Implicit Concept-based Image Indexing and Retrieval for Visual Information Systems

Ibrahim Ahmed Aref Azzam

This thesis is presented in fulfillment of

the requirements for the degree of

Doctor of Philosophy

School of Computer and Mathematical Sciences

Faculty of Health, Engineering and Science

Victoria University

2006

Abstract

This thesis focuses on Implicit Concept-based Image Indexing and Retrieval (ICIIR), and the development of a novel method for the indexing and retrieval of images.

Image indexing and retrieval using a concept-based approach involves extraction, modelling and indexing of image content information. Computer vision offers a variety of techniques for searching images in large collections. We propose a method, which involves the development of techniques to enable components of an image to be categorised on the basis of their relative importance within the image in combination with filtered representations. Our method concentrates on matching subparts of images, defined in a variety of ways, in order to find particular objects.

The storage of images involves an implicit, rather than an explicit, indexing scheme. Retrieval of images will then be achieved by application of an algorithm based on this categorisation, which will allow relevant images to be identified and retrieved accurately and efficiently.

We focus on Implicit Concept-based Image Indexing and Retrieval, using fuzzy expert systems, density measure, supporting factors, weights and other attributes of image components to identify and retrieve images.

Declaration

"I, Ibrahim Ahmed Aref Azzam, declare that the PhD thesis entitled Implicit Concept-Based Image Indexing and Retrieval for Visual Information Systems is no more than 100,000 words in length, exclusive of tables, figures, appendices, references and footnotes. This thesis contains no material that has been submitted previously, in whole or part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work".

Signature Date

A List of External Publications

[1] I. A. Azzam, C.H.C.Leung, J.F. Horwood, "Implicit Concept- based Image Indexing and Retrieval", *The 10th International Multimedia Modeling Conference MMM2004*, Jan-2004, Brisbane, pp 354-359.

[2] I. A. Azzam, A. Charlapally, C.H.C.Leung, J.F. Horwood, "Content-based Image Indexing and Retrieval using XML Representation", *2004 International Symposium on Intelligent Multimedia, Video & Speech Processing*, October-2004, Hong Kong, pp 181-185.

[3] I. A. Azzam, C.H.C.Leung, J.F. Horwood, "A Fuzzy Expert System for Concept-based Image Indexing and Retrieval", The *11th International Multimedia Modeling Conference MMM2005*, Jan-2005, Melbourne, pp 452-457.

Acknowledgments

I am deeply indebted to my principal supervisor, Professor Clement Leung and my co-supervisor Dr. John Horwood, for their guidance, constant direction, and countless hours of time. I thank both of them for their patience and trust that allowed me perform at full potential. I also appreciate their encouragement and support through the difficult days. I have benefited immeasurably from their tutelage.

I greatly acknowledge Associate Professor Perter Cerone, Head of School of Computer Science and Mathematics and other staff of the School of Computer Science and Mathematics for being supportive and cooperative. I would like to thank my friend, Murad Karrar for his help and advice of technical discussions.

I deeply owe my appreciation to my wife, Laila Aziz Azzam, for her consistent support with trust and love in these years that I was able to do this research. I miss the moments that I could otherwise spend with my children, who understand their father's reason for returning home late.

Last and not least, I praise GOD for leading me to pursue this dream, answering all my prayers, and fulfilling all my needs during this study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Overview

1.1 Objectives

The objectives of this thesis are two-fold: The primary aim is to develop an improved method for the storage and retrieval of images. This method will involve the construction of a new algorithm, which involves the development of techniques to enable components of an image to be categorised on the basis of their relative importance. Thus the storage of images will involve an implicit, rather than an explicit, indexing scheme. Retrieval of images will then be achieved by application of an algorithm based on this categorisation, which will allow relevant images to be identified quickly, and thus allow the search for specific details of images. Consequently, the search time involved in locating target images will be considerably reduced, providing a more responsive and more productive system. A secondary aim is to research existing methods of storing and retrieving images from a database, and to quantify the performance characteristics of the most effective existing methods, in order to provide a basis upon which to evaluate the performance of the algorithm developed in this thesis.

Due to the rapid increase in hardware and Internet capability, visual information, incorporating image storage and retrieval, has become a feasible source of information for many consumers. The technology for Visual Information Systems (VIS) is more urgently needed than ever before. What is needed are new computational methods to retrieve, index, compress and uncover pictorial information, new algorithms to allow access to very large databases of digital pictures and videos, as well as new systems with visual interfaces integrating the above components in new types of image databases. The construction of VIS represents a radical departure from building conventional information systems, and its architecture will incorporate fundamentally new capabilities for organising, processing, searching, querying and presenting information.

Visual Information Systems has a large market with many professions, which regard this form of information storage of major importance to their activities. For example, the medical profession uses large volumes of digital images to improve patient care. Mapping Systems use high levels of spatial data for producing detailed maps. VIS is used in advertising, education and training, entertainment, and video clips. Satellites and Australian Technology Network (ATN) enable videoconferencing, remote database management, distributed image storage and processing, plus much more.

Many applications are empowered using visual representations of objects or situations, since both experts and naive users can obtain at first glance, more information than by reading textual descriptions.

Information in a visual form differs from traditional database information in many important ways. It requires more space for storage, is highly unstructured, and needs some kind of decoding process to determine its semantic content [SETH95], [LEUN92].

Management of visual information poses several challenges with respect to its storage and retrieval, its delivery and presentation, as well as its querying and searches. In order to satisfy the need of new requirements to retrieve images efficiently in large databases, many visual information retrieval systems have been developed, where the queries are based on both visual content and textual annotation in these systems.

In the past few years, the increase of picture data has encouraged research into the development of image retrieval systems. The limitation of the current image analysis techniques necessitates that most image retrieval systems use some form of text description provided by the users as the basis to index and retrieve images. These techniques are rather primitive at present and they need further development and refinement [HWAR00], [LIN99], [BARB95], [BARB94].

Many different methods and techniques that have been proposed, largely fall into two categories; concept-based methods, which is a text-based approach that allows users to post their query either simply using keywords or using natural language, and content-based methods, which is pixel-based that allows user to post their query by an example or by image contents (colour, shape, texture, etc.). In this research, the focus will be on the concept-based method.

Current approaches are not as efficient as is desired for a technology that has become increasingly important in the processing of Visual Information System. For example, content-based methods have many disadvantages, which are explained with more detail later on in this thesis. These disadvantages include: images returned from query could be totally different from the target images, ambiguities with a multiple domain recall, and restricted access because these methods are not rich in image semantics. With concept-based methods, problems exist when extracting meaning from images in database applications. It is sometimes difficult to derive an image, which captures the meaning and context that the database user desires.

Hence, a more effective and more efficient algorithm to store images, which will be easy to retrieve, is urgently required.

1.2 Thesis Aims

The main aim of this thesis is to develop algorithms for the retrieval of image data from stored databases, using a methodology that incorporates implicit image indexing and retrieval.

We focus on implicit concept-based image indexing as a storage mechanism and retrieval as a query mechanism, with the major aim to develop a methodology for enhancing the retrieval effectiveness and overcome many of the current problems involved with the retrieval of images in databases. In particular it aims to provide new mechanisms for building image data index, which will facilitate query processing and retrieval.

Since the search effectiveness in an image database is always a compromise between the indexing cost and semantic richness, a solution that provides a significant degree of semantic richness that simultaneously limits the indexing cost will be investigated.

We developed mathematical rules that can be used to define image structure by analysing image components and assigning weights, supporting factors, densities and other properties to the components based on their relative importance within the image. This is followed by the computing of the values for each image using the mathematical rules to be stored in the database, which can be used to efficiently retrieve the desired images.

An additional goal in this thesis is to provide a methodology that concentrates on matching subparts of images, defined in a variety of ways, in order to find particular objects. The approach used is to integrate our concept-based image indexing and retrieval method with the use of a fuzzy expert system. From the initial returned images, users can select images using partial or complete objects related to the images [TURK92].

As fuzzy logic is a multi-valued logic that deals with degrees of membership and degrees of truth, and uses a continuum of logic values between zero and one, it enables us to qualify the imprecise data represented by subparts of an image. In this thesis fuzzy logic is used to compute object densities, which are used for indexing and retrieving the images [COX94].

1.3 Contribution to Knowledge

This thesis develops an innovative approach to the indexing and retrieval of image data. Current retrieval methods rely primarily on extensive and exhaustive descriptions of image data at the point of storage, which is labour intensive and considerably reduces the flexibility of retrieval. The proposed research makes use of semi-automatic indexing which significantly raise the efficiency of indexing while providing richer semantic to aid image retrieval.

This approach provides a new and innovative way to specify and view visual data across the whole spectrum of applications that involve processing of visual data.

1.4 Summary of Significance

This thesis provides new methodologies that increase the performance of retrieval algorithms, as measured by the proportion of actual retrieved images to the total number of images that satisfy the retrieval criteria. The proportion of successful retrievals is increased thus enhancing the effectiveness of applications that incorporate these algorithms. Its significance lies both in terms of more efficient image retrieval methodologies and also in terms of a more effective means of indexing image data consequent to the point of storage.

1.5 Thesis Organization

This thesis is organized as follows.

Chapter 2, examines the methods and problems for modelling and retrieving images: the concept-based methods and the content-based methods. It also explains the reasons for selecting the concept-based method for our research together with retrieval engines and evaluation criteria.

Chapter 3, describes the image collections used in this research and rules construction. It shows that the image collection must represent a good selection of multi-object images and also be related to one another. Furthermore, it is shown how the rules used to determine the images categories are constructed. The image representation is described where the image index consists of many discrete object types (low-level objects, mid-level objects, high-level objects etc), which determines how different levels of objects were built. The object attributes and relationships between objects and its levels are explained as well.

Chapter 4, explains in detail the objects and their attributes, and present Mathematical Equations used for measuring and computing the different values of the objects' attributes. These attributes are: the object's weight, supporting factor, value, rule significant factor, availability, density, and weight in image. This chapter demonstrates how the main step of density and fuzzy indexing and retrieving can be used for measuring and computing the object density, the steps used to determine the density levels and numerical ranges, the degree of membership of density components, construction of the fuzzy rules, and the encoding of these rules.

Chapter 5 discusses the process of implicit image indexing and its advantages. It shows various approaches for image indexing and retrieval system compared with our approach. Our method is based on the object that several atomic indices will be able to define intermediate indices, which in turn will be able to define higher indices and so on. The benefits of this method are also explained.

The data structures, creating the tables holding the data collected and computed from different level of objects within images are presented and the indexing algorithm are developed.

Chapter 6 provides a methodology that concentrates on matching images defined in different ways in order to find particular objects. This chapter also explains that a competent retrieval system should retrieve images that satisfy the user's needs. Tables containing linguistic variables notion for objects' attributes and their ranges using the fuzzy logic are constructed. The supporting objects and the advantage of using them when retrieving images are explained. The use of a thesaurus, controlled and uncontrolled vocabulary are explained in detail.

Chapter 7, describes the system specification; including the hardware and software used. The programming language and the database used for implementing the system and the reasons for choosing them are explained. The system architecture and relationship and the roles of all mangers contained in the system architecture are explained in detail. We explain our image retrieval engine and evaluation criteria.

Chapter 8, experiments to validate different aspects of our research are provided. Analysis of Concept-based Image Indexing and retrieval System is discussed with precision and recall being used for evaluating our search strategies.

Chapter 9, ends the thesis with the conclusions, the limitation and direction for future research.

Chapter 2

Literature Review

2.1 Introduction

Present day information systems are mainly based on alphanumeric information. The effective use of visual information for organizational applications has been limited and mostly confined to highly specialized applications. Visual information systems (VIS) not only substantially enhance the value and usability of existing information, but also open up a new horizon of previously untapped information sources. Future information systems in commercial and scientific applications can be expected to comprise a high visual content, and it is important to integrate the visual and image components into the architecture of organizational information systems. Although there has always been a demand for visual information, the technologies for such systems were either insufficiently mature or unavailable in the past. VIS is now becoming increasingly feasible because of rapid advances in hardware and software technologies. The construction of VIS represents a radical departure from building conventional systems, and many novel, fundamental, and architectural issues will need to be addressed [LEUN99], [LEUN97a], [LEUN97b].

There is a popular maxim that "*a picture is worth a thousand words*". Not surprisingly, the demand for digital photos on Web portals has continued to increase as the users are looking for better interfaces. As more digital multimedia archives are developed, they require specialized search engines that can index and search for these items. Since many image collections are indexed or annotated for data storage and retrievals, there is a great need for automated methods that could help users retrieve, browse, or structure image databases [ENSE95], [HOWA95].

A computer's degree of visual interface has grown exponentially within the past decade, producing tools that enable the retrieval of visual information; especially the capture of objects with no accompanying structural, administrative, or descriptive text information has become the prime focus of developers. Internet has become a common channel for the transmission of graphical information that moves visual information rapidly from stand-alone workstations and databases into a networked environment. The use of the Web to provide access to the search and retrieval mechanisms for visual and other forms of information has spawned the development of emerging standards for metadata about these objects as well as the creation of commonly employed methods to achieve interoperability across the searching of visual, textual, and other multimedia repositories.

The construction of VIS represents a different approach from building conventional information systems, and its architecture will integrate primarily new capabilities for organising, processing, searching, querying and presenting information.

2.2 Methods of Modelling and Retrieving Images

Many different methods and techniques have been proposed for modelling and retrieving images [SUTA99], [JORG98], [YANG97], [GUDI97], [SHAK96]. They largely fall into two categories, namely, concept-based methods and content-based methods, and the approach taken in this thesis is concept-based.

- Concept-based methods are mainly text-based approaches, which allow users to post their query by either simply using keywords or by using a form of natural language.
- Content-based methods, on the other hand, are pixel-based which allow users to post their query by an example or by image contents using such characteristics as colour, shape or texture.

2.2.1 Concept-based Methods

The design and implementation of concept-based image retrieval systems is studied in [CHUA94]. Concept-based image retrieval includes the following, which are based on free-text description:

- The use of concept-based search engine, for the retrieval of concept and images.
- The use of relevance feedback to update the concept representations and image description.

A text-based query system, based on the Ternary Fact Model (TFM) database was proposed and developed in [SUTA99], [LEUN95], which makes use of visual entity-relationship index representations, rule-based conceptual hierarchy, and other features to support semi-automatic indexing. It also provides mechanisms to enhance query performance using a thesaurus system, relevance feedback mechanism, and user-tailored weighting components.

Ternary Fact Model is different from conventional text-based systems that rely solely on keywords for database index and query. Using the entity-attribute-relationship model, the main semantic concept of entity, attribute and relationship can be represented. These correspond very closely to the noun, adjective and verb, which are the essential components of a simple description of a picture. It also allows the concept of relationships between two or more objects within a picture to be represented in the database.

An image can be also retrieved using common names of entities that appear in it [SHAK96]. In many cases, a query is the formal expression method of label-based image retrieval request. An image is abstracted into a hierarchy of entity names and features, showing how relations are connected between entities visible in the image.

A fuzzy logic is a multi-valued logic that deals with degrees of membership and degrees of truth. It uses the continuum of logic value between zero and one. Classical binary logic can be considered as a special case of multi-valued fuzzy logic [COX94].

A fuzzy matching technique is used that compares semantic queries to image abstractions by assessing the similarity of context between the query and the candidate image. The objective of this matching technique is to distinguish between abstraction of different images that have the same labels but are different in context. From the initial returned images, users can select images using partial or complete objects related to the images [TURK92].

An Enigma information system is proposed to retrieve images on the basis of geometrical or iconical indexes consisting of an image indexing and retrieval component [GEVE92]. This image indexing consists of automatically generating semantic representation of an image.

A semantic image database system, which is designed by using semantic database techniques for images with inner structure and contents, is presented in [YANG97]. The database system describes the structure and contents of images by incorporating type constructors and functions between types and inheritances. It supports composite modelling for entities, which consists of pictorial as well as alphanumeric attributes.

In [HWAR00], two ways of forming conceptual descriptions from images are proposed.

- Monitoring, just follow the flow of data from data images to interpretation, having little need for top-level control.
- Watching, this emphasizes the use of top-level control and actively selects evidence for task-based description of the dynamic scene.

A simple Picture Description Language (PDL) based on the entity-attribute-relationship (EAR) model is given in [LEUN92]. It furnishes degrees of standardisation that facilitates the indexing and storage of pictorial features in relational database.

A Picture Description Language allows the content of pictures to be systematically represented, and the resultant representation can then be stored for subsequent processing. It starts from a position of man-machine cooperation whereby users perform the processing, while the computer manages the storage and retrieval of the results.

2.2.2 Content-based Methods

The searches in image content use different low-level image features such as shape, colour, and texture. A content-based image retrieval system uses information from the content of images for retrieval and helps the user to retrieve database images relevant to the contents of a query image [SHAH04].

Images can be retrieved by an example image using edge or colour information; the system can also retrieve images using sketch [ABDE96].

In retrieval by an example image, for example, one image can be chosen from the database as query. A query can also be composed from images in the database using copy and paste. The system retrieves images with similar colour composition or similar edge information.

The clustering mechanism can categorise the images into different clusters based on their heterogeneous features (shape, colour, texture…etc). Database images are then classified based on their heterogenous feature. A database image is included into a feature sub-cluster only if the image contains all the features under the same cluster [SHEI98].

A system that uses a modified geometric hashing technique is given in [HUAN98], and that is to retrieve similar shaped images from the image database. It can retrieve the similar images from the database by using scaled, rotated or mirrored sketched query images.

A colour-base image retrieval approach, by which a user can specify colour regions and their spatial information in a query picture, where the desired images that partially or exactly satisfy the user-specified query, can be retrieved. New regular-texture image retrieval approaches, where users can retrieve regular-texture image from a database, which are most similar to a sample query. In this approach, texture primitives and their displacement vectors are extracted from the query and each regular image in the database. This technique and these images that satisfy perceptual similarity to the users query can be retrieved from the database [LIN99].

2.3 Image Indexing and Retrieval

The number of digital images is rapidly increasing. Many years ago some researchers were estimating that as many as one million digital images were being produced every day [YONG99], [JAIN93]. Government agencies, educational institutions, museums, and the commercial sector are all creating image-bases. Images are being captured, converted and stored in digital format in diverse fields and disciplines such as medicine, geography, law-enforcement, art, space science, journalism, and mass communications.

The current interest in accessing digital visual materials is also spurred by the widening use of multimedia in education and entertainment, and by a broadening focus in classification and information retrieval research beyond text-based materials towards more non-traditional perceptual modalities such as motion and sound. The growth of the World Wide Web over this period has produced unprecedented opportunities for accessing and sharing digital images. However, the technologies for creating collections of digital images at present exceed our capabilities for effective and efficient retrieval of these images. End-user access is a topic of vital concern to all those involved in building image-bases.

The increasing availability of image and multimedia-oriented applications clearly impacts image/multimedia file and database systems. Image data are not well-defined keywords such as traditional text data used in searching and retrieving functions [CHAN01], [ORLA00], [SONG99], [KIM99], [ZIMB98]. Consequently, various indexing methodologies must be defined based on the characteristics of image data. Pictorial spatial database has been attracting growing interest; spatial relationships represent an important feature of objects in an image.

Some image collections are very large, like medicine (ultrasound, CAT scans), architecture (building plans), geography (aerial photos, maps), art (paintings, cartoons), business (trademarks), and history (photographs). Indexing collections of this size can be extremely time-consuming, and unlike text, images cannot be searched by keywords. Many automatic indexing systems have been developed, but what computers can currently extract from images are mostly low-level features like color, shape, and texture [RUI99].

Searching an image library can be difficult and a major difficulty lies in designing powerful features to represent images in the library. Manual annotation works only for small library collections and does not scale to large ensembles. Furthermore, it is not clear what aspects of an image would be of interest to a particular user and the possible image descriptions can vary. Many studies are geared toward solving this problem and a few are even considered to be performance standards. The difficulty in designing powerful features is further aggravated if:

- Query images and images in the database can appear different due to non-essential environment changes, such as rigid motion, affined shape deformation, change in explanation and perspective.
- The objects in the database are similar in appearance, requiring highly selective description to tell them apart.

Many current systems for image indexing do not address the above two difficulties satisfactorily [GRAH99], [HAMP97], [PENT96], [FLIC95].

A variety of features have been used to index and retrieve images on contents, keywords, textual description or captions [SMEA96], [SETH95], [LEUN92].

Research on the information needs of users and on human perception of images may, in time, contribute the knowledge needed to produce the most precise and efficient retrieval systems possible. In the meantime, search engines contending with image collections have to make decisions about how best to provide access to them. Currently, there is no universal consensus in search engines.

Efficient indexing and retrieval of multimedia data are necessary to a successful multimedia system. A competent retrieval system should retrieve images that satisfy the user's needs. It should measure image similarity in a manner consistent with human perception. The development of innovative retrieval approaches for access to visual information is among the most significant of technological, conceptual, and institutional challenges for information science community. From an institutional standpoint, libraries, archives, and museums have entered into digitization projects, varying in scope and scale, the results of which are beginning to challenge the manner in which information is described, stored, and delivered [LONG01], [LONG01a], [PAYN99]. The next section describes different methods of Image indexing and retrieval.

Image retrieval is one of the most exciting and fastest growing research areas in the field of visual information systems (VIS). Visual information, incorporating image retrieval, has become a preferred source of information for many consumers [AZZA04].

Research in image retrieval has, so far, been divided between two main categories; "concept-based" image retrieval, and "content-based" image retrieval. The former focuses on using words to retrieve images (e.g. title, subject heading, keyword, and caption), while the latter focuses on the visual features of the image (e.g. size, colours, textures, etc.) [SUTA99], [JORG98], [GUDI97], [YANG97], [SHAK96], [CHUA94].

On a conceptual level, it is clear that humans employ a variety of socio-cognitive processes, as well as sensory skills, in the retrieval and evaluation of visual information. Content-based indexing has the advantage of providing a higher-level analysis of the image content but is expensive to implement and suffers from a lack of inter-index consistency due to the subjective nature of image interpretation.

Concept-based indexing uses controlled vocabulary or natural language to express what an image is or what it is about [ZHEN99], [MA98].

A method of evaluating image similarities, based on the modified "longest common subsequence" algorithm, is presented by many researches, but the proposed evaluation method cannot only separate out those images of which all icons and their spatial relationships are in full agreement with query images. The representation model and similarity evaluation also simplify the retrieval progress of linear transformations, including rotation and reflection, of images [SAME90], [CHAN87], [GUTT84].

The ability for users to express their search needs accurately and easily is crucial in any retrieval system. Image retrieval is no exception to this, though it is by no means obvious how this can be achieved in practice [RABB89].

The ability to refine searches online in response to user indications of relevance, i.e. relevance feedback, is particularly useful in image retrieval, even though it was originally developed for text [LU00]. This is because users can normally judge the relevance of a set of images displayed on the screen within seconds, and because so few current systems are capable of matching users' needs accurately first time round. The usefulness of relevance feedback for image retrieval was first demonstrated within a keyword-based system [PRIC92]; the facility has now been successfully implemented in several image retrieval systems [SMIT97], [SMIT97a], [RUI97]. However, there is still considerable scope for more research into improved interfaces for image retrieval systems, in particular the development of better methods for users to convey individual notions of image similarity [SANT97].

2.3.1 Concept-based Image Indexing and Retrieval

Indexing and abstracting are fundamental concepts in image retrieval, since they decide the effectiveness and efficiency of the retrieval process and results. In many scientific database applications, the information content of images is not explicit, and it is not easily suitable for direct indexing. In particular, the large scale image databases emerge as the most challenging problem in the field of scientific databases [OBEI01].

Concept-based is an improvement over keyword based indexing because it removes the ambiguities between keyword senses due to synonymy. Concept-based indexing is a primarily verbal and abstract identification of high-level concepts in an image. This type of indexing requires the recognition of meaning and is primarily performed by humans. Most production-level library systems rely on concept-based indexing using keywords. Manual keyword indexing is, however, expensive and introduces problems with consistency [RASM97].

In this section, we explain image indexing and retrieval with the use of ‘Moving Picture Expert Group (MPEG)’, and also some various web search engines that are used for indexing and retrieval methods.

Moving Picture Expert Group (MPEG)

The Moving Picture Expert Group (MPEG) is a working group that is in charge of the development of international standards for compression, decompression, processing, and coded representation of moving pictures, audio and the combination of both. MPEG files use less memory compared to other coding format for the same quality, which is advantageous [HWAR00].

MPEG-7 offers a comprehensive set of audiovisual Description Tools like the Description Definition Language (DDL), Description Schemes (DSs), and Descriptors (Ds), which form the basis for applications enabling the effective and efficient access to multimedia content. MPEG-7 consists of description schemes based on XML (Extensible Mark-up Language) as the language used for the purpose of Structured Annotations. MPEG-1, MPEG-2 and MPEG-4 make the content available while MPEG-7 helps in finding the content [FRAU01].

The MPEG-7 structured annotations provide a way to describe in XML the important concepts relating to an image in order to facilitate indexing, searching and filtering. The general requirement within MPEG-7 is that it should be based on XML syntax [MPEG02]. MPEG-7 Description Tools allows the creation of descriptions of content that may include:

- Low-level feature information of the content like colours, textures, sound timbres, melody description.
- Conceptual information of the contents such as objects and subjects, relationships among objects.

There are two ways of presenting the XML documents on the browser, the first is to display XML documents by transforming from XML format to HTML format; and the second when using a transformation language such as XSLT that is part of XSL [AZZA04a].

MPEG-7 description tools are used to extract the content feature and also for effective and efficient searching, filtering and browsing of the content according to the user needs as shown in Figure 2.1.

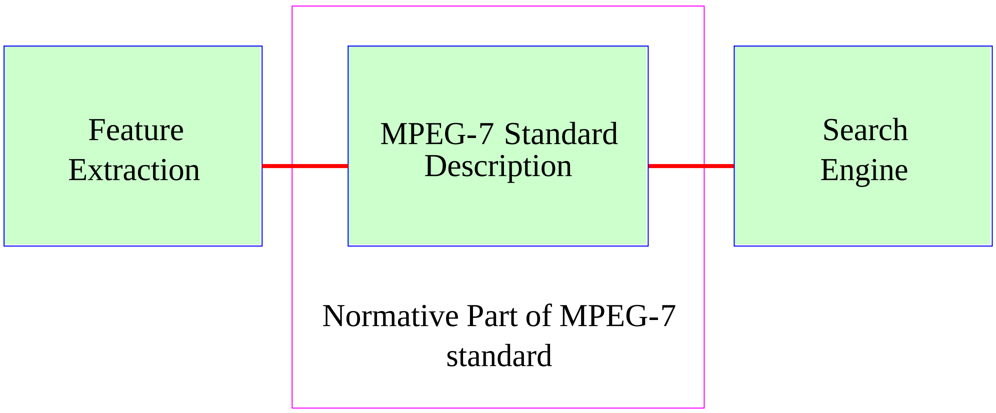


Figure 2.1 Scope of MPEG-7

MPEG-7 description tools include the use of the Structured Annotations Description Schemas to define the nature of XML schemas and their component parts. The Structured Annotation is treated as a data type of XML. Structured Annotations gives the images textual description of events, people, animals, objects, places, actions, purposes, times, attributes, behaviour and relations among them. It provides a structured format that is simple but is also an expressive and powerful annotation tool. Syntactic relations such as subject, direct object, indirect object, and verb modifiers between actions and objects can be described. In addition the modifiers can be described for people, animals, objects, places, actions and times [AZZA04a]. XML schemas and their component parts are explained as follows.

Subjects

A subject is referred to as the textual description of the grammatical ***Subject*** of visual data, often representing living beings (people, animals), for example, "*A **Koala** sitting on the tree*" and "*the **tree** shedding its leaves*".

<SUBJECT> koala </SUBJECT>

<SUBJECT> tree </SUBJECT>

Subjects can also have attributes like colour or size that can further describe the subject and can be specified by referencing the corresponding textual description of behaviour using the *modifier* attribute (grammatically speaking it is an adjective), for example, "*The **grey** koala sitting on the tree*" and "*The **tall** tree shedding its leaves*"

<SUBJECT modifier = grey> koala </SUBJECT>

<SUBJECT modifier = tall > tree </SUBJECT>

Objects

An Object is referred to as the textual description of the grammatical ***object*** of visual data. The objects can be divided into direct objects and indirect objects, e.g." *The batsman drives the **cricket ball** with his **bat**"*.

<OBJECT> cricket ball </OBJECT> (referring to direct object)

<OBJECT> bat </OBJECT> (referring to indirect object)

Similar to subjects, objects also have attributes covering all aspects of modifiers, For example, *"The batsman drives the **white** cricket ball with his bat"*.

<OBJECT modifier = white> *cricket* ball </OBJECT>

Actions

Actions are referred to as the textual description of the grammatically occurring ***Action*** of the visual data corresponding to verbs, e.g. "*The player **lifting** the Australian open trophy, his national flags are **waved** by the crowd."*

<ACTION> lifting</ACTION>

<ACTION> waved </ACTION>

Actions also have attributes, which can be specified by using the modifier attribute. Attributes of actions grammatically correspond to adverbs, e.g. "*The player **proudly** lifting the Australian open trophy, his national flags are waved by the crowd."*

<ACTION modifier = proudly> lifting</ACTION>

Sub-elements

Apart from the main elements like Subject, Object, and Action there can be Sub-elements such as: when, where, how, and why, which are used to describe more specifically the aspects corresponding to the Action. The Sub-elements can also have attributes, which can be specified by using the modifier attribute. Below is an example of the presentation for the data type as shown in Figure 2.2.

Figure 2.2 Playing Children Image

<TextAnnotation>

<FreeTextAnnotation xml:lang="en"> The children throwing the basketball. The children playing basketball </FreeTextAnnotation>

<StructuredAnnotation>

<Subject href=Image_location/img001.jpg">

<Name xml:lang="en">Children</Name> </Subject>

<Object> <Name xml:lang="en">Basketball</Name> </Object>

<Action> <Name xml:lang="en">throwing</Name> </Action>

<Action> <Name xml:lang="en">playing</Name> </Action>

</StructuredAnnotation>

</TextAnnotation>

Here is another example shown in Figure 2.3 of the presentation for the data type containing multi-subjects and modifiers.

Figure 2.3 Boxing Image

```
<TextAnnotation>
    <FreeTextAnnotation xml:lang="en"> The boxer in the white jersey punches
    the boxer in black jersey. The referee and scorer are watching the game.
     </FreeTextAnnotation>
    <StructuredAnnotation>
        <Subject modifier=”white jersey” href=Image_location/img002.jpg">
            <Name xml:lang="en">the boxer</Name> </Subject>
        <Object modifier=”black jersey”>
            <Name xml:lang=”en”>the boxer</Name>
        </Object>
        <Action> <Name xml:lang="en">punches</Name> </Action>
        <Subject> <Name xml:lang=”en”> referee</Name> </Subject>
        <Subject> <Name xml:lang=”en”> scorer</Name> </Subject
        <Object > <Name xml:lang="en"> game</Name> </Object >
        <Action> <Name xml:lang=”en”> watching </Name> </Action>
    </Structured Annotation>
</TextAnnotation>
```

Web Engines and Image Retrieval

Image search engines are based on existing search engine technology, but they use additional strategies to identify, categorize and rank images. A search engine's indexing of images is done automatically, rather than using human indexers, so it must find ways to predict the image's content. It might take into account its filename or any accompanying ‘ALT’ image tags (The ALT text provides alternative or substitute text, for use when the image is not being displayed). It might look for clues from the image's context, for example, the words or phrases that are close to the image, or the ‘META’ tags found at the top of the HTML coding. The characteristics of the Web site and its server will also often be taken into account [TASI03].

Analysis of an image's text and context can be used to exclude images as well as include them, for example, an image engine will usually consider an image's context and associated words when it is blocking out adult material.

2.3.2 Content-based Image Indexing and Retrieval

In content-based methods, images can be retrieved according to their superficial visual appearance, not their semantic meaning. In content-based images concept, features are extracted from the images during population of the image database. The features extracted are based on low-level features like shape of objects in the image, image colour and image texture. These features are used to index the images. When making a search, the user enters a query into the system and features are extracted from the query image. The query image is compared with the database image by the image retrieval system on the basis of the closeness of the extracted features and the best matched images which are presented to the user. Since the images are rich in information, integrated approach incorporating a number of features are required [SAJJ97].

Visual properties of an image are extracted into victors according to some low-level features, such as colours, textures and objects shape [RUI96], [SMIT96], [ZHAN04]. In such a way, images are transformed into one or several points in a high dimensional feature space. Some similarity functions can then be defined to measure the distance between points, and the images whose points have the shortest distance are deemed the most similar to each others; colour is one of the most important features of an image [STRI95].

The most commonly used colour representation is colour histogram, where the colour spectrum is divided into a number of bins and the value of each feature represents the number of pixels in an image in that colour range [SWAI91]. The colour sets approach offers fast search because colour set feature vectors is binary, and shape representation can be classified as boundary-based and region-based representations. Boundary-based shape representation uses outer boundary of the shape, while region-based shape presentation uses the entire shape interior [FENG04].

2.3.3 Search efficiency

A significant limitation of current Concept-Based Image Retrieval (CBIR) technology is the problem of efficiently retrieving the set of stored images most similar to a given query. One of the many fundamental ways in which CBIR differs from text retrieval is that it is based on a fundamentally different model of data. Most text retrieval systems associate each document with a variable number of descriptors representing its content.

Most current CBIR systems work on a completely different principle. Stored images are typically characterized by fixed-length real-valued multi-component feature vectors, each image having a value for every feature in the database. In this case, searching consists of calculating the similarity between feature vectors from query and stored images, a process of numerical computation. As [SANT97] pointed out, the primary aim of traditional text retrieval systems is to *partition* a database into two sets “relevant items and non-relevant items”, even if members of the first set may later be ranked by relevance.

By contrast, the prime aim of CBIR systems is to sort the database in order of similarity to the query. Finding index structures which allow efficient searching of an image database is still an unsolved problem [FALO94]. None of the index structures proposed for text retrieval has proved applicable to the problem.

The most promising approach so far has been multidimensional indexing, using structures such as the R*-tree [DICK98], the TV-tree [LIN94] and the SS $^+$-tree [KURN97]. A more recent approach, which seems to offer better prospects of success, is the use of similarity clustering of images, allowing hierarchical access for retrieval and providing a way of browsing the database as a bonus [JIN98], [VELL98].

2.3.4 Thesaurus

A Thesaurus is a list of synonyms a search engine can use to find matches for particular words if the words themselves don't appear in the databases.

In a database, a thesaurus uses the terms you choose to search on to lead you to other terms you may not have considered (related terms). While the book-type thesaurus presents you with a larger variety of words to choose from, the goal of a database thesaurus is to funnel your search into the unique, official language of the database “subject headings”. For example, a database thesaurus might suggest that you use the word “automobiles” instead of the words such as Car, Cars, Autos, Vehicles, etc.

In some search tools, the terms you choose to search on can lead you to other terms you may not have thought of. Different search tools have different ways of presenting this information, sometimes with suggested words you may choose amongst and sometimes automatically. The terms are based on the terms in the results of your search, not on some dictionary like thesaurus.

2.3.5 Controlled and Uncontrolled Vocabulary

When users' perform Internet searches they tend to look for general topics. This is because they may be unsure of what they are looking for exactly, or may want to collect as much information about a topic as they can to familiarise themselves with it. Studies conducted on users have found that they tend to do searches for general topics and then use only one or two of the resources (Controlled and Uncontrolled Vocabulary) retrieved.

Controlled Vocabularies

Controlled vocabularies are one of a variety of tools and techniques that are used to improve information retrieval accuracy, and handle the problem of mass retrieval. A controlled vocabulary is a set of standard terms used to describe a resource. They can be a simple list of defined terms from which a metadata creator chooses a suitable term. On the other hand, a controlled vocabulary might be a complex thesaurus made up of hierarchical relationships and synonyms. Controlled vocabularies can be used by query engines to help users define their subject area as closely as possible, and in a way that is as consistent as possible. Professional indexers add controlled vocabulary terms from a standard list to records as they are entered into the database, so that all the records on the same topic can be retrieved using the same search term. The controlled vocabulary terms are entered into a special field in each record. In library catalogues, controlled vocabulary terms are in the subject headings field. In other databases, they are often in fields labelled descriptors or subjects.

When you find a record that you like in a database you may look at its descriptor or subject field to find what controlled vocabulary terms were used to index that record. If any of the terms are appropriate to your topic, they can be used to expand your search. Another way to find controlled vocabulary terms is to search or browse the same standard list that professional indexers use. This list is called a thesaurus.

Uncontrolled Vocabularies

Uncontrolled vocabulary indexing is often referred to as free text indexing since the index term is usually derived from the documents themselves, this approach used for software reuse [FRAK88], on the other hand has taken the opposite approach by developing a controlled vocabulary using a faceted classification scheme [PRIE87]. While uncontrolled vocabularies will tend to be less expensive to implement, the performance tradeoffs between controlled and uncontrolled vocabularies are unknown for collection of reusable components.

Experiments with document retrieval systems beginning in the early sixties have shown that uncontrolled vocabularies produce retrieval results that are comparable to those produced with controlled vocabularies [SPAR81], [SALT83].

A database thesaurus is similar to a language thesaurus; it lists synonyms and indicates relationships among terms. It has additional features that help with searching:

- It often includes definitions, or scope notes, for the chosen terms, explaining their meaning in the database.
- It also lists synonyms that are not used in the database and directs the user from those to the chosen terms.
- Under a chosen term, it lists other chosen terms that are more specific or more general.
- It may also list equally specific terms that are used for another facet of the chosen term.

Controlled Versus Uncontrolled Vocabularies

The advantages and disadvantages of searching using controlled and uncontrolled vocabularies are as follows:

The main advantages of using a *controlled vocabulary* for searching are:

- Greater focus and level of relevancy
- Structure allows greater precision searches
- Lower recall
- Fewer missed citations because of terminology problems
- Not having to think of all the ways a topic maybe described
- Avoids problems with spelling variations
- Allows use of hierarchical searching (i.e. start broad and get narrower)

The main disadvantages of using a *controlled vocabulary* for searching are:

- Some citations may be missed if an inappropriate controlled vocabulary term is chosen.
- New or unusual topics are not well represented.
- Very specific and well-defined topics are not well represented.
- The constantly evolving nature of human knowledge means controlled vocabularies have to be updated regularly.
- Users need to be made aware that a controlled vocabulary is available when performing a search and need to learn how to use it.
- Searches that use controlled vocabulary can search the subject or descriptor fields only.
- Takes longer to perform a search using controlled vocabulary, as all terms related to a topic must be searched for.

The main advantages of using an *uncontrolled vocabulary* for searching are:

- Subject, title and abstract fields can be searched.
- Any term that is keyed in may be searched.
- Allows searches on novel topics for which there may be no controlled vocabulary available.
- Ability to perform very specific searches if the exact subject can be well-defined.
- Can often provide more results in a shorter time span, as there is no need to search through controlled vocabulary subject headings.

The main disadvantages of using an *uncontrolled vocabulary* for searching are:

- Low precision search.
- May retrieve many irrelevant items.
- No standard agreed terms for particular topics.
- Have to think of all the terms that could cover one topic, meaning that time must spent preparing the search strategy.
- In order to search successfully keywords , it should be used in conjunction with Boolean and proximity operators
- Specialist knowledge may be required in order to use these operators correctly and successfully.

