding, "Visual Informatics: Bridging Research and Practice", visual informatics conference, IVIC 2009, Kuala Lumpur, Malaysia, 2009.

[2] Masashi Inoue, "On the Need for Annotation-Based Image Retrieval", National Institute of Informatics, Tokyo, Japan, 2006

[3] Aixin Sun, Sourav S. Bhowmick, Khanh Tran Nam Nguyen, "Tag-Based Social Image Retrieval: An Empirical Evaluation", published in American Society for Information Science and Technology, USA, 2011.

[4] NeelaSawant, Jia Li and James Z. Wang, "Automatic Image Semantic Interpretation Using Social Action and Tagging Data", survey, Pennsylvania State University, USA,2011.

[5] Baojun Qiu, Kristinka Ivanova, John Yen, and Peng Liu, "Behavior Evolution and Event-driven Growth Dynamics in Social Networks"

[6] David S. Channin MD, Pattanasak Mongkolwat, Vladimir Kleper, and Daniel L. Rubin, "Computing Human Image Annotation"

[7] Chen-Ue Lee, Von-Wun Soo, and Yi-Ting Fu, "How to Annotate an Image? The Need of an Image Annotation Guide Agent"

[8] Nikolaos Papadakis, Klimis Ntalianis, Anastasios Doulamis, and George Stamoulis, "An Automatic Multi-Agent Web Image and Associated Keywords Retrieval System"

[9] Jian Zhang, Guang-Zhou zeng, and Zhi-Feng Li, " The Study of a Sociality Agent Architecture Based On Role"

[10] Takayuki Shiose, Tetsuo Sawaragi, Osamu Katai, and Michio Okada, "Dynamics of Reciprocal Learning by Bi-Referential Model within Multiagent Systems"

[11] Weng Zu Mao, and David A. Bell," Interactive Query for Image Contents by Semantic Descriptors and Multi-Agent".

[12] Jesus Favela and Victoria Meza,"Image-retrieval agent: integrating image content and text".

Automatic De-noising for Image Annotation Using Latent Semantic Analysis

Abbass. Khorsheed[1], Hussein Chible[1], Giorgio Giacinto[2], Luigi Raffo[2,]
[1]Doctoral School of science and Technology, Lebanese University, Hadtah, Beirut, Lebanon
[2]Department of Electrical and Electronic Engineering, University of Cagliari, Italy
abbasarab2000@yahoo.com & giacinto@diee.unica.it

Abstract: Images are important material accessed through the internet by a huge number of applications such as medical, social, mining applications. The biggest challenge facing the usage of those billion of images is the retrieving challenge. Two approaches are available to retrieve images over the internet: first one is by using textual matching between user query and image annotation, and second one is by using image contents. This paper introduces an approach to remove redundant words used to annotate images.

Keywords: Image Annotation, LSA, SVD, Automatic De-Noising, Semantic Space.

I. De-noising for Image Annotation

From the inspection of popular image search engines such as Google, Bing and Baidu, the retrieval paradigm employed by these search engines is still based on the keywords composing the query; this query is formulated by users to initiate image search process. Users use natural language words to describe requested image, or other multimedia contents, and the responsibility of a search engine is to scan databases for a proper match. The most crucial element is the search scenario is the indexing of images, or other multimedia contents, where natural language is demanded to achieve the labeling of available images with textual description; this process is called image annotation [1,2]. Content-based image retrieval, the problem of searching large image repositories according to their content, has been the subject of a significant amount of computer vision research in the recent past. While early retrieval architectures were based on the query-by-example paradigm, which formulates image retrieval as the search for the best database match to a user-provided query image, it was quickly realized that the design of fully functional retrieval systems would require support for semantic queries. These are systems where the database of images are annotated with semantic keywords, enabling the user to specify the query through a natural language description of the visual concepts of interest. This realization, combined with the cost of manual image labeling, generated significant interest in the problem of automatically extracting semantic descriptors from images [1,2,3]. Images are annotated using different methodologies, some are manually; this when clients comment on certain images and automatically such as mining the textual text in internet pages that hold that image. Crucial challenge in image annotation is the redundant words that increase false results such as the irrelevant images returned by Google search engine [3]. The earliest efforts in the area were directed to the reliable extraction of specific semantics. These efforts posed the problem of semantics extraction as one of supervised learning: a set of training images with and without the concept of interest was collected and a binary classifier trained to detect the concept of interest. The classifier was then applied to all database of images which were annotated with respect to the presence or absence of the concept [2,3].

This paper introduces a novel approach to remove redundant words used to annotate images; this is done by using Latent Semantic Analysis (LSA) to build the semantic space that combines queries and annotations, and then use Singular Value Decomposition (SVD) to determine variance produced by annotation words. As a last step, words with less variance are omitted. LSA can be used efficiently to filter annotation concepts (i.e., natural language words) where semantic similarities among annotations attached to certain images and the set of queries posted to search engines, is an effective approach to determine and omit redundant words. The accuracy of the results is corresponding to the distribution pattern of the natural language words over the query and the annotation at the same time, where semantic similarities among annotations and queries vectors should span the semantic space of a group of images that are to be de-noised. The most obstacle facing this approach is the intensive calculations required by the LSA when new image added to the group of images which have been de-noised against redundant words.

II. References

[1] NursuriatiJamil and SitiAisyahSa'adan, Proceeding, "Visual Informatics: Bridging Research and Practice", visual informatics conference, IVIC 2009, Kuala Lumpur, Malaysia, 2009.

[2] Masashi Inoue, "On the Need for Annotation-Based Image Retrieval", National Institute of Informatics, Tokyo, Japan, 2006

[3] Reinhard Koch and Fay Huang (Eds), " Computer Vision-ACCV 2010 Workshops: 'Two-Probabilistic Latent Semantic Model for Image Annotation and Retrieval' ", Springer, USA, 2011.

Abstract/Résumé

Automatic Image Annotation using Image Clustering in Multi Agent Society

A.A. Khorsheed [a], H. Chible[a], G. Giacinto [b], L. Raffo[b]

a. Microelectronics Research Team, Doctorate School of Sciences and Technologies, Lebanese University
hchible@ul.edu.lb
b. Microelectronics Lab, University of Cagliari, Sardinia, ITALY

Image annotation is the natural language description for essential blobs within an Image. Many methodologies have recruited to reveal image semantics and represent it as annotation. Due to the exclusionary growth of number of images distributed over massive repositories, the task of manual annotation is tedious and over killing in term of times and efforts.

This paper is presenting automatic annotation system based on the interaction between intelligent agents. Agent interaction is synonym to socialization behavior dominating Agent society. The presented system is exploiting knowledge evolution revenue due to the socialization to charge up the annotation process

Key words: Automatic Image Annotation, Image clustering, Multi Agent Society, Website.\

Technique for Image Deblurring Using Least Square Filter with MLMVN Network

Dr. Haider Kadhim Hoomod.

Mustansiriyah University/ Educational College

Abbas Akram Khorsheed

Mustansiriyah University / College of Science

Abstract

Image restoration is one of the image processing used to repaint the image or restore image information. One of image problem is the blur in the image. To solve this problem, a new deblurring technique was proposed in order to reduced or remove the blur in the images. The proposed filter was design from the Least Square interpolation calculation controlling by the neural network to select the blurred pixels. The wavelet decomposing technique was used in order to increase the performance of the proposed filter. The proposed filter was work with good results for fully and partially blurring region in images.

تقنية لرفع الضبابية من الصورة باستخدام مرشح المربعات الضغرى مع شبكة MLMVN

الخلاصة

استعادة الصورة واحدة من معالجة الصور المستخدمة لإعادة رسم الصورة أو استعادة معلومات الصورة. واحدة من مشاكل الصورة هو الضبابية في الصورة.لحل هذه المشكلة اقترحت تقنية لرفع الضبابية جديدة ولتخفيض أو إزالة التمويه في الصور. وتم تصميم المرشح المقترح من حساب الاستيفاء المربعات الصغرى ويتحكم بالمرشح الشبكة العصبية لتحديد البيكسل الغير واضحة المضببة. استخدمت تحليلات wavelet لزيادة أداء المرشح المقترح. المرشح المقترح حصل على نتائج تطبيقية جيدة لرفع الضابية للصورة كاملة او جزء من الصورة .

Here it is assumed that application of the force causes the spring to expand and, having derived the force constant by least squares fitting, the extension can be predicted from Hooke's law.

5. Multilayer Neural Network Based On Multi-Valued Neurons [7]

A multilayer neural network based on multi-valued neurons (MLMVN) has been introduced, investigated and developed. This network consists of multi-valued neurons (MVN). That is a neuron with complex-valued weights and an activation function, defined as a function of the argument of a weighted sum. This activation function was proposed in 1971 in the pioneer paper of N. Aizenberg et al. The multi-valued neuron was introduced based on the principles of multiple-valued threshold logic over the field of complex numbers formulated. The most important properties of MVN are: the complex-valued weights, inputs and output lying on the unit circle, and the activation function, which maps the complex plane into the unit circle. It is important that MVN learning is reduced to the movement along the unit circle. The MVN learning algorithm is based on a simple linear error correction rule and it does not require differentiability of the activation function. Different applications of MVN have been considered during recent years, e.g.: MVN as a basic neuron in the cellular neural networks, as the basic neuron of the neural-based associative memories, as the basic neuron in a variety of pattern recognition systems, and as a basic neuron of the MLMVN. The MLMVN outperforms a classical multilayer feedforward network and different kernel-based networks in the terms of learning speed, network complexity, and classification/prediction rate tested for such popular benchmarks problems as the parity n, the two spirals, the sonar, and the Mackey-Glass time series prediction. These properties of MLMVN show that it is more flexible and adapts faster in comparison with other solutions. In this paper we apply MLMVN to identify blur and its parameters, which is a key problem in image deblurring.