2.3.6 Image Retrieval Engines and Evaluation Criteria

In reviewing image search engines, it is helpful to use an evaluation criteria and a standard set of search terms. Although the exercise is still subjective, these will offer some basis for comparison. The evaluation criteria mainly depend on four points illustrated below; every point has its own requirement to satisfy the evaluation [TASI03]. The evaluation criteria, standard terms, and some of the major search engines offering image search options are shown later on in this section.

1. Scope

- *Size and range*: how big and comprehensive is the index? Is this clearly stated?
- *Currency*: how often is the index updated? Is this clearly stated?

2. Search Options

- *Simple and advanced*: are simple and advanced search options available?
- *Sophistication*: is it possible to conduct Boolean, phrase and wildcard or truncated searches?
- *Filtering*: can searches be limited before and after the search?

3. Performance

- *Speed:* the results quickly returned?
- *Number of hits*: Does the number of results seem reasonable given the size and scope of the index? Is it possible to view all the results?

- *Relevancy*: how well do the results match the query?
- *Quality*: how good are the images supplied?
- *Currency*: are there any missing images or dead links?
- *Duplication*: is there any obvious duplication of images?

4. Presentation

- *Thumbnails*: are the results presented as thumbnails or just text links?
- *File Information*: do the results include any of the following useful information: file format, file size, image dimensions and colour information?
- *Other Metadata*: do the results include any of the following: caption, filename, alt tag, page title or other associated text?
- *Copyright*: is it clear who owns the copyright and how the images can be used? Is it possible to link to the original source of the image?
- *Adverts*: is the page cluttered with adverts? Are the results skewed in favour of commercial images?

Currently, the main English-language-based search engines offering image searches in the Web are AllTheWeb, AltaVista, Google and Lycos. Yahoo (strictly speaking a directory rather than a search engine) offers a picture search. HotBot, Web Search, and MSN Search do not offer true image search features; they simply enable users to limit their search results to 'only pages with images' [TASI03].

Image search engines attempt to give access to the wide range of images available on the Internet. For those used to viewing well-indexed collections of quality images, the results of the large automated image search engines will probably disappoint. The poor quality of their offerings is not surprising, since they reflect the randomness and unevenness of the Web. The frequent irrelevancy of their results is also understandable, since the automated engines are guessing at their images' visual subject content using indirect textual clues

Anything, then, that enables the user to have more control over their image searching is helpful. The ability to filter a search, to include and exclude items, is important in any searching, but particularly so when searching for images. An image search engine also needs to return a reasonable number of results, since in any given search a fair amount of the images found are likely to be irrelevant or of insufficient quality. Here is where the meta-image search engines tend to fail their users. They deny them sophisticated filtered searching and generally only bring back a handful of the results they find. Internet search engines can be useful in identifying the best image search engine for a particular task, but the final searching is better done directly, using the source search engines themselves. Google performed well in this review, but it is worth trying some of the others. As TASI's and our evaluation suggestions, they each have strengths and weaknesses.

Collection-based image search engines include images selected for quality and indexed by hand. The images they contain are seldom found within the results of general Web search engines. Collection-based engines, then, will usually offer much better results than their search engine counterparts. The commercial and copyright issues will also be much clearer, although many users seem to prefer to look to image search engines for images they can freely reuse, as if easy access and absent copyright notices lets them off the moral and legal hook. True copyright-free images are rare on the Web, and many image search engines do operate on commercial imperatives, even subtly skew their results towards commercial ends. The Web is a fast changing environment, so it is likely that some of the information in this review may become outdated in short time. The reader is well advised to check the sites themselves and to ask the kind of questions listed in the evaluation criteria, done in this chapter.

2.4 Problems of Existing Image Indexing and Retrieval Methods

There are many problems and disadvantage of existing image indexing and retrieval systems in both “concept-based” and “content-based” methods; we state some of these problems and disadvantages for each method.

2.4.1 Problems of Concept-based Methods

The image indexing and retrieval system still suffers from a number of limitations [LEUN92], some of which are:

- The understanding of an image depends on different levels of knowledge and experience.

- Organising the description of a picture’s subject matter in such a way that the picture can be most certainly and efficiently retrieved has to be considered carefully.
- It is tedious and time-consuming to associate keywords or text with each image.
- Some features are very difficult to describe with text, some special textures and complex shapes cannot be clearly represented.
- Text description is sometimes incomplete. It is possible that some image feature may not be mentioned in textual description.
- Text description is sometime subjective. Different indexer or even the same indexer may describe the same feature with different term or different feature with the same term [ROBE94].

A computational model for exploring the interaction of textual and photographic information in document understanding, where textual captions are used as collateral information in the interpretation of the corresponding photographs, is given in [SRIH93]. The final understanding of the image and caption reflects a consolidation of the information obtained from each of the two sources and thus can be used in intelligent information retrieval tasks. Problems often occur when performing a general-purpose search without adequate a priori knowledge.

Most users wish to search images in term of semantic concepts rather than visual contents; work in image retrieval research has begun to shift from query-by-example (QBE) to query-by-keyword (QBK). QBK allows the users to search images by specifying their own query in term of a limited vocabulary of semantic concepts [FENG04], [SMIT03]. The problem with such approach is the need to annotate images with semantic concepts accurately and completely. The traditional approach is to manually annotate images, which is very tedious and time-consuming. Hence, it is desirable to automatically assign semantic concepts (Keywords) to images.

2.4.2 Problems of Content-based Methods

A typical content-based image retrieval system works basically on low-level visual features such as colour, texture, shape or regions. It is hard to link these low-level features to high-level concepts that correspond to objects and their semantics information within an image [CARS99], [SMIT96]. Content-base representation and retrieval provides only insufficient solutions to the problem of images retrieval [DENG04], [SMIT96a].

Naturally image search engines working on low-level features alone may sometime produce a great deal of nonsense results [SANT98]. Techniques such as spatial colour features, joint histograms, image classification and conceptual description have been proposed [HIRA00], [MUKE00], [CORR99], [PASS99], [HUAN98], [HUAN98a]. The difference in description and meaning between content-based representation and semantic description of images is still hard to overcome [SMEU00].

Performance of traditional content-based image retrieval systems is far from user's expectation due to the semantic gap between low-level visual features and the richness semantics [LIU05], [CHEN03], [MOJS01], [ZHOU00]. The success of content-based image retrieval is largely limited by the gap between low-level features and high-level semantics [QIN05].

The problem of content-based image searching has received considerable attention in the last few years. Thousands of images and multimedia are now available on the Internet, and many domain-specific applications require multimedia data searching. Sample domains include sports and entertainment, agriculture, e-commerce, weather prediction, medical imaging, and satellite imagery. Yet, in spite of the growing need for multimedia search capabilities, such technology is still largely unestablished as a mainstream field, nor is it widely used by search engines. Many of the major problems for this poor acceptance are inconsistent retrieval quality and poor usability. The content-based method suffers from a number of limitations [LEUN92], some of which are:

- Query-By-Example (QBE), for instance, it is suitable if a user has a similar image at hand, and a query will recall entities having similar image characteristics. However, depending on the meaning of similarity it would not perform well if the image is taken from a different angle, having a different scale, or placed in a different setting. It is often difficult to query by image content where users have to provide a colour composition, outline the drawing, select a texture etc. Because content-based methods are not rich in image semantics, it is difficult to use it to query high-level visual concepts like an image of a Melbourne Spring Festival. For the content-based disadvantage explained in this section, it is obvious that text-based queries are much better than QBE and Query-By-Image-Contents (QBIC) in this regard. Text-based queries are also very much faster than QBE and QBIC because text processing only takes a small fraction of time compared to image processing. Another advantage of text-based queries is the ease with which user could prepare the query, because it uses the same language as everyday human queries [SUTA99].

- One of the disadvantages of the content-based approach is that images returned from query could be totally different from the target images using shape concept [DICK98]. For example, users asking for images similar to 'doors' may receive images that semantically were not 'doors' at all.
- For the shape retrieval mechanism, many edge maps contain irrelevant edges, and the sketch comparison algorithm cannot calculate the similarity score very correctly.
- Multiple domain recall is another disadvantage of approaches, like QBIC using colour concept [BARB95], [FLIC95], [BARB94]. For example a query of 70% of blue colour and 30% of white colour may return an image of a white car parked behind a blue wall, and a white book on a blue table, and image of a beach with white sand, etc.
- Users can't specify that the absence of a colour from a certain region mean that the image is irrelevant. The position information is not taken into account in the algorithms. Since exact colour match is performed by the histogram intersection algorithms, score zero will be returned for images that do not contain the exact same shades of colour. The efficiency of the histogram intersection is not good; the response time will be poorer for larger database.

- In some methods like Two Dimension (2D) strings, which are used to express the image features and the query, the 2D strings cannot be generated automatically, because it is difficult to identify objects represented in the 2D strings without human intervention**.** Since objects are represented by points in 2D strings, some special relationship, such as DISJOINT, JOIN, CONTAIN, and BELONG between two objects, may not be determined correctly [TANG96].
- Another disadvantage of content-based methods is that they are not rich in image semantics; therefore it is difficult to use them to query high-level visual concepts.

2.4.3 Selection of Concept-based Method

Concept-based approach is used in many production library systems, while the content-based approach is dominant in research and in some newer systems. In the past, content-based indexing supported the identification of "low-level" features in an image. These features frequently do not require verbal labels. In many cases, current computer technology can create these indexes. Concept-based indexing, on the other hand, is a primarily verbal and abstract identification of "high-level" concepts in an image. This type of indexing requires the recognition of meaning and is primarily performed by humans. Most production-level library systems rely on concept-based indexing using keywords. The concept-based query provides an interesting technique for the user to incorporate domain knowledge into their queries. This incorporation overcomes the problem of incomplete knowledge in most knowledge-based systems. The domain knowledge may be stored in the database with no special effort from the designer, and the user may reuse his own concepts in another queries.

In addition to the disadvantages of the content-based methods, the high-level abstraction and concepts can be easily expressed in concept-based methods, the query can be easily issued and the user interface is very simple, and the standard text retrieval techniques can be used for image searching, where the text retrieval techniques have been highly developed in information retrieval system. These are additional advantages of concept-based retrieval system that leads us to select the concept-based retrieval system in our research.

Comparing to content-based image retrieval, text-based retrieval is more successful and mature, with successful application in many important areas, including Web search engines. Content-based image retrieval is still some way behind text-based retrieval in most criteria, including search precision and recall [ZHAN04].

2.5 Rule-Based Approach

In a rule-based system, the domain knowledge is represented by a set of IF-THEN production rules and data are represented by a set of objects about the current situation. The inference engine compares each rule stored in the knowledge base with objects present in the database, when the IF "condition" part of the rule matches an object, the rule is fired and its THEN "action" part is executed. The matching of the rule IF part to the objects produces inference chains; the inference chain indicates how rule-based system applies the rules to reach a conclusion. To evaluate the rules, the inference engine requires that all of the conditions be satisfied as shown in Figure 2.4. It is possible that more than one rule will apply to an image [DETE00]. Figure 2.5 shows an inference chain.

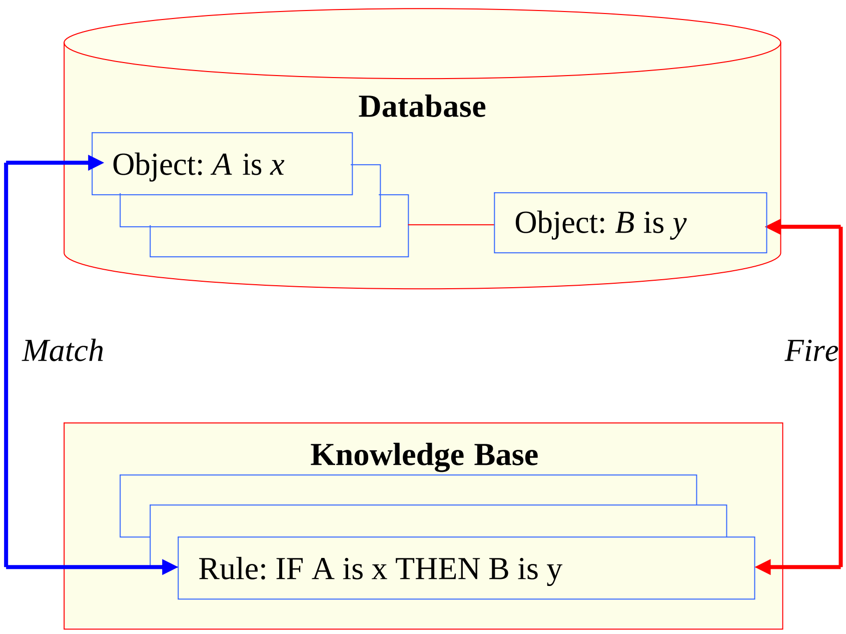


Figure 2.4 Inference Engine Match-Fire Cycles using a Match-Fire Procedure

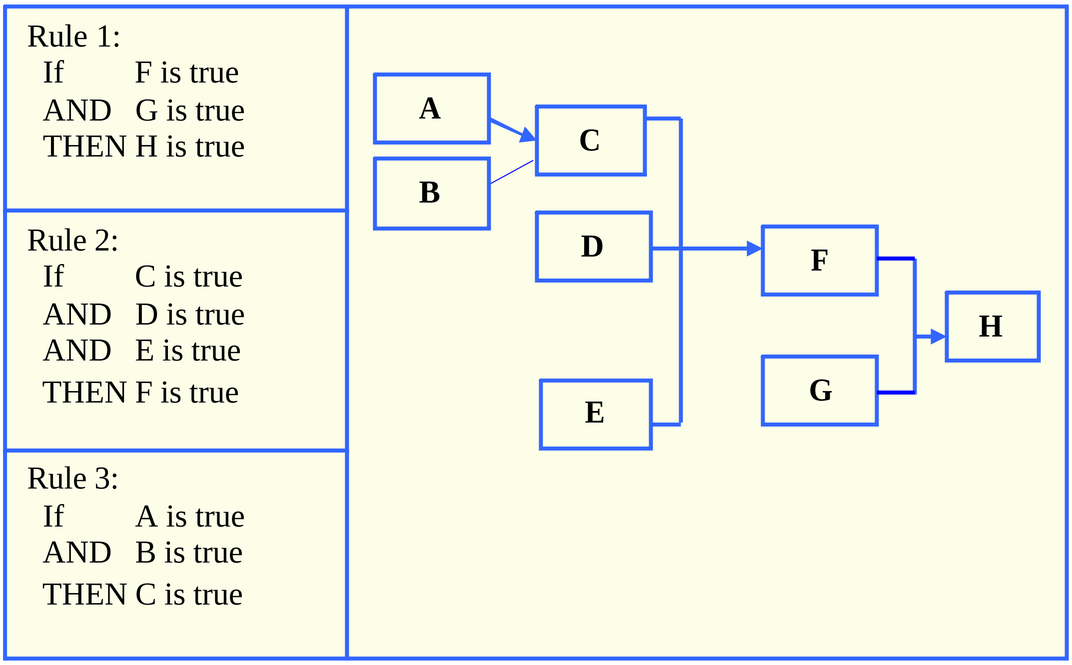


Figure 2.5 Inference Chain

2.5.1 Forward Chaining

Forward Chaining (FC) is data-driven reasoning. The reasoning starts from the known data and proceeds forward with that data, each time only the topmost rule is executed, and when fired, the rule adds a new object to the database. Any rule will be executed only once; the match-fire cycle stops when no further rules can be fired. An example of FC is shown in Figures 2.6, Figure 2.7, and Figure 2.8.

FC is a technique for gathering information and then inferring from it whatever can be inferred; however, in forward chaining, many rules may be executed that have nothing to do with the established goal. Therefore, if our goal is to infer only one particular object, the forward chaining inference technique would not be efficient.

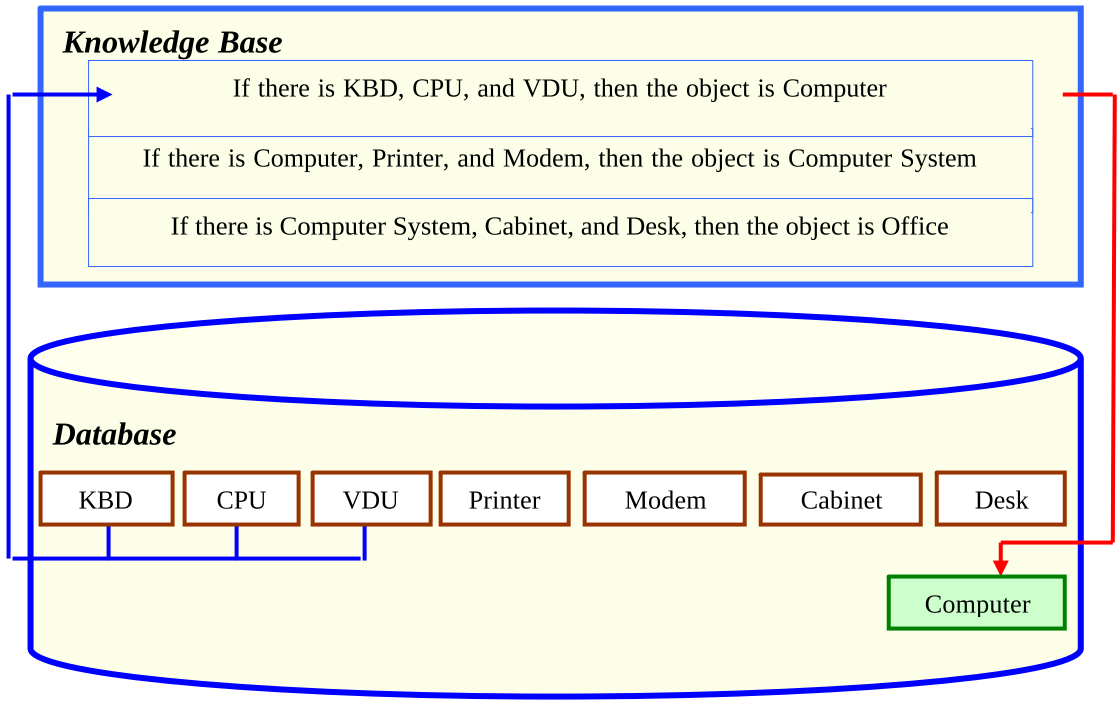


Figure 2.6 Forward “Rule 2.1”

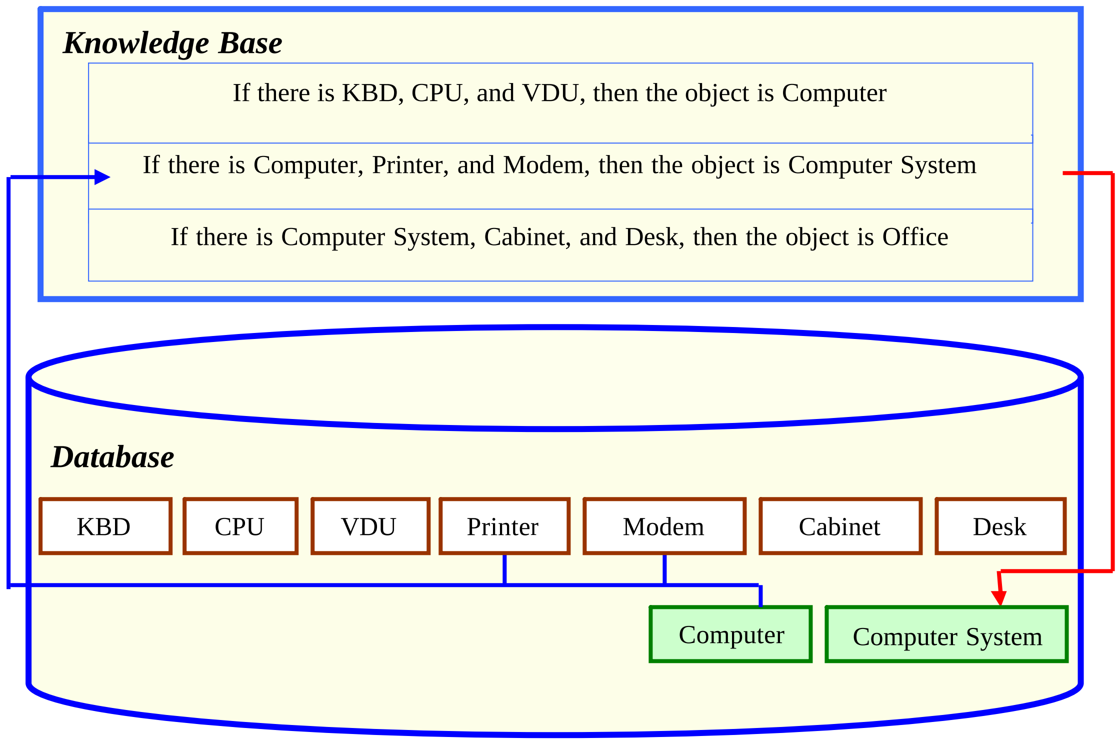


Figure 2.7 Forward “Rule 2.2”

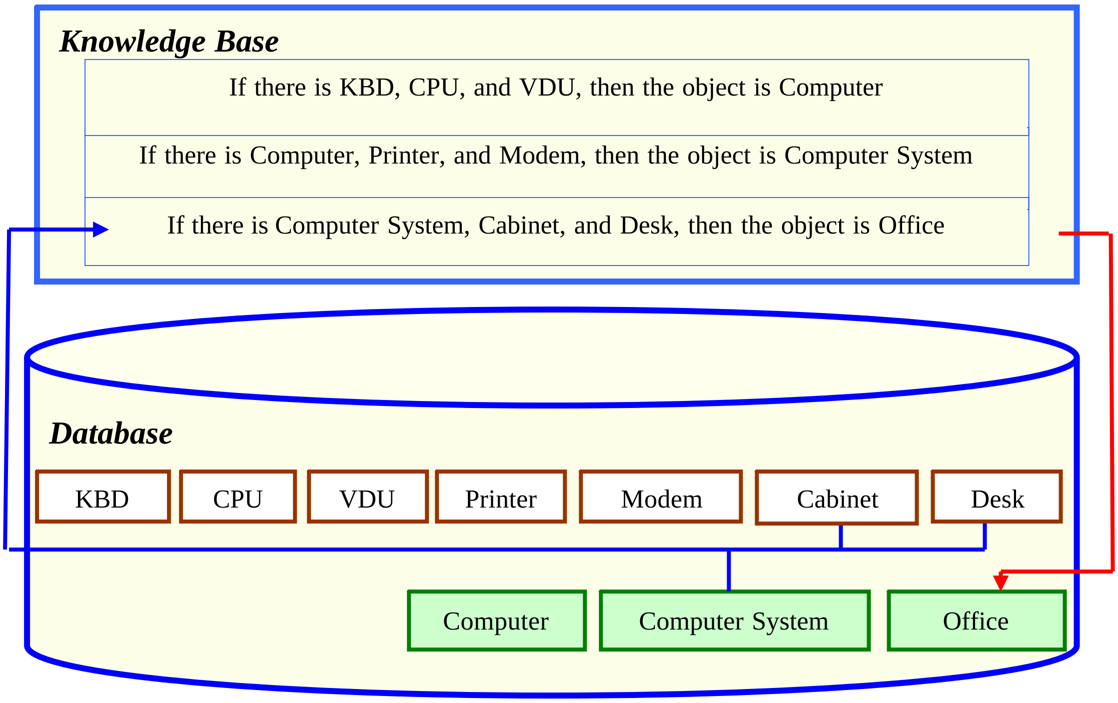


Figure 2.8 Forward “Rule 2.3”

2.5.2 Backward Chaining

Backward Chaining (BC) is goal-driven reasoning. In backward chaining, rule-based system sets the goal and the inference engine attempts to find the evidence to establish it. First, the knowledge base is searched to find rules that might have the desired solution. Such rules must have the goal in their THEN (action) parts. If such a rule is found and the IF (condition) part matches data in the database, then the rule is fired and the goal is proved. However, this is rarely the case. Thus the inference engine puts aside the rule it is working and sets up a new goal, a sub-goal, to prove the IF part of this rule, and then the knowledge base is searched again for rules that can achieve the sub-goal. The inference engine repeats the process of stacking the rules until no rules are found in the knowledge base to prove the current sub-goal. An example of Backward Chaining is shown in Figures 2.9, Figure 2.10, and 2.11.

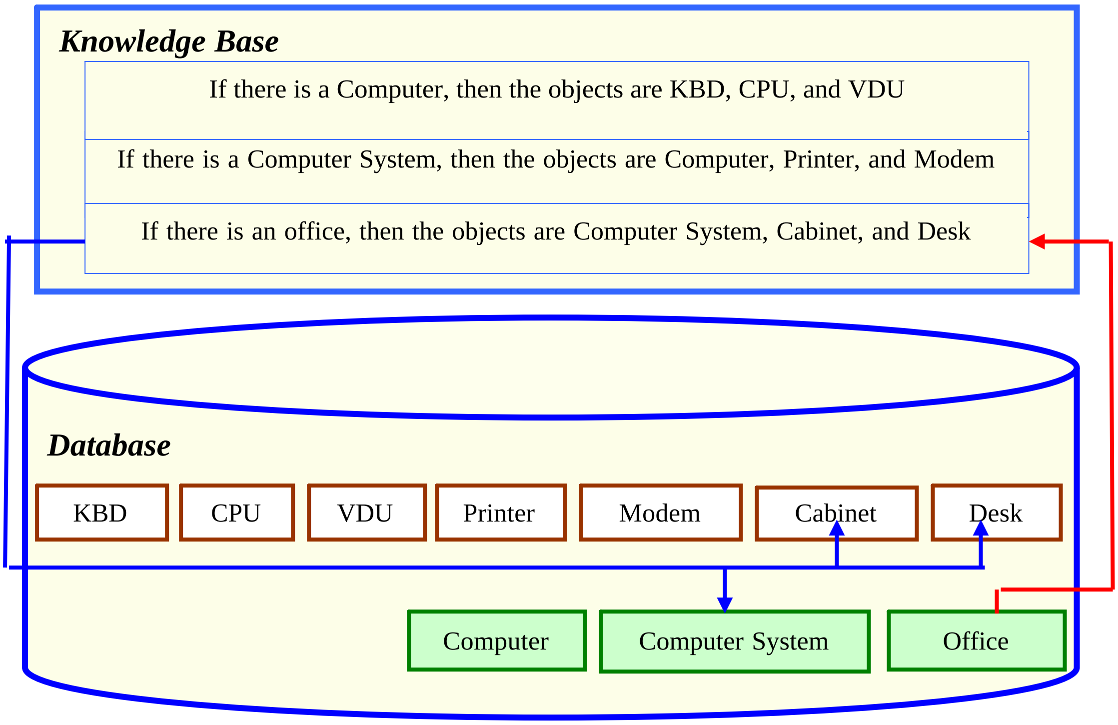


Figure 2.9 Backward "Rule 2.4"

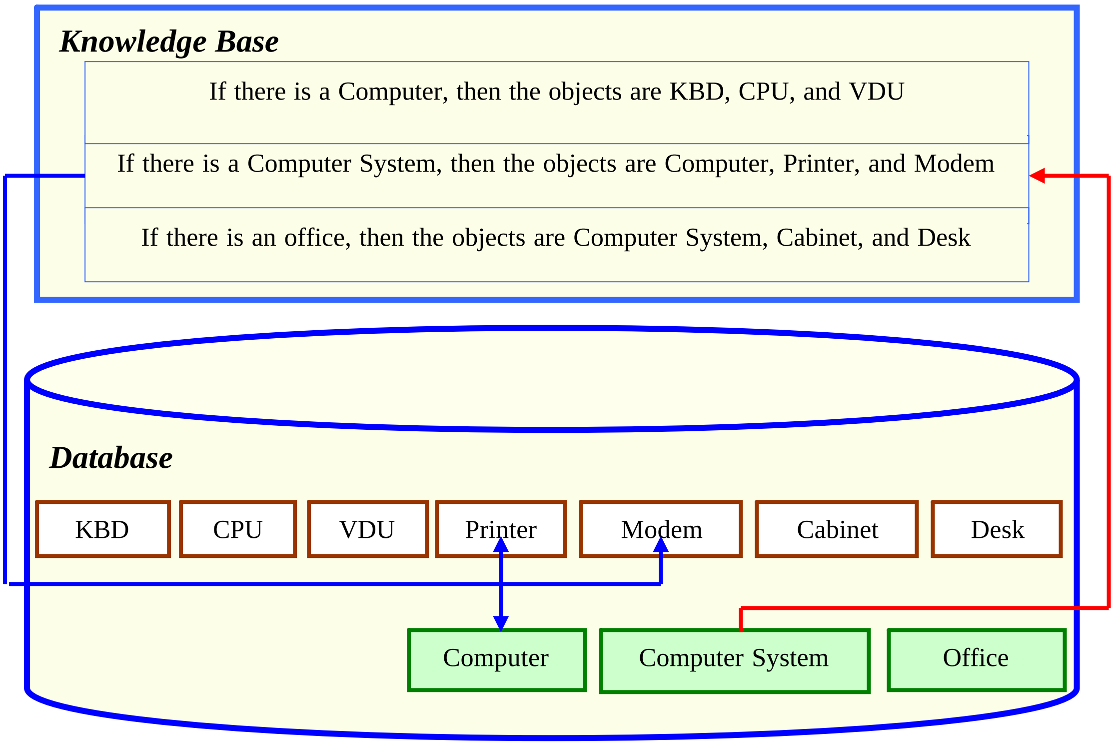


Figure 2.10 Backward "Rule 2.5"

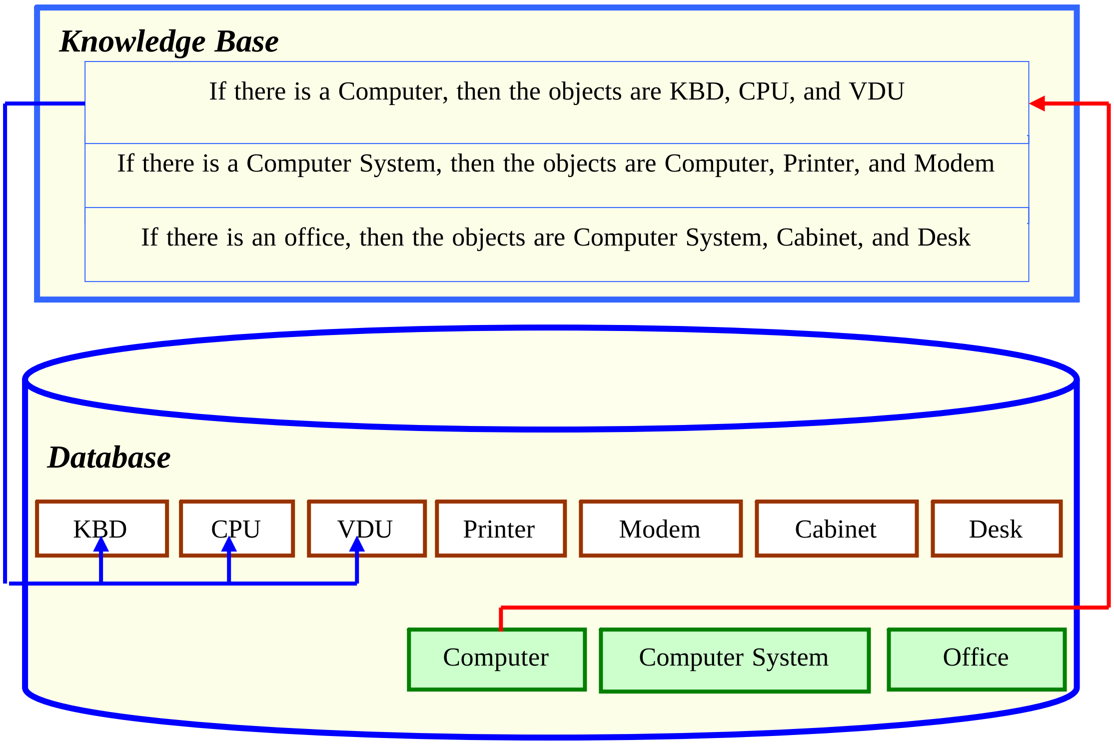


Figure 2.11 Backward "Rule 2.6"

Rules for Problem Solving

The human mental process is internal, and it is too complex to be represented as an algorithm. However, most experts are capable of expressing their knowledge in the form of rules for problem solving.

IF the road is open, THEN the action is go

IF the road is closed, THEN the action is use another road

Rules can represent the following aspect:

1) Relations. 2) Recommendations. 3) Directives. 4) Strategies. 5) Heuristics.

Here below are examples for each aspect.

Relation IF the ‘fuel tank’ is empty, THEN the car is dead.

Recommendation IF the season is autumn AND the sky is cloudy AND the forecast is drizzle, THEN the advice is ‘take an umbrella’.

Directive IF the car is dead AND the ‘fuel tank’ is empty, THEN the action is ‘refuel the car’.

Strategy IF the car is dead, THEN the action is ‘check the fuel tank’;

THEN step1 is complete.

IF step1 is complete AND the ‘fuel tank’ is full, THEN the action is ‘check the battery’; THEN step2 is complete.

Heuristic Heuristic rules can be applied unsuitably if some condition is omitted with some understanding of the problematic system, these inadequacies could be overcome.

IF the spill is liquid AND the ‘spill pH’ < 6 AND the ‘spill smell’ is vinegar, THEN the ‘spill material’ is ‘acetic acid’.

Decision Trees and Rule Generation

Decision tree is a classifier in the form of a tree structure (Figure 2.12), where each node is either: A leaf node, which indicates the value of the target attribute of examples, or a decision node, which specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test. A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the classification of the instance. Decision tree induction is a typical inductive approach to learn knowledge on classification. The key requirements to do mining with decision trees are:

- *Attribute-value description*: object or case must be expressible in terms of a fixed collection of properties or attributes; this must have been provided in the algorithm.
- *Predefined classes (target attribute values):* The categories to which examples are to be assigned must have been established beforehand (supervised data).
- *Discrete classes:* A case does or does not belong to a particular class, and there must be more cases than classes.
- *Sufficient data:* Usually hundreds or even thousands of training cases.

A decision tree takes as input an object or situation described by a set of properties, and outputs a yes/no decision. Decision trees therefore represent Boolean functions. Functions with a larger range of outputs can also be represented. Similar are applied in our system when indexing and searching images with different aspects.

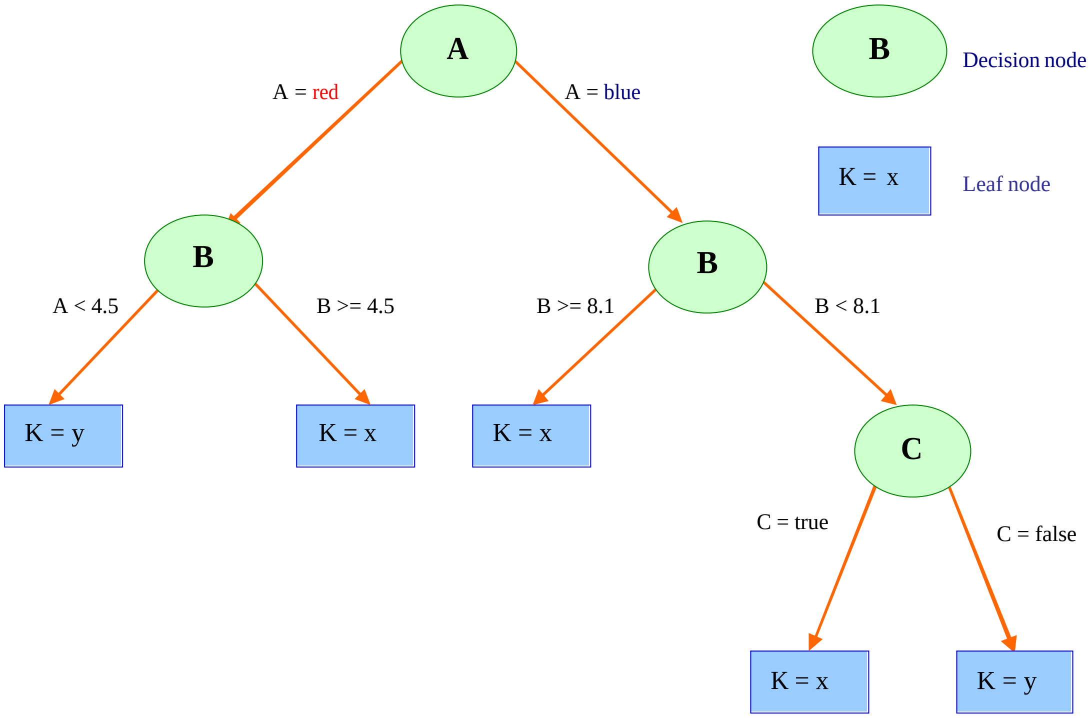


Figure 2.12 Simple Decision Tree of Rule Generation

A decision tree takes as input an object or situation described by a set of properties, and outputs a yes/no decision. Decision trees therefore represent Boolean functions. Functions with a larger range of outputs can also be represented. Similar are applied in our system when indexing and searching images with different aspects. Decision trees are powerful and popular tools for classification and prediction. The attractiveness of decision trees is due to the object that, in contrast to neural networks, decision trees represent rules. Rules can readily be expressed so that humans can understand them or even directly used in a database access language like SQL so that records or images falling into a particular category may be retrieved.

In some applications, the accuracy of a classification or prediction is the only thing that matters. In such situations is not necessarily of concern how or why the model works. In other situations, the ability to explain the reason for a decision is essential. In marketing one has described the customer segments to marketing professionals, so that they can utilize this knowledge in launching a successful marketing campaign. These domain experts must recognize and approve this discovered knowledge, and for this we need good descriptions. There are a variety of algorithms for building decision trees that share the desirable quality of interpretability. Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. Decision tree programs construct a decision tree *T* from a set of training cases [KONW02], [QUIN86].

2.6 Summary

In this chapter, we explained the existing methods of modelling and retrieving images in details, the technologies used in these methods, which go with user's needs, and in addition the approaches represented by the visual information system has been illustrated. This chapter also explores the variety of features which have been used to index and retrieve images based on contents, keywords, textual description or captions as well as many more.

An overview of image indexing and retrieval has been introduced showing the existence methods and how these methods deal with the image indexing and retrieval. We end with the efficient indexing and retrieval of the image database, which is necessary for success in image indexing and retrieval systems.

The Moving Picture Expert Group has taken place, its development, and offers are explained in this chapter. The ways of performing the XML documents are provided as well as the MPEG-7 tools. The XML schema and their component parts are thoroughly explained. In addition the structure annotation providing examples of the presentation of data type, which showing how to store information in XML document, are presented.

The use of the thesaurus in our retrieval system is outlined, which is used when the user searches for images where the entered name is not found in our database. The controlled and uncontrolled vocabulary has certain advantages and disadvantages explained in details.

An overview of image retrieval engines, the evaluation criteria and standard terms are elaborated in this chapter. We went through the main English-Language-based search engines offering images search in the Web as well. The limitation of the current image analysis techniques necessitates that most image retrieval systems use some form of text description provided by the users as the basis to index and retrieve images. All methods have these limitations; hence we explain choice of the selection of the concept-based method in our research.

We explained the rule-based approach, and the forward and backward chaining which show us the way of storing and retrieving the image components. We explained many rules aspects for problem solving, which include the heuristic rules and explained its function in more detail; this aspect is used when building our rules to determine the images to which categories could fall in. Finally we explain what the decision tree and rule generation is and what the use of it in our research side by side with the rule-based expert system. We end the chapter by this summary.

Chapter 3

Image Organization

3.1 Introduction

Throughout this thesis, we shall use a collection of images to illustrate and validate our methods. In this chapter, we describe in some details the characteristics of the image collection and issues related to it.

Images need to be stored in such a way that multi-resolution images can be efficiently retrieved, and so that ideally, one can quickly retrieve only portions of these images without having to decompress the whole image. This extraction of sub-images needs to be performed in parallel in order for the system to have a fast response time. One can develop several storage and compression strategies that could be efficiently used in a parallel computer system with multiple storage devices. For example, to allow multiple simultaneous accesses to different portions of an image, one can hierarchically partition the image [MARK92], [BODL91].

Categorizing images based on various criteria is a key task for all image management systems. Most current systems use some type of key-wording to categorize images in a collection. The key-wording concept has proven to be useful but limited [IMAT04], [HAN96], [SELO90]. A drawback of this concept is that there exist mismatches between indexing terms and query terms. In practice, it is often much more convenient to create a hierarchical structure that is optimal to emphasize relations between images or to order images based on projects, clients or themes. In the real world, images tend to fall into multiple categories. For example, an image showing a group of people on the beach may be assigned to categories like "Beach", "People", "Female", "Male", and to themes like "Leisure", "Fun", "Recreation", "Holidays", “Swimming", etc. If the image also shows a car, you may want to denote that by assigning the image into the category "Cars" as well.

From the feature of images belonging to the same category, the system finds the pattern of interest in the form of rules based on two statistical measures (conditional probability and implication intensity). This is helpful for automatic classification of new images during their insertion in a large database, for obtaining results with more semantics and for improvement in the retrieval [BOUE98].

Image classification helps the selection of proper features and descriptors for indexing and retrieval. Semantic categorization provides image organization according to high-level meanings. Efficient organization enhances not only the retrieval accuracy but also the retrieval speed. Current work of image classification relies on either low-level features or knowledge rules [SZUM98], [LIPS97]; a hierarchical clustering technique for natural image database organization and summarization, where images were grouped via low-level feature clustering has been proposed [VELL98].

A text-based query system based on the Ternary Object Model (TFM) database was developed by [LEUN99], [SUTA99]. TFM comprises an underlying visual entity-relationship index representation, rule-based conceptual hierarchy, and other features to support semi-automatic indexing and to enhance query performance using a thesaurus system, relevance feedback mechanism, and user-tailored weighting components. The TFM Modeling approach has been proven to give component performance in terms of recall and precision in the retrieval of images. TFM is used by expanding and restructuring elementary objects and outlining objects into a hierarchy.

Our approach to image indexing and retrieval system is to develop algorithms for the storage and retrieval of image data from stored databases using a methodology that incorporates implicit image indexing. We develop techniques for the categorization of image components, and the assignment of weights, supporting factors, etc to these components on a proportional scale based on their relative importance within the image [AZZAM04], [HWAR00].

3.2 Image Collection and Rules Construction

The selected images used in our database represent a good selection of multi-object images, since the selected images comprise two or more objects, repetitive objects, objects having relationships to each other, and so on. Some of these objects could be primitive and some others may be non-primitive objects. Primitive objects are defined as the smallest entity that cannot be decomposed further into components. Non-primitive objects could consist of two or more of the primitive objects; could also consist of two or more primitive and/or non-primitive objects. We collect these images from many sources; these sources are generally free of charge if the images were used for research purposes.

3.2.1 Image Collection

We have approximately 1000 images collected in our database, which are divided into 20 different categories, with each category consisting of around 50 images varying in size. Images belong to certain categories depending on which category rule they fall under. The rules and image belonging are determined and explained later on in this section. The main criteria of the collected images are: the collected images are selected from a large number of different images; every selected image must satisfy the target categories rules.

The selected images follow the image selection rules listed below:

1. *Number of objects*: a good cross section of the image collection contains images with different number of objects.
2. *Repetitiveness of objects*: there should be images, in which all objects could be different from each other, and others the same (many birds).
3. *Number of relationships*: the images should have relationships between objects present within them.
4. *Categorization*: the images should contains both, images that can only belong to one category "Garden" for example, and images that could be put into many several different categories, for example, "Garden, Party, and Wedding". Images are related to the certain categories if they satisfy the category rules. For example, if an image contains Trees, Grass, Flowers, Open Area, Bride, Groom, and People, then this image can be put into two categories, "Garden" and "Wedding", since it satisfies the rules for both simultaneously. The Garden category contains Plants (Trees, Grass, and Flowers) and Open Area as essential objects, whereas the Wedding category contains Bride and People as essential objects. The images of different categories are selected, but the categories should not be disjoint, for example, an image containing snow may fall in the sports category of winter games and the holiday category [AZZA04a].

Other images are collected from different Web pages in order to measure the performance of the ICIIR system. Every image contains three attributes, the image number, the image name (taken from the title found with the image in the Web) and the destination. This information can easily be stored in the database for our testing purposes; the images are stored in separate image collection folders.

3.2.2 Rules Construction

Applying the heuristic rules, the image can fall in a category if it satisfies its creation rule. For example, the image can be categorized in the Garden category if it contains the essential objects which satisfy the Garden category rule; every Garden image must contain the essential objects and may contain other objects. All other categories follow the same technique, the rule numbers and the minimum objects to create the categories are listed below in this section.

Every category needs at least two rules; one to create the mid-level object, and the other for the high-level object; this for any three-level image. The number of rules may increase when the number of levels increases.

The category always means the highest level object created by rules for the certain image; other created objects could be considered sub-categories, which must belong to a category. We explain below the use of all rules and the objects created by these rules.

1. Garden Category

$R1m_1$: IF there are Trees, and Grass, then the object is “Plants”

$R1m_2$: IF there are Trees, and Flowers, then the object is “Plants”

$R1m_3$: IF there is Grass, and Flowers, then the object is “Plants”

$R1m_4$: IF there are Trees, Grass, and Flowers, then the object is “Plants”

$R1h_1$: IF there are Plants and Open Area, then the object is “Garden”

2. Wedding Category

$R2m_1$: IF there is Female and Wedding Clothes, then the object is “Bride”

$R2h_1$: IF there is Bride and People, then the object is “Wedding”

$R2h_2$: IF there is Bride and Groom, then the object is “Wedding”

3. Soccer Category

$R3m_1$: IF there are People and Soccer Clothes, then the object is “Soccer Players”

$R3h_1$: IF there are Soccer Players and Field, then the object is “Soccer”

$R3h_2$: IF there are Soccer Players and Referee, then the object is “Soccer”

$R3h_3$: IF there are Soccer Players and Soccer Ball, then the object is “Soccer”

$R3h_4$: IF there are Soccer Players and Crowd, then the object is “Soccer”

$R3h_5$: IF there are Soccer Players, Field and Referee, then the object is “Soccer”

$R3h_6$: IF there are Soccer Players, Field and Soccer Ball, then the object is “Soccer”

$R3h_7$: IF there are Soccer Players, Field and Crowd, then the object is “Soccer”

$R3h_8$: IF there are Soccer Players, Referee and Soccer Ball, then the object is “Soccer”

$R3h_9$: IF there are Soccer Players, Referee and Crowd, then the object is “Soccer”

$R3h_{10}$: IF there are Soccer Players, Soccer Ball and Crowd, then the object is “Soccer”

$R3h_{11}$: IF there are Soccer Players, Field, Referee and Soccer Ball, then the object is “Soccer”

$R3h_{12}$: IF there are Soccer Players, Field, Referee and Crowd, then the object is "Soccer"

$R3h_{13}$: IF there are Soccer Players, Field, Soccer Ball and Crowd, then the object is "Soccer"

$R3h_{14}$: IF there are Soccer Players, Referee, Soccer Ball and Crowd, then the object is "Soccer"

$R3h_{15}$: IF there are Soccer Players, Field, Referee, Soccer Ball and Crowd, then the object is "Soccer"

4. Football Category

$R4m_1$: IF there are People and Football Clothes, then the object is "Football Players"

$R4h_1$: IF there are Football Players and Field, then the object is "Football"

$R4h_2$: IF there are Football Players and Football, then the object is "Football"

$R4h_3$: IF there are Football Players and Referee, then the object is "Football"

5. Volleyball Category

$R5m_1$: IF there are People and Volleyball Clothes, then the object is "Volleyball Players"

$R5h_1$: IF there are Volleyball Players and Net, then the object is "Volleyball"

$R5h_2$: IF there are Volleyball Players and Volleyball, then the object is "Volleyball"

$R5h_3$: IF there are Volleyball Players and Referee, then the object is "Volleyball"

6. Fishing Category

$R6m_1$: IF there are People and Fishing Rods, then the object is "Fishermen"

$R6h_1$: IF there are Fishermen and Water, then the object is "Fishing"

7. Zoo Category

$R7m_1$: IF there are Animals and Habitat, then the object is "Animal Habitat"

$R7h_1$: IF there is Animal Habitat and Open Area, then the object is "Zoo"

8. Party Category

$R8m_1$: IF there is Food and Drink, then the object is "Nutrition"

$R8h_1$: IF there is Nutrition and People, then the object is "Party"

9. Birthday Category

$R9m_1$: IF there is Cake and Candles, then the object is "Birthday Cake"

$R9h_1$: IF there is Birthday Cake and People, then the object is “Birthday”

10. Beach Category

$R10\,m_1$: IF there is Water and Sand, then the object is “Shore”

$R10\,h_1$: IF there is Shore and Open Area, then the object is “Beach”

11. Car Racing Category

$R11h_1$: IF there is Racing Road and Cars, then the object is “Car Racing”

$R11m_1$: IF there is Track and Road, then the object is “Racing Road”

12. Computer System Category

$R12\,m_1$: IF there is Keyboard, CPU, and Monitor, then the object is “Computer”

$R12\,m_2$: IF there is Keyboard, CPU, Monitor and Mouse, then the object is “Computer”

$R12\,h_1$: IF there is Computer and Printer, then the object is “Computer System”

13. Playground Category

$R13\,m_1$: IF there is Play Set and Bark, then the object is “Playing Area”

$R13\,h_1$: IF there is Playing Area and Open Area, then the object is “Playground”

14. Restaurant Category

$R14\,m_1$: IF there are Tables and Chairs, then the object is “Meal Tables”

$R14\,h_1$: IF there are Meal Tables and Cutlery, then the object is “Restaurant”

$R14\,h_2$: IF there are Meal Tables and Waiter, then the object is “Restaurant”

15. War Category

$R15\,m_1$: IF there are People and Army Uniform, then the object is “Army”

$R15\,h_1$: IF there is Army and Smoke, then the object is “War”

$R15\,h_2$: IF there is Weapon and Smoke, then the object is “War”

$R15\,h_3$: IF there are Troops and Open Area, then the object is “War”

16. Class Room Category

$R16\,m_1$: IF there are Desks and Chairs, then the object is “Study Area”

$R16\,h_1$: IF there is Study Area and Board, then the object is “Class Room”

17. Waterfall Category

$R17\,m_1$: IF there are Rocks and Cliffs, then the object is “Fall”

$R17\,h_1$: IF there is Fall and Water, then the object is “Waterfall”

18. Wrestling Category

$R18\,m_1$: IF there are Athletes and Attire, then the object is “Wrestler”

$R18\,h_1$: IF there is Wrestler and Ring, then the object is “Wrestling”

$R18\,h_2$: IF there is Wrestler, Ring and Crowd, then the object is “Wrestling”

19. Snowboarding Category

$R19\,m_1$: IF there are People and Snow Gear, then the object is “Snow Border”

$R19\,h_1$: IF there is Snow Boarder and Snow, then the object is “Snowboarding”

$R19\,h_2$: IF there is Snow Border, Snow and Boarder, then the object is “Snowboarding”

20. Cricket Category

$R20\,m_1$: IF there are People and Cricket Clothes, then the object is “Cricket Player”

$R20\,h_1$: IF there are Cricket Players and Wickets, then the object is “Cricket”

$R20\,h_2$: IF there are Cricket Players and Field, then the object is “Cricket”

$R20\,h_3$: IF there are Cricket Players and Cricket Ball, then the object is “Cricket”

3.3 Image Representation

In this research, the representation for image indexing consists of many discrete object types: low-level objects, mid-level objects, high-level objects and so on. We expect that three levels of object types will give a very good representation, nevertheless; fewer or more levels (two or four) can also be used to present some images.

Low-level objects are primitive objects within the image that are meaningful to the human users. Mid-level objects can be primitive; they can also consist of two or more low-level objects being non-primitive objects. High-level objects can also be primitive; they can also consist of two or more mid-level objects. In this research, we use the ternary object model by expanding and restructuring low-level objects, mid-level objects, and high-level objects (in the case of three levels, which could differ) into a hierarchy. This expansion of the index makes use of rules to automate the generation of a mid-level objects from low-level objects, and high-level objects from mid-level objects. At the same time, this expansion of index makes use of rules to work in the opposite direction in order to automate the generation of a low-level object from mid-level objects, and mid-level objects from high-level objects and so on. This expansion is depicted in Figure 3.1.

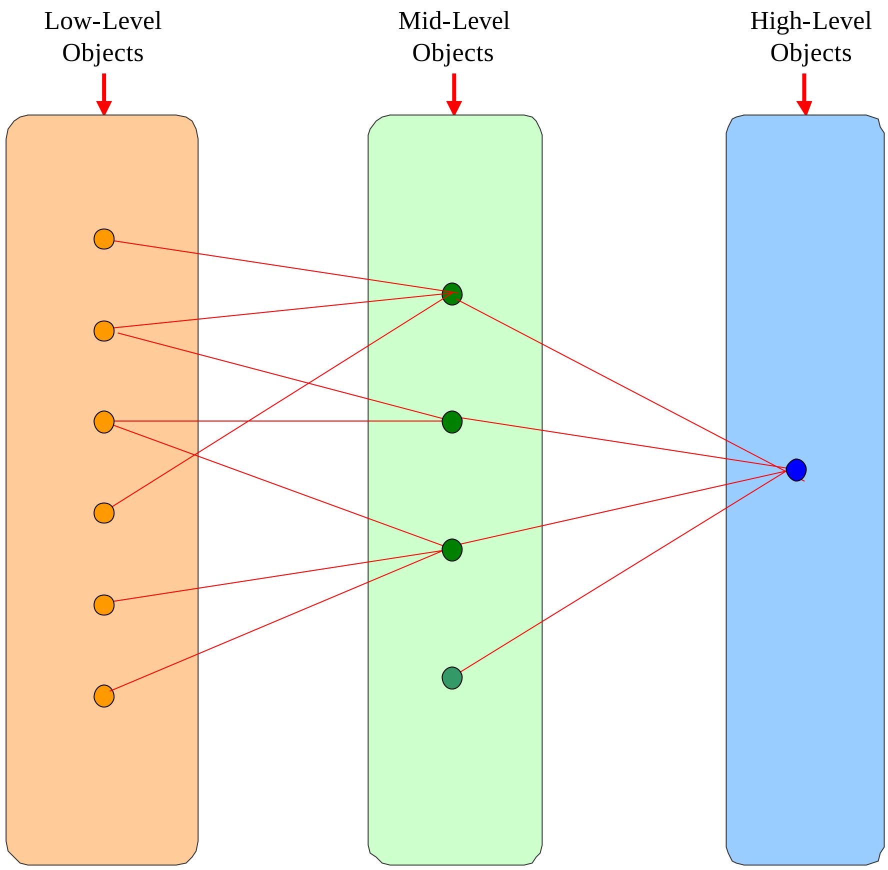


Figure 3.1 Index Expansions

Apart from the left most set, which contains all primitive objects labelled as low-level objects, the remaining sets contain mid-level objects, and high-level objects. A dot represents an object and a line to the left of the object relates that particular object to its components. From this representation, it is clear that the higher the level of the concept, the less the number of objects in the set. This is an important characteristic that will be exploited in database search.

Initially, the indexing is entered manually or extracted automatically from the image by human or computer in terms of primitive objects. Depending on the application, an indexer will have to predetermine to what extent they want to decompose an image object into primitive objects, which will be used to automatically develop higher-level objects using a knowledge-based system [SUTA99]. It is the task of the rules to perform the creation of the higher-level objects. A rule consists of an IF part, which lists all the conditions that must be satisfied, and a THEN part, which concludes the rule given that the conditions are met. Assigning lower-level objects in the IF part and higher-level objects in the THEN part creates a rule. Boolean "AND" or "OR" can be used to define the relationship among the primitive objects in the condition of the rule. By validating the rule conditions with existing objects and obtaining higher-level objects from the hypothesis of the rule, we can create a new index entry automatically.

In other words, we build a higher-level object from lower-level objects that might be directly recognizable from the image. This indexing mechanism will avoid inconsistency in human perception of the image concept when the process is performed manually. For retrieval, users can take the advantage of these high-level objects to speed up the searching time and narrow down the search space.

We apply the knowledge-based system using the previous rules listed in Section 3.2.2 of this chapter. Consider the Soccer image shown in Figure 3.2, containing the atomic objects: People, Soccer Clothes, Soccer Ball, Field, Referee, and Crowd. We illustrated below how to construct a typical rule to define a high-level object, "Soccer".

Figure 3.2 Soccer Image

This image falls in the Soccer category because it satisfies all rules creating the Soccer object. Below is an explanation of how this image falls in the soccer category, by using Rule $R3m_1$:

$R3m_1$: IF there are People and Soccer Clothes, then the object is “Soccer Players”

The image contains People and Soccer Clothes, the object Soccer Players is created using Rule $R3m_1$. In turn, we can treat the object “Soccer Players” as a mid-level object, and then use it as a condition for a higher-level object description as in the following examples. By using the rule “ $R3m_1$ ” and one of the following rules ($R3h_1$, $R3h_2$, $R3h_3$ and $R3h_4$), we can build the object of Soccer.

1. $R3h_1$: IF there are Soccer Players and Field, then the object is "Soccer"

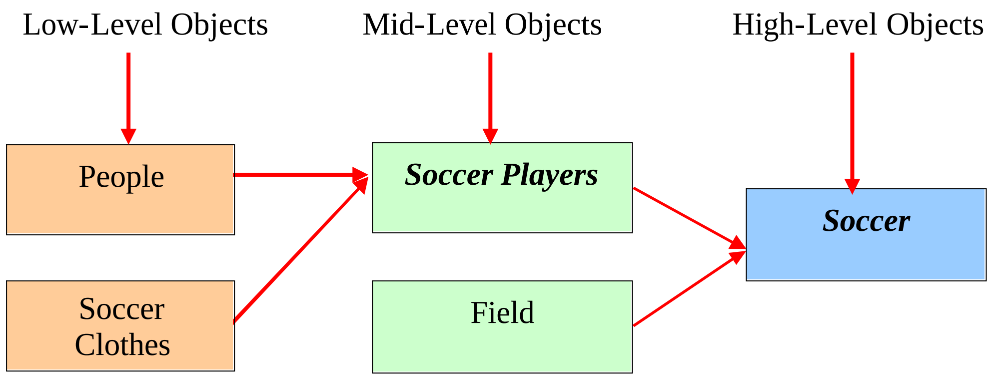


Figure 3.3 Rule Hierarchy using Essential Objects of Soccer Category (A)

2. $R3h_2$: IF there are Soccer Players and Referee, then the object is "Soccer"

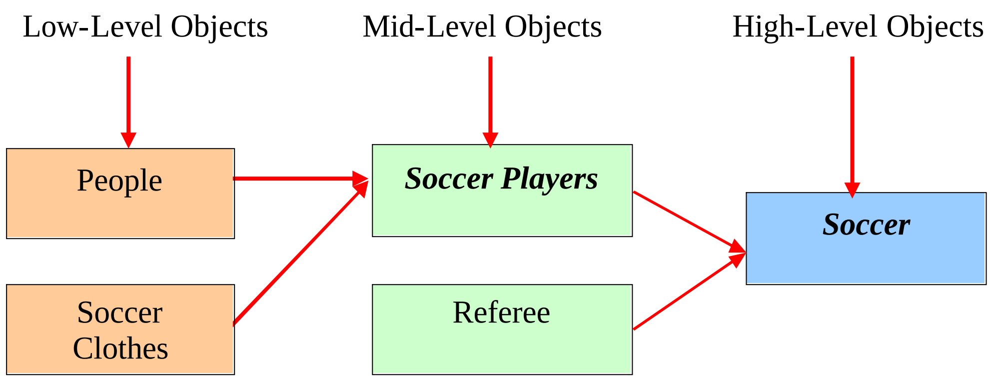


Figure 3.4 Rule Hierarchy using Essential Objects of Soccer Category (B)

3. $R3h_3$: IF there are Soccer Players and Soccer Ball, then the object is "Soccer"

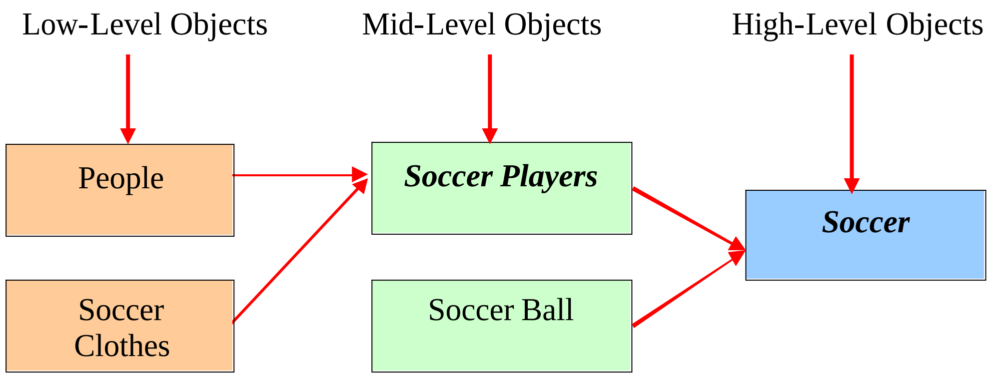


Figure 3.5 Rule Hierarchy using Essential Objects of Soccer Category (C)

4. $R3h_4$: IF there are Soccer Players and Crowd, then the object is "Soccer"

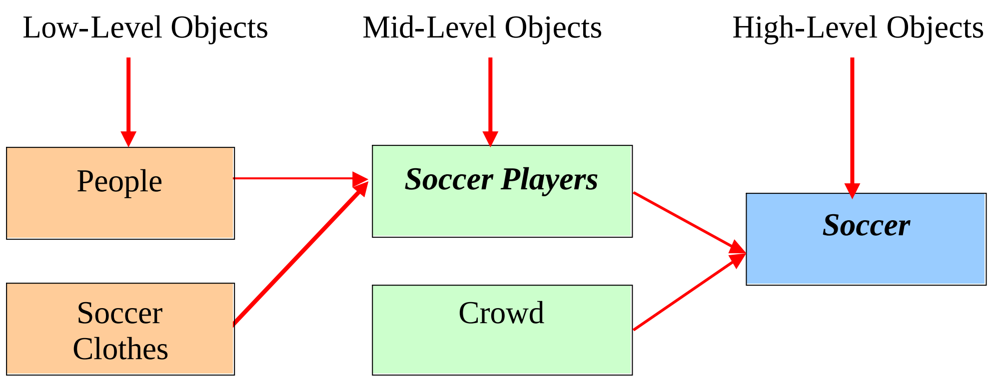


Figure 3.6 Rule Hierarchy using Essential Objects of Soccer Category (D)

Satisfying one of these rules is enough to make this image fall under the Soccer category. This image satisfied several of the rules giving it the characteristics to fall under the "Soccer" Category.

To build the rule hierarchy, we use some of the essential objects required for the category as shown in one of Figure 3.3, Figure 3.4, Figure 3.5, and Figure 3.6. The created objects are always appearing in the ***italic*** style font (Soccer Players and Soccer).

In this way we structure the index representations into a hierarchy. Therefore, several atomic objects will be able to define mid-level object and several mid-level objects will be able to define higher-level objects and so on. The benefit of this method is the reusability of the rules. Once a rule that defines an object is created, it can be used to define several other higher-level objects [TAM01], [HWAR00].

The remaining objects (if there are any) of the image that are not used to create the mid-level objects can be used to create a high-level object, we will explain this case in detail later on in this chapter. Figure 3.7 presents an example of rule hierarchy using all objects within the image shown in Figure 3.2, also satisfying rules $R3m_1$ and $R3h_{15}$, in creating the objects “Soccer Player” and “Soccer”.

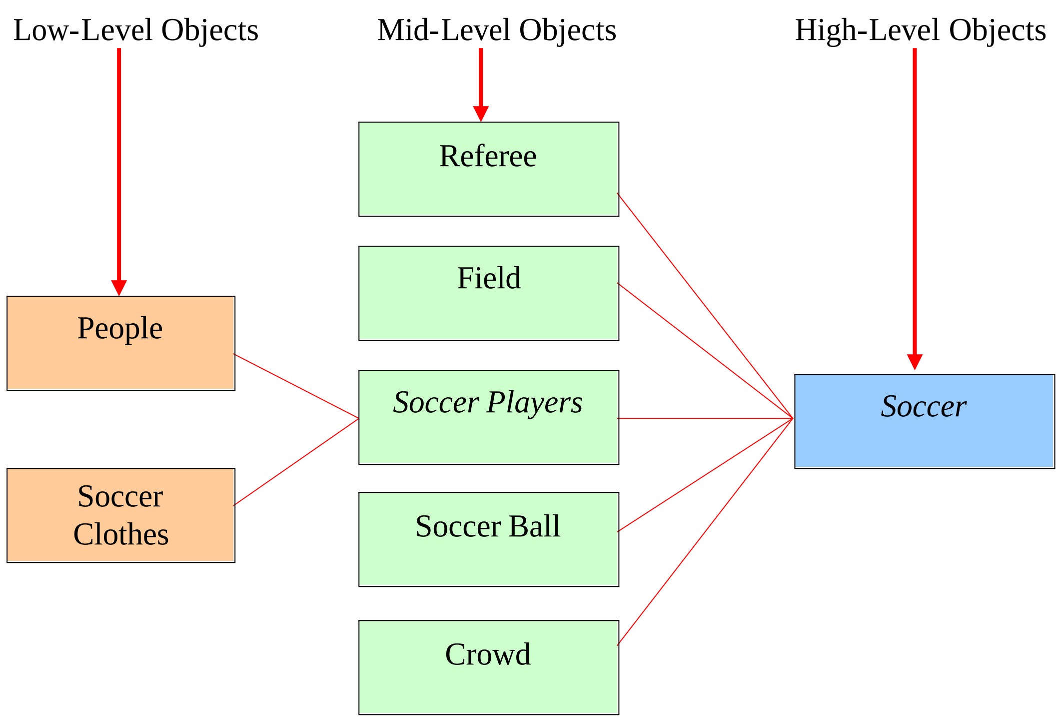


Figure 3.7 Rule Hierarchy using all Objects within Soccer Image

3.4 Object's Relationships

There are three kinds of relationships between object levels: Composite Relationship (CR), Is-A Relationship (IAR) and Aggregation Relationship (AR). The uses of these relationships illustrate the idea of grouping objects together when creating mid-level objects, high-level objects, etc. Some primitive objects could group with the essential objects to create higher-level objects; the created objects could group with some other objects to create higher-level objects. The remaining objects may be grouped together to create the highest level object. These relationships are explained below.

3.4.1 Composite Relationship

A composite object is an object that can be broken down into component objects (all are essential). The Composite Relationship (CR) is shown in figure 3.8.

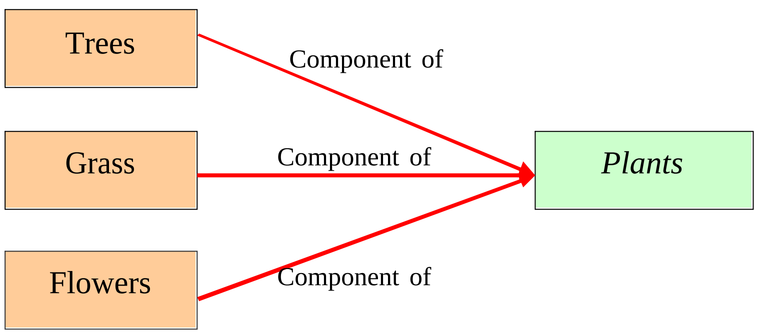


Figure 3.8 Composite Relationship

3.4.2 IS-A Relationship

We say A Is-A B if object set B is a generalization of object set A, or equivalently, A is a special kind of B. (e.g. Apple is-a Fruit). IS-A Relationship (IAR) is shown in Figure 3.9.

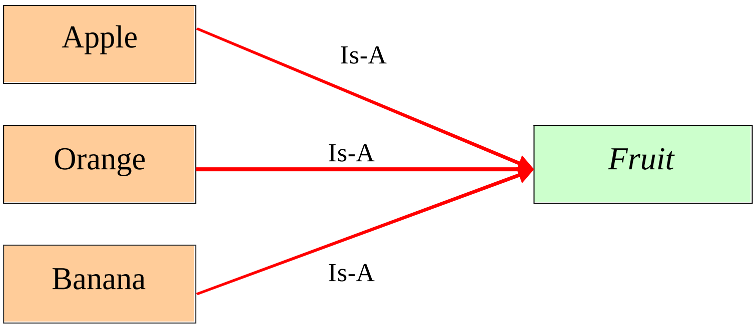


Figure 3.9 Is-A Relationship

3.4.3 Aggregation (Member-Of) Relationship

An object can contain many components (an object is part of a greater whole), some of these components are essential and some are not (Weapon and Smoke are essential but Trees are not essential in the object "War"); the Aggregation Relationship (AR) is shown in Figure 3.10.

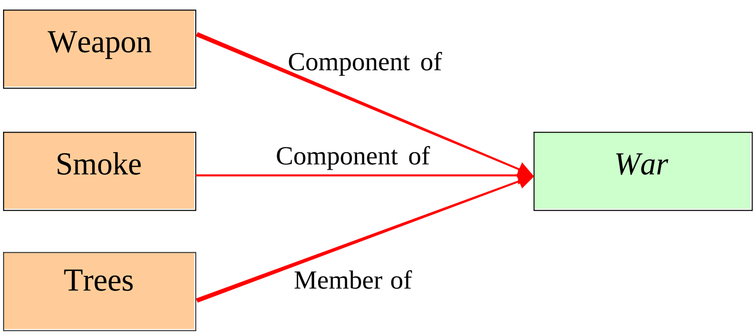


Figure 3.10 Aggregations (Member-Of) Relationship

3.5 Object Levels

As we mentioned in Section 3.2 of this Chapter, the index is entered manually or extracted automatically from the image by human or computer in term of atomic objects. Atomic objects are defined as the smallest entity that cannot be decomposed further into components. For example if a table is defined as an atomic component, then in the future we would not be able to recognize a table leg or a table top from the database. Therefore, depending on the application, an indexer will have to predetermine to what extent to decompose an image object into atomic objects. From these atomic objects, the machine will develop higher-level objects using its knowledge-based system [SUTA99].

Suppose that we have an image containing the atomic objects: Trees, Grass, Flowers, Open Area, Pathway and Cloud as shown in Figure 3.11.

Figure 3.11 Garden Image

We illustrate below how to construct a typical rule to define a mid-level object "Plants" and high-level object "Garden", as shown in Figure 3.12.

- Using rule $R1m_4$, we can create the mid-level object "Plants" from the essential objects of the low-level objects found in the given image.

$R1m_4$: IF there are Trees, Grass, and Flowers, then the object is "Plants"

- The objects not used to create the mid-level object moved from the low-level object to the mid-level object.
- Using rule $R1h_1$, we create the high-level object "Garden" from the essential objects found in image of the mid-level objects.

$R1h_1$: IF there are Plants and Open Area, then the object is "Garden"

- The objects that are not used to create the highest-level objects are moved down and categorised as a mid-level object since they can not be used to create any further objects, they are dropped back and are used a member-of "Garden" object. The "Garden" object will remain alone as a high-level object.
- The numbers beside the arrows indicate the number of steps taken when using the object found on the left side of the arrows, the solid arrows represent that of the essential objects and the dashed arrows for the non-essential objects.

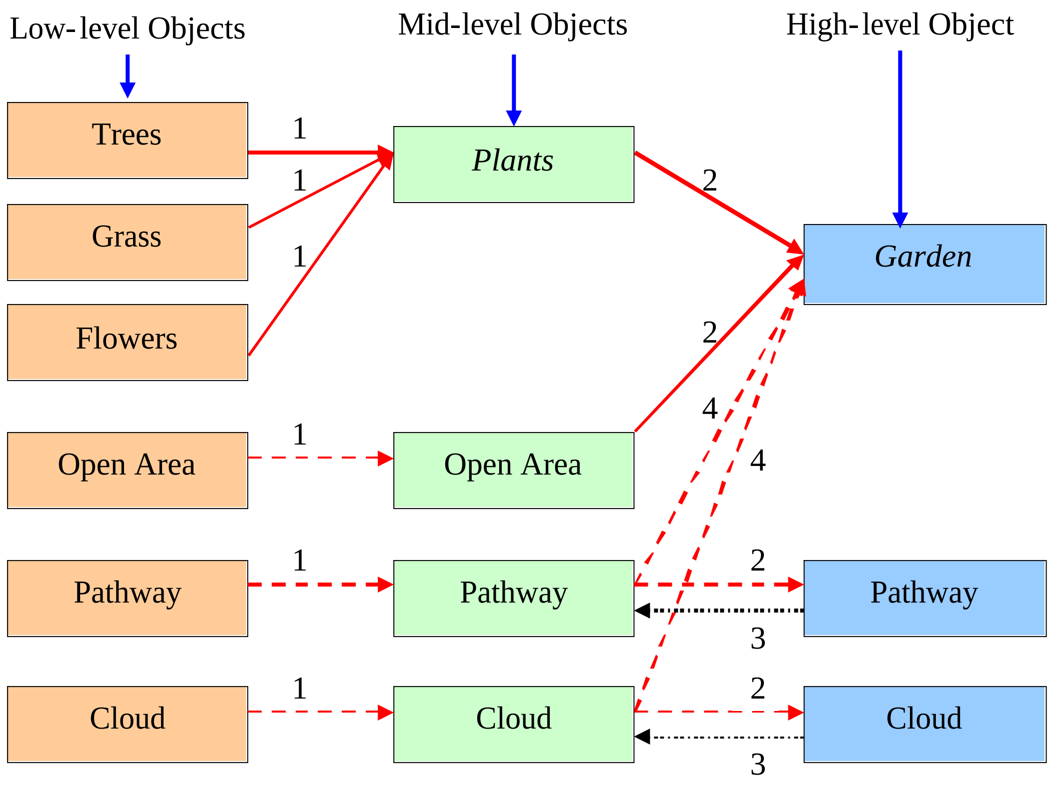


Figure 3.12 Steps of Constructing Three Levels of Objects

The final Figure of Figure 3.12 could be reorganizing as in Figure 3.13.

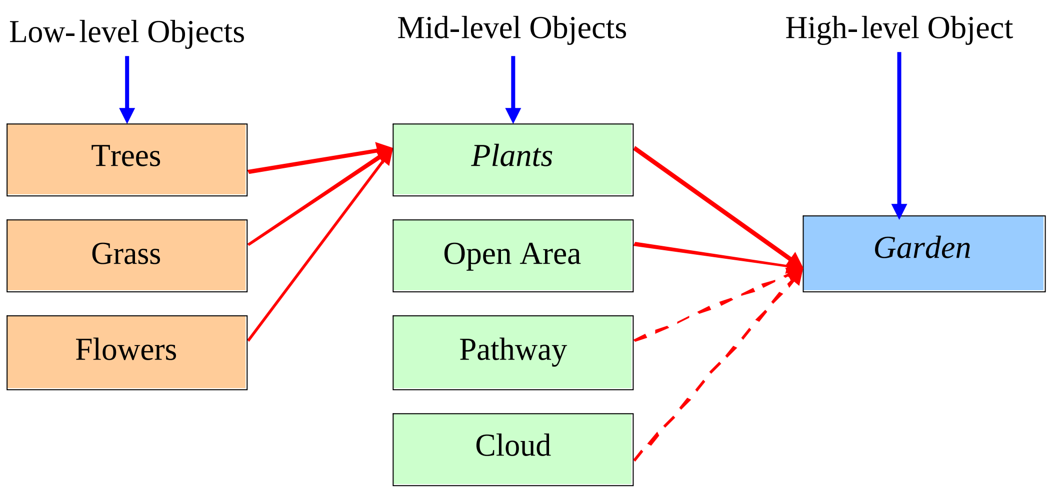


Figure 3.13 Three Levels of Objects

Another way of how to construct a typical rule to define a mid-level object "Plants" and high-level object "Garden" is illustrated below.

3.5.1 Low-Level Objects

Several primitive objects are considered as a low-level objects if they can be used together to build a mid-level object. In our case, the image contains Trees, Grass, and Flowers, which can be used to create an object of Plants. It satisfies the rule $R1m_4$ of building the Plants object. So Trees, Grass, and Flowers are of the Low-level objects as shown in Figure 3.14.

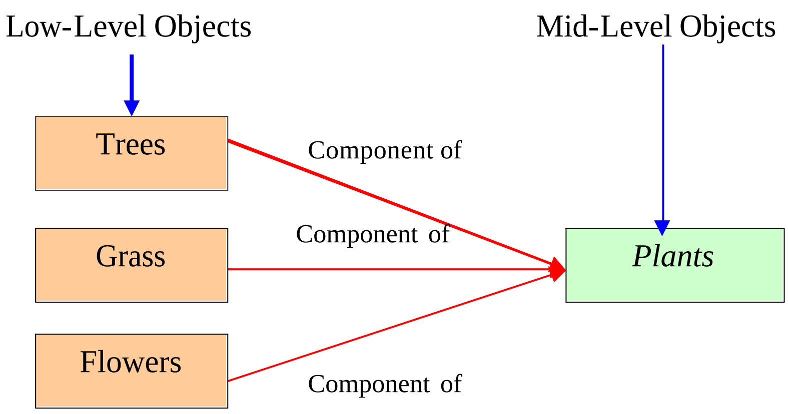


Figure 3.14 Low-level Objects

3.5.2 Mid-Level Objects

In turn, we can treat the created object "Plants" as a mid-level object, and then use it as a condition with the other object "Open Area" in this image as mid-level objects for a high-level object "Garden". Rule $R1h_1$ is used to create the Garden object from its essential components (Plants and Open Area), as shown in Figure 3.15.

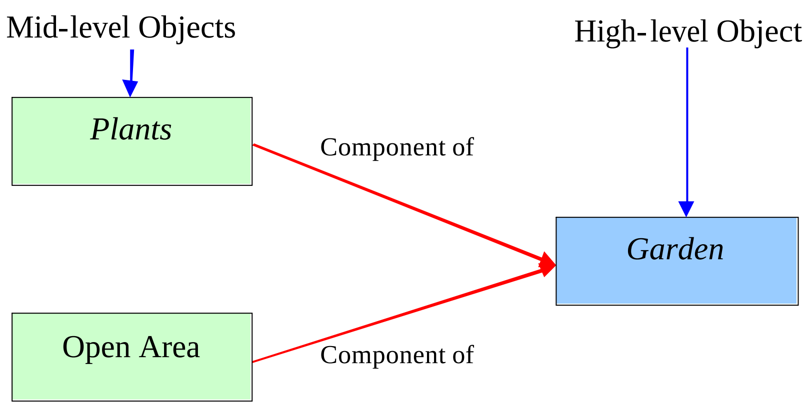


Figure 3.15 Essential Mid-level Objects

The remaining objects, which are not used to build any level of objects, are grouped with the essential objects to build the highest level object in an image. So Pathway and Cloud are not used to build any other object; we include these objects together with the essential objects (Plants and Open Area) to build the same high-level object "Garden" as shown in Figure 3.16.