Usually blur refers to the low-pass distortions introduced into an image. It can be caused, e.g., by the relative motion between the camera and the original scene, by the optical system which is out of focus, by atmospheric turbulence (optical satellite imaging), aberrations in the optical system, etc. Any type of blur, which is spatially invariant, can be expressed by the convolution kernel in the integral equation. Hence, deblurring (restoration) of a blurred image is an ill-posed inverse problem, and regularization is commonly used when solving this problem. There is a variety of sophisticated and efficient deblurring techniques such as deconvolution based on the Wiener filter, nonparametric image deblurring using local polynomial approximation with spatially-adaptive scale selection based on the intersection of confidence intervals rule, Fourier-wavelet regularized deconvolution, expectation-maximization algorithm for wavelet-based image deconvolution, etc. All these techniques assume a prior knowledge of the blurring kernel or its point spread function (PSF) and its parameter. When the blurring operator is unknown, the image restoration becomes a blind deconvolution problem. Most of the methods to solve it are iterative, and, therefore, they are computationally costly. Due to the presence of noise they suffer from the stability and convergence problems. The original solutions of blur identification problem that are based on the use of MVN-based neural networks were proposed in [7]. Two different single-layer MVN-based networks have been used to identify blur and its parameter (e.g., variation for the Gaussian blur, extent for motion blur, etc.). The results were good, but this approach had some disadvantages. For instance, the networks used have specific architecture with no universal learning algorithm, thus each neuron was trained separately. Another disadvantage is the use of too many spectral coefficients as features (quarter of image size). Thus the learning process was heavy.

A single neural network (the discrete-valued MLMVN) with the original backpropagation training scheme was used to identify both smoothing operator and its parameter on a single observed noisy image. However, the discrete-valued MLMVN had such a drawback as discrete inputs which results in quantization error of pattern vectors.

6. Multilayer Neural Network Based On Multi-Valued Neurons[7]

A continuous-valued MVN has been introduced in [7]. It performs a mapping between n inputs and a single output using n+1 complex-valued weights

$$f(x_1, ..., x_n) = P(w_0 + w_1 x_1 + ...+ w_n x_n), ...(21)$$

where X = x_1 ,...,x_n is a vector of complex-valued inputs (a pattern vector) and W = w_0 ,w_1 , ...,w_n is a weighting vector. P is the activation function of the neuron:

$$P(z) = \exp(i\ (\arg z)) = e^{iArg\ z} = \frac{z}{|z|}\ . \qquad ...(22)$$

where z = w_0 + w_1 x_1 + ...+ w_n x_n is a weighted sum, arg z is an argument of the complex number z, Arg z is a main value of the argument of the complex number z and |z| is its modulo.

The MVN learning is reduced to the movement along the unit circle. This movement does not require differentiability of the activation function. Any direction along the circle always leads to the target. The shortest way of this movement is completely determined by an error that is a difference between the desired and actual outputs. The corresponding learning rule is:

$$W_{r+1} = W_r + \frac{C_r}{(n+1)}(\varepsilon^q - e^{iArg\ z})\bar{X} = W_r + \frac{C_r}{(n+1)}\left(\varepsilon^q - \frac{z}{|z|}\right)\bar{X}.. \qquad ..(23)$$

where $\bar{X}$ denotes vector with the complex-conjugated elements to input pattern vector X, W_r is a current weighting vector, W_{r+1} is a weighting vector after correction, C_r is a learning rate. A modified learning rule is:

$$W_{r+1} = W_r + \frac{C_r}{(n+1)|z_r|}\left(\varepsilon^q - \frac{z}{|z|}\right)\bar{X}\ . \qquad ..(24)$$

where z_r is a current value of the weighted sum.

A multilayer feedforward neural network based on multi-valued neurons (MLMVN) has been proposed. It refers to the basic principles of the network with a feedforward dataflow through nodes proposed by D. E. Rumelhart and J. L. McClelland. The most important is that there is a full connection between the consecutive layers (the outputs of neurons from the preceding layer are connected with the corresponding inputs of neurons from the following layer). The network contains one input layer, *m*-1 hidden layers and one output layer. Let us use here the following notations. Let *km T* be a desired output of the k^{th} neuron from the m^{th} (output) layer; Y_{km} be an actual

output of the k^{th} neuron from the m^{th} (output) layer. Then the global error of the network taken from the k^{th} neuron of the m^{th} (output) layer is calculated as follows:
*

$$\delta^*_{km} = T_{km} - Y_{km} \quad \ldots(25)$$

The square error functional for the s^{th} pattern $X_s = x_1 ,\ldots,x_n$ is as follows:

$$E_s = \sum_k (\delta^*_{km})^2 (W) , \quad \ldots(26)$$

where δ^*_{km} is a global error taken from the k^{th} neuron of the m^{th} (output) layer, E_s is a square error of the network for the s^{th} pattern, and W denotes all the weighting vectors of all the neurons of the network. The mean square error functional for the network is defined as follows:

$$E = \frac{1}{N}\sum_{s=1}^{N} E_s . \quad \ldots(27)$$

Where N is a total number of patterns in the training set.

The back propagation learning algorithm for the MLMVN was used; the errors of all the neurons from the network are determined by the global errors of the network (eqn.25). Finally, the MLMVN learning is based on the minimization of the error functional (eqn.27). It is fundamental that the global error of the network consists not only of the output neurons errors, but of the local errors of the output neurons and hidden neurons. It means that in order to obtain the local errors for all neurons, the global error must be shared among these neurons.

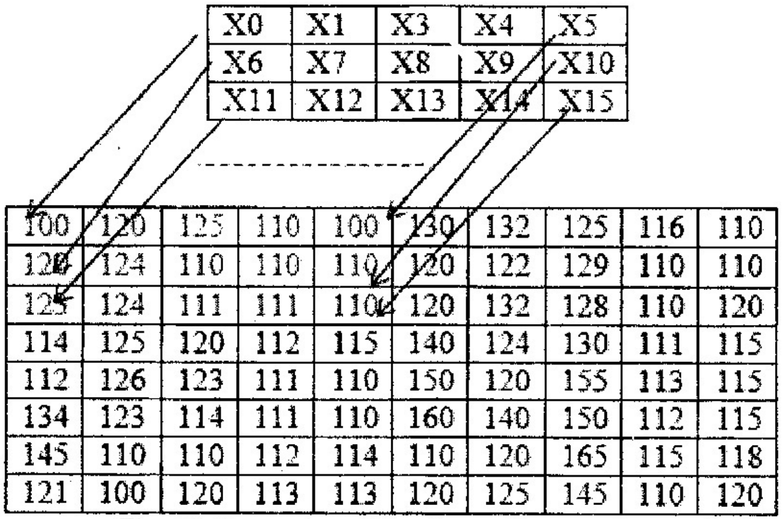
7. The Proposed Least Squares Interpolation Filter

In this section, the Least Square filter design will explain. This filter was proposed to repaint image depend on the least square calculation method. The result from this filter will used to remove or reduce the blurring in the blurred images. The main equation of this proposed filter is develop to the eq.(5) and (6) applied to the suggested mask. The Leas square developed equation are :

$$S = \sum_{i=1}^{n} r_i^{2}$$

$$r_i = ((f(x_i) - y_i) + (M - y_i))/2 \qquad \ldots(28)$$

Where M is the average pixels for the suggested mask. The Least Squares interpolation resulted values that used to replace the selected pixels in the original image depend on the net of neighbors pixels effect. The replacing operation will covered all pixels on the suggested mask. The suggested mask design to gives the best pixels effect calculation. The suggested mask contains three rows 5 image pixels (3x5). Figure (2) shows an example of the design mask. The mask will apply to whole image until deblurring operation complete.



X0	X1	X3	X4	X5
X6	X7	X8	X9	X10
X11	X12	X13	X14	X15

100	120	125	110	100	130	132	125	116	110
120	124	110	110	110	120	122	129	110	110
125	124	111	111	110	120	132	128	110	120
114	125	120	112	115	140	124	130	111	115
112	126	123	111	110	150	120	155	113	115
134	123	114	111	110	160	140	150	112	115
145	110	110	112	114	110	120	165	115	118
121	100	120	113	113	120	125	145	110	120

Fig.(2) Example of The Least Squares Interpolation Mask

As shown in Fig.(2), the mask points to selected named by x0, x1, x2, x3, ...x15. These selected pixels will replaced by the resulted values from the Least Square calculation. (Note: the replacing operation will control by the MLMVN network).

8. The Proposed Image Deblurring System

Restoration or deblurring average blur from images is a very difficult problem to resolve. In this research we describe a strategy

that can be used for solving such image blurring problems. This strategy is used to evaluate the proposed filter to deblur image (explain in section 7) control by neural network explain in section 6. The structure of the neural network used in this proposed system was explained bellow.