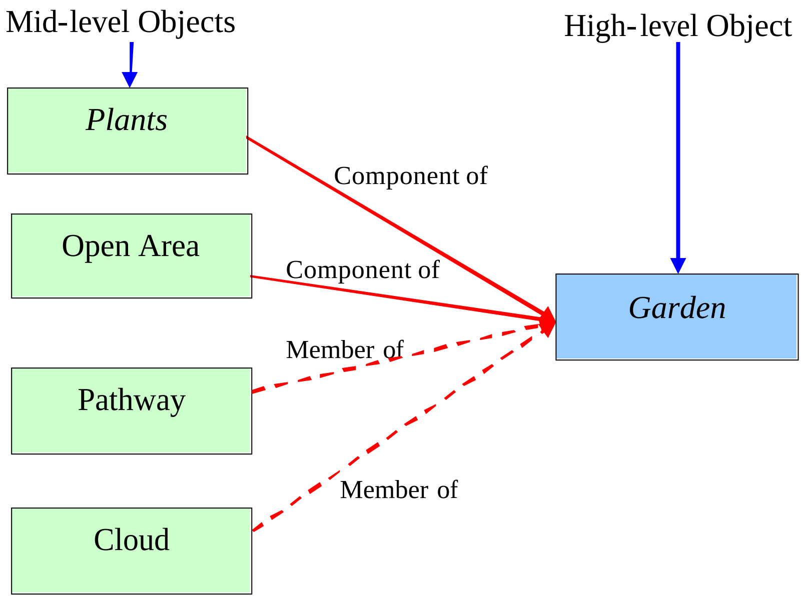


Figure 3.16 Essential and Non-Essential Mid-level Objects

3.5.3 High-Level Objects

The created object “Garden”, which was created from the mid-level objects Plants, Open Area, Pathway and Cloud, is considered a high-level object.

The three levels of objects we explained in this section are shown together in Figure 3.13 in this Section.

As mentioned in an earlier section, the advantage of this method is the reusability of the rules, where a rule that defines an object is created, can also be used to define several other higher-level objects.

We can build Higher-level objects using the same technique used to create mid and high-levels of objects, however components must be objects of a high-level.

3.6 Building the Object Levels

We develop techniques for the categorization of image components, and the assignment of different attributes (weights, supporting factors, density, etc) to these components on a proportional scale based on their relative importance within the image [JORG98].

Suppose that we have an image shown in Figure 3.17, containing the atomic objects: Female, Wedding Clothes, Groom, People, Flowers and Candles.

Figure 3.17 Wedding Image

In an example, we explain below how we can build the possible different levels of objects using the given objects within the image.

3.6.1 Building Mid-level Objects

The Low-level objects must satisfy the rule of creating a mid-level object in order for us to be able to use them to build the particular mid-level object, for example, the essential objects such as Female and Wedding Clothes are considered components of the object, Bride and hence can create the Bride object here, using rule ($R2m_1$) as illustrated in Figure 3.18.

$R2m_1$: IF there is Female and Wedding Clothes, then the object is "Bride"

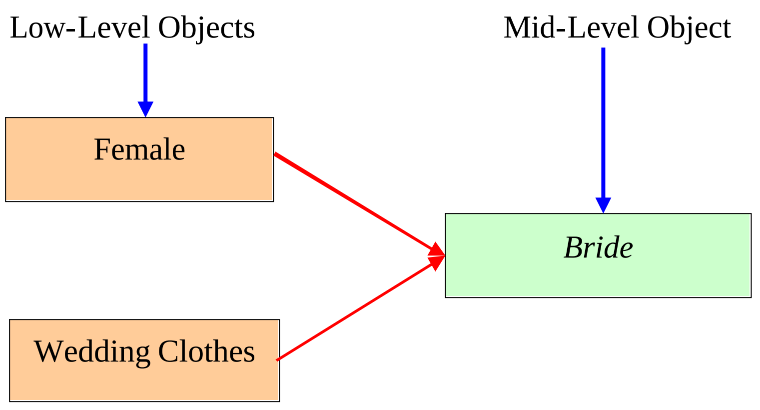


Figure 3.18 Building Mid-Level Object

We have to include all low-level objects that could be used to build the mid-level object (the essentials could be components of the mid-level object). The minimum amount of objects required to create a higher-level object are always needed (the essential objects) with the option of adding other components that are not always required.

3.6.2 Building High-level Objects

Using the same technique when building mid-level object, the created mid-level object “Bride” and some of the remaining objects of the image that are not used to create the mid-level object Bride (Groom and People) can be used to create a high-level object “Wedding” using rule $R2h_3$, as illustrated in Figure 3.19.

$R2h_3$: IF there is Bride, Groom and People, then the object is “Wedding”

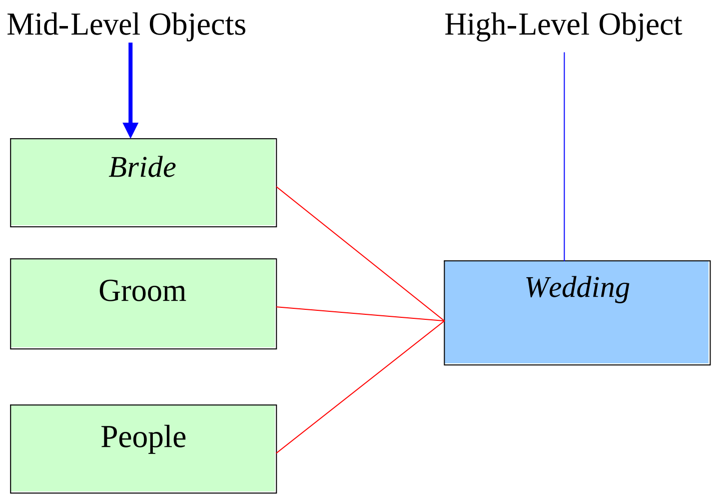


Figure 3.19 Building High-level Object

Figure 3.20 illustrates the same high-level object “Wedding” built in Figure 3.19, in addition to the use of the non-essential object (Flowers and Candles), since we cannot use them to build any further object.

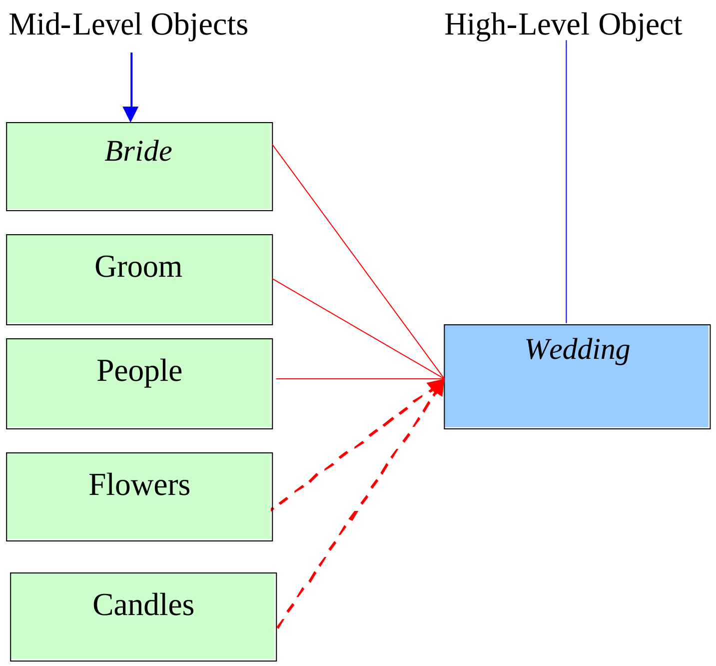


Figure 3.20 Building High-level Object using all Mid-Level Objects within Image

The created high-level object “Wedding” and any other objects (if there is any) that is not available in the given image; could be used to create a higher-level object, if it satisfy the specific rule using the same technique of building mid-level and high-level objects.

Upon the execution of these rules, which are used in creating objects of different levels, we will generate new objects that represent mid-level or high-level objects. These objects will be stored in different tables corresponding to the level of abstraction or the stage of creation of the index.

To evaluate the rule, the inference engine requires that all of the conditions be satisfied. It is possible that more than one rule will apply to an image. The final shape contains all levels we build is shown in the figure 3.21.

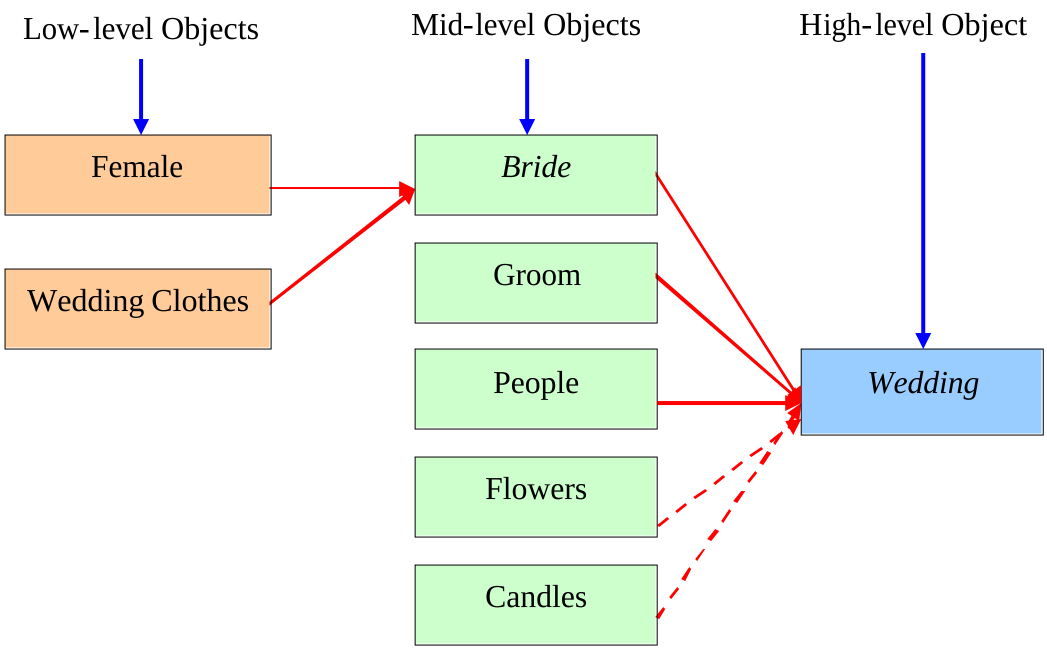


Figure 3.21 All Levels of Objects Contained within Wedding Image

As shown from building different levels of objects, we build a higher-level index from lower-level objects that might be directly recognizable from the image. This indexing mechanism will avoid inconsistency in human perception of the image concept when the process is performed manually. For retrieval, users can take the advantage of these high-level objects to speed up the searching time and narrow down the search space.

3.7 Image Organization and Categorization

The images in our database could be put into one category, or they could be put into many different categories, For example, if an image contained Trees, Grass, and Open Area, this image can put into one category "Garden" because it satisfy the Garden image rule only, but if it contained Trees, Grass, Flowers, Open Area, Bride, and People, this image can put into two categories "Garden" and "Wedding". It satisfies both the Garden and the Wedding rules simultaneously.

The following example demonstrates the way to manipulate the image organization: Suppose that we have two images in our database (Figure 3.22), with the following objects as shown in Table 3.1

Image1

Image2

Figure 3.22 Garden and Wedding Images

Table 3.1 Image descriptions

Image1	Image2
Trees	Trees
Grass	Flowers
Flowers	People
Open Area	Female
Cloud	Wedding Clothes
	Open Area

Image Falling in One Category

Suppose that we have created a knowledge-based system using the following rules (Section 3.2.2 in this Chapter) and the objects contained in image1:

$R1m_4$: IF there are Trees, Grass, and Flowers, then the object is “Plants”

$R1h_1$: IF there are Plants and Open Area, then the object is “Garden”

Image1 falls under only one category, the “Garden” category; since it satisfies the Garden rule creation as shown in $R1m_4$ and $R1h_1$. As creation of different levels of object, we included the remaining objects of the image “Cloud”, as mid-level object when representing the relationship between objects as shown in Figure 3.23.

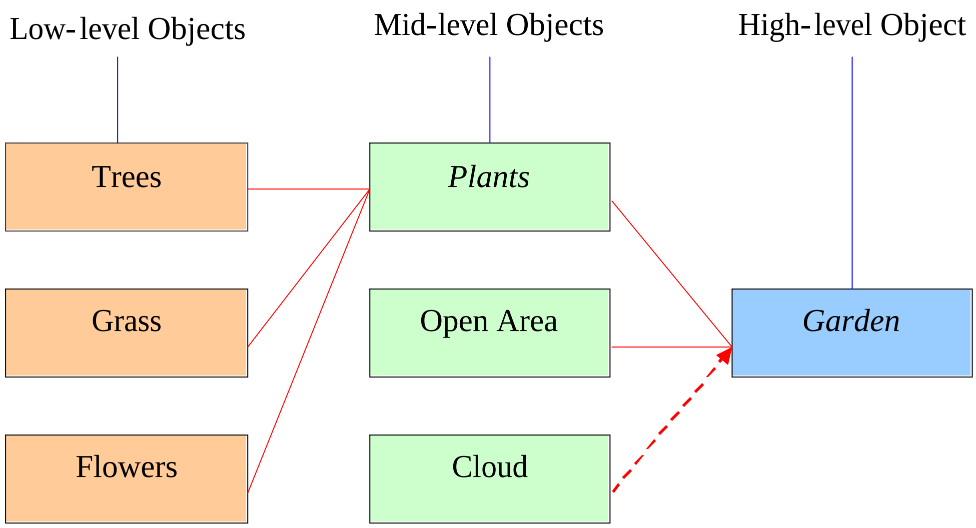


Figure 3.23 Image Falling in One Category “Garden” Category

Image Falling in Two Categories

By the same way, suppose that we have created a knowledge-based system using the following rules and the objects contained in image2:

$R1m_2$: IF there are Trees, and Flowers, then the object is “Plants”

$R1h_1$: IF there are Plants and Open Area, then the object is “Garden”

This means that image2 falls under the “Garden” category: It satisfies the Garden rule creation as in rules $R1m_2$ and $R1h_1$, it contained the essential objects of the Garden object, as shown in Figure 3.24.

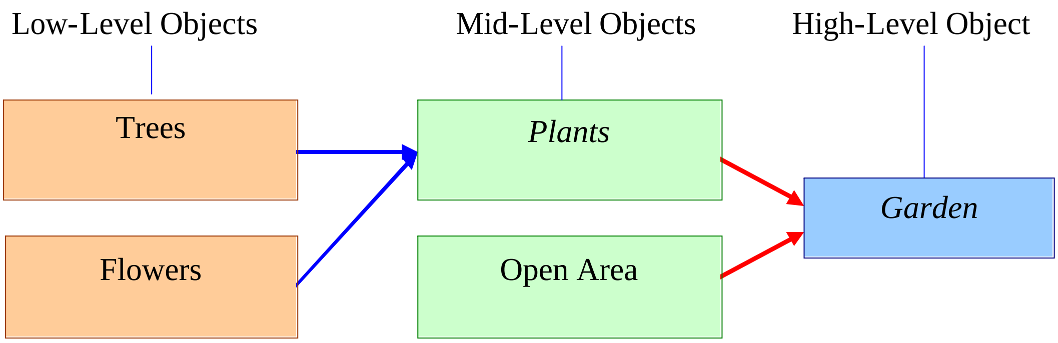


Figure 3.24 Image Falling in Two Categories (1/2 Garden)

For the same image (image2), suppose that we have created a knowledge-based system using the following rules and the objects contained in image2:

$R2m_1$: IF there is Female and Wedding Clothes, then the object is "Bride"

$R2h_1$: IF there is Bride and People, then the object is "Wedding"

This means that image2 falls in the "Wedding" category also, it satisfies the wedding rule creation as in rules $R2m_1$ and $R2h_1$, it contained all objects needs to create the Wedding, as shown in Figure 3.25.

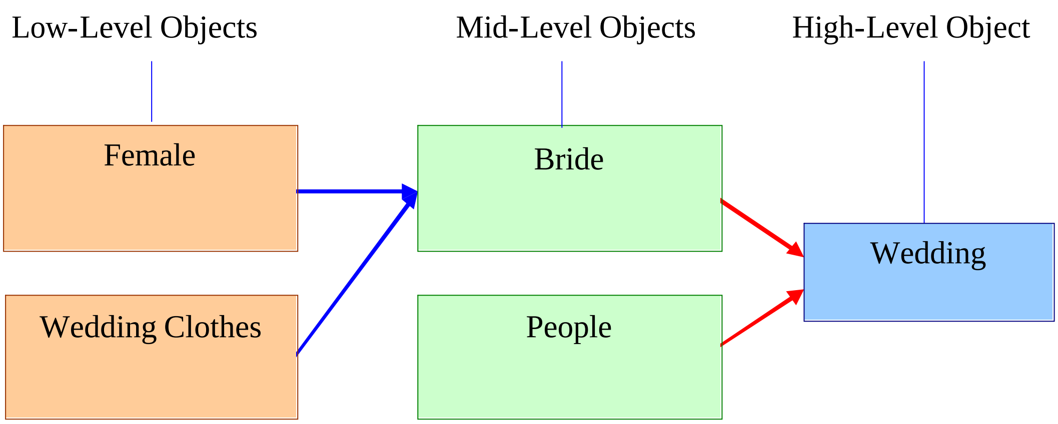


Figure 3.25 Image Falling in Two Categories (2/2 Wedding)

Rules $R1m_2$, $R1h_1$, $R2m_1$ and $R2h_1$ illustrated that image2 falls in two categories, the Garden category and the Wedding category. These rules briefly illustrate how we construct the rule hierarchies. We can create mid-level objects "Plants" and "Bride", using some of the image components as in $R1m_2$ and $R2m_1$. The created mid-level object, Plants, and Open Area can be used to create the high level object "Garden" as in $R1h_1$. Plants and Open Area and the remaining objects of the image that are not used to create the mid-level object (Bride (Female and Wedding Clothes), People), can be used to create the same high-level object "Garden" as shown in Figure 3.26.

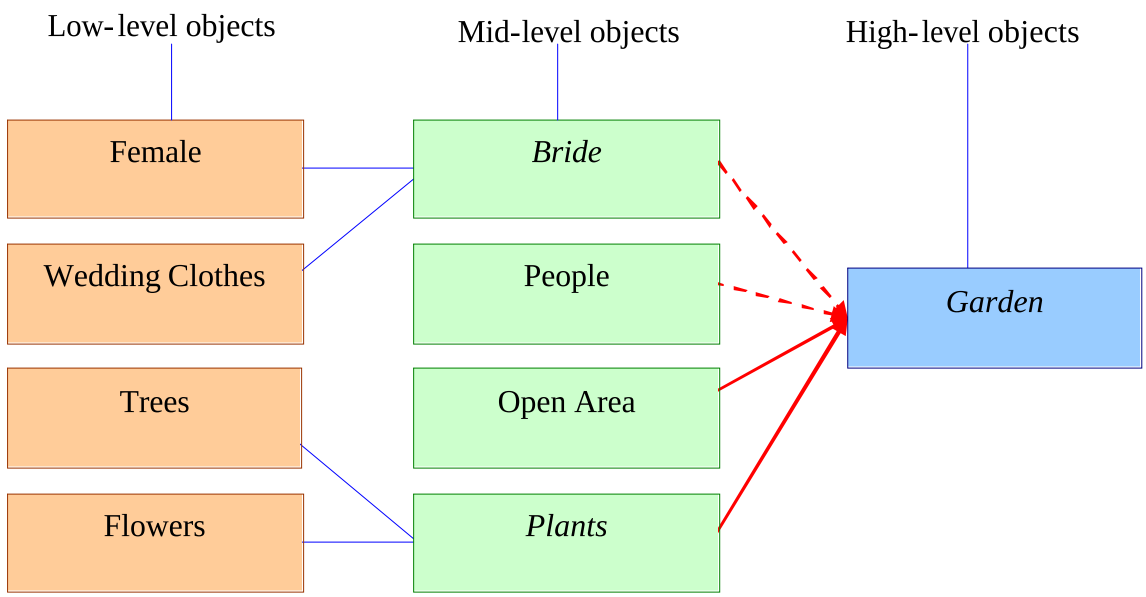


Figure 3.26 Image Falling in Two Categories using All Objects within Image

By the same way, the created mid-level objects, Bride, and People can be used to create the high level object "Wedding" as in $R2h_1$. Bride and People and the remaining objects of the image that are not used to create the mid-level object (Open Area, Plants (Trees and Flowers)), can be used to create the same high-level object "Wedding".

To build the highest level object in the image, we have to use all objects inside the image even if some of them are not essential to be components of the created object. For this, all objects inside image2 can build the "Garden" objects, at the meantime the same objects inside image2 can build the object "Wedding", since some of them can build the Garden and the Wedding objects.

Image Falling in Three Categories

Another example, an image falling under more than two categories is explained below. Suppose that we have an image in our database with the following primitive objects: Trees, Flowers, Open Area, People, Female, Wedding Clothes, Playing Set, and Bark.

This image can fall under the Garden category using the following rules:

$R1m_2$: IF there are Trees and Flowers, then the object is "Plants"

$R1h_1$: IF there are Plants and Open Area, then the object is "Garden"

Figure 3.27 shows the relationship between the created objects (Plants and Garden) and its essential components only.

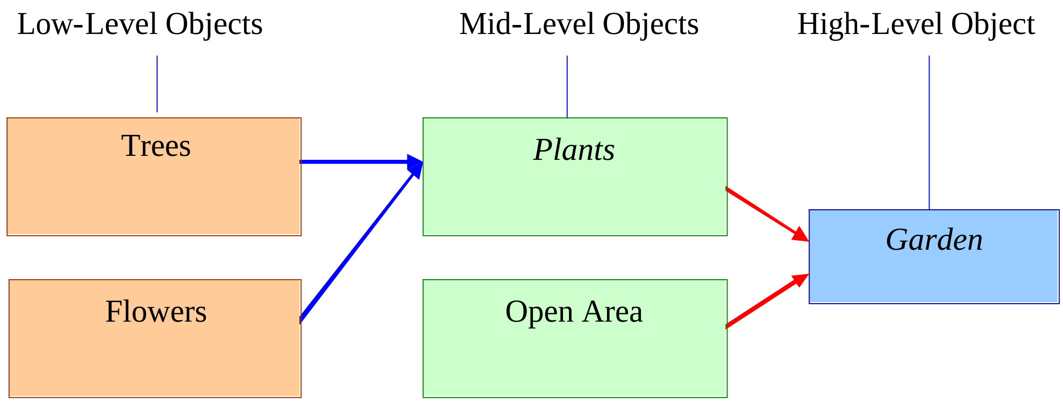


Figure 3.27 Image Falling in Three Categories (1/3 Garden)

In the same way, when using the following rules, the same image falls under the Wedding category as shown in Figure 3.28.

$R2m_1$: IF there is Female and Wedding Clothes, then the object is "Bride"

$R2h_1$: IF there is Bride and People, then the object is "Wedding"

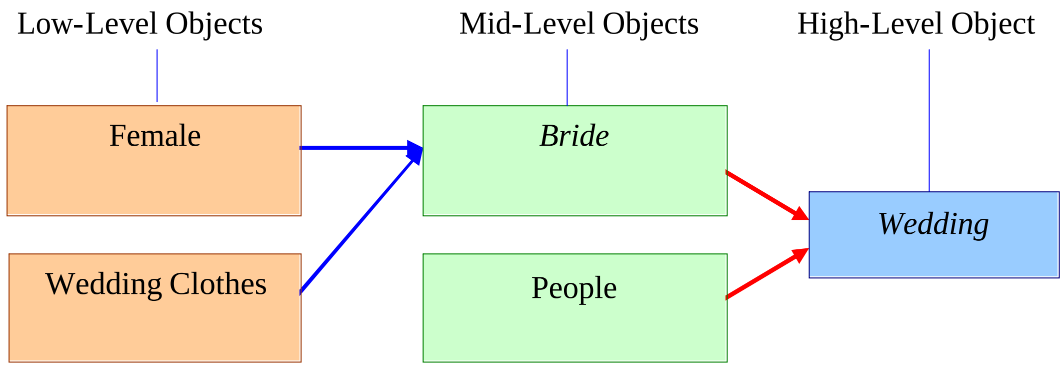


Figure 3.28 Image Falling in Three Categories (2/3 Wedding)

Using the following Rules, the same image falls under the Playground category as well as shown in Figure 3.29.

$R13m_1$: IF there is Playing Set and Bark, then the object is "Playing Area"

$R13h_1$: IF there is Playing Area and Open Area, then the object is "Playground"

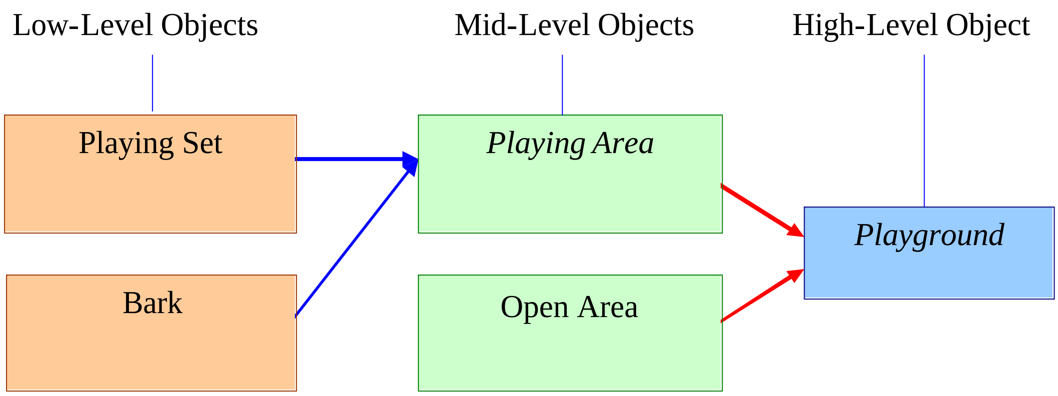


Figure 3.29 Image Falling in Three Categories (3/3 Playground)

It is obvious from the previous rules that one image could fall under many categories. When using all objects to build the different levels, Figure 3.30 shows the image falls under the Garden category. This could be applied for the Wedding and Playground categories following the same technique.

This image can be stored once, and its information will be in three different categories, which will save a lot of space since the images need a lot of space for storage. This idea leads us to store our images in one folder, not in numerous categories, not only to save more spaces but also for easier access.

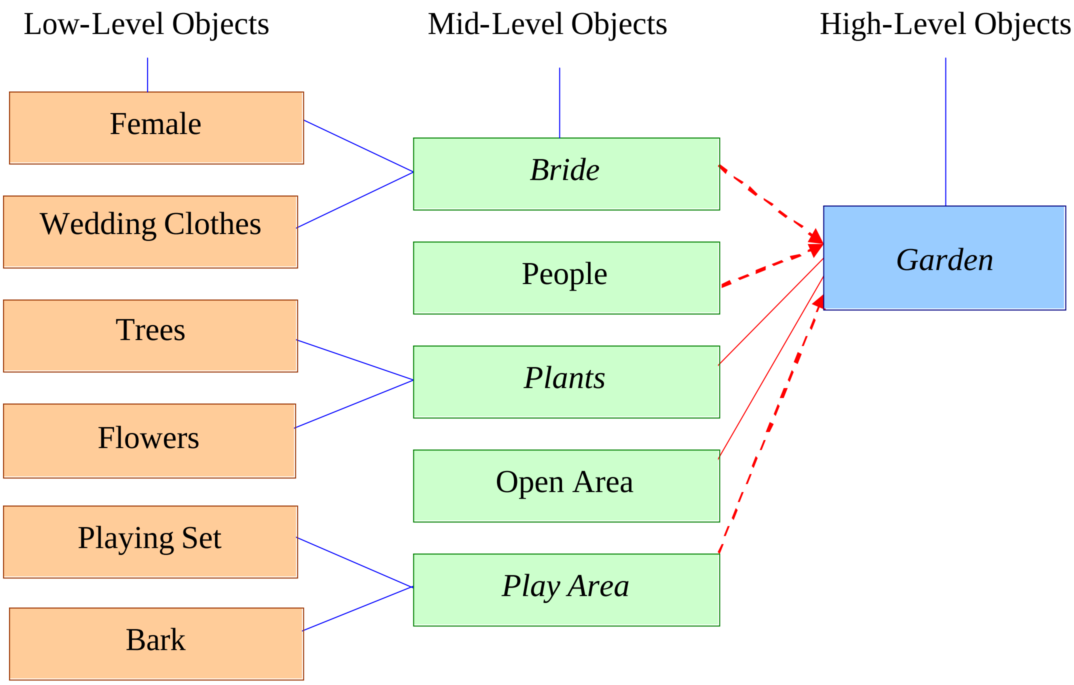


Figure 3.30 Image Falling in Three Categories using All Objects within Image

3.8 Summary

In this chapter, we present the image collection and the selection rules, the current systems used to categorize images in collections and where images tend to fall under multiple categories in the real world. We show that the images need to be stored in such a way that multiple resolution images can be efficiently retrieved.

We explained the developments of the text-based query system, which is based on the Ternary Object Model database.

We showed that our collected images contain multi-objects, and the rules control the image categories. We also presented the image descriptions and the relationship between objects contained in images.

The image representation has taken a place, where image indexing consists of many discrete object types, we show these object types in a figure furthermore, the type levels of these objects, the primitive and non-primitive objects and the relationship between them are explained as well, where we expect that three objects types (Low-level object, Mid-level object, and High-level object) will give a good representation.

We determined the different levels of objects, the understanding of every level, and the way of creating the higher-level objects from its components of the Lower-level objects are explained in this chapter. The relationship between objects, the type of relationship, and the different objects attributes is identified.

We explained in complete examples the way of building the different object levels and their relationships. We show in these examples how some images can fall in more than one category, where every category must contain its essential components and may be more of non-essential.

Chapter 4

Measures of the Reliability of Objects

4.1 Introduction

In order to gain satisfactory results for the objects' attributes values, which will be stored in the database and will be the only data used to retrieve the desired images measures using Mathematical Equations (MEs) have been created which depend on the object's significance in images of a certain category.

As explained in Section 3.3 of Chapter 3, objects in images can be primitive or non-primitive, the primitive objects are defined as the smallest entity that cannot be decomposed further into components, and the non-primitive objects usually contain many components.

Later on in this chapter, the creation of the Mathematical Equations and how they work is shown in order to compute the different values of objects' attributes by using examples with explanations. We also explain the objects' attributes shown in Figure 4.1 in more detail.

The created Mathematical Equations are used to compute the objects attributes' values (weights, supporting factors, primitive availabilities, values, rule significant factors, non-primitive availabilities, the weight in image, and the primitive and non-primitive densities) of different levels of objects. The computed values of all attributes and other non-computed attribute data are stored in different tables. Once we input all of the object attributes' values and other data, the search engine is ready to perform a search on the database. In a particular image with several categorisations, each object will have different values for each attribute depending on the categorisation.

4.2 Object's Attributes

We associate with every object many attributes, which are: number, name, image address, weight, availability, supporting factor, value, rule significant factor, weight in image, and density as shown in Figure 4.1. Each attribute is defined as follows:

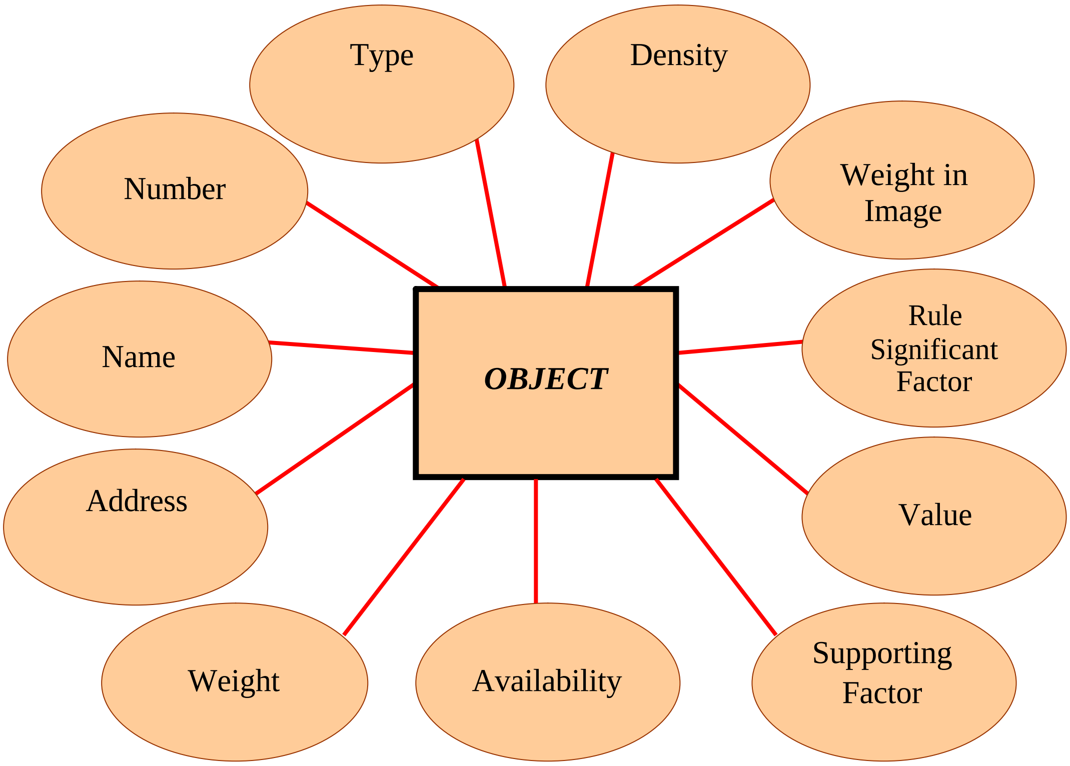


Figure 4.1 Objects' Attributes

Object's Attributes and Relative Importance

Objects' Number:

An Object's number is a unique number, the principal means of identifying objects within an object set.

*Objects' Name***:**

An object's name is a suitable description of an object.

*Objects' Address***:**

An object's address is the image address, which contains this particular object.

Objects' Weight **(**W_i**):**

An object's weight is a measure of the significance/ prominence of an object in its set in an image. These weights appear inside the closed brackets [] in figures.

Objects' Availability **(** A_i **):**

An object's availability is a measure of the expected probability of the object's appearance in an image.

Objects' Supporting Factor **(**S_i**):**

An object's supporting factor is a measure of the dependence of the appearance of a high-level object on this object. The value of the supporting factor appears beside the arrow connection between objects on figures.

Objects' Value **(V_i):**

An object's value is computed by multiplying together the object's weight, availability, and supporting factor.

Objects' Rule Significant Factor **(Rsf_j):**

An object's rule significant factor is a measure of the total appearance of all object components in a certain image divided by the total appearance of all objects which were components for the same object's name in the database.

Objects' Weight in Image **(Wii):**

An object's weight in an image is a measure of the importance of an object in the image. These weights appear inside the double closed brackets [[]] in figures. It describes the object's weight to the whole image.

Object's Density **(D_i):**

An object's density is related to the object's width and its height in the image, which is equivalent to the area occupied by the object in the image.

Objects' Type (T_i)**:**

An object's type describes the object level in an image (low-level, mid-level, high-level objects, etc).

4.3 The Measurements of Object's Attributes

As mentioned in Section 4.2 of this chapter, every object in an image has many attributes associated with it. Every one of them has been defined; many of the objects attributes' values need to be computed, which are: weight, availability, supporting factor, value, rule significant factor, weight in image, and density.

In a complete example below, we illustrate the ways to define, measure, and compute all object attribute values using the Mathematical Equations created in this chapter.

Suppose that we have an image of a Garden containing the primitive objects: Trees, Grass, Flowers, Open Area and Cloud as shown in Figure 4.2, with the image description listed in Table 4.1.

Figure 4.2 Garden Image

Table 4.1 Garden Image Descriptions

Object Name
Trees
Grass
Flowers
Open Area
Cloud

4.3.1 Object's Weight

A non-primitive object can contain many components; the weight of each component may vary from one to another. The total weight of the object's components is equal to 1. Figure 4.3 shows where the weight must appear inside the closed brackets, which is not known yet, the total weight of Plants' components (Grass, Trees, and Flowers) is shown in this figure is equal to 1.

$$W_{Trees} + W_{Grass} + W_{Flowers} = 1$$

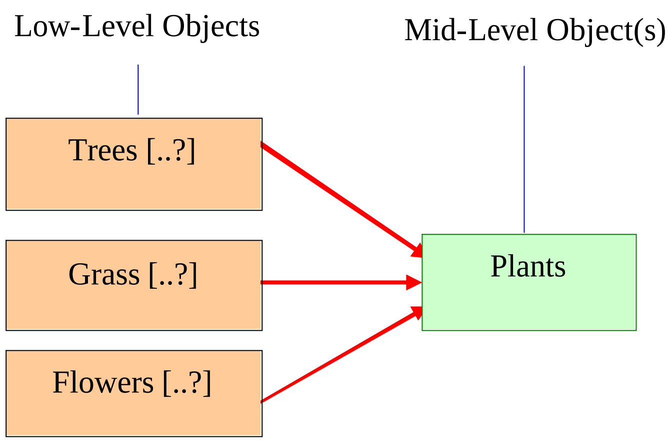


Figure 4.3 Objects without Weights

Measuring Objects' Weight

Let P_i = the number of appearances of object *i* (a low-level object belonging to a certain mid-level object), and P_t = the total number of appearances for all low-level objects that belong to the same mid-level object.

Then $P_t = (P_1 + P_2 + ... + P_n)$

$$P_t = \sum_{i=1}^{n} P_i \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (ME1)$$

And,

$$W_i = \frac{P_i}{P_t} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (ME2)$$

Uses of ME1

ME1 computes the total appearance (P_t) of all low-level objects that belong to a certain mid-level object.

Uses of ME2

ME2 computes the weight (W_i) of the low-level objects, which belong to the same mid-level object, by dividing its number of appearances (P_i) by the total number of appearances for all low-level objects that belong to the same mid-level object.

Computing the Object's Weights

To compute the weights for the low-level objects, we need to know how many times every low-level object appears as a subset of the mid-level object in our database.

Suppose that in 77 images containing Plants in the database, 'Trees' appeared 74 times as a subset of Plants; 'Grass' 56 times and 'Flowers' 57 times.

The weight for every lower-level object will be computed using a heuristic based on the frequency of appearance of each object relative to the total appearance of all objects that create that mid-level object. Below is the method of computing the weights of Plants' components in Figure 4.2 and Figure 4.3.

The total appearance of the Plants objects' components in this image = $P_t = P_{Trees} + P_{Grass} + P_{Flowers} = 74 + 56 + 57 = 187$

$$W_{Trees} = \frac{P_{Trees}}{P_{Trees} + P_{Grass} + P_{Flowers}}$$

Trees' weight can be indexed with the value:

$$W_{Trees} = \frac{74}{187} = 0.396$$

Similarly, Grass' weight can be indexed with the value:

$$W_{Grass} = \frac{56}{187} = 0.299 \ ,$$

And Flowers' weight can be indexed with the value:

$$W_{Flowers} = \frac{57}{187} = 0.305$$

As mentioned before, $W_{Trees} + W_{Grass} + W_{Flowers} = 1$ and our computing showed that the total weights of Trees, Grass, and Flowers are as follows: $W_{Trees} + W_{Grass} + W_{Flowers} = 0.396 + 0.299 + 0.305 = 1$

Figure 4.4 shows the Plants components' (Trees, Grass, and Flowers) weights which were computed appear inside the closed brackets [].