As explain in section 7, the proposed filter was design using the principles and theory of the Least polynomial interpolation to estimate the pixels values to replace in the mask window of the filter.

The operation of the proposed is as shown: firstly, the proposed system applied the wavelet transform(the wavelet transform used is the Daubechies 4D basis wavelet functions applied on the loaded image) on the loaded image. Then select the details subband (LL subband) to applied the deblurring operation.

Second, for each mask window, the proposed system will use the MLMVN to test each pixel in the window if pixel blur or not blur. The MLMVN network was learned to cover three types of the blurred (Gaussian, rectangular, and the motion linear uniform horizontal). In this proposed system, we used the continuous-valued MLMVN (further simply MLMVN) describe in [7] to solve both the blur and its parameters identification problems in order to overcome the disadvantages mentioned above. The modification of the MLMVN results in significant improvement of the functionality.

In the same time, the proposed system applied the proposed Least Square filter to calculate the new replacement pixels values. If the pixel is assigned as blur with the blur type, the proposed system will replace the pixel by the proposed filter result. Else, the pixel will leave without replace. The proposed filter and MLMVN network will apply to whole image. Finally, the proposed system recomposes the resulted image and calculates the RMSE measure to check the variation between the original image and resulted image.

The main proposed system steps are as follows:

- Applying Wavelet Transform on the loaded image using Daubechies 4D basis function. Then, a window of 3x5 coefficients blocks for each sub bands was used.
- Apply MLMVN neural network on the decomposed image mask.

- **Applying adaptive Least Square filter on the decomposed image mask.**
- **If MLMVN decision id blur replace with result of the proposed Least Square filter.**
- **Recomposed the deblurred image subbands.**
- **Calculate RMSE**

Figure 3 shows the block diagram of the proposed system block:

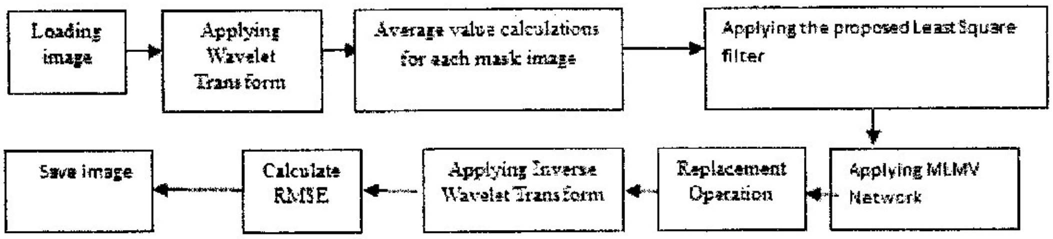

Loading image
Applying Wavelet Trans form
Average value calculations for each mask image
Applying the proposed Least Square filter
Applying MLMV Network
Replacement Operation
Applying Inverse Wavelet Trans form
Calculate RMSE
Save image

Fig.(3) The proposed system block diagram

In this proposed system, the Daubechies 4D basis filter was used to decomposed loaded image into 2 level using the Daubechies 4D scaling and wavelet functions are as show in equations (29,30) [8]. Figure (4) shows some results of the 2-level decomposed image.

$$\alpha_1 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad \alpha_2 = \frac{3+\sqrt{3}}{4\sqrt{2}} \qquad .....(29)$$

β

$$\alpha_3 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad \alpha_4 = \frac{1-\sqrt{3}}{4\sqrt{2}} \qquad .....(30)$$

$$\beta_3 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \beta_4 = \frac{-1-\sqrt{3}}{4\sqrt{2}}$$

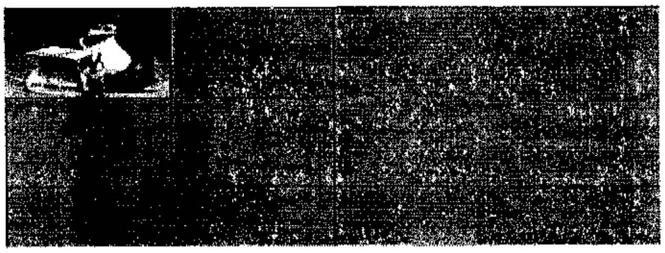


Fig.(4) Examples of the Daubechies 4D basis filter(2 – level).

The RMSE can calculate from :

$$RMSE = \sqrt{\frac{\sum_{x=0}^{n-1}\sum_{y=0}^{n-1}[\overline{X}(x,y)-X(x,y)]^2}{N^2}} \qquad .....(31)$$

Where$\overline{X}$, X represent the deblurred and blurred images, respectively.

MLMVN Neural Network Structure

As point above the MLMVN network used in the proposed system was explain in[7].But we provide some development on the input nodes number and the blurring types due the increase the mask design properties and to increase the performance of the neural network. In this proposed system, we learned the neural network with the three types of blur with the following parameters. The Gaussian blur is considered with τ ∈{1, 1.33, 1.66, 2, 2.33, 2.66, 3} for

$$v(t) = \frac{1}{2\pi\tau^2}\exp\left(-\frac{t_1^2+t_2^2}{\tau^2}\right) \qquad ...(32)$$

where τ^2 is a parameter of the PSF (the variance of the Gaussian function); the linear uniform horizontal φ = 0 motion blur of the lengths 3, 5, 7, 9, for

$$v(t) = \begin{cases} \frac{1}{h}, \sqrt{t_1^2+t_2^2} \cdot h/2, \ t_1\cos\phi = t_2\sin\phi, \\ 0, \qquad \text{otherwise}, \end{cases} \qquad ...(32);$$

Rectangular blur window has sizes 3×3 , 5×5 , 7× 7 , 9×9, for

$$v(t) \doteq \begin{cases} \frac{1}{h^2}, \ |t_1| \quad \frac{h}{2}, \ |t_2| < \frac{h}{2}, \\ 0, \qquad \text{otherwise}, \end{cases} \qquad ...(33)$$

Where parameter h defines the size of smoothing area.

The MLMVN has two hidden layers consisting of 5 and 35 neurons, respectively, and the output layer which consists of the same number of neurons as the number of classes, i.e. types of blur. Since we consider three types of blur (Gaussian, rectangular, and the motion linear uniform horizontal, $\varphi = 0$) the output layer contains three neurons. Therefore, the structure of network is 15,5, 35, 3 (input layer, hidden layer1, hidden layer2, output layer). Each neural element of the output layer has to classify a parameter of the corresponding type of blur, and reject other blurs (as well, as an unblurred image).
The MVN activation function (eqn.22) for the output layer neurons has a specific form: the equal subdomains (non-overlapping sectors) of the complex plane are reserved to classify a particular blur and its parameters and to reject other blurs and deblurred images.

7. Experimental Results & Discussions

In this research a Least Square interpolation calculation was used with Multilayer Neural Network Based On Multi-Valued Neurons to deblurring images with a full blurring or partial blurring. The proposed filter and MLMVN network give very good results in removing the fully/partially blurring from images. Also, a good RMSE values are resulted. Fig(5) shows some proposed system results. The RMSE of the different samples deblurring by proposed Least Square filter is compared in Table 1.

Table 1 RMSE Results from applies the proposed Least Square filter

Sample name	RMSE	Size
S1	31.22	640x480
S2	30.78	600x800
S3	34.67	600x800
S4	32.21	512x 512
S5	32.35	512x512
S6	37.54	600x800

1. Introduction

The de-blurring method developed relies on the facts that natural images have sparse leading edges, and edges of the blurred image are lesser sparse than that of the de-blurred images as the edges of the blurred images occupy a larger area due to blurring. So, a prior which tend to make the edges sparser will tend to make the images sharper also.[1]

Image blur is caused either by the camera motion or by the object motion. Camera motion is the camera vibration when shutter is pressed. The corresponding motion blur is usually modeled as a linear image degradation process

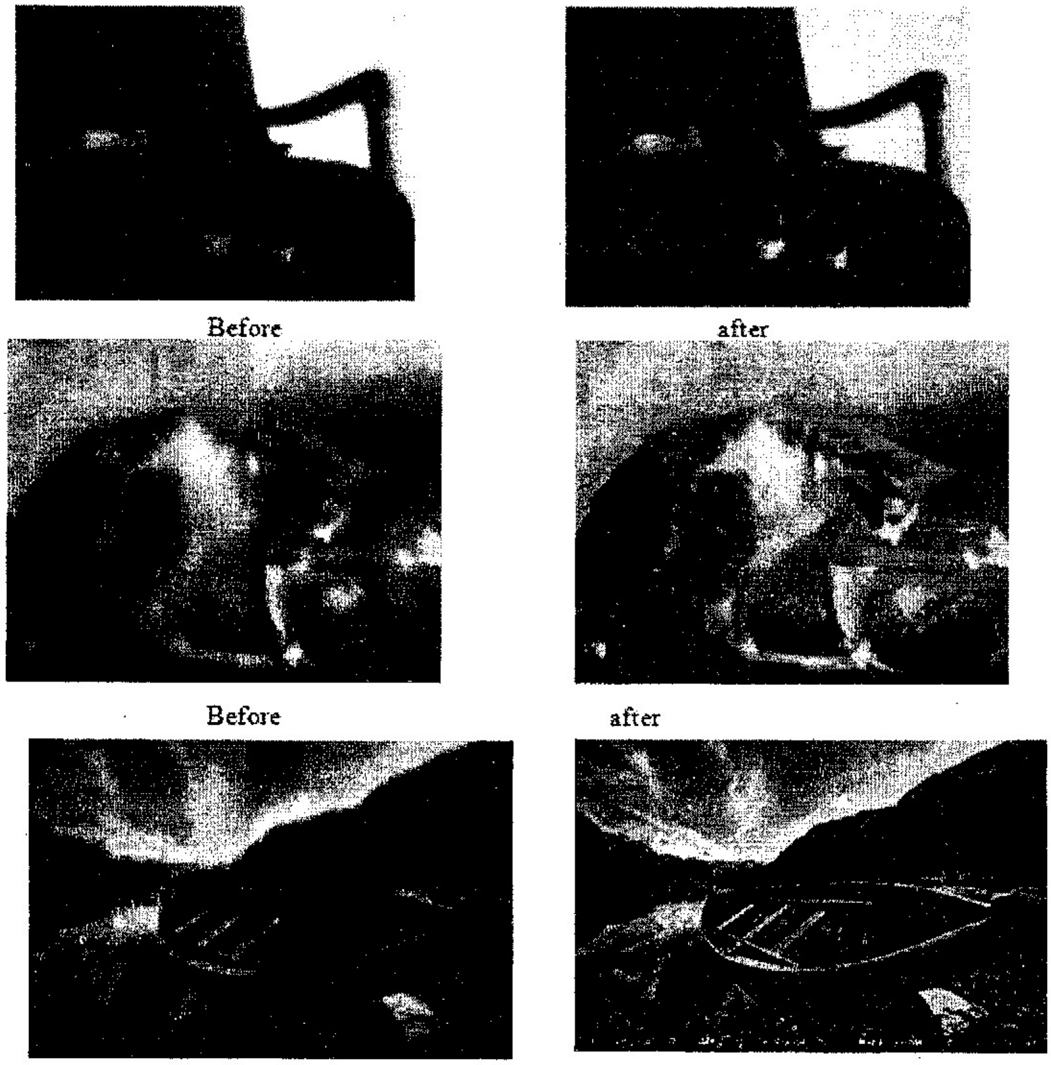
$$I = L \otimes f + n; \ldots(1)$$

where I, L, and n represent the degraded image, unblurred(or latent) image, and the additive noise respectively. $\otimes$ is the convolution operator and f is an unknown linear shift invariant point spread function (PSF). Conventional blind deconvolution approaches focus on the estimation off to deconvolve I using image intensities or gradients. [2]

The assumption that edges are sparser in natural sharp images may be an over simplification of the problem but the results obtained on this analysis are promising. The prior should be adjusted and the cost function should be altered such that initial iteration which estimate the lower frequency components of the image are regularized with bigger weight whereas later iterations which corresponds to higher frequency components should be regularized with smaller weights. This adjustment ensures noise suppression at the same time allows for better sparseness.[3]

Blur is unsharp image area caused by camera or subject movement, inaccurate focusing, or the use of an aperture that gives shallow depth of field. The Blur effects are filters that smooth transitions and decrease contrast by averaging the pixels next to hard edges of defined lines and areas where there are significant color transition.[3]

The problem of restoring a still image containing a motion blurred object cannot be completely solved by the blind deconvolution techniques because the background may not undergo the same motion. The PSF has a uniform definition only on the



Before after

Before after

Before after

Fig(5) Some results of the proposed Least Square filter.

References

[1] Andr´e Jalobeanu, Laure Blanc-F´eraud, Josian eZerubia," Satellite image deblurring using complex wavelet packets", International Journal Of Computer Vision
Volume 51, Number 3, 205-217, DOI: 10.1023/A:1021801918603, February 2003 .

[2] Jiaya Jia, " Single Image Motion Deblurring Using Transparency", IEEE,07/7-1180-4244-1 /, 2007

[3] Adarsh Nagaraja," Blind and Semi-Blind De-blurring of Natural Images", www.cs.ucf.edu/~adarsh/blindDeconv_v2.pdf

[4] *Stan Reeves* ," Image Deblurring-Introduction", 2007, http://blogs.mathworks.com/steve/2007/08/13/image-deblurring-introduction/

[5]Mathieu Lamard, Guy Cazuguel et al, "Content Based Image Retrieval based on Wavelet Transform coefficients distribution ", Conference proceedings : Annual International Conference of the IEEE Engineering, 2005

[6] Khalid Ali Hussien and Jolan Rokan Naif, " The Least Squares Interpolation Polynomial for Images", 2010

[7] Igor Aizenberg, Dmitriy Paliy, Claudio Moraga, and Jaakko Astola, "Blur Identification Using Neural Network for Image Restoration", Computational Intelligence, Theory and Applications 2006, Part 17, 441-455, DOI: 10.1007/3-540-34783-6_45.

[8]http://www.bearcave.com/misl/misl_tech/wavelets/daubechies/index.html.

moving object. Raskar et al propose a solution to this problem, in part of their method, assuming that the background is known or has constant colors inside the blurred region. [2]

Although there have been many image restoration techniques proposed, without knowing the blur filter, few of them can be readily applied to solve both of the above two motion deblurring problems. In this paper, as a first attempt, a unified approach is proposed to estimate the motion blur filter from a transparency point of view. Suppose that an object, which is originally opaque and has solid boundary, is motion blurred. Its boundary is blended to the background. The transparency on this blurred object is primarily caused by its motion during image capture. [2]

2. Image Deb luring [4]

Image deblurring (or restoration) is an old problem in image processing, but it continues to attract the attention of researchers and practitioners alike. A number of real-world problems from astronomy to consumer imaging find applications for image restoration algorithms. Plus, image restoration is an easily visualized example of a larger class of inverse problems that arise in all kinds of scientific, medical, industrial and theoretical problems.

To deblur the image, we need a mathematical description of how it was blurred. (If that's not available, there are algorithms to estimate the blur. But that's for another day.) We usually start with a shift-invariant model, meaning that every point in the original image spreads out the same way in forming the blurry image. We model this with convolution:

$$g(m,n) = h(m,n)*f(m,n) + u(m,n) \quad \dots (2)$$

where * is 2-D convolution, h(m,n) is the point-spread function (PSF), f(m,n) is the original image, and u(m,n) is noise (usually considered independent identically distributed Gaussian). This equation originates in continuous space but is shown already discretized for convenience.

Actually, a blurred image is usually a windowed version of the output g(m,n) above, since the original image f(m,n) isn't ordinarily zero outside of a rectangular array. Let's go ahead and synthesize a blurred image so we'll have something to work with. If we assume f(m,n) is periodic (generally a rather poor assumption!), the convolution becomes circular convolution, which can be implemented with FFTs via the convolution theorem.

3. Wavelet Transform and Image Processing[5]

Wavelets decomposition: Wavelets are families of functions generated from one single prototype function (mother wavelet) ψ by dilation and translation operations: ψ is constructed from the so-called scaling functionϕ, satisfying the two-scale difference equation

$$\phi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} h(k)\phi(2t-k) \qquad .... (3)$$

where h(k) are the scaling coefficients. Then, the mother wavelet ψ(t) is defined as

$$\psi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} g(k)\phi(2t-k) \qquad .... (4)$$

Where the wavelet coefficients g(k) = $(-1)^k$h(1 - k). Several different sets of coefficients h(k) can be found, which are used to build a unique and orthonormal wavelet basis. The wavelet transform represents the decomposition of a function into a family of wavelet functions $\psi_{m,n}$(t) where m is the scale/dilation index and n the time/space index). In other words, using the wavelet transform, any arbitrary function can be written as a super position of wavelets.

Many constructions of wavelets have been introduced in mathematical and signal processing literature (in the context of quadrature mirror filters). In the mid-eighties, the introduction of multi resolution analysis and the fast wavelet transform by Mallet and Meyer provided the connection between the two approaches. The wavelet transform may be seen as a filter bank and illustrated as follows, on a one dimensional signal x[n]:

•x[n] is high-pass and low-pass filtered, producing two signals d[n] (detail) and c[n] (coarse approximation).

•d[n] and c[n] may be subsampled (decimated by 2: ↓2) otherwise the transform is called translation invariant wavelet transform

•the process is iterated on the low-pass signal c[n].
This process is illustrated in figure 1. We have extract information at several scales (sub bands) plus an approximation of the signal (the last c[n]) In the case of images, the filtering operations are both performed on rows and columns, leading to the decomposition.

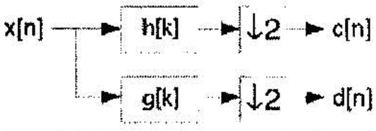


Fig. 1 Two-channel filter bank involving sub sampling

4. The Least Squares Estimation Method[6]

The method of least squares is a standard approach to the approximate solution of over determined systems, i.e. sets of equations in which there are more equations than unknowns. "Least squares" means that the overall solution minimizes the sum of the squares of the errors made in solving every single equation.

The most important application is in data fitting. The best fit in the least-squares sense minimizes the sum of squared residuals, a residual being the difference between an observed value and the fitted value provided by a model.

Least squares problems fall into two categories: linear least squares and nonlinear least squares, depending on whether or not the residuals are linear in all unknowns. The linear least-squares problem occurs in statistical regression analysis; it has a closed-form solution. The non-linear problem has no closed solution and is usually solved by iterative refinement; at each iteration the system is approximated by a linear one, thus the core calculation is similar in both cases.