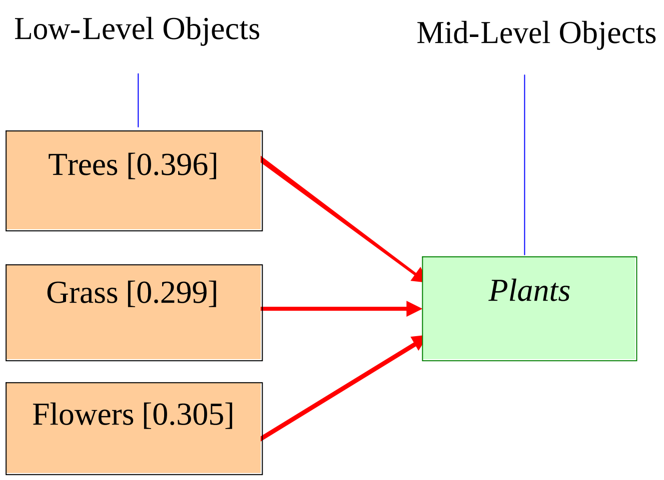


Figure 4.4 Weights of Low-Level Objects

Using ME1 and ME2, we can also compute the weights of the mid-level objects that belong to a high-level object, with the same process, this is shown below:

Suppose that, in 77 images containing Garden in the database, ‘Plants’ appeared 77 times as a subset of Garden; ‘Open Area’ 77 times and ‘Cloud’ 4 times.

The way of computing the weights of Garden components in Figure 4.5 is illustrated below.

P_t of the Garden objects’ components in this image = (77 + 77 + 4) = 158

For our example, $W_{Plants} = \dfrac{P_{Plants}}{P_{Plants} + P_{Open\ \ area} + P_{Cloud}}$

Plants’ weight can be indexed with the value:

$$W_{Plants} = \frac{77}{158} \cong 0.487 \ ,$$

Similarly, Open Area’s weight can be indexed with the value:

$$W_{Open\ Area} = \frac{77}{158} \cong 0.487$$

And Cloud’s weight can be indexed with the value:

$$W_{Cloud} = \frac{4}{158} \cong 0.025$$

Figure 4.5 shows the Garden components’ (Plants, Open Area, and Cloud) weights which were computed appear inside the closed brackets [].

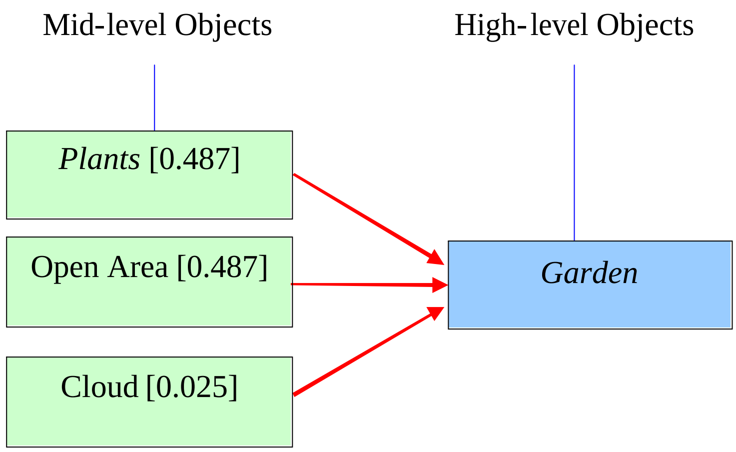


Figure 4.5 Weights of Mid-Level Objects

Figure 4.6 contained low-level and mid-level objects' weights, which were computed using ME1 and ME2. The weight of the highest-level "Garden" = 1.00 (the sum of Garden components' weights).

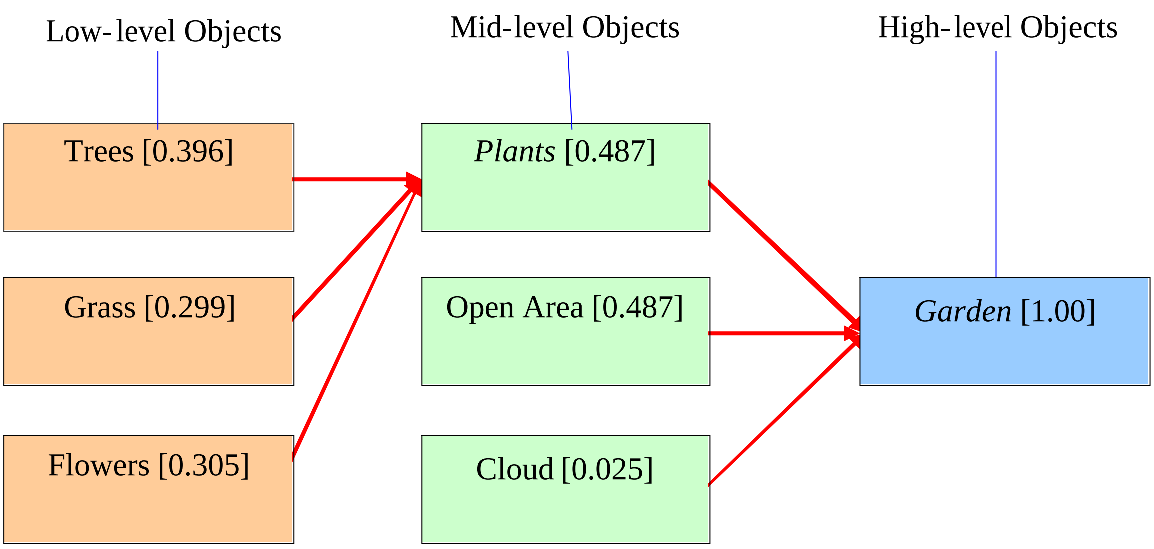


Figure 4.6 Weights of All Objects within Image

4.3.2 Object's Supporting Factor

Objects' Supporting factor (S_i) is a measure of the dependence of the appearance of a high-level object on this object. The value of the supporting factor appears beside the arrow connection between objects in figures.

Measuring an Object's Supporting Factor

Object's Supporting Factor = lower-level object's appearances (subset) divided by the mid-level object's appearances (set).

$$S_i = \frac{P_i}{P_j} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (ME3)$$

Uses of ME3

ME3 computes the supporting factor (S_i) of the low-level object (components of the mid-level object), by dividing its number of appearances (P_i) by the number of appearances of the mid-level object (P_j). Using the same method, other levels of objects' supporting factors can be computed.

Computing an Object's Supporting Factor

Computing the supporting factor for Trees, Grass, and Flowers using ME3 where in 77 images containing Plants in the database, 'Trees' appeared 74 times as a subset of Plants; 'Grass' 56 times and 'Flowers' 57 times.

Trees' supporting factor can be indexed with the value:

$$S_{Trees} = \frac{P_{Trees}}{P_{Plants}} = \frac{74}{77} = 96\ \% ,$$

Similarly, Grass' supporting factor can be indexed with the value:

$$S_{Grass} = \frac{P_{Grass}}{P_{Plants}} = \frac{56}{77} = 73\ \% ,$$

And Flowers' supporting factor can be indexed with the value:

$$S_{Flowers} = \frac{P_{Flowers}}{P_{Plants}} = \frac{57}{77} = 74\ \%$$

By adding weights and supporting factors to Figure 4.4, we can build Figure 4.7 shown below.

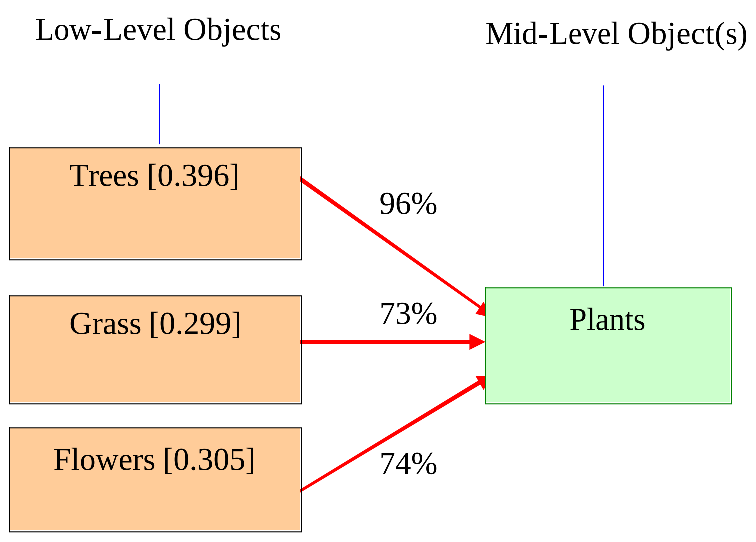


Figure 4.7 Weights and Supporting Factors of Low-Level Objects

ME3 can be used to compute the supporting factors of the mid-level objects (Plants, Open Area, and Cloud), which belong to a high-level object, “Garden”, using the same technique that is used to compute the supporting factors of the low-level objects as follows:

Plants’ supporting factor can be indexed with the value:

$$S_{Plants} = \frac{p_{Plants}}{P_{Garden}} = \frac{77}{77} = 100\ \%\ ,$$

Similarly, Open Area’s supporting factor can be indexed with the value:

$$S_{Open\ \ Area} = \frac{77}{77} = 100\ \%$$

And Cloud’s supporting factor can be indexed with the value:

$$S_{Cloud} = \frac{4}{77} \cong 5\,\%$$

Figure 4.8 contains the weights and the supporting factors of all mid-level objects, which were computed using ME1, ME2, and ME3.

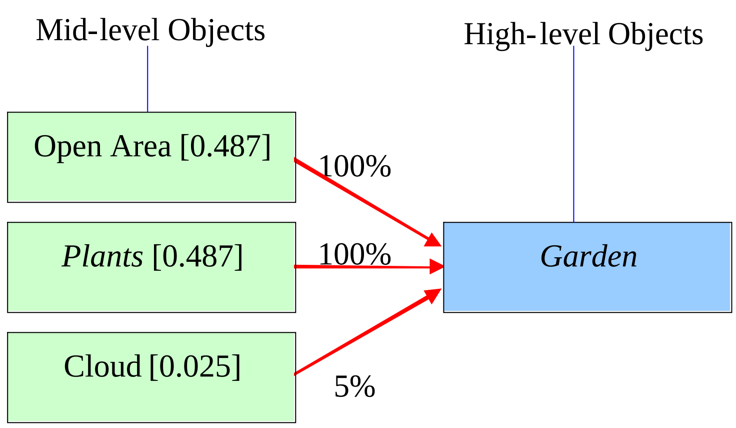


Figure 4.8 Weights and Supporting Factors of Mid-Level Objects

Figure 4.9 contains the weights and supporting factors of the low-level and the mid-level objects together.

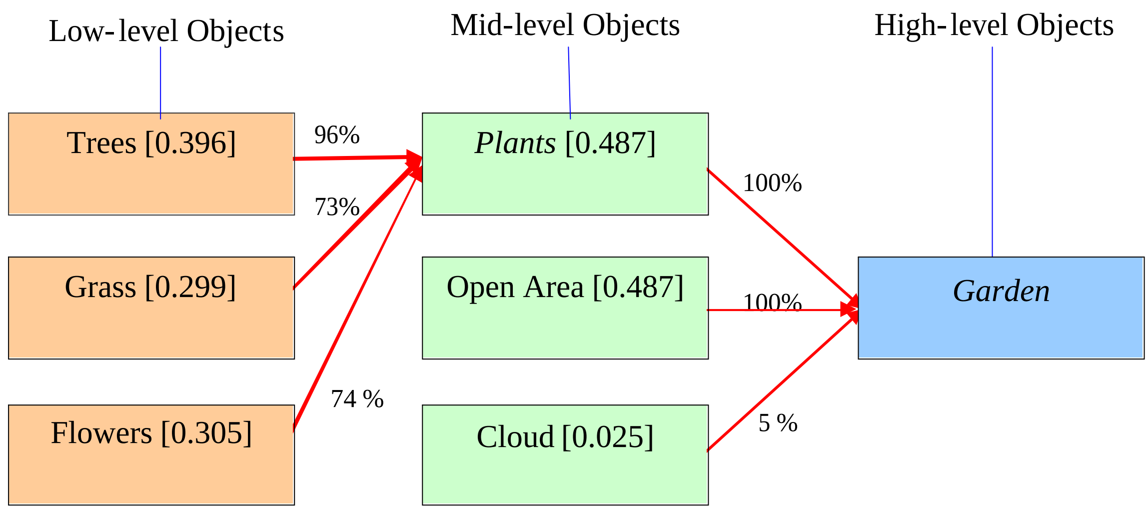


Figure 4.9 Weights and Supporting Factors of Low-level and Mid-Level Objects

4.3.3 Availability of Primitive Objects

Objects' availability (A_i) is a measure of the importance of objects inside the image comparing with the objects in image of the same category and containing all possible objects to be in this category (expected probability of the objects' appearance in an image).

Measuring an Object's Availability

The availability (A_i) of a low-level, or of any primitive object from any level is taken to be 1 because it is directly indexed primitive content (since its appearance is certain); ME4 gives the value of the availability of any primitive object.

$$A_i = 1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(ME4)$$

Uses of ME4

ME4 is used to give values to the availability of any primitive object; this value is equal to one.

Computing the Primitive Objects' Availability

Trees, Grass, and Flowers are low-level objects that are categorised as primitive. Using ME4, we can see that each object's availability will be equal to one.

Trees' availability can be indexed with the value:

$$A_{Trees} = 1$$

Similarly, Grass' availability can be indexed with the value:

$$A_{Grass} = 1,$$

And Flowers' availability can be indexed with the value:

$$A_{Flowers} = 1.$$

The availability (A_j) of a non-primitive object, is defined as the sum of the values of all its components multiplied by its rule significant factor. It has a value between 0 and 1($0 < A_j \leq 1$); this case will be discussed later.

4.3.4 Object's Values

An object's value (V_i) is computed by multiplying together the object's weight, availability, and supporting factor.

Measuring an Object's values

ME5 defines the value of each low-level object as the product of the object's weight, its availability, and its supporting factor, while ME6 computes the total value that belongs to the one mid-level object.

$$V_i = W_i \times A_i \times S_i \text{ ....}(ME5)$$

And,

$$V_t = \sum_{i=1}^{n} V_i \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (ME6)$$

Uses of ME5 & ME6

ME5 computes the value of the lower-level object by the product of the object's weight, its availability, and its supporting factor.

ME6 compute the total values of the lower-level objects that belong to the one higher-level object, these values are used to compute the availability of the higher-level object (non-primitive) by multiplying the computed values by the higher-level object's rule significant factor, explained in further detail later in this chapter.

Computing the Object's Values

Using ME5, we can compute the value of a low-level object:

$$V_{Trees} = W_{Trees} \times A_{Trees} \times S_{Trees}$$

Trees' value can be indexed with the value:

$$V_{Trees} = (0.396 \times 1.00 \times 0.96) \cong 0.380$$

Where the Trees weight = 0.396, and the availability is taken to be 1 because it is directly indexed primitive content, and the supporting factor = 0.96 (Computed in Section 4.3.2 of this chapter).

Similarly, Grass' value can be indexed with the value:

$$V_{Grass} = (0.299 \times 1.00 \times 0.73) \cong 0.218$$

Where the Grass weight = 0.299, the availability = 1, and the supporting factor = 0.73.

And, Flowers' value can be indexed with the value:

$$V_{Flowers} = (0.305 \times 1.00 \times 0.74) \cong 0.226$$

Where Flowers' weight = 0.305, the availability = 1, and the supporting factor = 0.74.

ME6 is used to compute the total values of Plants' components, which is used to compute the Plants availability then multiply the total values of the Plants components by the Plants' rule significant factor. Later on in this chapter we illustrate how we compute the non-primitive mid-level, high-level and higher-level objects' availability.

In our case study, the total value of Plants components is computed below:

$$V_t = V_{Trees} + V_{Grass} + V_{Flowers} = 0.380 + 0.218 + 0.226 = 0.824$$

ME5 is used to compute the objects values of the mid-level objects (Plants, Open Area, and Cloud), which belong to a high-level object, "Garden" using the same technique that is used to compute the values of the low-level objects as follows:

$$V_{Plants} = W_{Plants} \times A_{Plants} \times S_{Plants}$$

Plants' value can be indexed with the value:

$$V_{Plants} = (0.487 \times 0.824 \times 1.00) \cong 0.403$$

Where Plants' weight = 0.487, and the availability = 0.824 (Plants is a non-primitive object, the rule significant factor for Plants is equal to one since it contain all possible objects that could be found in Plants, the value of it availability will not change), and supporting factor = 1.00.

Similarly, Open Area's value can be indexed with the value:

$$V_{Open\ \ Area} = (0.487 \times 1.00 \times 1.00) \cong 0.487$$

Where Open Area's weight = 0.487, the availability is taken to be 1 because it is directly indexed primitive content, and the supporting factor = 1.

And, Cloud's value can be indexed with the value:

$$V_{Cloud} = (0.025 \times 1.00 \times 0.05) \cong 0.001$$

Where Cloud's weight = 0.025, the availability is taken to be 1, and the supporting factor = 0.05.

In our case study, the total value of Garden components is computed below:

$$V_t = V_{Plants} + V_{Open\ \ Area} + V_{Cloud} = 0.403 + 0.487 + 0.001 = 0.891$$

4.3.5 Object's Rule Significant Factor

An object's Rule Significant Factor (Rsf) is a measure of the total appearance of all object components in a certain image divided by the total appearances of all objects for components of the same object's name in the database. Rsf needed for the non-primitive objects to compute its availabilities, Rsf of the primitive objects is equal to 1.

Measuring an Object's Rule Significant Factor

ME7 is used to compute the object's Rsf by dividing the total number of appearances for all low-level objects that belong to a certain mid-level object in one case (image), to the number of appearance for all objects that belong to the same mid-level object name in the database, then:

$$Rsf_j = \frac{P_1 + P_2 + \ldots + P_n}{P_1 + P_2 + \ldots + P_m}, where \ \ m \geq n$$

$$Rsf_j = \frac{\sum_{i=1}^{n} P_i}{\sum_{i=1}^{m} P_i} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (ME7)$$

$\sum_{i=1}^{n} P_i$ Stands for the total number of appearances of mid-level object components, in our case, "Plants", and $\sum_{i=1}^{m} P_i$ stands for the total number of appearances of all objects that are components of an object named Plants in the database, and so on.

Uses of ME7

ME7 is used to compute the object's rule significant factor (Rsf) of the non-primitive objects, the result is multiplied by the total values of its components results in an object's availability.

Computing the Object's Rule Significant Factor

In our example, 77 images containing Plants in the database, 'Trees' appeared 74 times as a component of Plants; 'Grass' 56 times and 'Flowers' 57 times. The Plants in database has only three objects (Trees, Grass, and Flowers) as components of it.

The total number of appearances of Plants' components in this case = 187, and the total number of appearance of Plants' components in the database = 187. Using ME7, Rsf of Plants in this image computed as below:

$$Rsf_j = \frac{\sum_{i=1}^{n} P_i}{\sum_{i=1}^{m} P_i} = \frac{P_{Trees} + P_{Grass} + P_{Flowers}}{P_{Trees} + P_{Grass} + P_{Flowers} + P_{Others}}$$

Plants' Rsf can be indexed with the value computed below:

$$Rsf_{Plants} = \frac{74+56+57}{74+56+57+0} = \frac{187}{187} \cong 1.00$$

In the case of the Plants components including Trees and Grass only, their appearances = 74 + 56 = 130.

$$Rsf \text{ of Plants in this case } = \frac{74+56}{74+56+57} = \frac{130}{187} \cong 0.695 .$$

The total number of appearances of Garden's components in our example is computed below. The total number of appearances of all objects which are components of objects named Garden in the database = 290 (the appearances of 20 objects were components of objects named Garden in database). ME7 can be used to compute the object's rule significant factor (Rsf) of the high-level object "Garden", with the same technique that is used to compute Rsf of Plants object (mid-level object):

$$Rsf_j = \frac{\sum_{i=1}^{n} P_i}{\sum_{i=1}^{m} P_i} = \frac{P_{Plants} + P_{Open\ Area} + P_{Cloud}}{P_{Plants} + P_{Open\ Area} + P_{Cloud} + P_{Others}}$$

$$Rsf_{Garden} = \frac{77+77+4}{290} = \frac{158}{290} \cong 0.545$$

4.3.6 Availability of Non-Primitive Objects

As mentioned before, objects' availability is a measure of the expected probability of the objects' appearance in an image, and the availability of the primitive objects is equal to 1, the availability of the non-primitive objects may be different from 1 ($0 < A_j \leq 1$).

Measuring an Object's Availability

The availability (A_j) of a non-primitive mid-level object is defined as the sum of the values of all its low-level objects' components multiplied by the mid-level object's rule significant factor. It has a value between 0 and 1; ME8 or ME9 computes the availability of any non-primitive object.

$$A_j = \left(\sum_{i=1}^{n} V_i \right) \times Rsf_j \cdots\cdots\cdots\cdots\cdots\cdots (ME8)$$

$$A_j = V_t \times Rsf_j \cdots\cdots\cdots\cdots\cdots\cdots (ME9)$$

$$where \quad V_t = \sum_{i=1}^{n} V_i$$

Uses of ME8 & ME9

ME8 or ME9 compute the non-primitive objects' availability by multiplying the total value of its components by its rule significant factor. The result is the value of the component contained inside the object.

Computing the Object's Availability

In a previous section, the values of Plants' components (Trees, Grass, and Flowers) were computed as follows:

$V_{Trees} = 0.380$, $V_{Grass} = 0.218$, and $V_{Flowers} = 0.226$, then

$$V_t = V_{Trees} + V_{Grass} + V_{Flowers} = 0.380 + 0.218 + 0.226 = 0.824 ,$$

And the Plants' rule significant factor indexed with the value:

$$Rsf_{Plants} = \frac{187}{187} = 1.00 .$$

Using ME8, Plants' availability can be indexed with the value:

$$A_{Plants} = V_t \times Rsf_{Plants} = 0.824 \times 1.00 = 0.824$$

Using the same technique, we can compute the availability of Garden (high-level object) from its components (mid-level objects) as follows:

Since $V_t \cong 0.891$, and $Rsf_{Garden} \cong 0.545$, then

Garden's availability can be indexed with the value:

$$A_{Garden} = V_t \times Rsf_{Garden} = 0.891 \times 0.545 \cong 0.486$$

4.3.7 Object's Weight in Image

Objects' weight in image (Wii_i) is a measure of the importance of an object in an image.

Measuring an Object's Weight in Image

The weight in image is computed by multiplying the object's weight by the weight of the set in which it is a component of and the result follows the same procedure and so on until the last level in the image, using ME10.

$$Wii_i = W_1 \times W_2 \times ... \times W_n \cdots\cdots\cdots\cdots\cdots\cdots\cdots (ME10)$$

Where W_1 stands for the weight of the lower-level object, W_2 stands for the weight of the higher-level object where the lower-level object is one of its components, and W_n stands for the weight of the highest-level object present in an image, the highest-level object's weight's value is considered to be =1.00 .

Uses of ME10

ME10 is used to compute the weight in image of any object compared with the whole image weight, by multiplying its weight by the object's weight which it belongs to, and the result being multiplied by the higher-level weight and so on until the last level of objects in the image.

Computing the Object's Weight in Image

As shown in Figure 4.9, the Trees weight (W_1) = 0.396, Plants weight (W_2) = 0.487, and the Garden weight (W_n) = 1.00 since it is the highest level of this image.

Using ME10, $Wii_i = W_1 \times W_2 \times \ldots \times W_n$

Then the Trees' weight in image can be indexed with the value:

$Wii_{Trees} = W_{Trees} \times W_{Plants} \times W_{Garden} = 0.396 \times 0.487 \times 1.00 \cong 0.193$,

Similarly, Grass' weight in image can be indexed with the value:

$Wii_{Grass} = 0.229 \times 0.487 \times 1.00 \cong 0.146$

And Flowers' weight in image can be indexed with the value:

$Wii_{Flowers} = 0.305 \times 0.487 \times 1.00 \cong 0.149$.

Using the same technique, we can compute the weight in image of mid-level objects as follows, where the Garden weight is equal to 1.

Then the Plants' weight in image can be indexed with the value:

$Wii_{Plants} = W_{Plants} \times W_{Garden} = 0.487 \times 1.00 \cong 0.487$, Similarly, Open Area's weight in image can be indexed with the value:

$$Wii_{Open\ Area} = 0.487 \times 1.00 \cong 0.487$$

And Cloud's weight in image can be indexed with the value:

$$Wii_{Cloud} = 0.025 \times 1.00 \cong 0.025,$$

The weights in image for all objects inside the given image (Figure 4.2) are shown in Figure 4.10, where these weights appear inside the double closed brackets [[]] in figures.

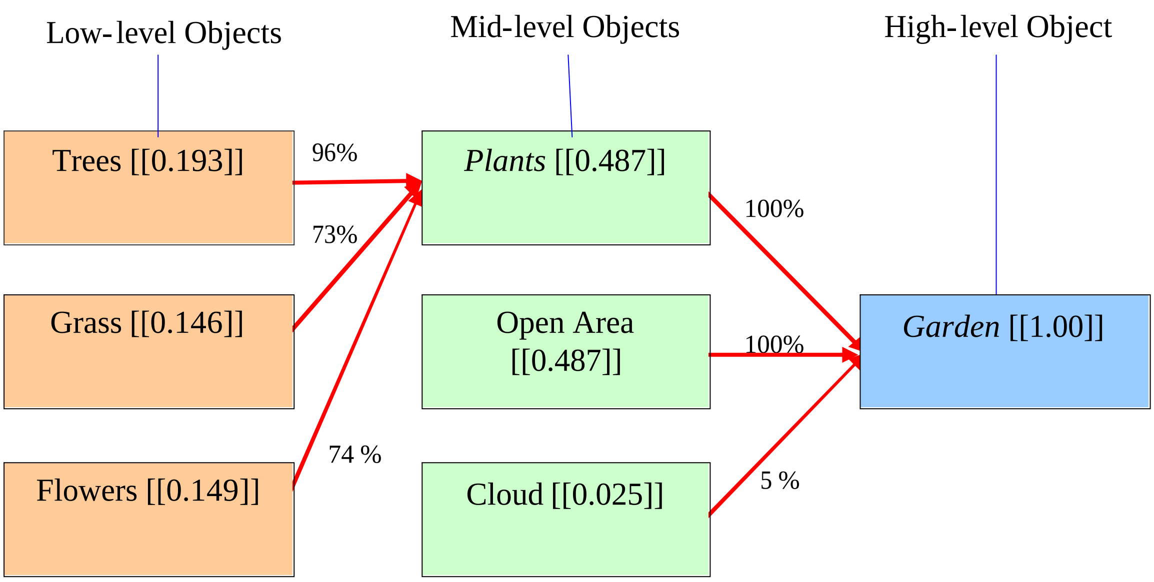


Figure 4.10 Weights in Image and Supporting Factors of All Levels of Objects within Image

4.3.8 Object's Density

One option of image retrieval in our system is to retrieve images containing objects of various sizes (Density); in this section we explain the objects' density and the way of computing this density that is to be stored in the database. An object's density can take any value between 0 and 1, and the density value can fall in one of three levels as described below.

Object density (D_i) is related to the object's height and width in an image, which is equivalent to the area occupied by the object in the image.

The object density takes one of three values as described below:

$$D_i = \begin{cases} [0.00]\,to\,[0.20]\,\{\ for\ low\ density \\ [0.15]\,to\,[0.40]\,\{\ for\ medium\ density \\ [0.30]\,to\,[1.00]\,\{\ for\ high\ density \end{cases}$$

The above values are considered to be reasonable in the present context. In general, of course, these can be fine-tuned or modified, which may be on the form:

$$D_i = \begin{cases} [0.0]\,to\,[\alpha]\,\{\ for\ low\ density \\ [\beta]\,to\,[\gamma]\,\{\ for\ medium\ density \\ [\delta]\,to\,[1.0]\,\{\ for\ high\ density \end{cases} \cdots\cdots\cdots\cdots (ME11)$$

$$Where\ \ 0 < \beta \le \alpha < 1, \quad \delta \le \gamma$$

Density and Fuzzy Indexing & Retrieval

The object density provides users with a means to give priorities to selected query components in order to optimise the results of the query. Users may want to further tailor the query results by widening the scope of the result or narrowing the result.

In order to measure and compute the object's density, we need to go through the main step of fuzzy indexing and retrieving, these steps are used to determine the density notation and numerical ranges, the degree of membership of density components (height and width), construction of the fuzzy rules, and finally encode these rules. We summarized these step in a form illustrated in the following paragraphs.

1. Specify the Problem and Define Linguistic Variable

In building any expert system, we need to determine the problem's input and output variables and then ranges in order to determine the density level. Table 4.2 specifies the range of our linguistic variables, which has the overall range of [0, 1].

Table 4.2 Linguistic Variables and their Ranges

Linguistic variable: Object's weight in image		
Linguistic value	**Notation**	**Numerical range**
Low	L	[0.00, 0.40]
Medium	M	[0.30, 0.70]
High	H	[0.60, 1.00]

2. Determining the Fuzzy Sets

Determining the initial value of an object's density D_i depends on its height and width in image. A triangle or trapezoid shape can often provide an adequate representation of the expert knowledge and at the same time significantly simplifies the process of computation. One of the key points is to maintain sufficient overlap in adjacent fuzzy sets, Figure 4.11.

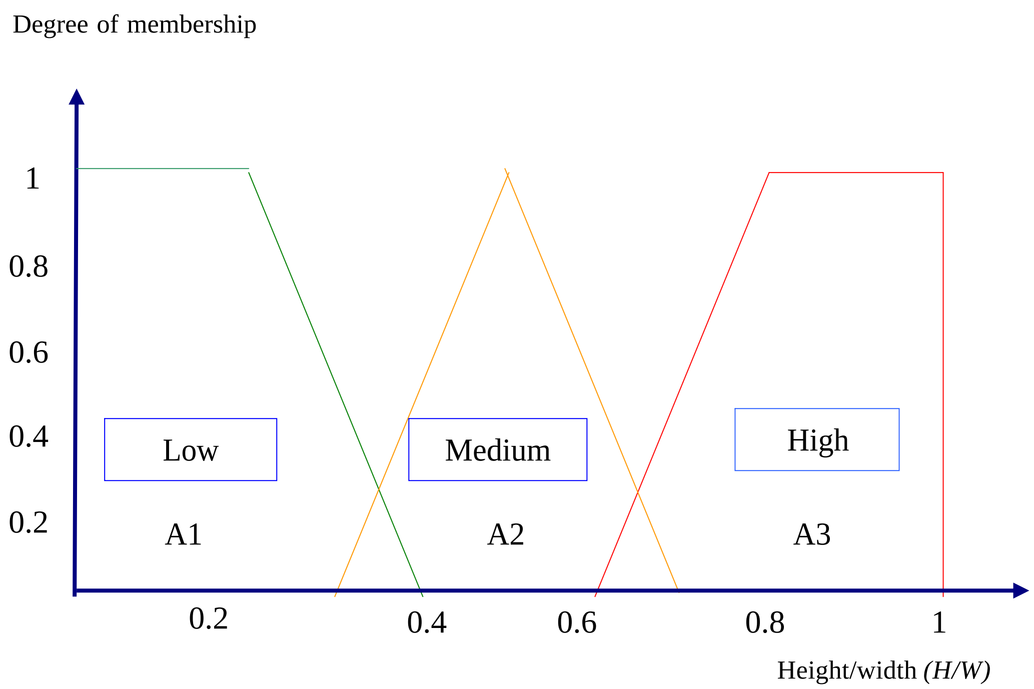


Figure 4.11 Different Levels of Densities

3. Constructing Fuzzy Rules

We consider an example, having two inputs and one output; it is often convenient to represent fuzzy rules in a matrix form. The representation takes the shape of an M x N matrix. We make use of a very basic relation between the height "*h*" and the width "*w*" of the same object in the image; assuming that other input variables are fixed. This relation can be expressed in the following form:

If w increases, and h increase, the object density-concept will increase. Thus we could write nine rules as follows:

If the height is X, Width is Y, then D is Z, and so on as in Figure 4.12, where *X* and *Y* can each take on the values L, M, or H.

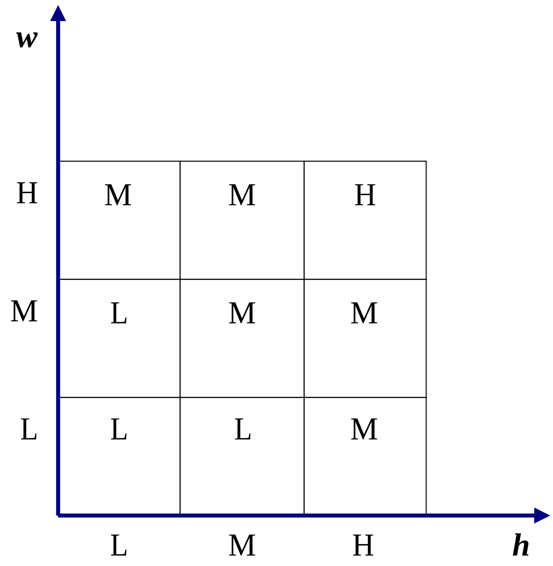


Figure 4.12 Density Assignments

For three inputs and one output, the representation takes the shape of an M x N x K cube, where “K” represents the third dimension of the cube. If we increase the number of inputs to three and one output, we then have a Fuzzy Associative Memory (FAM), which enables us to derive twenty seven rules that represent complex relationships between all variables used in the expert system. Since images can be understood and interpreted at multiple levels, they can also be indexed at multiple levels.

4. Encoding the fuzzy sets, Fuzzy Rules and Procedure

To encode the fuzzy sets, fuzzy rules and procedures, we may choose one of two options: use a programming language, or to apply a fuzzy logic development tool to encode the fuzzy sets, and the fuzzy rules and procedures to perform fuzzy inference in the expert system. Most experienced fuzzy system builders prefer the programming language approach [WITT99], [LI95].

The Primitive Object's Density

It is related to the primitive object size, which is computed using the object's height and width in the image. This is equivalent to the area occupied by the object in the image.

Measuring an Object's Density

ME12 computes the primitive object's density using the product of the height and width values with its degree of membership, the degree of membership can take the value between 0 and 1.

$$D_i = \frac{\sum_{M_h=0}^{1} M_h(x_1)\, x_1 + \sum_{M_w=0}^{1} M_w(x_2) x_2}{\sum_{M_h=0}^{1} M_h(x_1) + \sum_{M_w=0}^{1} M_w(x_2)} \text{.................... } (ME12)$$

Where $M_h(x_1)$ represents the degree of membership for the height on the *y*-axis, which has the value of x_1 on the *x*-axis, the degree of membership takes the value between 0 and 1. $M_w(x_2)$ represents the degree of membership for the width on the *y*-axis, which has the value of x_2 on the *x*-axis.

Uses of ME12

ME12 is used to compute the primitive object density of different objects levels.

Computing the Primitive Object's Density

Suppose that we have a number of objects (i.e. Trees) having the height value of x_1 = 0.8 at point "b" on the *x*-axis, and the width value of x_2 = 0.32 at point "a" on the *x*-axis, where both values of x_1 and x_2 can have a value between 0 and 1.

The degree of membership for the Trees' height in A3, $M_h(x_1)$ = 0.91 on the *y*-axis, and the degree of membership for the Trees' width have two values, the first value in A1, $M_w(x_2)$ = 0.38 on the *y*-axis, and the second value of width degree of membership in A2, $M_w(x_2)$ = 0.1 on the *y*-axis., as shown in Figure 4.13.

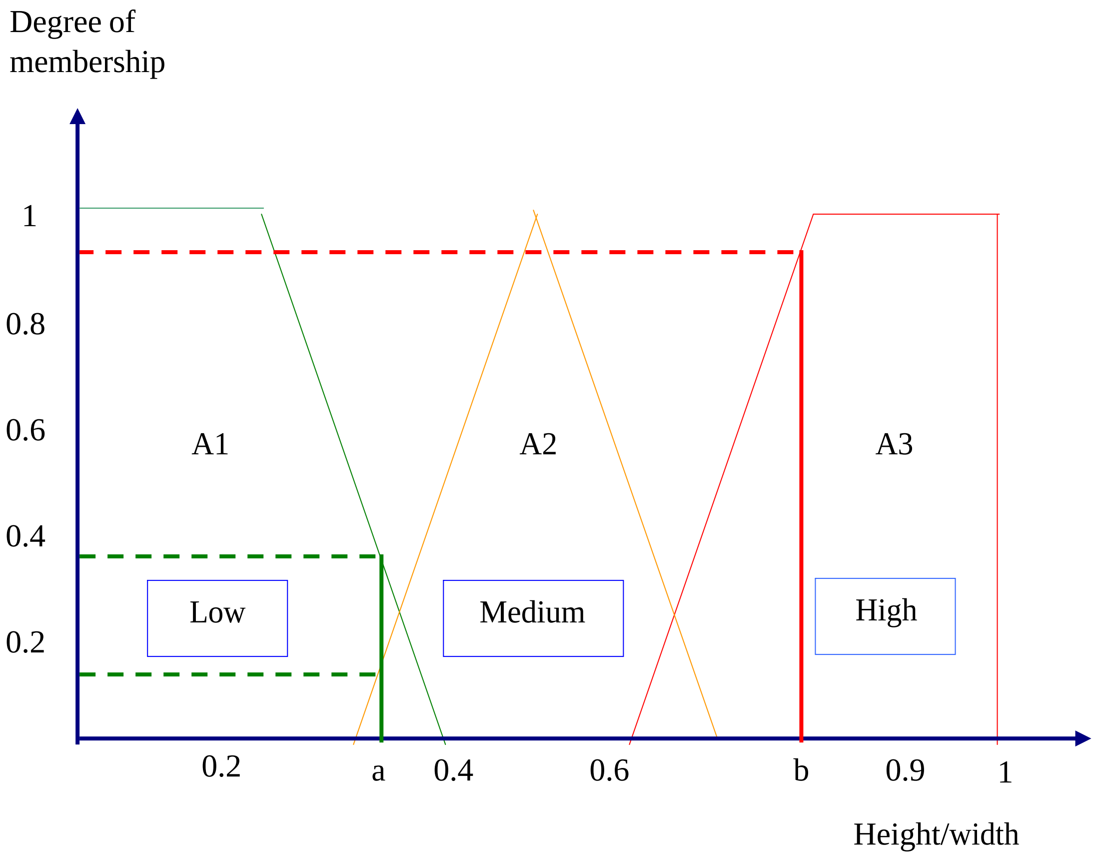


Figure 4.13 Density Levels and Degree of Memberships

Using ME12, the density of trees will be computed as follows:

$$D_{Trees} = \frac{(0.1 \times 0.32 + 0.38 \times 0.32) + 0.91 \times 0.8}{(0.1 + 0.38) + 0.91} \cong 0.63$$

The Non-Primitive Object's Density

After computing the primitive object's densities, the non-primitive object's density can be computed from the sum of its component object's density values.

Measuring non-primitive Object's Density

ME13 computes the non-primitive object density using the total sum value of it components' object densities.

$$D_j = \sum_{i=1}^{n} D_i \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (ME13)$$

Uses of ME13

ME13 computes the non-primitive object density for different object levels from the sum of its component object's density values.