The least-squares method was first described by Carl Friedrich Gauss around 1794. Least squares correspond to the maximum likelihood criterion if the experimental errors have a normal distribution and can also be derived as a method of moments estimator.

The objective consists of adjusting the parameters of a model function to best fit a data set. A simple data set consists of n points (data pairs) (x_i, y_i), i = 1, ..., n, where x_i is an independent variable and y_i is a dependent variable whose value is found by observation. The model function has the form f(x,β), where the m adjustable parameters are held in the vector β. The goal is to find the parameter values for the model which "best" fits the data. The least squares method finds its optimum when the sum, S, of squared residuals

$$S = \sum_{i=1}^{n} r_i^2 \qquad ....(5)$$

is a minimum. A residual is defined as the difference between the value predicted by the model and the actual value of the dependent variable

ri = f(xi,β) − yi. ....(6)

An example of a model is that of the straight line. Denoting the intercept as β_0 and the slope as β1, the model function is given by $f(x,\beta) = \beta_0 + \beta_1 x$. A data point may consist of more than one independent variable. For an example, when fitting a plane to a set of height measurements, the plane is a function of two independent variables, x and z, say. In the most general case there may be one or more independent variables and one or more dependent variables at each data point. The minimum of the sum of squares is found by setting the gradient to zero. Since the model contains m parameters there are m gradient equations.

$$\frac{\partial S}{\partial \beta_j} = 2\sum_i r_i \frac{\partial r_i}{\partial \beta_j} = 0,\ j = 1, \ldots, m \qquad ...(7)$$

and since $r_i = y_i - f(x_i, \beta)$the gradient equations become

$$-2\sum_i \frac{\partial f(x_i,\boldsymbol{\beta})}{\partial\beta_j} r_i = 0,\ j = 1,\ldots,m \quad \text{...(8)}$$

The gradient equations apply to all least squares problems. Each particular problem requires particular expressions for the model and its partial derivatives.

4.1 Linear least squares[6]

A regression model is a linear one when the model comprises a linear combination of the parameters, i.e.

$$f(x_i,\beta) = \sum_{j=1}^{m} \beta_j \phi_j(x_i) \quad \text{...(9)}$$

where the coefficients, φj, are functions of xi.

Letting

$$X_{ij} = \frac{\partial f(x_i,\boldsymbol{\beta})}{\partial\beta_j} = \phi_j(x_i). \quad \text{....(10)}$$

we can then see that in that case the least squares estimate (or estimator, in the context of a random sample), β is given by

$$\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \boldsymbol{y} \quad \text{...(11)}$$

A generalization to approximation of a data set is the approximation of a function by a sum of other functions, usually an orthogonal set:

$$f(x) \approx f_n(x) = a_1\phi_1(x) + a_2\phi_2(x) + \cdots + a_n\phi_n(x), \quad \text{...(12)}$$

with the set of functions {$\phi_j(x)$} an orthonormal set over the interval of interest, say [a, b]. The coefficients {a_j} are selected to make the magnitude of the difference ||f – f n ||2 as small as possible.

4.2 Non-linear least squares[6]

There is no closed-form solution to a non-linear least squares problem. Instead, numerical algorithms are used to find the value of the parameters β which minimize the objective. Most algorithms involve choosing initial values for the parameters. Then, the parameters are refined iteratively, that is, the values are obtained by successive approximation.

$$\beta_j^{k+1} = \beta_j^{\,k} + \Delta\beta_j \qquad \text{...(13)}$$

k is an iteration number and the vector of increments, $\Delta\beta_j$is known as the shift vector. In some commonly used algorithms, at each iteration the model may be linearized by approximation to a first-order Taylor series expansion about β^k

$$\begin{aligned} f(x_i,\boldsymbol{\beta}) &= f^k(x_i,\boldsymbol{\beta}) + \sum_j \frac{\partial f(x_i,\boldsymbol{\beta})}{\partial \beta_j}\left(\beta_j - \beta_j^{\,k}\right) \\ &= f^k(x_i,\boldsymbol{\beta}) + \sum_j J_{ij}\Delta\beta_j. \end{aligned} \qquad \text{...(14)}$$

The Jacobian, J, is a function of constants, the independent variable and the parameters, so it changes from one iteration to the next. The residuals are given by

$$r_i = y_i - f^k(x_i,\boldsymbol{\beta}) - \sum_{j=1}^{m} J_{ij}\Delta\beta_j = \Delta y_i - \sum_{j=1}^{m} J_{ij}\Delta\beta_j \quad . \qquad \text{...(15)}$$

To minimizo the sum of squares of ri, the gradient equation is set to zero and solved for $\Delta\beta_j$

$$-2\sum_{i=1}^{n} J_{ij}\left(\Delta y_i - \sum_{j=1}^{m} J_{ij}\Delta\beta_j\right) = 0 \qquad \text{...(16)}$$

Example:

The methods of least squares and regression analysis are conceptually different. However, the method of least squares is often used to generate estimators and other statistics in regression analysis. Consider a simple example drawn from physics. A spring should obey Hooke's law which states that the extension of a spring is proportional to the force, F, applied to it.

$$f(F_i, k) = kF_i \qquad ...(17)$$

constitutes the model, where F is the independent variable. To estimate the force constant, k, a series of n measurements with different forces will produce a set of data, $(F_i, y_i), i = 1, n$, where yi is a measured spring extension. Each experimental observation will contain some error. If we denote this error ε, we may specify an empirical model for our observations,

$$y_i = kF_i + \varepsilon_i. \qquad ...(18)$$

There are many methods we might use to estimate the unknown parameter k. Noting that the n equations in the m variables in our data comprise an overdetermined system with one unknown and n equations, to estimate k may choose to using least squares. The sum of squares to be minimized is

$$S = \sum_{i=1}^{n} (y_i - kF_i)^2 . \qquad ...(19)$$

The least squares estimate of the force constant, k, is given by

$$\hat{k} = \frac{\sum_i F_i y_i}{\sum_i F_i^2}. \qquad ...(20)$$

Conceptual Image Indexing and Searching Technique Based on Collaborative Multi-Agents platform

MR. ABBAS AKRAM KHORSHEED

Baghdad- Almustansiriya University- College of science – Computer science dep. E-mail : abbasarab2000@yahoo.com.

تقديـــــم البحث : 2011/12/1

قبول نشر البحث : 2012/ 2/7

Abstract

The rapid growth of the internet provides tremendous resource for information in different domains (text, image, voice, and many others). This growth introduces new challenge to hit an exact match due to huge number of document returned by search engines where millions of items can be returned for certain subject. Images have been important resources for information, and billions of images are searched to fulfill user demands which faces the mentioned challenge.

This paper presenting online intelligent indexing for image repositories based on their contents, although content based indexing and retrieving systems have been introduced, this paper is adding an intelligent technique to re-index images upon better understanding for its composed concepts. Collaborative Agent scheme has been developed to promote objects of an image to concepts and re-index it according to domain specifications.

تقنية فهرسة الصور والبحث عنها بصيغة المفاهيم بالاعتماد على التعاون في وسط متعدد الوكلاء

م.م عباس اكرم خورشيد

الجامعه المستنصريه/كلية العلوم/ قسم علوم الحاسوب

الخلاصة

ان النمو السريع للانترنيت وفر مصدر هائل للمعلومات في مختلف المجالات (النصوص، الصورة ، الصوت وغيرها)، هذا النمو نتج عنه تحدي جديد والذي يتمثل في مطابقة نتائج البحث في الانترنيت عبر محركات البحث والموضوع المقصود من البحث حيث ان الناتج احيانا من محركات البحث لموضوع معين قد يتجاوز الملايين. الصورة واحدة من المصادر المهمة على الانترنيت والتي تستخدم المليارات منها يوميا من قبل الباحثين والتي تواجه التحدي المذكور.

ان هذا البحث يقدم تقنية جديدة في عملية فهرسة الصور في مستودعات الخزن بالاعتماد على محتوياتها وبالرغم من ان هذا الموضوع قد تم طرحه سابقا فان هذا البحث يضيف بعد جديد في استخدام تقنية اعادة الفهرسة الذكية بالاعتماد على فهم اكبر لمكونات الصورة. لقد تم تطوير وسط يتعاون فيه مجموعة من الوكلاء البرمجيين لغرض اعادة الفهرسة من خلال الارتقاء بكيانات الصورة الى مفاهيم وحسب مواصفات المجال .

1- Introduction to Image Retrieval

Images are a major information source in the real world, and represent features of objects like their color, shape and other attributes [1,2]. Enormous amount of applications are counting on images as the main interactions with environment such satellite applications, medical applications, security applications.. etc., The rapid growth of the repositories, holding images introduces new challenges such as indexing and retrieving images, where the number may exceed billions when it comes to Web applications, Thus a considerable researches have been presented to propose effective mechanisms for image retrieval, which mainly two paradigms are adopted: text-based metadata and content-based image retrieval(CBIR) [1] .

The purpose of an image database is to store and retrieve an image or image sequences that are relevant to a query. There are a variety of domains such as information retrieval, computer graphics, database management and user behavior which have evolved separately but are interrelated and provide a valuable contribution to this research subject. As more and more visual information is available in digital archives, the need for effective image retrieval has become clear [2, 3, 4, 5]. In image retrieval research, researchers are moving from keyword based to content based then towards semantic based image retrieval and the main problem encountered in the content based image retrieval research is the semantic gap between the low-level feature representing and high-level semantics in the images [3,5].