Computing the Non-Primitive Object's Density

Suppose that we have a number of objects (Trees, Grass, and Flowers) as components of a higher-level object (Plants), with density values of (0.65, 0.05, and 0.10). Consequently the density of Plants will be computed as follows:

$$D_{Plant} = D_{Trees} + D_{Grass} + D_{Flowers}$$

Then the Plants density can be indexed with the value:

$$D_{Plant} = 0.63 + 0.05 + 0.10 = 0.78$$

Suppose that the density of Open Area = 0.15 and the density of Cloud = 0.07, when applying the same ME13, consequently the density of Garden will be computed as follows

$$D_{Garden} = D_{Plants} + D_{Open\ Area} + D_{Cloud}$$

Then the Garden object can be indexed with the value:

$$D_{Garden} = 0.78 + 0.15 + 0.07 = 1.00$$

4.4 Summary

This chapter explains in detail the meaning and significance of objects and their attributes. We created many Mathematical Equations to measure and compute the different values of objects' attributes, including the object's weight, supporting factor, value, rule significant factor, availability for the primitive and non-primitive objects, weight in image and density.

The method of using every Mathematical Equation was also explained via some examples. It is shown how the main step of density and fuzzy indexing and retrieving can be used for measuring and computing the object density. These steps are used to determine the density notion and numerical ranges, the degree of membership of density components, construction of the fuzzy rules, and then encoding these rules. By showing examples, we illustrate the way of computing the density of primitive and non-primitive objects. Finally a Mathematical Equations summary table is shown in Table 4.3.

Table 4.3 Mathematical Equations

Mathematical Equation #	Uses of Mathematical Equations
ME1	Computing the total repetition of object's components.
ME2	Computing the object's weights which belong to a certain higher-level object.
ME3	Computing the supporting factors.
ME4	Determine that the primitive object's availability is equal to 1.
ME5	Computing the object's values.
ME6	Computing the total values of lower-level objects which belong to a certain higher-level object.
ME7	Computing the rule significant factor of non-primitive object.
ME8 or ME9	Computing the non-primitive object's availability.
ME10	Computing the object's weight in image.
ME11	Determine the level of density due to its value.
ME12	Computing the primitive object's density.
ME13	Computing the non-primitive object's density.

Chapter 5

Image Indexing Paradigm

5.1 Introduction

The main premise of this chapter is to design the process of data indexing in an easy way to store the image database. The approach process of image indexing and its advantages are important factors to be discussed in this chapter. We look at the way of indexing the image components in a suitable manner where the different index levels will be defined for simplicity in finding an image that we are looking for. Index levels will then be used to define indices of higher levels. This will be done so that the indices are organized in a manner allowing for fast and easy retrieval of any query. The indexing model and its sections shown in Figure 5.1 in this chapter are very important to explain the path of indexing our data, where the module's part has its own role.

The method of indexing the primitive objects and the created objects is explained in detail in this chapter, the extraction of data and the different object's attribute values need to be explained and how to store it in different tables in a database. The steps of entering the data in the database are listed and examples of this data are shown within the different tables.

Our approach provides a new and innovative way to specify and view visual data across the whole spectrum of applications that involve the processing of visual data. This approach has significantly increased the performance of the retrieval algorithm, as measured by the proportion of actual retrieved images versus the total number of images that satisfy the retrieval criteria, the performance optimisation.

5.2 Image Indexing

Our approach to image indexing and retrieval system is to develop algorithms for the storage and retrieval of image data from stored databases using a methodology that incorporates implicit image indexing. We developed techniques for the categorization of image components, and the assignment of weights, supporting objects, densities, and other attributes to these components on a proportional scale based on their relative importance within the image [AZZA05], [AZZA04].

Images are indexed based on the component objects within the particular image. By extracting a high-level object from the base objects, we develop a set of high-level indices, which can be used for image retrieval.

The primitive objects which are used to create the mid-level object are of low-level and their types are equal to 1, the created objects and some of the remaining objects could be used to create high-level objects that are of mid-level objects and their types are equal to 2. The high-level objects are of type 3 and so on. Table 5.1 shows the objects levels and its types to be store in the database.

Table 5.1 Objects Levels and their Types

Object Level	Object Type
Low-level objects	1
Mid-level objects	2
High-level Objects	3
Higher-level objects	4

As discussed in Chapter 3 Section 3.3, the image indexing consists of many discrete object types: low-level objects, mid-level objects, high-level objects, and so on. We can build a higher-level index from lower level (atomic/primitive) indices that might be directly recognizable from the image. This indexing mechanism will avoid inconsistency in human perception of the image concept when the process is performed manually. For retrieval, a user can take the advantages of these high-level indices to speed up the searching time and narrow down the search space.

5.3 Database Structure

In this section we explain our Database Structure, which shows the creation of some of the tables which hold the data used for object indexing and retrieval in the database.

Two tables in the database (FACTS and FACTSDATA) are designed to hold the data that belongs to the different object attribute values, this data is stored in the mentioned tables and then used to retrieve all information needed for different options of image search. These values are extracted and computed from different objects levels within images with the images being stored in a particular folder. The FACTS and FACTSDATA tables with description of all objects’ attributes are shown in Table 5.2 and Table 5.3.

Table 5.2 Objects in FACTS Table with Description

Attribute No.	Attribute Name	Attribute Description
01	FID	Object identification
02	FName	Object name
03	TypeID	Type level
04	Density	Object density

Table 5.3 Objects in FACTSDATA Table with Description

Attribute No.	Attribute Name	Attribute Description
01	FID_Child	Child object identification
02	FID_Parent	Parent object identification
03	Image Name	Image address containing the object
04	Weight	Object weight due to its object parent
05	Incl	Object supporting factor
06	Avail	Object availablity
07	Value	Object value
08	RSF	Object rule significant factor
09	WII	Object weight to the whole image

The FACTS table must be created before the FACTSDATA table due to the foreign key constraint found in the FACTSDATA table. In the FACTSDATA table, the relationship between different objects is built. The way of creating these tables using Oracl9i is illustrated below:

Creating FACTS *Table*

```
CREATE TABLE FACTS (
    FID         VARCHAR (20) NOT NULL,
    FName       VARCHAR (50) NOT NULL,
    TypeID      NUMBER,
    Density     NUMBER (6, 3),
    CONSTRAINT PK_FID PRIMARY KEY(FID));
```

Creating FACTSDATA *Table*

```
CREATE TABLE FACTSDATA (
        FID_Child    VARCHAR(20) NOT NULL,
        FID_Parent   VARCHAR(20) NOT NULL,
        ImageName    VARCHAR(50) NOT NULL,
        Weight       NUMBER(6,3),
        Incl         NUMBER(6,3),
        Avail        NUMBER(6,3),
        Value        NUMBER(6,3),
        Rsf          NUMBER(6,3),
        WII          NUMBER(6,3),
        CONSTRAINT PK_CID_PID PRIMARY KEY(FID_Child,
        FID_Parent),
        CONSTRAINT FK_FID_Child FOREIGN KEY(FID_Child)
        REFERENCES FACTS (FID),
        CONSTRAINT FK_FID_Parent FOREIGN KEY(FID_Parent)
        REFERENCES FACTS (FID));
```

The FACTS table has a primary key, “FID” and the FACTSDATA table has the composite primary key, “FID_Child and FID_Parent”. In the FACTSDATA table, there are two foreign keys, “FID_Child” and, “FID_Parent” and each foreign key has a reference in the FACTS table, “FID”.

5.4 Indexing Algorithm

Our indexing module consists of four main sections (Images Collection, Data Extraction, Objects Attributes' Values Computation and Data Storing and Indexing). Every section of this module has its own role and is explained in detail later in this chapter. Figure 5.1 illustrates the algorithm sections needed to store the data of the object's attributes for all levels.

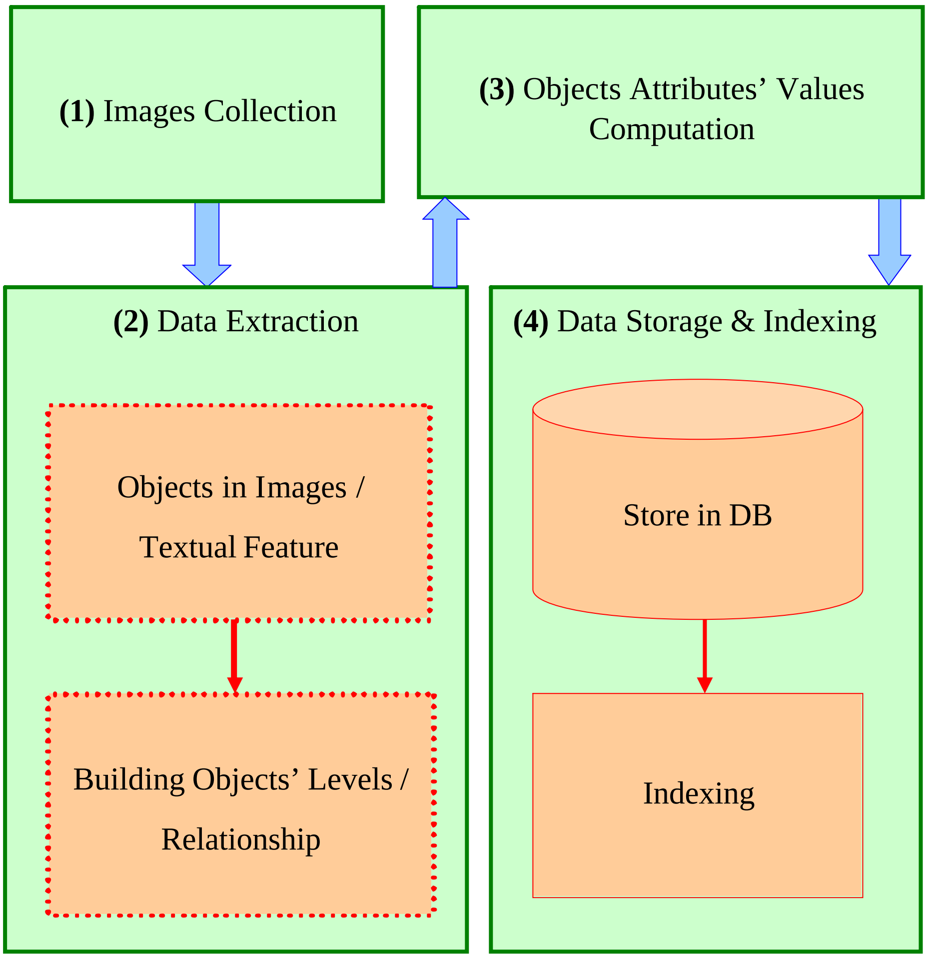


Figure 5.1 Indexing Modules

Thus our indexing algorithm consists of the following steps:

1. Collect the suitable images and store them in an image folder.
2. Data Extraction:

    - Determine the primitive objects presented within the image.
    - Create the mid-level, the high-level and higher-level objects (when possible), which are not primitive.
    - Assign image names (Categories) to the images stored in the image folder (normally the highest created objects names).

3. Compute the objects' attributes' values (objects' weights, supporting factors, availabilities, values, weights in images and densities).
4. Save the objects' extracted data and the computed values in the database and perform the indexing.

The sections of our Indexing Modules are explained in depth below, showing how it works.

5.4.1 Image Collection

The selected images used in our database represent a good selection of multi-object images, since the selected images have images with two or more objects, images with repetitive objects, objects having relationships to each other, and so on.

No images with one object have been selected, since we are looking for images from which we can build different levels of objects. The example used in Chapter 3, Section 3.7, when an image containing the primitive objects: Trees, Grass, Flowers, Open Area and Cloud in which it has repetitive objects (Trees and Flowers), and there is relationship present between objects like: Trees, Grass and flowers from one side and the Plants from other side, the same with Plants, Open Area and Cloud with Garden is shown in Figure 5.4.

5.4.2 Data Extraction

Initially, the object is entered manually or extracted automatically from the image by human or computer in terms of primitive objects. Depending on the application, an indexer will have to predetermine to what extent they want to decompose an image object into primitive objects, which will be used to automatically develop higher-level objects using a knowledge-based system.

For example, applying the knowledge-based system to the example mentioned in Section 5.4.1 in this chapter, we can create the mid-level object, "Plants" using rule "$R1m_4$":

$R1m_4$: IF there are Trees, Grass, and Flowers, then the object is "Plants"

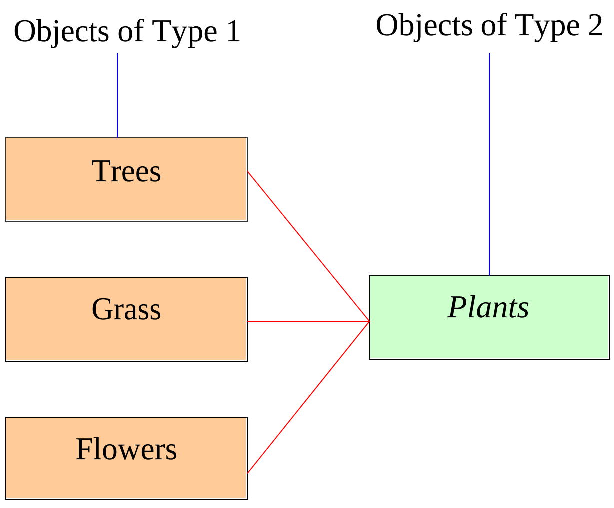


Figure 5.2 Objects of Type 1 and Type 2

And the high-level object, "Garden" using the rule "$R1h_1$":

$R1h_1$: IF there are Plants and Open Area, then the object is "Garden"

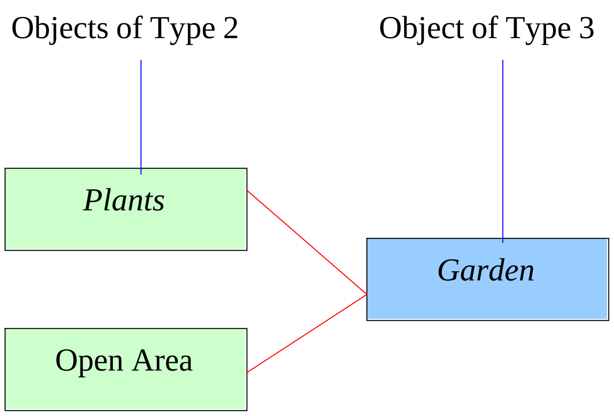


Figure 5.3 Objects of Type 2 and Type 3

When using all objects inside the image including the "Cloud" object, which is not used for the creation of any higher level objects, the final figure containing all primitive objects including the created (non-primitive) objects, their relationship and levels is shown in Figure 5.4

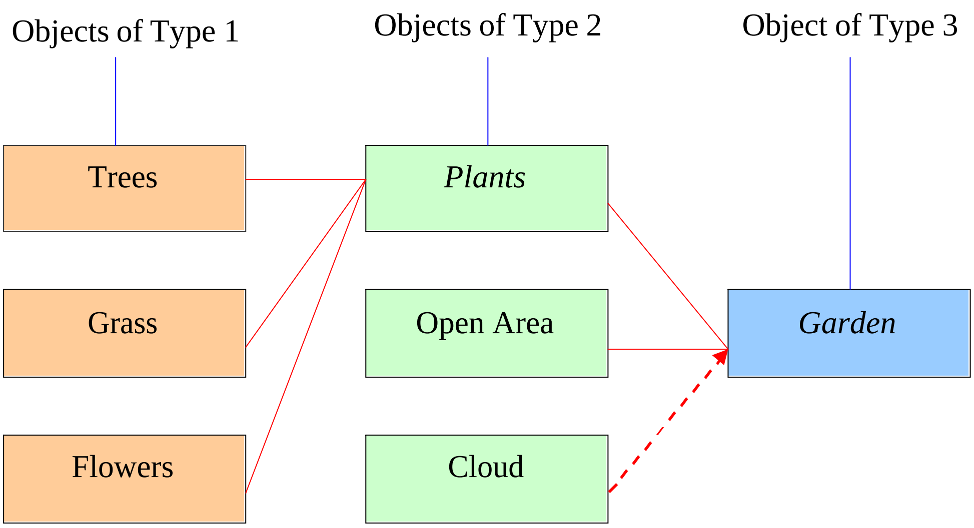


Figure 5.4 Objects of all Types Contained within Image

As explained in Section 5.2 in this chapter, these objects are organized in different types as follows:

- Grass, Flowers, and Trees are of low-level objects, which take the object type value of 1.
- The created object, “Plants”, “Open Area” (primitive) and the remaining object “Cloud”, which is not a necessary a component of a higher level object, are considered as mid-level objects; they take the object types value of 2.
- The Garden object, being a high-level object, will take the object type value of 3.

Table 5.4 shows the objects in Figure 5.4 and their types in addition to an object description.

Table 5.4 Image Objects, their Types and Descriptions

Object Number	Object Name	Object Type	Object Description
001	Trees	1	Primitive
002	Grass	1	Primitive
003	Flowers	1	Primitive
004	Open Area	2	Primitive
005	Cloud	2	Primitive
006	Plants	2	Non-Primitive
007	Garden	3	Non-Primitive

Data Extraction Procedure

The primitive objects are entered in the specific table in the database, object numbers are assigned to all objects; all primitive objects have the initial type of “1”, their status is “P” as primitive. The current type for all primitive objects is equal to “1”, and their parent number equal to “None” since a parent is not assigned to any of them at this point in time. This data is shown in Table 5.5.

Table 5.5 Primitive Objects of an Image

Object Number	Object Name	Initial Type	Status	Current Type	Parent Number
001	Trees	1	P	1	None
002	Grass	1	P	1	None
003	Flowers	1	P	1	None
004	Open Area	1	P	1	None
005	Cloud	1	P	1	None

Using the knowledge based system; we can create the mid-level object, "Plants" from its components (Trees, Grass and Flowers) by using the rule "$R1m_4$". The object number follows the series and follows the last number present in the database, in this situation the created object will be '006'. The created object has the initial and current type of "2" and the status of "C1" as created first with the parent number of "None". The object number of the created object is assigned to its components as parent number. The current type of other objects, which are not used as components of the other objects, is changed to be "2". The updated data is shown in Table 5.6.

Table 5.6 Primitive Objects and the Created Mid-Level Object

Object Number	Object Name	Initial Type	Status	Current Type	Parent Number
001	Trees	1	P	1	006
002	Grass	1	P	1	006
003	Flowers	1	P	1	006
004	Open Area	1	P	2	None
005	Cloud	1	P	2	None
006	Plants	2	C1	2	None

Using the same technique, we can create a high-level object, "Garden" from its components (Open Area and Plants) using the rule "$R1h_1$" having the object number of "007" with initial and current type of "3". The created object has the status "C2" as created second. The object number of the created object is assigned to its components as their parent number. The parent number of the created object is "None" and the current type of all objects, which are not used as components of a higher level object, in this case "Cloud", are changed to be "3" as shown in Table 5.7.

Table 5.7 Primitive Objects, Mid-Level Objects and the Created High-Level Object

Object Number	Object Name	Initial Type	Status	Current Type	Parent Number
001	Trees	1	P	1	006
002	Grass	1	P	1	006
003	Flowers	1	P	1	006
004	Open Area	1	P	2	007
005	Cloud	1	P	3	None
006	Plants	2	C1	2	007
007	Garden	3	C2	3	None

If no further rules can be used to create higher level objects from those that are not components of one with a current type of 3 and the object is not a parent itself then the current type is decreased by one (from three to two). The parent number of this object is set to be that of the highest level object, in this case "Garden". The most recent created object, being the highest level object is given the parent number of itself (object number = parent number) as it can not be a component to further created objects. The final object numbers, names, types, parent numbers and other data for all objects within the image are shown in Table 5.8.

Table 5.8 Final List of Objects' Data Extracting

Object Number	Object Name	Initial Type	Status	Current Type	Parent Number
001	Trees	1	P	1	006
002	Grass	1	P	1	006
003	Flowers	1	P	1	006
004	Open Area	1	P	2	007
005	Cloud	1	P	2	007
006	Plants	2	C1	2	007
007	Garden	3	C2	3	007

5.4.3 Objects Attributes' Values Computation

This section concentrates on computing the object's attribute values. The computing is performed using the different Mathematical Equations (MEs) shown in Section 4.3 of Chapter 4. As soon as the Data is extracted, the Object Attribute Value Computation begins.

The data processing follows the next steps:

1. The objects numbers, names, types and the density of all objects within the image including the created objects are entered, and the objects density is computed and entered together with the data extracted. All this data is entered in the FACTS table as shown in Table 5.9 found in Section 5.4.4 of this chapter. Other data is stored in the FACTSDATA table as shown in Table 5.10 found in Section 5.4.4 of this chapter as well.
2. In the FACTSDATA table, the object numbers and the parent number of every object is entered; the image name (image destination) is assigned and entered for all objects, which will be the same for all objects within the image.
3. The weight of all objects is computed.
4. The supporting factor of all objects is computed.
5. The availability and the rule significant factor are assigned to all primitive objects as 1.
6. The values of all non-primitive objects are computed.
7. The total value for all components of the created object of mid-level objects is computed and added to the mid-level object availability.
8. The rule significant factor for the created mid-level object is computed.
9. The availability of the mid-level object is computed.
10. The value for all components of the created object of high-level object is computed and added to the high-level object availability.
11. The rule significant factor for the created high-level object is computed.
12. The availability of the high-level object is computed.

13. The weight in image for all objects (primitive and non-primitive) is computed.

All values of the objects' weights, supporting factors, availabilities, values, densities and weights in images of the objects used in the example mentioned in Section 5.4.2 of this chapter are computed and explained in detail in Section 4.3 of Chapter 4.

We summarized these values as follows:

The weights of the low-level objects were:

$W_{Trees} = 0.396$, $W_{Grass} = 0.299$ and $W_{Flowers} = 0.305$

The weights of the mid-level objects were:

$W_{Plants} = 0.487$, $W_{Open\ Area} = 0.487$ and $W_{Cloud} = 0.025$.

The weight of the high-level object was:

$W_{Garden} = 1.00$

The supporting factors of the low-level objects were:

$S_{Trees} = 0.96$, $S_{Grass} = 0.73$ and $S_{Flowers} = 0.74$

The supporting factors of the mid-level objects were:

$S_{Plants} = 1.00$, $S_{Open\ Area} = 1.00$ and $S_{Cloud} = 0.05$

The supporting factor of the high-level object was:

$$S_{Garden} = 1.00$$

The availabilities of the low level objects were:

$$A_{Trees} = 1,\ A_{Grass} = 1,\ A_{Flowers} = 1$$

The Values of the low-level objects were:

$$V_{Trees} = 0.380\ , V_{Grass} = 0.218 \text{ and } V_{Flowers} = 0.226$$

The total values of the low-level objects were:

$$V_t = V_{Trees} + V_{Grass} + V_{Flowers} = 0.824$$

The Values of the mid-level objects were:

$$V_{Plants} = 0.403\ , V_{Open\ Area} = 0.487 \text{ and } V_{Cloud} = 0.001$$

The total values of the mid-level objects were:

$$V_t = V_{Plants} + V_{Open\ Area} + V_{Cloud} = 0.891$$

The rule significant factors of primitive objects were:

$$Rsf_{Trees} = Rsf_{Grass} = Rsf_{Flowers} = Rsf_{Open\ Area} = Rsf_{Cloud} = 1.00$$

The rule significant factors of non-primitive objects were:

$Rsf_{Plants} = 1.00$ And $Rsf_{Garden} = 0.545$

The availabilities of primitive objects were:

$A_{Grass} = A_{Trees} = A_{Flowes} = A_{Open\ Area} = A_{Cloud} = 1$

The availabilities of non-primitive objects were:

$A_{Plants} = 0.824$, and $A_{Garden} = 0.486$

The weights in image of the low-level were:

$Wii_{Trees} = 0.193, Wii_{Grass} = 0.146$ and $Wii_{Flowers} = 0.149$

The weights in image of the mid-level objects were:

$Wii_{Plants} = 0.487, Wii_{Open\ Area} = 0.487$ and $Wii_{Cloud} = 0.025$,

The weight in image of the high-level object was:

$Wii_{Garden} = 1.00$

The densities of the low-level objects were:

$D_{Trees} = 0.63$, $D_{Grass} = 0.05$ and $D_{Flowers} = 0.10$

The densities of the mid-level objects were:

$D_{Plants} = 0.78$, $D_{Open\ Area} = 0.15$ and $D_{Cloud} = 0.07$

The density of the high-level object was:

$D_{Garden} = 1.00$

5.4.4 Data Storage and Indexing

When the collection object data of an image is completed, the different level of objects are built, every level has its own type and the objects attributes' values are computed.

The data collected and computed is stored in two different tables in the database. For example: objects' numbers, names, object types, and densities are stored in the FACTS table, as shown in Table 5.9. Other attribute values such as objects numbers, parent's numbers, image name, weight, availability, supporting factors, values, rule significant factors, and weight in image are stored in the FACTSDATA table, as shown in Table 5.10.

Table 5.9 and Table 5.10 are holding the data of the Garden image which have three levels of objects and their attributes. The data shown in these tables have been taken and summarised from the example worked out in Section 5.4, and the uncomputed data (object number, object name, and image name) from the example completed in Section 4.3 of Chapter 4.

Once we input all of the objects' information listed in Table 5.9 and Table 5.10, the search engine is ready to perform a search on the database.

Table 5.9 FACTS Table Contained Data

Object ID	Object name	Type ID	Density
001	Trees	1	0.65
002	Grass	1	0.05
003	Flowers	1	0.10
004	Open Area	2	0.15
005	Cloud	2	0.07
006	Plants	2	0.78
007	Garden	3	1.00

Table 5.10 FACTSDATA Table Contained Data

FID Child	FID Parent	Image Name	Weight	Incl	Avail	Value	Rsf	Wii
001	006	001.jpg	0.396	0.96	1	0.380	1	0.193
002	006	001.jpg	0.229	0.73	1	0.218	1	0.146
003	006	001.jpg	0.305	0.74	1	0.226	1	0.149
004	007	001.jpg	0.487	1	1	0.487	1	0.487
005	007	001.jpg	0.025	0.05	1	0.001	1	0.025
006	007	001.jpg	0.487	1	1	0.403	1	0.487
007	007	001.jpg	1	1	0.486	0.891	0.545	1

5.5 Summary

This chapter explains the approach process of image indexing and its advantages. An explanation of the way of indexing the image components in a suitable matter is described. We explained that our method is based on the objects that contain several atomic indices, which will be able to define intermediate indices and the intermediate indices will be able to define higher indices and so on. The benefits of this method are highlighted. Via examples the hierarchy and object's types is shown. We explained the database structure, the relation schema, the primary keys and other attributes in the created tables etc.

This chapter organizes the indices in such a way that it will facilitate fast and easy retrieval for a query. It explains our indexing model and its sections, each with its own role. In an example a summary of the different object's attribute values are listed and it is shown how these values are stored on different tables in a database. The steps of entering the data in the database are listed and examples of this data are shown inside the different tables.

Our approach provides a new and innovative way to specify and view visual data across the whole spectrum of applications that involve the processing of visual data. This approach has significantly increased the performance of the retrieval algorithm, as measured by the proportion of actual retrieved images versus the total number of images that satisfy the retrieval criteria, the performance optimisation has also been looked into throughout this chapter.

Chapter 6

Image Retrieval Paradigm

6.1 Introduction

Our goal in this chapter is to provide a methodology that concentrates on matching images or subparts of it, defined in a variety of ways, in order to find particular objects.

The image retrieval service should be an open standard, requiring no centralized control, allowing any one to access an image service. Security and authentication issues, including limitation access to proprietary images, will be addressed initially; all search images are initially assumed to be public. The interface used in this research for image retrieval is simple to implement and use for simple image services, but capable of supporting more complex services.

Searching images is organized in a way that will be easy for the user to understand, easy to access the system and retrieve the desired images. Many users are from different backgrounds hence this system will allow them to easily retrieve the images they desire even if their English language is not the same as the language used in the database using the thesaurus explained later on in this chapter.

6.2 Image Retrieval

Many aspects of image retrieval are used in this research in order to satisfy all types of user needs. The first aspect of image search includes displaying images that may contain specific objects; the user can retrieve images that contain specific features. This aspect itself has many methods, “Joint Objects Search”, where the displayed image may contain specific objects together, the “Separate Objects Search”, where the displayed images may contain one or many of the specified objects and the “Backward Objects Search”, where the users search for objects that may be contained within a specific image.

The second aspect of image search is an option which allows for a specific object to be included or excluded from retrieved images. The user can retrieve images without specifying any object; or images containing or not containing various objects. Images may also be retrieved with specific weight levels of objects, availabilities or densities.

Specifying supporting objects will help to reduce ambiguities, and retrieve the most relevant images as explained more thoroughly later on in this section.

The object's weights, supporting factors, availabilities, densities, and other attributes, used in different search methods, provide users with a means to give priorities to selected query components in order to optimise the results of the query. Applying the values of the higher-level object's weight, for example, will increase the value of the object recall rate. Users may want to further tailor the query results by widening the scope of the result or narrowing the result.

The approach used is to integrate our concept-based image indexing and retrieval method with the use of many factors; these factors are listed and then explained below.

- The supporting objects, which help to reduce the ambiguity and retrieve the most relevant images.
- A fuzzy expert system.
- The thesaurus, which is used when the entered object name is not found in the database.

Supporting Objects

When the user is searching for an object, they enter the object name and then submit their search. Some non-relevant images may be retrieved. To avoid these images from being retrieved, the user can use supporting objects that can be either included or excluded from the displayed images.

Using a supporting object such as “Smoke”, which has a high value supporting factor for Plant (as in Factory) and low value of supporting factor for the Plant (as in Plant life), will help to reduce the ambiguity, and retrieve the most relevant images.

Below we illustrate the supporting objects and how they can be used to support the desired images. The images retrieved depend on which supporting objects are used and their relevance to the specific images.

Table 6.1 Supporting Factors

	Existence of Supporting Object (S.O.) % in Image				
Image Name	Smoke	Trees	Workers	Pathway	Forklift
Plant (Factory)	70	15	20	3	72
Plant (Plant Life)	10	95	0	50	0

Referring to the supporting objects in Table 6.1, if the user enters Plant as the object name, all images of Plant, if it is Plant, “Factory”, or Plant, “Plant life”, will be retrieved. When using supporting objects like “Smoke” and “Workers”, the retrieved images will be only that of the Plant “Factory” since the supporting existence values of Smoke and Workers is high for Plant “Factory” and is either absent or very low for Plant “Plant Life”. Entering Plant as the object name and using the supporting objects “Tress” and “Pathway”, which have a high existence value in Plant “Plant life” and a low existence value in Plant “Factory” will results in only Plant “Plant Life” images being retrieved and displayed.

All other methods of image retrieval explained later on in this chapter are additional features for the supporting objects. One method concerning image retrieval having the entered object name with supporting objects is the varying of weights or densities of the supporting objects. Another method is to retrieve images which exclude specific objects which aid narrowing down the results to optimise the search.

Linguistic Variables Notations and Ranges

In this research, the fuzzy logic is used in order to determine the Linguistic variable notations of the objects' weights, availabilities and densities as well as some other object attributes and then facilitates their ranges. The linguistic variable notations and their numerical ranges are used for image retrieval. Table 6.2 provides an example of the approach to the linguistic variable notation of one of the objects' attributes including an example of the overlapping ranges.

Table 6.2 Linguistic Variables Notations and Ranges

Linguistic variable: objects' attributes		
Linguistic value	Notation	Numerical range
Low	L	[0.00 - 0.25]
Medium	M	[0.20 - 0.45]
High	H	[0.40 - 1.00]

The Thesaurus

The Thesaurus is used in this research in cases where the user searches for an objects and the entered name is not found in the database. The system will check the entered name in the thesaurus and will search the alternative word or sentence in our database having the same meaning. All image names matched in the thesaurus will be displayed providing they satisfy the other conditions like the supporting object names as their features, similar to that of a normal image search. Figure 6.1 illustrates the thesaurus cycle.

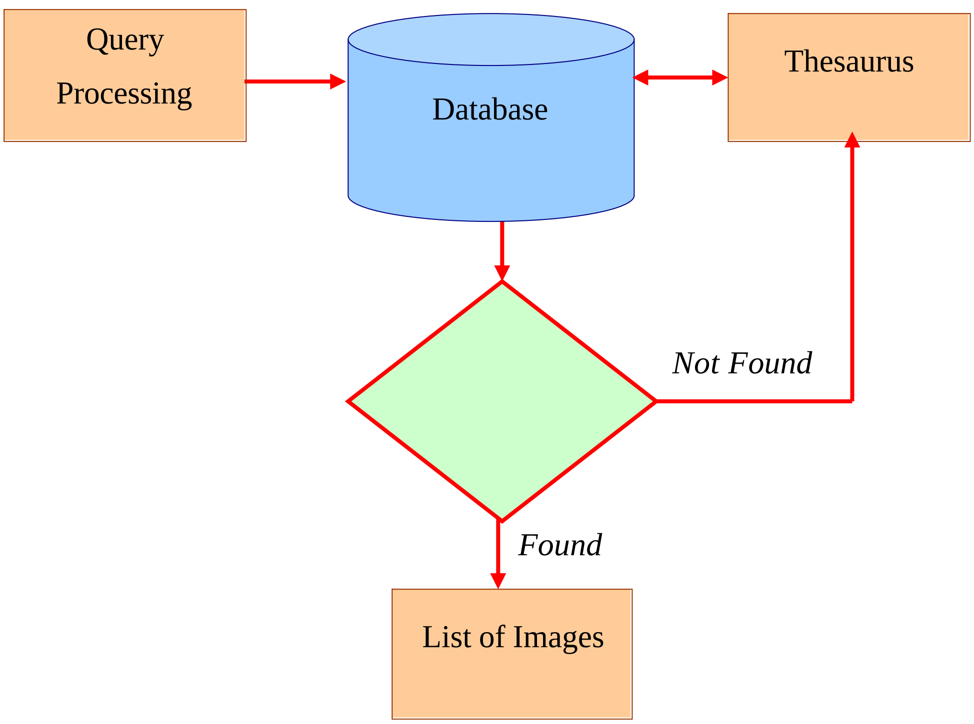


Figure 6.1 Thesaurus Cycle

6.3 Retrieval Algorithm

As mentioned in Section 6.2 of this chapter, our retrieval algorithm consists of two main aspects. The first aspect is used to retrieve images that may contain specific objects, while the second is used to retrieve images containing or excluding specific objects in addition to object features (weights, availabilities, densities etc)

Figure 6.2 contains the Retrieval Modules; the figure illustrates the algorithm needed to retrieve images from the database.

The Retrieval Module consists of four main parts, every part of this module has it own role, which are:

- User Interface
- Data Processing
- Output (List of Images)
- Feedback

The aspects of the Retrieval Module are explained in detail below.

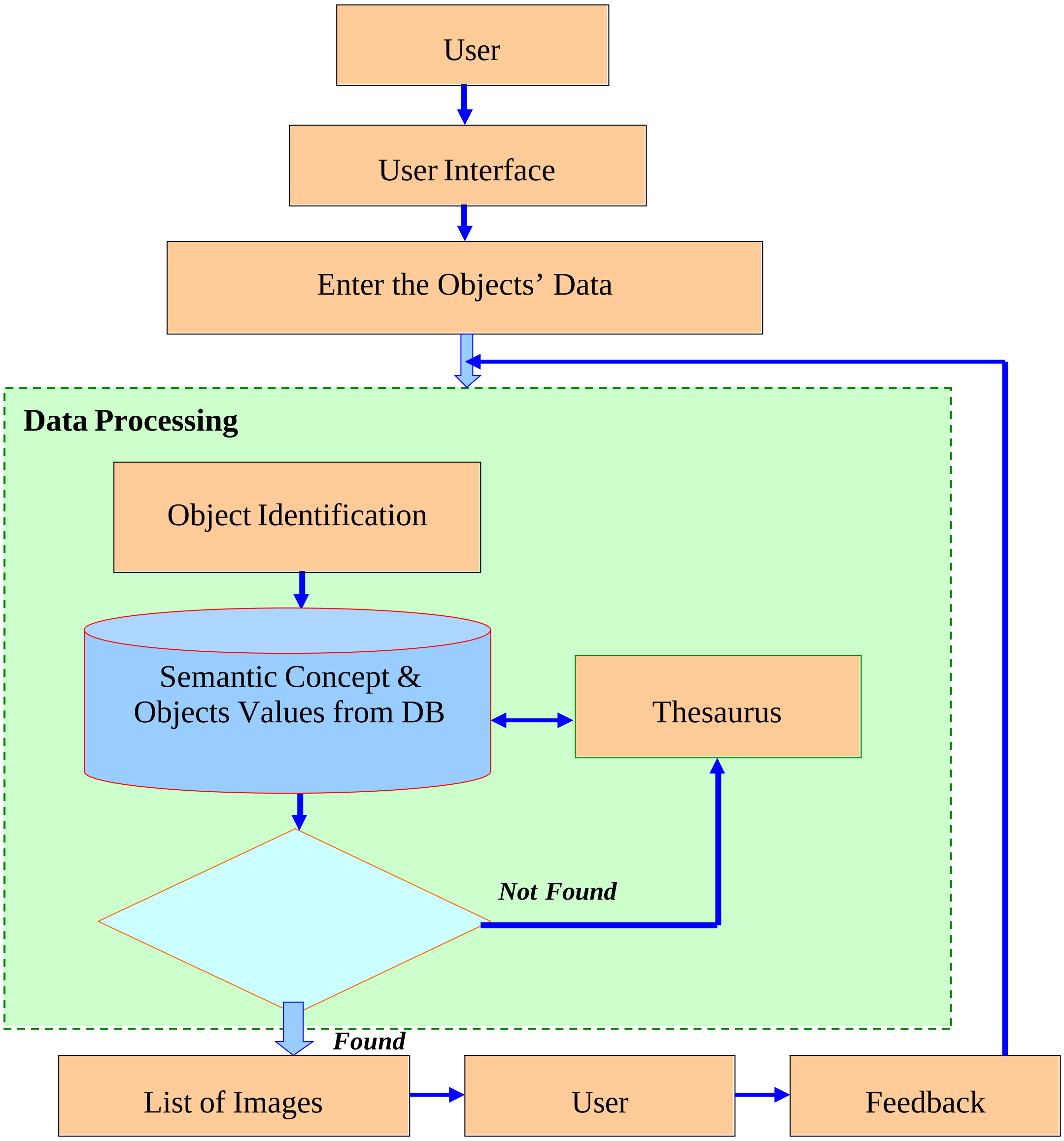


Figure 6.2 Retrieval Modules

6.3.1 User Interface

- Image search consists of two main aspects, every aspect contains many methods.
- The user enters the objects names.
- The user must enter at least three letters in the field of the first object name.

- When the user is attempting to retrieve certain images from the database, the user can enter up to three objects, the first object is mandatory while the other two are optional.
- The same technique is used when attempting to retrieve images from the image collection.
- The user also has the option of entering the objects name along with supporting objects to be included or excluded from the retrieved images, reducing the results to more relevant images.
- Another option of image search consists of the user entering the objects name, as well as the objects weight and availability for a more precise search. The user may also enter supporting objects and their densities in order to narrow the displayed images and retrieve more accurate results.
- All these methods will be explained in detail in Section 6.4 of this chapter.

6.3.2 Data Processing

- When a search is submitted, the systems checks the database for images corresponding to the entered object name, if found, the object will display all relevant images. If the entered name is not found in the database, the system searches the database for synonyms of the entered object name. When these are found, the system will display all images matching the new object names found in thesaurus having the equivalent meaning.

- Checking the names of the supporting objects follows the same process as for the objects names.
- The values of other data (supporting factors, weights, availabilities and densities) of the supporting objects are also checked as well to optimise the result of the search and retrieve more precise images.

6.3.3 Output (List of Images)

- When the data processing is finished, the process is finalised with the relevant images being displayed in an organised table.
- The images are displayed in pages, enabling the user to select a specific page number by selecting the appropriate link.
- Every image displayed is accompanied by its data (image name, weight, etc, depending on the search method used).
- The user can enlarge any image by clicking on it.

6.3.4 Feedback

The search may be refined online according to the user’s indications of relevance, meaning the user can obtain more relevant images.

6.4 Image Retrieval Procedure/Methodologies

Object searching may approach image retrieval differently. In this research we used two major aspects of image retrieval and these aspects are explained below.

1. The first aspect is used for searching images that may include specific objects.
2. The second aspect is used for searching images that include or exclude specific objects; objects of different weights, availabilities and densities.

The image search hierarchy and all aspects and their methods are shown in Figure 6.3.

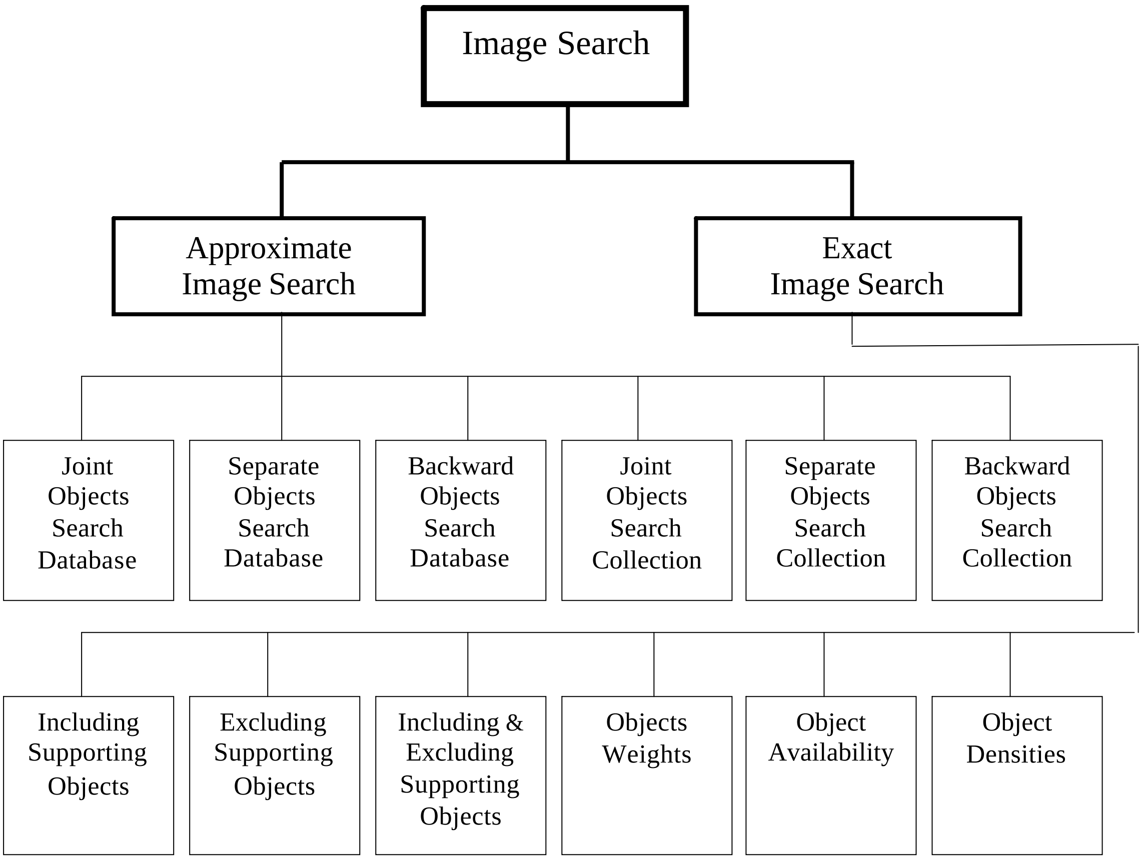


Figure 6.3 Image Search Hierarchy

6.4.1 Approximate Image Search

The Approximate Image Search is an option which retrieves images that may contain specific objects. However, some of the retrieved images may not contain these specific objects but objects that are related to those entered in the search.

This aspect of image search begins when the user enters the specific object names in the appropriate field. When the user submits the search, a list of images is displayed using an approach which determines images that are likely of including this object. An example of this approach is demonstrated below.

Suppose that the user is searching for images with the object “Flowers”. The user enters the object name “Flowers” and then submits the search.

Suppose that there are three categories of images in the database containing “Flowers” as follows:

- 77% of the “Garden” category contains the object “Flowers”.
- 28% of the “Wedding” category contains the object “Flowers”.
- 20% of the “Party” category contains the object “Flowers”.

The system will index the images which may contain the object “Flowers” due to the supporting factor value of the object “Flowers” in these categories from the high to the low value as shown in Table 6.3. All images in the Garden category are indexed first without varying between the image containing and not containing the object "Flowers" in this category. The second category indexed is the Wedding category, and the third category is the Party category, and so on.

Table 6.3 Image Index May Contain "Flowers" Object

Category Name	The Images in Different Categories may contain the Object of Flowers									# of images in a category
Garden	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10	66
	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20	
	G21	G22	G23	G24	G25	G26	G27	G28	G29	G30	
	G31	G32	G33	G34	G35	G36	G37	G38	G39	G40	
	G41	G42	G43	G44	G45	G46	G47	G48	G49	G50	
	G51	G52	G53	G54	G55	G56	G57	G58	G59	G60	
	G61	G62	G63	G64	G65	G66	-	-	-	-	
Wedding	W01	W02	W03	W04	W05	W06	W07	W08	W09	W10	56
	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	
	W21	W22	W23	W24	W25	W26	W27	W28	W29	W30	
	W31	W32	W33	W34	W35	W36	W37	W38	W39	W40	
	W41	W42	W43	W44	W45	W46	W47	W48	W49	W50	
	W51	W52	W53	W54	W55	W56	-	-	-	-	
Party	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	60
	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	
	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	
	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	
	P41	P42	P43	P44	P45	P46	P47	P48	P49	P50	
	P51	P52	P53	P54	P55	P56	P57	P58	P59	P60	

The matched images are displayed in a paged table, every page containing 20 images, and the user could go through these pages by clicking on the page number located above the displayed images.

A category containing some images with the object “Flowers” may not be displayed if the supporting factor value of the object Flowers in this category is too low, due to the low possibility of relevant images within this category.

Another example illustrating the categories containing specific objects is explained below:

Suppose that there are four categories (C1, C2, C3 and C4) that may contain the object “Trees” as shown in Figure 6.4, where the *x*-axis represents the various categories containing some images with the object “Trees”, and the *y*-axis is representing the object’s supporting factor’s values as a percentage.

When the user searches for objects containing Trees, these categories are indexed as follows: The category “C1” first, category “C2” second, category “C3” third and category “C4” fourth. These categories are indexed according to their probability of containing the “Trees” object.

- Category “C1” contains the Trees object of percentage of 95%
- Category “C2” contains the Trees object of percentage of 80%
- Category “C3” contains the Trees object of percentage of 50%
- Category “C4” contains the Trees object of percentage of 30%

The system indexes these categories from highest to the lowest level of the Trees supporting factors’ values as in Figure 6.4.

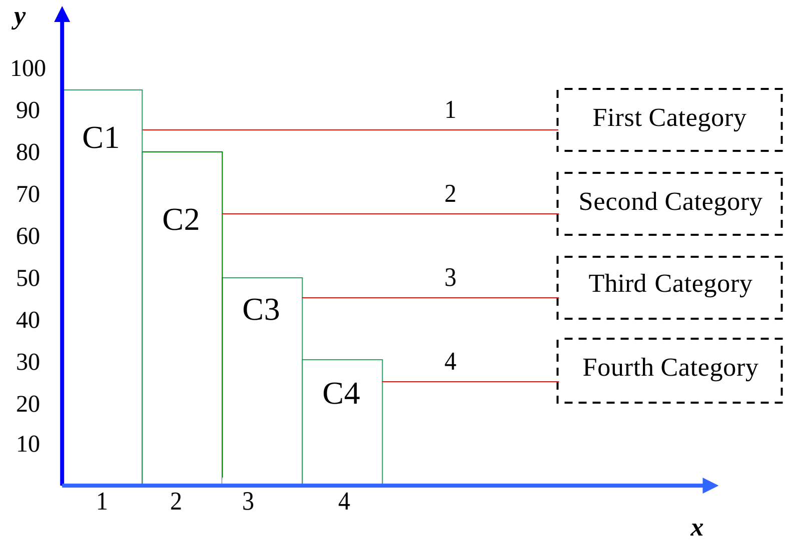


Figure 6.4 Objects in Categories

To show how the system displays the images which may contain the entered objects, the following example explains the selection process. Suppose that all images in the database fall in the positive parts of the ***x*** and ***y*** axes. Three sets of images are shown, the set of all images in the database (A), the relevant set (B) and the retrieved set (C) as shown in Figure 6.5

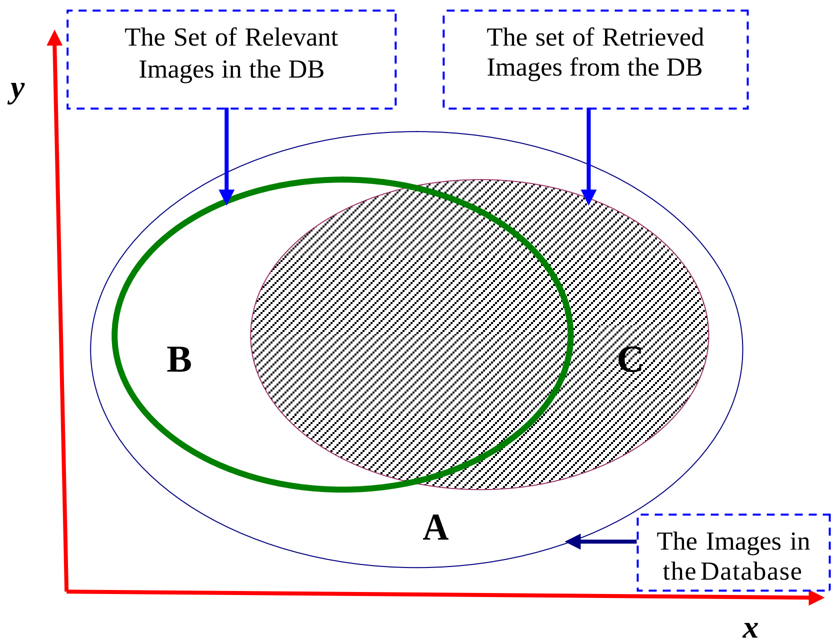


Figure 6.5 Relevant and Retrieved Images

When the user searches for images that may contain a specific object, a set of images is retrieved. Some of these images are relevant and some others are irrelevant. The set which is not retrieved may contain both relevant and irrelevant images as shown in Figure 6.6

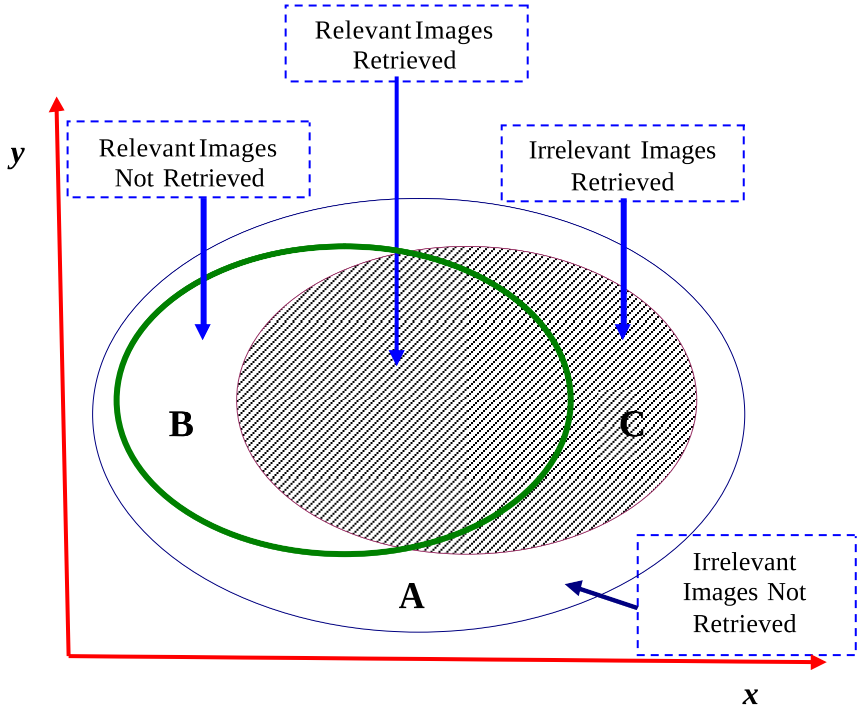


Figure 6.6 (Relevant & Irrelevant) Retrieved and Not Retrieved Images

The Query Design of object retrieval

The query model is important in conjunction with the data model in order to complete the framework of a database. As explained earlier in this chapter, two categories of queries are needed, the first query is referring to Boolean queries, and the second is exact query in nature. In the first category of queries the results are ranked / indexed, in the second query category, the results have to be very precise.

Query of Approximate Image Search

The user interface for Approximate Image Search aspect contains three fields; the number of entered objects in these fields is optional. The search could be used to match the images which may contain all the entered objects, as in Joint Objects Search; or it could retrieve images with any of the entered objects, as in Separate Objects Search.

Joint Objects Search Query Design

This method of image search is used to retrieve images that contain multiple objects, both within all images retrieved. All images that satisfy the search are displayed. At the end of the application a filter is used to avoid image duplication as a result of multiple objects being entered matching the same image. Two tables need to be accessed (FACTS and FACTSDATA), which hold the image data. Below is the query design for the Joint Objects Search shown in Figure 6.7, the queries of other search will be similar with little differences.

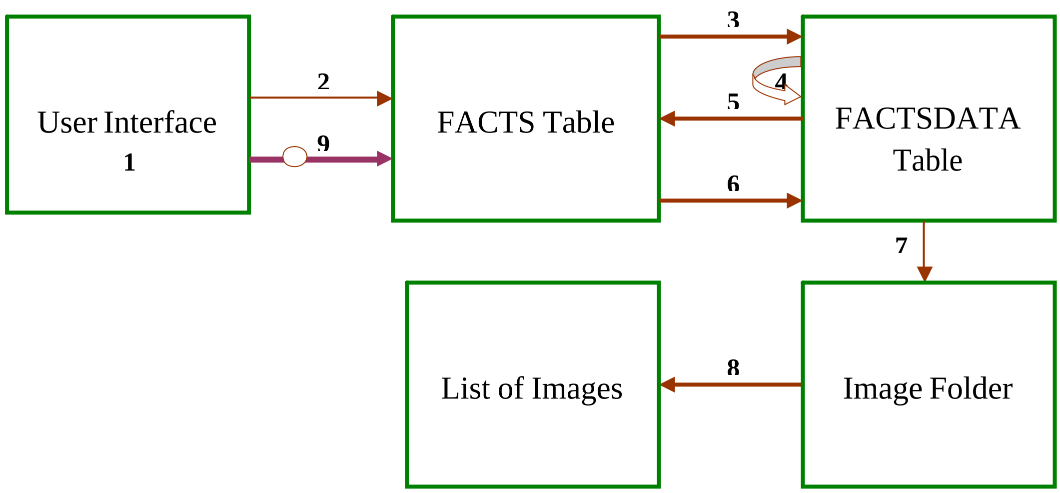


Figure 6.7 Joint Objects Search Query Design

The Query steps are as follows:

1. Enter the objects names.
2. Find the objects names in FACTS table.
3. Find the supporting factor's values in the FACTSDATA table where object numbers FID in FACTS table = FID-Child in FACTSDATA table and the objects names in FACTS table = the entered objects names.
4. Get the FID-Parent from FACTSDATA table where FID in FACTS table = FID-Child in FACTSDATA table and the objects names in FACTS table = the entered objects names.
5. Get the objects names from FACTS table where FID-Parent numbers in FACTSDATA table = FID numbers in FACTS table.
6. Get the image address from FACTSDATA table where FID numbers of the found objects in FACTS table = FID-Child in FACTSDATA table.
7. Search all images having the objects name in the image folder.
8. Display all the matched images.
9. Loop for the higher objects using the found objects names if possible as in step 2 to step 6 (○ represents a loop).

The query steps are summarised in two SQL queries in addition to many functions used in the specific program applied for this method of image retrieval. The frame of the SQL queries used for the Joint Image Search method are listed below, which need some adjustment when applied in the program.

```
Image1Query =
AND factsdata.imagename IN
(SELECT     distinct (imagename)
FROM        azzam_asp.factsdata factsdata, azzam_asp.facts facts
WHERE       facts.fid = factsdata.fid_child
            AND UPPER(facts.fname) like imageName1%
            AND  image2Query, image3Query)

Filter1Query =
factsdata.fid_child IN
            (SELECT     facts.fid
            FROM        azzam_asp.facts facts
            WHERE       UPPER(facts.fname) like imageName1%)

FinalSQL =
(SELECT     facts.fname, facts.fid, factsdata.imagename
FROM        azzam_asp.facts facts, azzam_asp.factsdata factsdata
WHERE       facts.fid=factsdata.fid_parent and facts.fname in
            (SELECT     facts.fname
            FROM        azzam_asp.facts facts, azzam_asp.factsdata  factsdata
            WHERE       facts.fid=factsdata.fid_parent
                        AND factsdata.incl>=0.15
                        AND (filter1Query, filter2Query, filter3Query )
                        AND (image1Query, image2Query, image3Query)
                        ORDER BY factsdata.imagename asc))
```

Separate Objects Search Query Design

This method of image retrieval utilizes the entered objects separately. The query design is quite similar to the design used in the Joint Object Search; the difference is that there is an extra loop for this method since the application is used in the retrieving objects separately (the loop "number 10" of Separate Objects Search Query Design shown in Figure 6.8). The query design is shown in Figure 6.8.

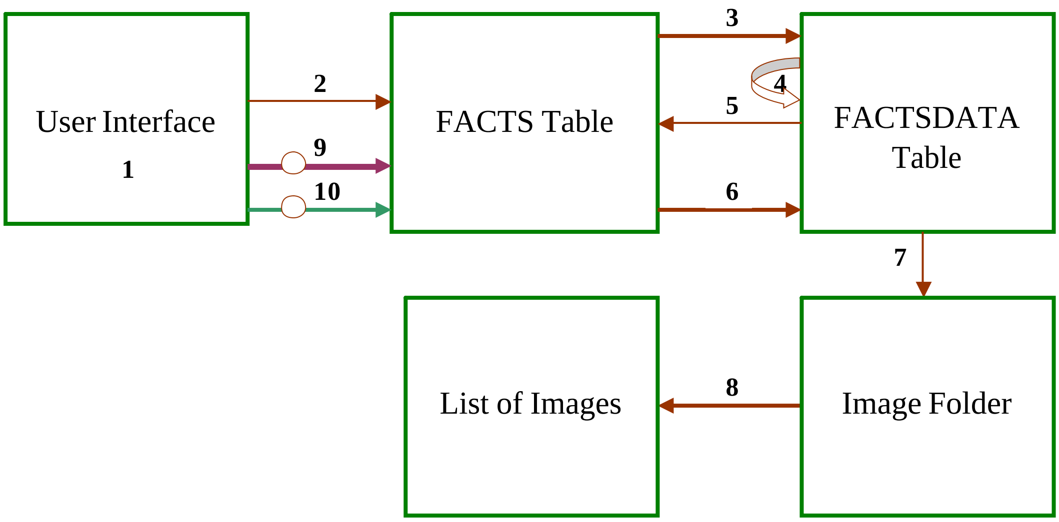


Figure 6.8 Separate Objects Search Query Design

Backward Objects Search

This method searches for objects which make up a higher-level object, for example, the user may need to search for objects which constitute the object "Garden". Up to 50 different component objects could be found in the object Garden. We could selectively display images containing some of these objects providing they have certain supporting factor values. The query design in this case is similar to the Joint Objects Search method.

Collection Object Search

There are three methods for this object search; it works similar to the Joint Object Search, Separate Object Search and Backward Object Search. The difference is in the final query. The selected images are displayed from the image collection. The following query shows the final query of the first method, Joint Object Search from the collection.

```
FinalSQL =
(SELECT imagecollection.fid, imagecollection.fname, imagecollection.imagename
FROM    azzam_asp.imagecollection imagecollection, azzam_asp.factsdata
        factsdata
WHERE   imagecollection.fname in
            (SELECT facts.fname
            FROM        azzam_asp.factsdata factsdata, azzam_asp.facts facts
            Where       facts.fid=factsdata.fid_child
            AND         factsdata.incl>=0.15  AND image1Query)
            ORDER BY imagecollection.imagename asc)
```

6.4.2 Exact Image Search

In this aspect, the user is given many methods on how they would like to retrieve an image. They can make a search using only the image name; this method retrieves all images that match the same name or synonyms of the enquiry made by the user. The user can also make a search using supporting objects. This happened when the user requires a particular object in the image and only images that match the name searched by the user that include and/or exclude the specific supporting objects, depending on the user’s specifications, will be displayed. This technique can also be used when searching for images including objects of certain weight levels, objects of certain availability levels, and images including objects of certain density levels. Each application of Exact Image Search is explained below.

Image Retrieval Including Supporting Objects

Due to the use of the supporting objects, one of several number of image groups are retrieved depending on the entered object name, the entered supporting object’s names, the number of supporting objects and many other factors.

Image Retrieval Including One Specific Supporting Object

When using this method of image search, the retrieved images are those which correspond to the entered object name that must also include the entered supporting object. An example of this method is shown in Figure 6.9.

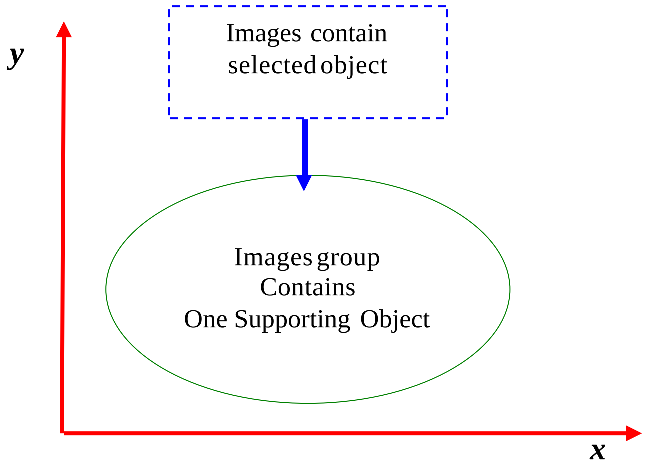


Figure 6.9 Images Including One Specific Supporting Object

Image Retrieval Including Two Specific Supporting Objects

The second group of images is displayed when the user enters the object name and enters the names of two specific supporting objects as components of the desired images, and then submits the search. The displayed images retrieved by the system will be images that correspond to the entered image name that also include both entered supporting objects. The displayed images fall in the intersection between the groups containing the first and the second supporting objects as shown in Figure 6.10.

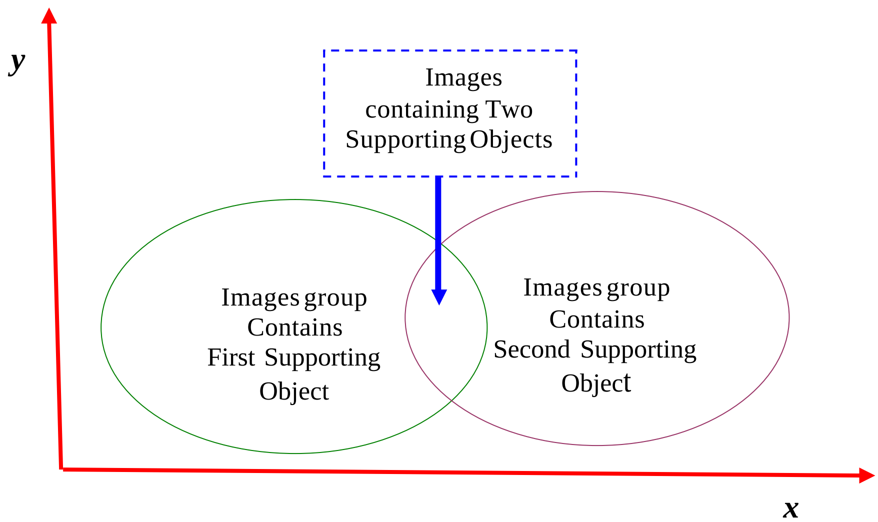


Figure 6.10 Images Including Two Specific Supporting Objects

Image Retrieval Including Three Specific Supporting Objects

The third group of images is displayed when the user enters the object name and enters the names of three supporting objects as components of the desired images in the search. The group of images displayed will contain images that correspond to the entered object name that also include all three entered supporting objects. The retrieved images will be the intersection between the groups containing the first, second, and third supporting object as shown in Figure 6.11.

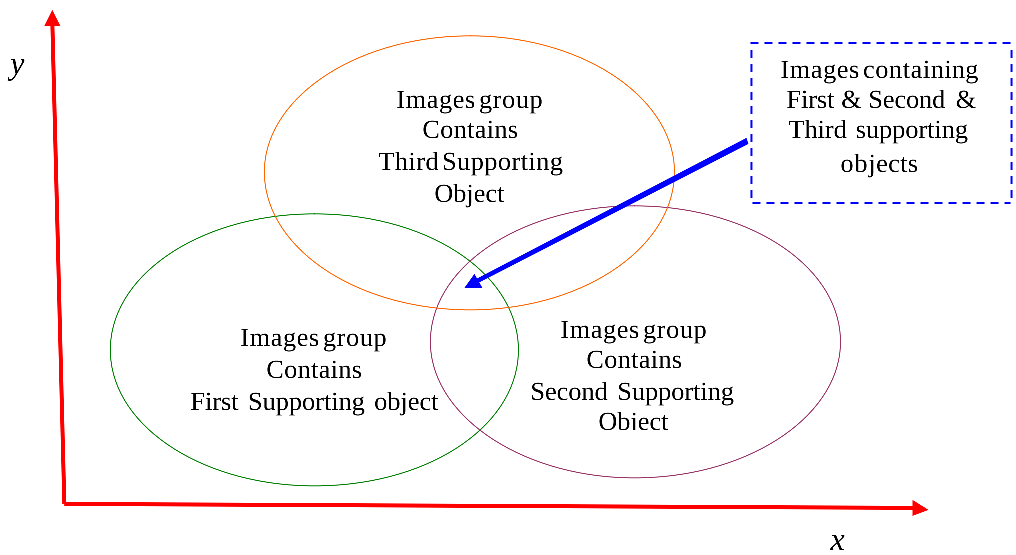


Figure 6.11 Images Including Three Specific Supporting Objects

If the user makes any changes to the name of any of the previously entered supporting objects, a new group may be displayed differing from the previous group as the intersection between the groups containing the supporting objects varies. In other words, any changes to any of the entered supporting objects may affect the number of the displayed images as well as which specific images are displayed. If the number of supporting objects is changed, the number of displayed images is also most likely going to vary.

Using more than three supporting objects will follow the same procedure as when using three in all cases of images retrieval aspects.

Image Retrieval Excluding Supporting Objects

The same techniques used for image retrieval using supporting objects are used when retrieving images specifically excluding certain supporting objects, differing with retrieved images not including the specified supporting entered.

Image Retrieval Excluding One Specific Supporting Object

The group of images that are displayed when an object name is entered followed by a supporting object, which is not to be included in the image will include all image from the databases that correspond to the entered object name that do not have the entered supporting object as one of it is components. This is shown in Figure 6.12.

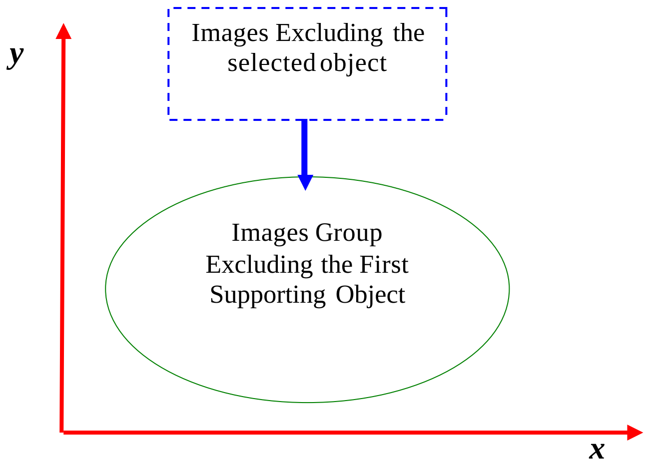


Figure 6.12 Images Excluding One Specific Supporting Object

Image Retrieval Excluding Two Specific Supporting Objects

The second group of images is displayed when the user enters the object name and enters the names of two supporting objects to be excluded before submitting the query. This group will display images that match the object name but exclude the entered supporting objects; the images fall in the intersection between the groups which do not contain the first and the second supporting objects as shown in Figure 6.13.

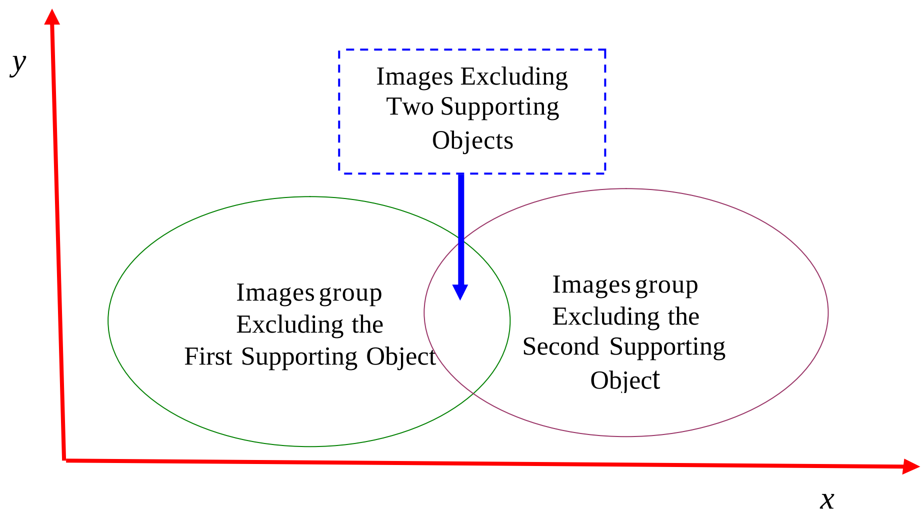


Figure 6.13 Images Excluding Two Specific Supporting Objects

Image Retrieval Excluding Three Specific Supporting Objects

The third group of images is displayed when the user enters the object name and enters the names of three supporting objects to be excluded and then submits the query. This group will display images that match the object name but exclude the entered supporting objects, these images fall in the intersection between the groups that do not contain the first, second, and third supporting objects as shown in Figure 6.14.

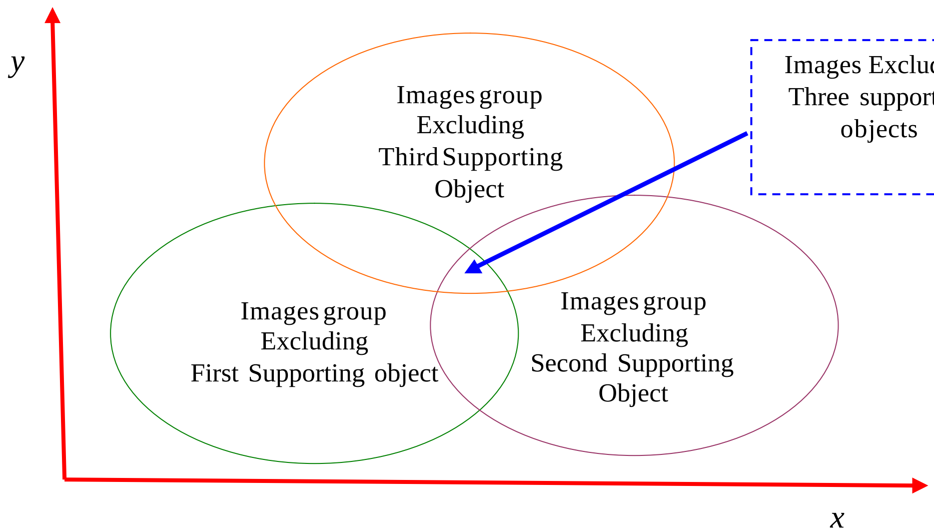


Figure 6.14 Images Excluding Three Specific Supporting Objects

Image Retrieval Including and Excluding Supporting Objects

The same techniques used for image retrieval using supporting objects are used when retrieving images specifically including certain supporting objects and excluding other certain objects together, where the displayed images must contain the first number of entered supporting objects and exclude the second group of entered supporting objects.

Image Retrieval using Objects' Weights

Since it is inappropriate and difficult for the user to use precise numerical values when attempting to retrieve a desired image, for example, the object's weight in a certain image is equal to 0.33, all possibilities of weight values that are stored in the database are replaced with three easy linguistic variable notations and their ranges as shown in Table 6.4. This way the user can select any of them describing the supporting object weight's levels, where every notation has a weight value between two values.

Table 6.4 Linguistic Variable of Object's Weight in Image

Linguistic variable: Object's weight in image		
Linguistic value	Notation	Numerical range
Low	L	[0.00 - 0.20]
Medium	M	[0.15 - 0.30]
High	H	[0.25 - 1.00]

The object's weight takes one of three values as below:

$$W_i = \begin{cases} [0.00]\,to\,[0.20]\,\{\ \ for\ low\ \ weight \\ [0.15]\,to\,[0.30]\,\{\ \ for\ medium\ \ weight \\ [0.25]\,to\,[1.00]\,\{\ \ for\ high\ \ weight \end{cases}$$

The above values are considered to be reasonable in the present context. In general, of course, these can be fine-tuned or modified, which may be of the form:

$$W_i = \begin{cases} [0.0]\,to\,[\,\alpha\,]\{\ \textit{for low weight} \\ [\,\beta\,]\,to\,[\,\gamma\,]\{\ \textit{for medium weight} \\ [\,\delta\,]\,to\,[1.0]\{\ \textit{for high weight} \end{cases}$$
$$\textit{Where } 0 < \beta \le \alpha < \gamma < 1, \quad \beta < \delta \le \gamma < 1$$

Due to the use of the different objects' weights to be included in the desired images, several kinds of image groups could be retrieved. If the user uses certain levels of object weights to be included within the displayed images, one of three groups could be matched depending on the objects weights' levels selected as shown in Table 6.5.

Table 6.5 Probability of Selected Groups

Objects weights inside the desired images		
Group No.	Notation	Objects weight
1	L	L + M+ H
2	M	M + H
3	H	H

Due to the image selection criteria, every image must contain some objects of a high-level weight, which is the essential objects, so we cannot retrieve images that contain objects with low-level or even mid-level weights only.

Figure 6.15 shows the three levels of objects' weights. Low-level weights consist of objects of all levels hence all the images. Mid-level weights consist of medium and high-level objects. High-level weights are those containing only high-level objects.

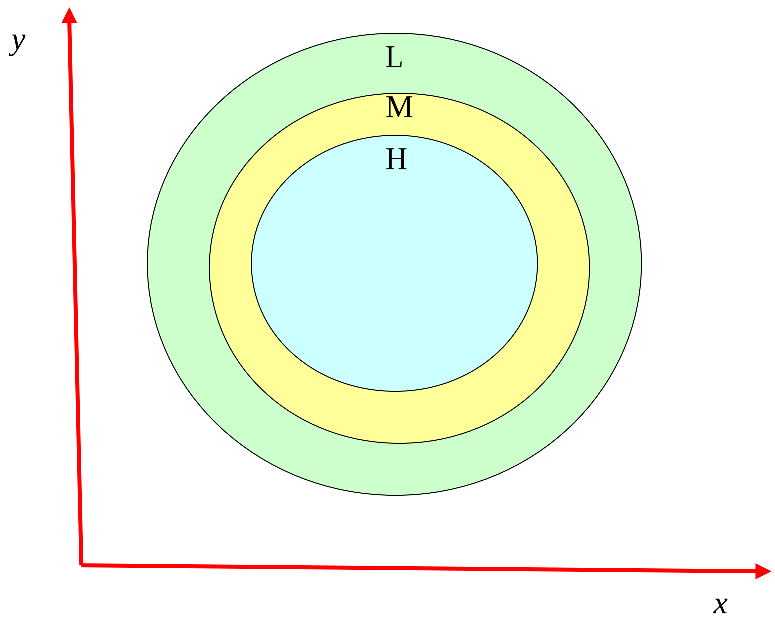


Figure 6.15 Three Levels of Objects Weights within Images

From Figure 6.15, it is obvious that within the “L” level, the number of images would usually be more than the images within the “M” level, and the number of images within the “M” level would usually be more than the images within the “H” Level. The three groups of images that could be retrieved when searching for images of different levels of objects are dependent on the object name and the selected level of objects.

Image Retrieval Including Low-Level Weights

The first group of images is displayed when the user enters the object name and selects the low-level-weights “L” for the objects to be components of the retrieved image. This group will display images that match the entered object name and contain objects of all weight levels (high-level weights, mid-level weights and low-level weights)

Image Retrieval Including Mid-Level Weights

The second group of images is displayed when the user enters the object name and selects the mid-level weight (M). Objects of high-level weights and mid-level weights will be displayed. This request narrows down the list of images to be displayed, eliminating low-level objects.

Image Retrieval Including High-Level Weights

The third group is displayed when the user enters the object name and selects high-level (H) weights for the images. This group consists of the essentials that an image must consist of as well as some other of high-level weight objects to be categorized in those particular retrieved images. This attempts to eliminate all non-required objects located in the image so that the images retrieved are a precise description of what the user is searching for.

Search Images using Availability

Searching images using the object availabilities will follow the same technique of searching images using the object weights. The difference is in the value of the numerical ranges for every notion and the number of levels used. Also when using objects availabilities there is no use of the supporting objects. Table 6.6 shows the notion and its numerical ranges. This method of search applies for the non-primitive objects only, since the primitive objects always have the availability value of one.

The object availability takes one of five values as below:

$$A_i = \begin{cases} [0.00]\ to\ [0.35]\ \ for\ low\ availability \\ [0.30]\ to\ [0.55]\ \ for\ low/mid\ availablity \\ [0.50]\ to\ [0.65]\ \ for\ mid\ availability \\ [0.60]\ to\ [0.75]\ \ for\ mid/high\ availablity \\ [0.70]\ to\ [1.00]\ \ for\ high\ availablity \end{cases}$$

The above values are considered to be reasonable in the present context. In general, these can be fine-tuned or modified.

Table 6.6 Linguistic Variable: Object's Availability

Linguistic variable: Object's availability		
Linguistic value	Notation	Numerical range
Low	L	[0.00 - 0.35]
Low/Medium	L/M	[0.30 - 0.55]
Medium	M	[0.5 - 0.65]
Medium/High	M/H	[0.60 - 0.75]
High	H	[0.70 - 1.00]

Searching images using availabilities starts when the user enters the object name and selects the level of the object availability. The system will check the object name and its range of availability in which it falls under, and then it will display the relevant images in a list.

The greater the value of availability of an object, the more the number of component objects present for the specific object and/or the more amount of component objects with a high repetition value. For example, if a "Garden" object has an availability value of 1; it means that this object contains all its possible component objects, but if it has an availability value of 0.70, it means that the total number of repetition of all objects making up this object divided by the total number of repetition of all component objects of "Garden" in the database is equal to 0.70.