2- Image Retrieval Methodologies

In the past retrieval was done manually by experts who store and index images and they receive requests from people seeking these images, forcing them to search for relevant images using their own knowledge[1]. Today, due to the information boom and the massive number of images been stored and retrieved, an automation methodologies have been the demand [2,5].

The basic idea of automation methodologies is that documents (i.e., images) are indexed according to pre-defined features extracted from the document itself. Then, given a query that reflects a user's information need, the retrieval system will search the document for relevant match. The main problem affecting the early attempts in designing image retrieval systems was lack of rigorous evaluation [5].

2-1 Keyword Based Image Retrieval

In 1970s, the conventional image retrieval system used keyword as descriptors to index an image however the content of an image is much richer than what any set of keywords can express [2].

Text-based image retrieval techniques employ text to describe the content of the image which often causes ambiguity and inadequacy in performing an image database search and query processing. This problem is due to the difficulty in specifying exact terms and phrases in describing the content of images as the content of an image is much richer than what any set of keywords can express. Since the textual annotations are based on language, variations in annotation will pose challenges to image retrieval [2,5].

2-2 Content Based Image Retrieval

In 1990s, Content-based image retrieval (CBIR) then has been used as an alternative to text based image retrieval. Unlike keywords-based system, visual features for contents-based system are extracted from the image itself. CBIR can be categorized based on the type of features used for retrieval which could be either low level or high level features. At early years, low level features include colour, texture, shape and spatial relations were used. The summary of CBIR researches done in retrieving the image based on their visual content can be found in our paper[1,3,5] .

Although there are many sophisticated algorithms to describe color, shape and texture features approaches, these algorithms do not satisfied and comfort to human perception[2]. This is mainly due to the unavailability of low level image features in describing high level concepts in the users' mind. For an example finding an image of a little boy is playing a ball in the garden. The only way a machine is able to perform automatic extraction is by extracting the low level features that represented by the color, texture, shape and spatial from images with a good degree of efficiency [4,5] .

2-3 Semantic Based Image retrieval

In 2000s, semantic based image retrieval has been introduced. This is due to neither a single features nor a combination of multiple visual features could fully capture high level concept of images. Besides, the performance of image retrieval system based on low level features are not satisfactory, there is a need for the mainstream of the research converges to retrieve based on semantic meaning by trying to extract the cognitive concept of a human to map the low level image features to high level concept (semantic gap). In addition, representing the image content with semantic terms allows users to access images through text query which is more intuitive, easier and preferred by the front end users to express their mind compare with using images. The review and general framework of semantic based image retrieval can be found our paper in [3,4,5] .

3- Semantic Gap

Bridging the semantic gap for image retrieval is a very challenging problem yet to be solved [6,7]. Describing images in semantic terms is an important and challenging task that needed to carry out to fulfill human satisfaction besides to have more intelligent image retrieval system. Human beings are able to interpret images at different levels, both in low level features (colour, shape, texture and object detection) and high level semantics (abstract objects, an event). However, a machine is only able to interpret images based on low level image features. Besides, users prefer to articulate high-level queries [5], but CBIR systems index images using lowlevel features. Hence, introducing an interpretation inconsistency between image descriptors and high-level semantics that is known as the semantic gap [3,5]. The semantic gap is the lack of correlation between the semantic categories that a user requires and the low-level features that CBIR systems offer [5].

The semantic gap between the low-level visual features (color, shape, texture, etc.) and semantic concepts identified by the user remains a major problem in content based image retrieval [5]. Semantic content representation has been identified as an important issue to bridge the semantic gap in visual information access. It has been addressed as a good description and representation of an image, it able to capture meaningful contents of the image. Current researches often represent images in terms of labeled regions or images, but pay little attention to the spatial positions or relationships between those regions or objects [3,5].

Spatial relationship is needed in order to further increase the confidence in image understanding. Besides, users preferred to express their information needs at the semantic level instead of the level of preliminary image features. Moreover textual queries usually provide more accurate description of users' information needs **[5].**

4- Software Agent

The term 'agent', or software agent, has found its way into a number of technologies and has been widely used, for example, in artificial intelligence, databases, operating systems and computer networks literature. Although there is no single definition of an agent [9], all definitions agree that an agent is essentially a special software component that has autonomy that provides an interoperable interface to an arbitrary system and/or behaves like a human agent, working for some clients in pursuit of its own agenda. Even if an agent system can be based on a solitary agent working within an environment and if necessary interacting with its users, usually they consist of multiple agents [8].

These multi-agent systems (MAS) can model complex systems and introduce the possibility of agents having common or conflicting goals. These agents may interact with each other both indirectly (by acting on the environment) or directly (via communication and negotiation). Agents may decide to cooperate for mutual benefit or may compete to serve their own interests [8,9] .

5- Java Agent Development(JADE)

The first software developments, that eventually became the JADE platform, were started by Telecom Italia (formerly CSELT) in late 1998, motivated by the need to validate the early FIPA specifications [8].

Partially funded by European Commission (FACTS project, ACTS AC317) a team composed of Fabio Bellifemine, Agostino Poggi and Giovanni Rimassa were gathered with the good will and dedications to promote the concepts of JADE and its compliant to FIPA. At a certain point it was decided to move beyond a means of simply validating the

FIPA specifications towards developing a fully fledged middleware platform. The vision was to provide services to application developers and that were readily accessible and usable by both seasoned developers and newcomers with little or no knowledge of the FIPA specifications. Emphasis was placed on the simplicity and usability of the software APIs [8,9].

In order to better facilitate industrial involvement, in May 2003 Telecom Italia Lab and Motorola Inc. defined a collaboration agreement and formed the JADE Governing Board, a not-for-profit organization of companies committed to contributing to the development and promotion of JADE. The Board was formed as a contractual consortium with well-defined rules governing the rights and obligations toward generated IPR. The Board is open with members able to join and leave according to their needs. At the time of writing, Telecom Italia, Motorola, France Telecom R&D, Whitestein Technologies AG and Profactor GmbH have all become members of the Board [8].

When JADE was first made public by Telecom Italia, it was used almost exclusively by the FIPA community but as its feature set grew far beyond the FIPA specifications, so did its usage by a globally distributed developer community. It is interesting to note that JADE contributed to widespread diffusion of the FIPA specifications by providing a set of software abstractions and tools that hid the specifications themselves; programmers could essentially implement according to the specifications without the need to study them. This is considered as one of the main strengths of JADE with respect to FIPA [8,9].

6- AGENT COMMUNICATION

Agent communication is probably the most fundamental feature of JADE and is implemented in accordance with the FIPA specifications [8,9]. The communication paradigm is based on asynchronous message passing. Thus, each agent has a 'mailbox' (the agent message queue) where the JADE run-time posts messages sent by other agents. Whenever a message is posted in the mailbox message queue the receiving agent is notified. However, when, or if, the agent picks up the message from the queue for processing is a design choice of the agent programmer [8].

The particular format of messages in JADE is compliant with that defined by the FIPA-ACL message structure, where each message includes the following fields:

- The sender of the message.
- The list of receivers.
- The communicative act (also called the 'performative') indicating what the sender intends to

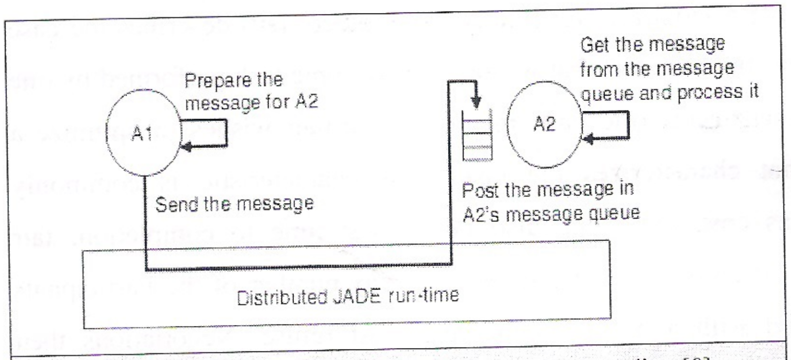

Prepare the message for A2
A1
Send the message
Post the message in A2's message queue
A2
Get the message from the message queue and process it
Distributed JADE run-time

Figure 1: JADE asynchronous message passing paradigm[8]

- The content containing the actual information to be exchanged by the message (e.g., the action to be performed in a REQUEST message, or the fact that the sender wants to disclose in an INFORM message, etc.).
- The content language indicating the syntax used to express the content. Both the sender and the receiver must be able to encode and parse expressions compliant with this syntax for the communication to be effective.
- The ontology indicating the vocabulary of the symbols used in the content. Both the sender and the receiver must ascribe the same meaning to these symbols for the communication to be effective.

Some additional fields used to control several concurrent conversations and to

specify timeouts for receiving a reply such as conversation-id, reply-with,

in-reply-to and reply-by.[8]

7- FIPA Contract Net Interaction Protocol Specification

The FIPA Contract Net Interaction Protocol (IP) describes the case of one agent (the Initiator) that wishes to have some task performed by one or more other agents (the Participants) and further wishes to optimize a function that characterizes the task. This characteristic is commonly expressed as cost, but could also be soonest time to completion, fair distribution of tasks, etc. For a given task, any number of the Participants may respond with a proposal; the rest must refuse. Negotiations then continue with the Participants that proposed. The IP is depicted in Figure (2)

The Initiator solicits m proposals from other agents by issuing a call for proposals (cfp) CA which specifies the task and any conditions the Initiator places upon the execution of the task. Participants receiving the call for proposals are viewed as potential contractors and are able to generate n responses. Of these, j are proposals to perform the task, specified as propose CAs.

The Participant's proposal includes the preconditions that the Participant is setting out for the task, which may be the price, time when the task will be done, etc. Alternatively, the i = n −j Participants may refuse to propose. Once the deadline passes, the Initiator evaluates the received. [8]

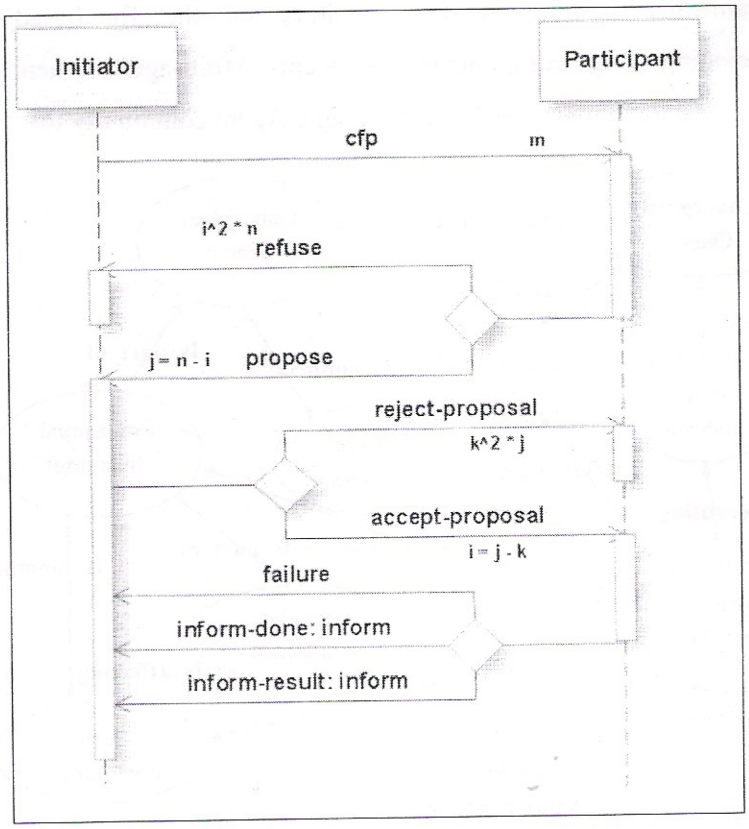


Figure 2 : FIPA-net interaction protocol (IP)[8]

8- Conceptual Image Indexing and searching

Conceptual mapping is a technique that combines knowledge from different domains and map it to human level semantics, this way the machine will interact with human requests at a very high level of semantics. Indexing is a technique used to facilitate accessing the designations resources in a short time, many approaches are used by database engines to index data. Conceptual indexing in other hand is the process to organize document based of their semantics, this approach is applied to image data base too.

This paper is proposing a technique as figure (3) presents to dynamically re-index image repository on the fly based on developing image components to concepts. Multi-agent system has been used to, autonomously, socializing e Agent community to

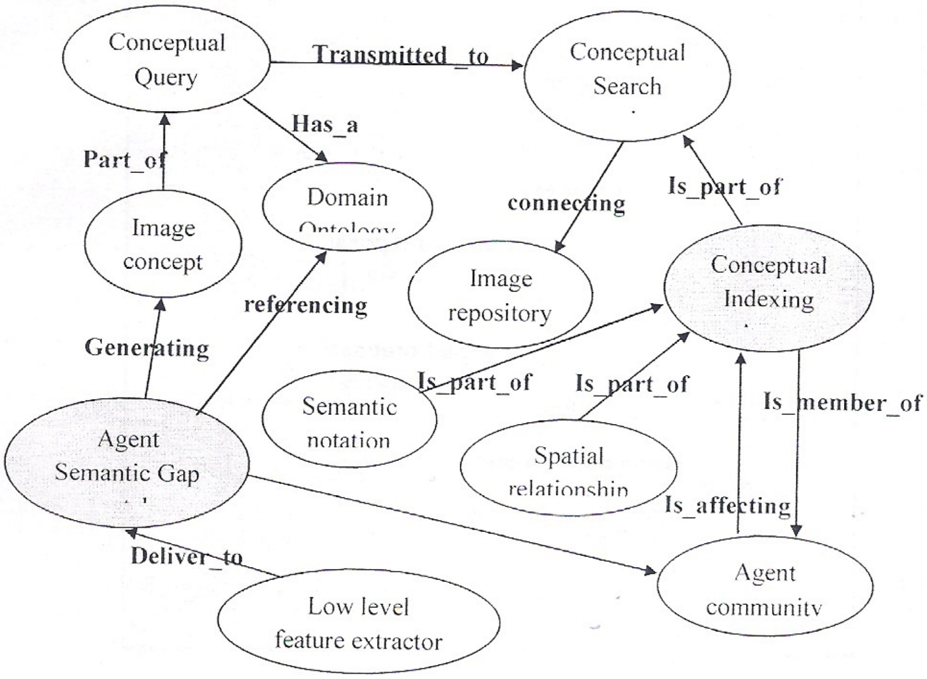


Figure 3: Conceptual Indexing and Searching Ontology

Figure (4) depicts the Multi-Agent structure and components where Agents are collaborated to reveal more concepts regarding images being indexed. The indexing is not static and can be changed frequently based indentify more objects within images and map it to semantic level concept. As figure (presents) Agents have been classified according to their assignment and functionalities, where two types are founded: Indexing Agent and region/concepts promotion Agent.

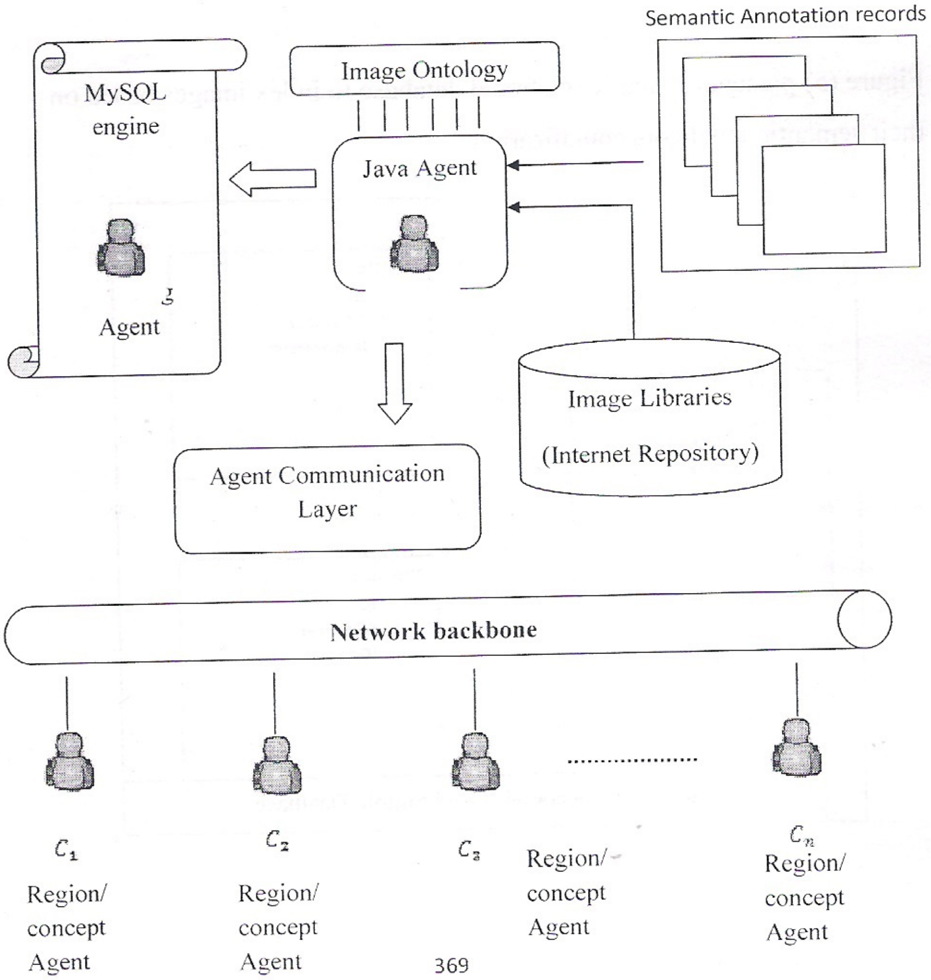

Semantic Annotation records
Image Ontology
MySQL engine
Agent
Java Agent
Image Libraries
(Internet Repository)
Agent Communication Layer
Network backbone
C_1 Region/ concept Agent
C_2 Region/ concept Agent
C_3 Region/ concept Agent
....
C_n Region/ concept Agent

Figure 4 : Mutli-Agent Image Indexing and searching block diagram

$$I_{concept} = (c_1 , c_2 , c_3 ,\ldots, c_n) = \sum_{1}^{n} c_i \qquad (1)$$

$$\mathrm{I} = (o_1 , o_2 , o_3 ,\ldots, o) \qquad = \sum_{1}^{n} o_i \qquad (2)$$

^{u} Iconcept ^{u} Ispatial $^{\exists}$ R index ϵ Image Repository (3)

Figure (5) presents a simple relational database to index images based on their semantic level conceptualization

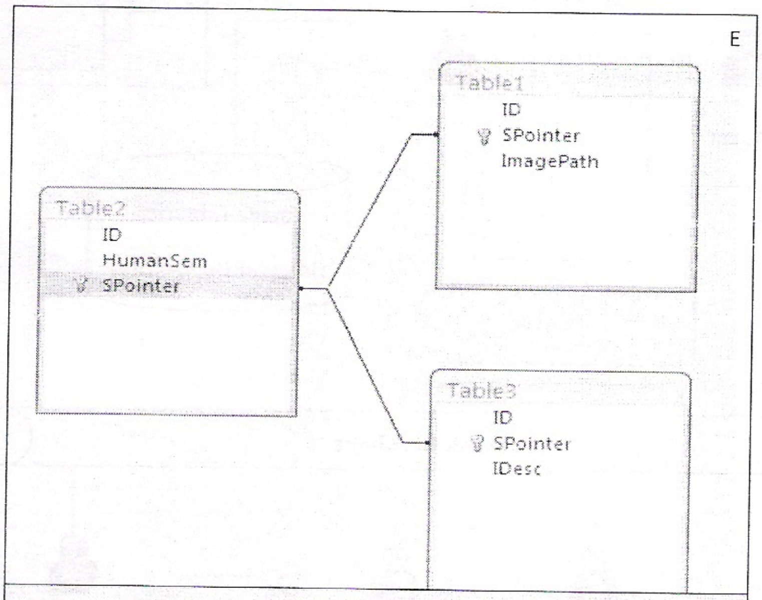


Figure 5: Conceptual Based Simple Database

SPointer	HumanSem	ID
321234	Man_With_dog_Meeting_Friend	1
652113	Man_Walking_With_Dog	2
762391	Dog_Chases_People	3
171283	Two_Friends_Walking	4

Table 1: relational table to code ontology

The most important field is within Table3 presented in figure (5) which is 'IDesc' (Image Descriptor), this field contains XML code to represent Images in term of their concepts and spatial relationship. The following are the ontology used to define each semantic level :

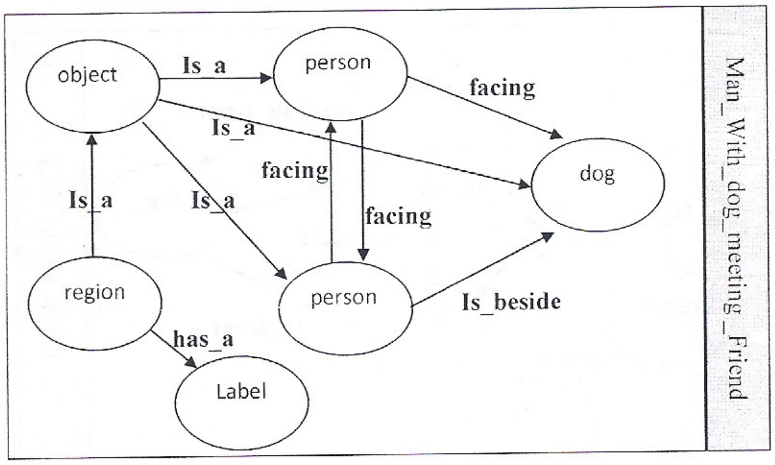


Figure 6: Man with dog meeting_Friend ontology

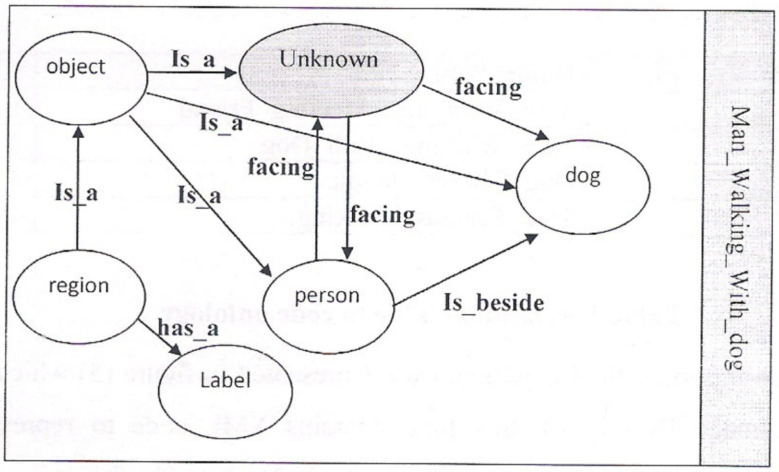


Figure 7: Man Walking With dog ontology

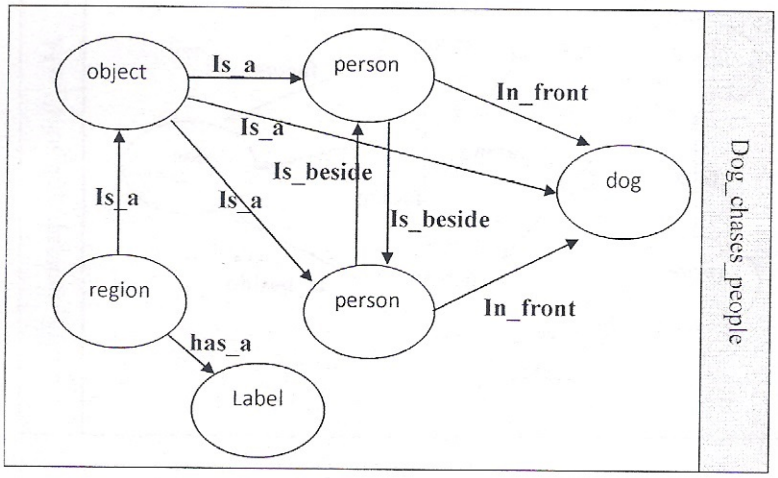


Figure 8: Dog chases people ontology

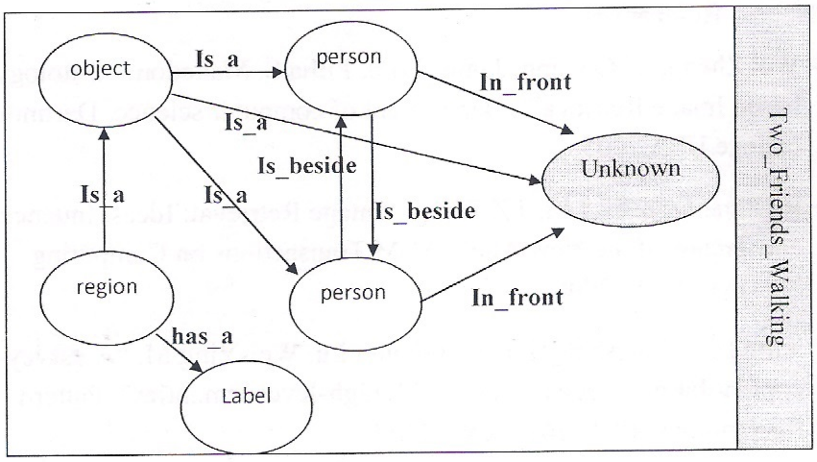


Figure 9: Tow Friends Walking ontology

9- Conclusions

1- Automatic Image Retrieving system still needs more methodologies to index images based on their content and extracted features. Topology depends very much on extracting objects and determination the geometry of the image.

2- Topology of objects constructing the image can serve excellent in binding low level image features to high level semantics if the view port is evaluated correctly. View port will be the crucial point in determining spatial relationship among objects within the image.

3- Socializing image indexing can be integrated to internet search engines to provide accessibility to rich repository on the web, results can be graded relatively to matching scale determined as a collaboration outcome of Agent society.

10- References

1- Wei Zheng, Yi Ouyang, James Ford, Fillia S. Makedon." Ontology-based Image Retrieval", Department of computer science, Dartmouth college,USA,2005

2- R.Datta, D.Joshi, J.Li, J.Z.Wang. "Image Retrieval: Ideas, Inuences, and Trends of the New Age". ACM Transactions on Computing Surveys, 40(2), 2008

3- Ying Liu, Dengsheng Zhang, Guojun Lu, Wei-Ying M. "A survey of content-based image retrieval with high-level semantics". Pattern Recognition, 40(1):262 – 282, 2007

4- Semantic Gap in CBIR: Automatic Objects Spatial Relationships Semantic Extraction and Representation,2004

5- Masashi INOUE , " Image Retrival: Research and use in the information explosion", National Institute of Information, 2009.

6- C.Y.Chang, H.Y.Wang, C.F.Li. "Semantic analysis of real world images using support vector machine". Expert systems with application : An international journal, 2009.

7- Najlae Idrissi. "Bridging the Semantic Gap for Texture-based Image Retrieval and Navigation", Journal of Multimedia University, Malaysia , 2009.

8- Fabio Bellifemine, Giovanni Caire and Dominic Greenwood, "Developing Multi-Agent Systems with JADE", John Wiley & Sons, Ltd, 2007.

9- Bordini, Rafael H, Dastani, Mehdi, El Fallah Seghrouchni,"Multi-Agent programming, languages, performance and applications", 2005