Image Retrieval using Objects' Densities

Searching images by using the objects' densities follows the same procedure as when combining image retrieval using supporting objects. The user enters the object name, the supporting objects and then selects the levels of density for the supporting objects. The value of the numerical range for every notion of the densities is different from the numerical range for every notion of the weights. Table 6.7 shows the densities notion and its numerical ranges.

The object density takes one of three values as below:

$$D_i = \begin{cases} [0.0]\,to\,[0.4] \{ \ \textit{for low density} \\ [0.3]\,to\,[0.7] \{ \ \textit{for medium density} \\ [0.6]\,to\,[1.0] \{ \ \textit{for high density} \end{cases}$$

The above values are considered to be reasonable in the present context. In general, certainly, these can be fine-tuned or modified, which may take on the form:

$$D_i = \begin{cases} [0.0]\,to\,[\alpha] \{ \ \textit{for low density} \\ [\beta]\,to\,[\gamma] \{ \ \textit{for medium density} \\ [\delta]\,to\,[1.0] \{ \ \textit{for high density} \\ \textit{Where} \quad 0 < \beta \le \alpha < \gamma < 1, \ \beta < \delta \le \gamma < 1 \end{cases}$$

Table 6.7 Linguistic Variable: Object's Density

Linguistic variable: Object's density		
Linguistic value	Notation	Numerical range
Low	L	[0.0, 0.4]
Medium	M	[0.3, 0.7]
High	H	[0.6, 1.0]

Image Retrieval Including One Supporting Object with Densities

Searching images using densities starts when the user enters the object name and the first supporting object with its density. Four groups of images may be displayed depending on the selected level of density of the supporting object (L, M, H, and A) as shown in Table 6.8.

Table 6.8 Probability of Selected Groups using One Supporting Object

Level of densities of one Supporting Objects (SO)	
Group No.	Selected Level
1	A
2	L
3	M
4	H

Image Retrieval Including Two Supporting Objects with Densities

When the user applies two supporting objects, one of sixteen groups (4 x 4) may be displayed depending on the selected level of densities of both supporting objects. For example, if no level is selected for both supporting objects, it means all levels (A) will be selected at once for both supporting objects, as in group number 1 shown in Table 6.9 and so on.

Table 6.9 Probability of Selected Groups using Two Supporting Objects

Level of Densities for Two Supporting Objects (S.O.)		
Group Number	1St S.O. level	2nd S.O. level
1	A	A
2	A	L
3	A	M
4	A	H
5	L	A
6	L	L
7	L	M
8	L	H
9	M	A
10	M	L
11	M	M
12	M	H
13	H	A
14	H	L
15	H	M
16	H	H

Image Retrieval Including Three Supporting Object with Densities

For three inputs and one output, the representation takes the shape of an M x N x K cube. If we increase the input to three inputs and one output, we have a fuzzy associative memory (FAM), which enables us to derive sixty-four rules that represent complex relationships between all variables used in the expert system

[HWAR00]. Since images can be understood and interpreted at multiple levels, they can also be indexed at multiple levels.

Table 6.10 contains four groups out of the sixty four possible combinations for the density levels of three supporting objects in conjunction.

Table 6.10 Probability of Selected groups using Three Supporting factors

(Four rows out of 64 rows)

Level of Densities for Three Supporting Objects (SO)			
Group No.	1^{St} SO	2^{nd} SO	3^{rd} SO
1	A	A	A
2	H	H	H
3	M	M	M
L	L	L	L

Searching Process

If the user applies three supporting objects with different densities, one of sixty-four groups may be displayed depending on the user's selection.

When the user enters the object name, the names of the supporting objects and selects their density levels (A, L, M, or H), then submits their search, the system will search the object name in the database. The system will check the name of the first supporting object and its density in the database. The same technique will follow for other supporting objects, when found, a list of images will be displayed which satisfy all condition of the user's query.

6.5 Summary

Our method concentrates on matching the user's desired images, defined in a variety of ways, in order to find particular objects. The use of fuzzy logic and supporting factors lead to improvements of both precision and recall. This approach provides a new and innovative way to specify and view visual data across the whole spectrum of applications that involve the processing of visual data.

Our approach has significantly increased the performance of the retrieval algorithm, as measured by the proportion of actual retrieved images versus the total number of images that satisfy the retrieval criteria.

In this chapter we explained that competent retrieval system should retrieve images that satisfy the user's needs. Tables that contain linguistic variables notion for objects' attributes and their ranges using the fuzzy logic are created.

All methods of image search may contain different number of objects and retrieving images from specific image collection. Using supporting objects is a great advantage in order to be specific when searching for particular images.

The use of the thesaurus in our retrieval system is outlined, which is used when the user searches for images where the object name is not found in our database.

Retrieving images can make use of supporting objects, which are to be included or excluded within the retrieved images. The supporting objects can also have specific weight levels to be included within the retrieved images. Using specific levels of object availability and supporting objects with specific density levels can be further used for a more precise search.

Finally the chapter had explained the retrieval algorithms and the data retrieval in depth, as well as retrieval modules and its components in detail.

Chapter 7

System Implementation

7.1 Introduction

Implicit Concept-based Image Indexing and Retrieval System ICIIR is provides easy access to a collection of images in a database. It uses a simple and comprehensive indexing and retrieval method which facilitates the building of easy to use user and/or staff interfaces. Underlying this is a model of the indexing and retrieval process which allows the user and/or the staff to move between analysis queries, retrieve and browse the images in a natural and flexible manner. The users have the ability to use many options of image retrieval that can be found in the interface, according to their preference.

The ICIIR system was developed to run on a Microsoft Windows-based personal computer. The system was not intended for use by a single group of users but rather developed for use by a large and varied group.

Our method concentrates on matching subparts of images, defined in a variety of ways in order to find particular objects. These ideas have been explained with a variety of examples in this chapter. The use of the fuzzy logic and supporting factors lead to improvements of both precision and recall and is also explained in further detail.

7.2 Environment

In this section we thoroughly explain the programming language used in our project and how the system works. We describe the system specifications software and hardware' and explain the database used "Oracle9i" for storing our data, discussing the issues involved with using the programming language and Oracle9i. Below we explain the components of this environment in detail.

Active Server Pages (ASP)

ASP, the programming language used in this research is a very powerful, easy-to-learn, server-side scripting language. This language enables us to create a dynamic Website, fast and interactive without needing to be concerned with the capabilities of our client's browsers, which becomes a major issue when using client-side scripting such client-side Visual Basic Scripts (VBScript) or client-side JavaScript. Processing of an Active Server Pages is shown in Figure 7.1.

An ASP file has an “asp” extension instead of “htm” or “html” to denote the server-side code to the hosting Web server. Because an ASP file is a text file with the extension “asp” that contains a combination of text, HTML tags, and ASP script commands, we can use any text editor to create or change an ASP file.

ASP runs on four platforms: Personal Web Server (PWS) for Windows 95-98, Window NT Workstation, Internet Information Server (IIS) on Windows NT Servers and Chilli! ASP for running ASP on various Web servers. Each of these platforms has slight differences in functionality, with IIS being the most powerful Web server available from Microsoft.

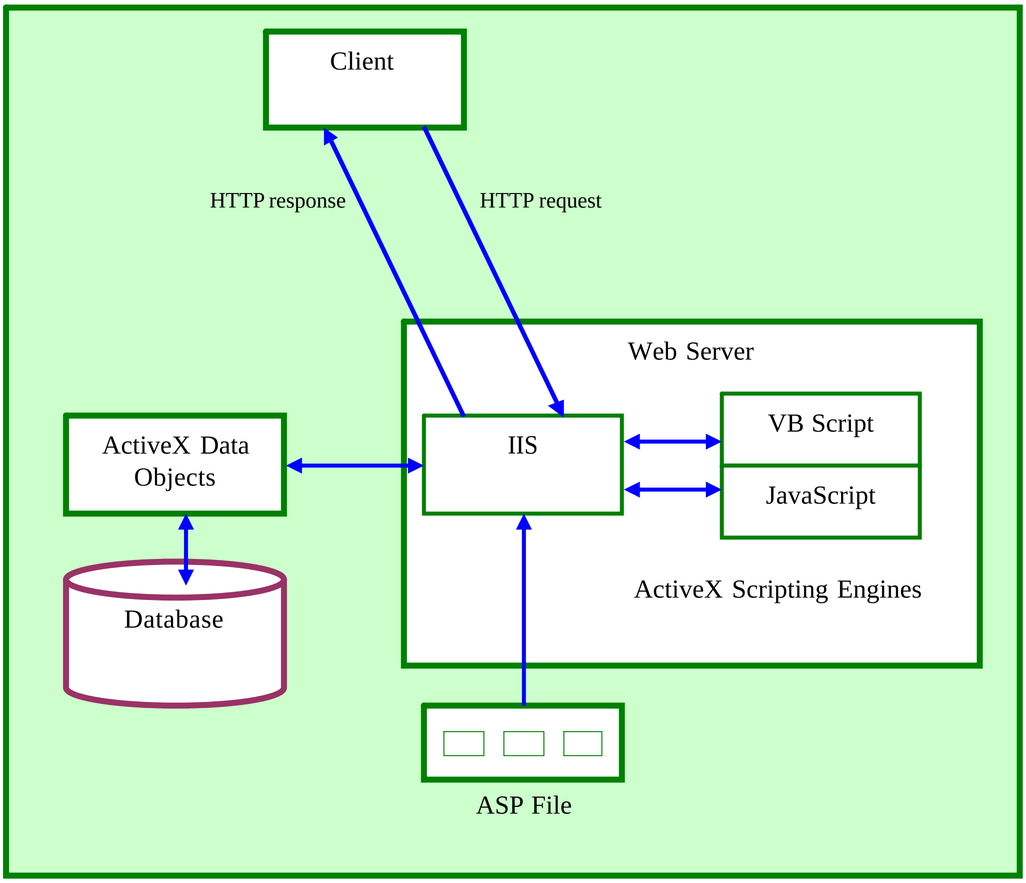


Figure 7.1 Processing of an Active Server Pages

When a client requests a page 'X.asp' from the server, the Web server checks the file extension to see whether a special program (such as the Active Server Page engine) must be invoked to process the request. If there is an "asp" extension, the Web server determines that it should invoke ASP to process this page.

If this page has never been requested before or has been changed since the last request, it must be parsed and the syntax checked and compiled by the Web server. Otherwise, the page might be read from a cache of recently processed pages, which aids in performance. During the parsing process, the HTML and scripting code are separated. IIS determines which scripting engine is responsible for which part of the script and delegates the work of syntax checking and compiling to the proper scripting engine.

After executing the codes using the appropriate scripting engine with resources from the IIS, which is hosting the scripting engines, all objects that the language engine cannot handle are requested from IIS, which is also responsible for handling inputs and outputs for external ActiveX objects that are created and used inside the script. If it is unable to supply the object, an error is generated. Script output and static Hypertext Mark-up the HTML language code in the ASP file and merge them. The final HTML is sent back to the user in an HTTP response.

7.3 System Specifications

The hardware and software used in our project in order for it to operate and function are listed below.

7.3.1 Hardware Specifications

- Intel Pentium IV, 2.8GHz processor
- 80 GB Hard Disk
- 256MB of RAM
- CD-ROM 24X speed and 256MB USB flash drive.

7.3.2 Software Specifications

- Microsoft Windows XP
- Microsoft Internet Explorer for Personal Web Server
- Microsoft Visual Studio.NET and EditPlus 2
- Oracle9i software.

7.4 System Architecture

The overall architecture of the ICIIR system can be divided into a number of managers. Every manager has its different purposes and is used for indexing and accessing the system. The access parts provide a user interface to accept users' instructions. The access part of the system access the image database operating the indexing database to output the retrieval results. The system managers are as follows:

- User Interface (UI)
- Transaction Manager (TM)
- Storage/Retrieval Manager (SRM)
- Query Manager (QM)
- Display Manager (DM)
- Interpretation Manager (IM)

The overall organization of our system is shown in Figure 7.2; the connection between parts describes the dependency of every manager to the others. Every manager of the model is identified and its purpose is explained in details later on in this chapter.

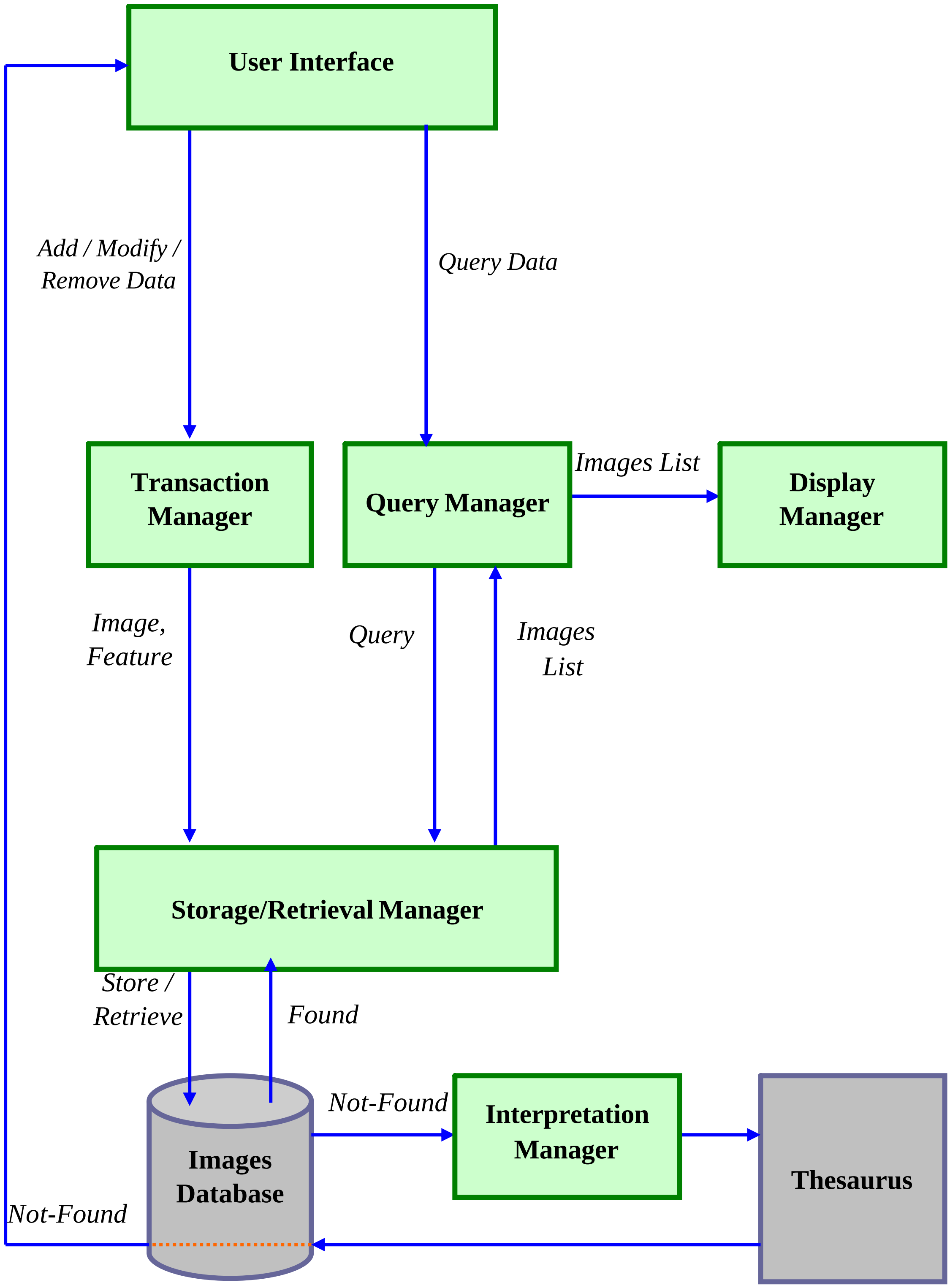


Figure 7.2 Overall Model of the System (System Architecture)

7.4.1 User Interface

The User Interface is used for the image/object search, maintaining the database and enabling the user to contact the administrators and the research group. The User Interface is connected with two managers; the Transaction Manager and the Query Manager. The User Interface is displayed in all pages created by this system; these pages are connected to each others as shown in Appendix A. Every page performs a specific role “Add, Modify, Delete and Display records, etc” and all pages are explained in detail with appropriate diagrams.

7.4.2 Transaction Manager

The Transaction Manager receives requests from the different interface pages, which are used to add, modify and delete records from the different files in the database. To add new records, the Transaction Manager is used to compute some of the data and then transmit the computed and non-computed data to the Storage/Retrieval Manager in order to store it in the database. Some of this data cannot be accepted unless some other adjacent data is entered. For example, the system checks the existence of object numbers in the FACTS file before any further process in the FACTSDATA file are undertaken for the same object number. If the type of any of the entered data does not match the specific type for every field, it will not be accepted. When the Transaction Manager is used to modify and delete records from the different files, it receives the request (the record number to be modified or deleted) from the User Interface and conveys it Storage/Retrieval Manager, which forward it back to the User Interface when a full data record is found, giving users the ability to modify or delete the records. The users are then able to send the modifications of the record or the request of deleting the particular record to the Storage/Retrieval Manager to finalise the process.

7.4.3 Storage/Retrieval Manager

The Storage/Retrieval Manager is connected with two parts of the system (Transaction Manager and Query Manager) in addition to the database. When the Storage/Retrieval Manager receives requests from the Transaction Manager for addition, a modification or deletion of records, the Storage/Retrieval Manager processes the query and sends it to the database for storing, modifying or deleting. The Storage/Retrieval Manager also receives the query from the Query Manager to retrieve data, the Storage/Retrieval Manager processes the query and transmits it to the database and waits for the result. When Storage/Retrieval Manager receives the result from the database it will forward it to the Query Manager.

7.4.4 Query Manager

The Query Manager receives the query data from the User Interface, the query will be parsed and processed by the Query Manager, which will interpret it and then send it to the Storage/Retrieval Manager. The Query Manager then receives the results of the query from the Storage/Retrieval Manager, which will be processed and sent to the database. The Query Manager finalises the process by sending the received result to the Display Manager (if there is any).

7.4.5 Display Manager

The Display Manager's purpose is to receive results of queries from the Query Manager and display the results for the user.

7.4.6 Interpretation Manager

The Interpretation Manager is a connection between the database and the thesaurus. As the Storage/Retrieval Manager sends the query to the database, if the entered object or/and supporting objects names are not found, the entered names are checked in the thesaurus; when found, the system will look for all images having the name found in the thesaurus and processes the query. If no images are found in the database and the thesaurus, a message is sent to the User Interface suggesting no relevant images were found.

The indexing and retrieval of images was implemented to perform different interactive indexing and retrieval strategies based on the use of image indices. The indexing and the layout are divided into many windows as illustrated in the next section.

7.5 System Pages

This section explains all pages used in the system and how to use them. The pages consist of four levels explained in Appendix A.

- The first level is the Home page.
- The second level is the Image Search, Image Maintenance, Contact us, and the Research Group.
- The third level is the sub-options of the Image Search and the Maintenance.
- The fourth level is the sub-option of the third level category (sub-options of Image Search and Image Maintenance's sub-options).

The first page in the system is the home page which explains what every page in the system is, their uses and how the user can implement the particular page.

The image search options are also explained in detail with particular reference to the method of enlarging the image and the differences between options.

An explanation of how the staff and administrators can access their pages to maintain the database through the many options and sub-options is available. Add, delete, modify and display records from different files are explained thoroughly in order for the staff to maintain the system.

Information on how to contact us will also be available as well as information about the research group. Most pages contain notes for easier option selections. These pages are given in the appendix A.

7.6 ICIIR Evaluation and Search Criteria

ICIIR is the research engine used in our thesis. It has concentrated on providing the best search facilities, resisting the temptation to become a portal. The different options of image search make the search favoured by many users; the user can select the preferred option of image search that they desire. In this section we examine our system using the evaluation criteria, and explaining all of them in details.

7.6.1 System Scope

Our database contains approximately 1000 images. These images are stored in one folder, not in separated categories in order for images to be retrieved from different angles rather than storing the same image several times. Our method of image indexing saves more space and prevents the duplication which could happen at some stage. Updating, deletion and modification can be done when needed, and all of them will be automatically done when any new images are inserted at a later stage (when the content of images is automatically determined).

7.6.2 System Search Options

There are many options of searching images in our system, for several of them, we use the filtering mechanisms. There are no images in the image collections that are to be excluded for a particular group of people. We could increase the Image Search options if required at any time in order to satisfy the user's requirements.

7.6.3 System Performance

Since our database contains a small number of images, the results are quickly returned. It is possible to view all the results and the results always match the query. The image quality is good and is displayed in a table. Queries are very accurate and since the administrator is always logged on, there are no dead links.

7.6.4 System Presentation

The results are presented as thumbnails; every image contains some information relating to it and depending on the query. Images used in our system are for research purposes and are collected from different free sources enabling anyone to use them as they are not restricted by copyright laws.

7.6.5 System Support

In every search option, there is some information to show how to use the search; in addition there are two different links, one for the administration and the other for the research group enabling the user to contact any of them for queries about the services.

A summary of the criteria and the overall evaluation can be found in Table 7.1, where, ICIIR test refer to the result of the test performed by "Implicit Concept-based Image Indexing and Retrieval system (ICIIR).

Table 7.1 Summary of the Evaluation Criteria

Criteria	Overall Evaluation	Comments
Scope	Small	There are 1000 images used that can easily be increased at any time.
Search Options	Good	All options are simple to use. Some options of ICIIR search engine actually allow the users to view all images under the entered name in the Database.
Performance	Good	A good result was obtained, the data is small, and the repose to the query was very fast.
Presentation	Good	The images are displayed as thumbnails in a table of four columns and a number of rows within a number of pages which is Limited and determined for every search.
Support	Good	Help for every option is available on the site with online help.
ICIIR test	V. Good	Find all images needed, and the number of displayed images depending on whether the option used in retrieving images was reasonable. The results were good.

7.7 Summary

This chapter starts with the introduction, which explains our search engine, and the advantages that make users use this engine, a web based search engine that is a great development in the use of image indexing and retrieval.

A description of the system's specifications is also available; the hardware and software used are listed. The programming language and the database used for implementing the system and the reasons of selecting them are also thoroughly explained.

The system architecture, its components and the implementation of the system are explained in detail.

We applied the evaluation criteria to the ICIIR engine, and explained all of them in detail, and we summarized this evaluation in a table.

Chapter 8

Experimental Results

8.1 Introduction

We have conducted experiments using various image collections to measure the performance of the Implicit Concept-based Image Indexing and Retrieval system (ICIIR). The image collection criteria and the image resources are explained in details in Sections 3.2.1 and 3.2.3 of Chapter 3.

Different aspects can be used to display images in our search engine in order to test the ICIIR system; every aspect uses different methods of image retrieval explained in Section 6.4 of Chapter 6. We will use the "Approximate Image Search" aspect in this research, where one aspect allows the retrieval of images that may contain specific objects. Basically, when a certain object is entered, if the supporting factor value of this object in this category is above a certain value, then all the images in this category will be displayed. This is why some of the images displayed may not contain the entered object. Another method of this aspect is to retrieve images containing certain objects that constitute the entered object name, even if it is in a different image category. Examples for all the methods are shown later in the chapter.

Approximate Image Search

This aspect uses different methods of image retrieval and these are as follows:

Joint objects Search

This method of image retrieval is used to retrieve images which may contain a specific number of objects jointly. If the user uses one object, a number of images are displayed; most of them containing the used object, while some others do not. When two objects are used, the search engine combines the two objects, searching for images which may contain both objects. Again a number of images without the specified objects will be displayed. The same technique is also applied when using three objects. The user must enter at least one object.

Separate Objects Search

This method of image retrieval is used to retrieve images which may contain a specific number of objects separately. If the user uses one object, a number of images are displayed; most of them containing the used object but some others could possibly not include it. When two objects are used, the search engine searches for images containing each object as separate searches. The images retrieved may include both objects together, only one of the objects or possibly without any of the entered objects. The same technique is also applied when using three objects.

Backward Objects Search

This method of image retrieval is used to retrieve images using the backward technique. For example, a user may want to search for some objects, which are usually included in the "Garden" images. The user enters the object name as "Garden"; many images could be displayed, with some of them not being in the Garden category, but contain objects that may be found in images of Garden.

In the next sections, all of these techniques are tested and the results are reported.

8.2 Experimental Results

All methods of "Approximate Image Search" mentioned in the introduction of this chapter are tested fully in this section using the ICIIR system. For every method we used the maximum possibility of testing criteria in order to investigate each technique from different stand points. We recorded the results of every method in different tables, and some graphical figures were utilized in order to clarify our results.

The experiments tested the existence of the entered objects names, the intersection (Joint) of the used objects and the union (Separation) of the used objects, as well as the supporting factors of the objects in order to meet the users' requests.

One important issue covered in our experiments is to test the ratio of relevant retrieved images to the retrieved images (Precision). The ratio of relevant retrieved images to the relevant images (Recall) is not tested because in open collections it is not physically measuring the Recall. This test is a good way to investigate how relevant the retrieved images are to the user's desire.

The results from the users' queries regarding the methods used in Approximate Image Search from the new collection and Approximate Image Search from the database are reported in different tables. Two examples for each image search are reported later in this chapter.

For all methods used in the examples, only the first 120 of the retrieved images are tested and reported for their relevance because in some cases the number of images retrieved is very large.

Table 8.1 shows different columns each representing a certain result concerning the number and relevance of the retrieved image when a certain search is used. Each column is described in more detail below. This same table format is used throughout the next examples in order to determine important features about each search.

Table 8.1 Retrieved Images from Image Collection
Using Different Objects

Number and Percentage of Retrieved Images using Different Objects					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Trees	120	73	60.83	47	39.17
Flowers	120	57	47.50	63	52.50
Car	120	36	30.00	84	70.00

- The first column lists the names of the used objects when searching the desired images.
- The second column is used to report the number of retrieval images (120 images in all tables).
- The third column is used to report the number of relevant images.
- The fourth column is used to report the percentage of the relevant images.
- The fifth column is used to report number of irrelevant images.
- The sixth column is used to report the percentage of the irrelevance images.

8.2.1 Approximate Image Search using Non-Indexed Objects

The images used to test the methods under this aspect are collected randomly from the Google Web site. The objects contained within the collected images are not fully indexed (Implicitly indexed).

Example 1A:

As mentioned in the introduction of this chapter, this aspect uses three methods of image retrieval, these methods are as follows:

Joint Objects Search

Retrieving images using objects that may contain specific objects begins when the user enters the objects names that may be present in some images and submits the query. The system checks the objects names in the database, if not found, the system will check the entered names in the thesaurus, which searches the alternative words or sentences in the image collection with the equivalent meaning for the word or the sentence. When found it will search the image under the new name displaying all images that match the new name. Many images could be retrieved; the number of retrieved images could vary depending on the entered objects names, as well as the amount of objects used. Numerous testing of the system were carried out using different numbers of objects, up to three. This is shown in the following:

Using One Object

In this case, the user enters the object name and then submits the query. A number of images that match the user's query are displayed, most of these images are relevant, hence they contain the entered object, while some others are irrelevant, meaning they do not contain the entered object as shown in the first row of Table 8.2. In this case there are no joint objects since only one object is used.

Using Two Objects

In this case, the user enters two objects names and then submits the query; a number of images that match the user's query are displayed with many of them containing the entered objects but some others do not posses them as in the second row shown in Table 8.2. Relevant images are those containing both objects in the same image.

Using Three Objects

In this case, the user enters three objects names and then submits their query; a number of images that match the users query are displayed, with a small majority being relevant to the required search by containing all three desired objects, as in third row shown in Table 8.2.

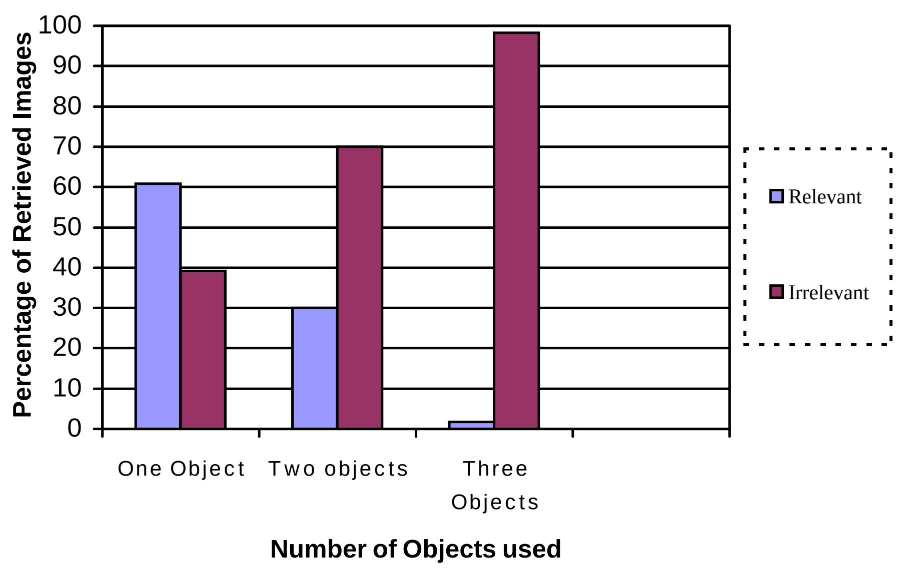
Table 8.2 Retrieved Images from Image Collection
Using Joint Objects Search – Example 1A

Number and Percentage of Retrieved Images, Relevant and Irrelevant , using Different Number of Joint Objects from Image Collection					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Trees	120	73	60.83	47	39.17
Trees **AND** Flowers	120	36	30.00	84	70.00
Trees **AND** Flowers **AND** Cars	120	2	01.66	118	98.33

In Table 8.2, the low percentage of the retrieved and relevant images does not mean that all other images are irrelevant but commonly partially relevant. For instance entering three objects may see the retrieval of some images with one or two of the entered objects which are theoretically irrelevant but actually have partial relevance.

Figure 8.1 displays the information presented in Table 8.2 in a graphical form for the retrieved (relevant and irrelevant) images using different number of objects.

Figure 8.1 Joint Objects Search - Example 1A



Separate Objects Search

This method of searching images is similar to “Joint Objects Search”; the difference being that the search engine, searches for the images with each object separately. The user enters the objects names and submits their query. Many images could be retrieved; the number of retrieved images could vary depending on the entered object name, as well as which and the amount of objects used. Numerous testing of the system were carried out using different numbers of objects as follow:

Using One Object

This case of image search is the same as “Joint Objects Search” when using only one object, the results are shown in the first row of Table 8.3.

Using Two Objects

The user enters two objects names and then submits the query; a number of images that match the user’s query that may contain the entered objects are displayed as shown in the second row of Table 8.3.

Using Three Objects

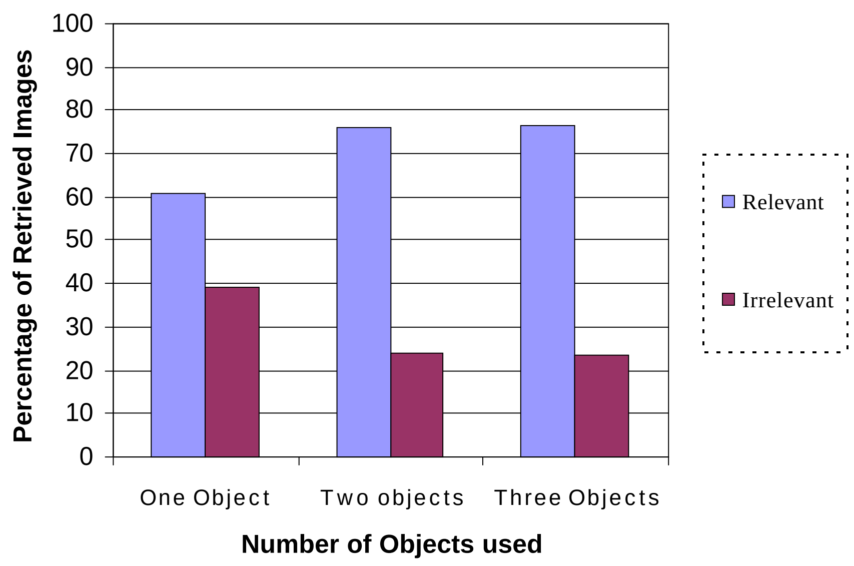
This case follows the same technique as when using two objects, the user enters three object names and then submits the query, and a number of images that match the user’s query are displayed as shown in the third row of Table 8.3.

Table 8.3 Retrieved Images from Image Collection
Using Separate Objects Search – Example 1A

Number and Percentage of Retrieved Images, Relevant and Irrelevant, using Different Number of Separate Objects from Image Collection					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Trees	120	73	60.83	47	39.17
Trees **OR** Flowers	120	91	75.83	29	24.17
Trees **OR** Flowers **OR** Cars	120	92	76.67	28	23.33

Figure 8.2 displays the information presented in Table 8.3 in a graphical form for the retrieved (relevant and irrelevant) images using different number of objects.

Figure 8.2 Separate Objects Search - Example 1A



The number of retrieved images (relevant and irrelevant) are most likely going to vary from the displayed images when using two or three objects with the same name in “Joint Objects Search”. In this case, the objects don’t have to be in the same image together, the presence of one object is enough to classify the image as relevant.

Backward Objects Search

This method of image search requires retrieving images which contain certain objects that constitute the entered object name. The objects found are displayed, some of them could be found in the same category as the entered object and some of them could be found in some other categories. In this case the number of displayed images is usually large. The user enters the object name and then submits the query; a number of images that match the user’s query are displayed as shown in Table 8.4.

Table 8.4 Retrieved Images from Image Collection
Using Backward Objects Search

Number and Percentage of Retrieved Images, Relevant and Irrelevant, of Object Components					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Garden	120	96	80.00	24	20.00
Soccer	120	84	70.00	36	30.00

Example 2A:

Below, another example with different objects is used for Approximate Image Search using the same technique as in all the methods used in the previous example (1 A). The results for this example are also reported in the same fashion.

Table 8.5 displays the results for retrieved images (relevant and irrelevant) for the objects used in this example.

Table 8.5 Retrieved Images from Image Collection
Using Different Objects – Example 2A

Number and Percentage of Retrieved Images using Different Number of Objects					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Soccer Ball	120	69	57.50	51	42.50
Field	120	53	44.17	67	55.83
Goals	120	27	22.50	93	77.50

Joint Objects Search

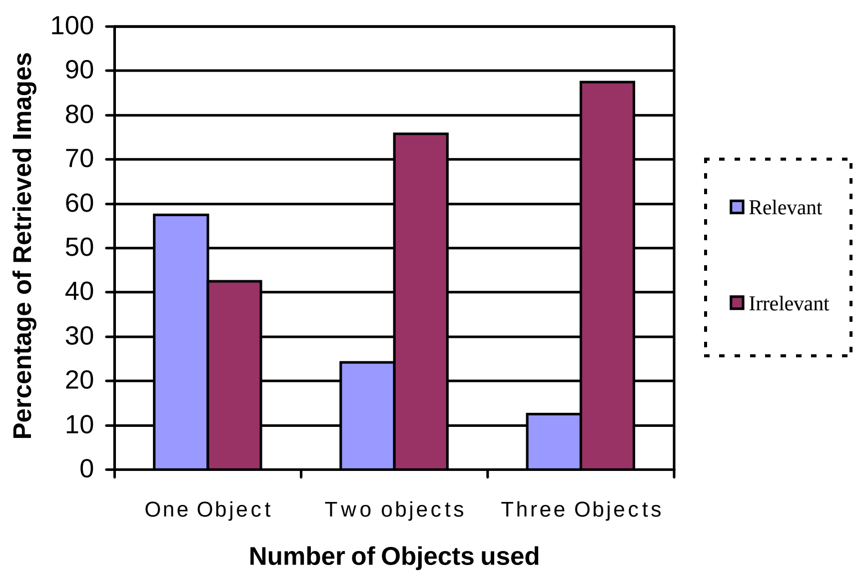
The results for a varying number of entered objects are shown in Table 8.6 using the Joint Objects Search method.

Table 8.6 Retrieved Images from Image Collection
Using Joint Objects Search – Example 2A

Number and Percentage of Retrieved Images, Relevant and Irrelevant , using Different Number of Joint Objects from Image Collection					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Soccer Ball	120	69	57.50	51	42.50
Soccer Ball **AND** Field	120	29	24.17	91	75.83
Soccer Ball **AND** Field **AND** Goals	120	15	12.50	105	87.50

Figure 8.3 displays the information presented in Table 8.6 in a graphical form for the retrieved (relevant and irrelevant) images using different number of objects.

Figure 8.3 Joint Objects Search - Example 2A



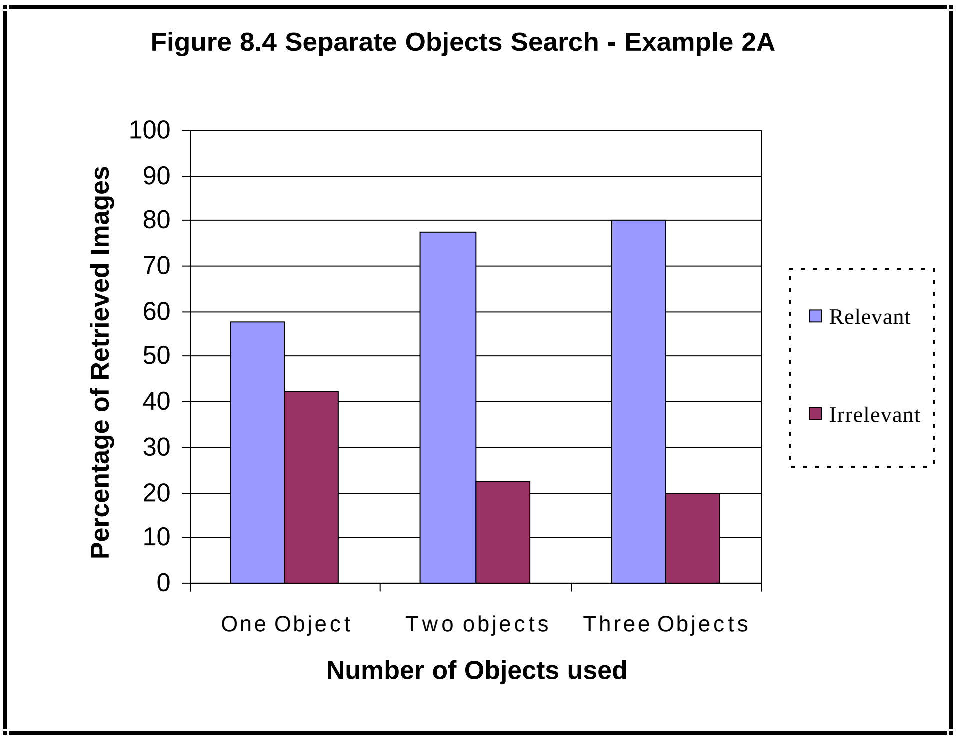
Separate Objects Search

The results for a varying number of entered objects are shown in Table 8.7 using the Separate Objects Search method.

Table 8.7 Retrieved Images from Image Collection
Using Separate Objects Search – Example 2A

Number and Percentage of Retrieved Images, Relevant and Irrelevant, using Different Number of Separate Objects from Image Collection					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Soccer Ball	120	69	57.50	51	42.50
Soccer Ball **OR** Field	120	93	77.50	27	22.50
Soccer Ball **OR** Field **OR** Goals	120	96	80.00	24	20.00

Figure 8.4 displays the information presented in Table 8.7 in a graphical form for the retrieved (relevant and irrelevant) images using different number of objects.



8.2.2 Approximate Image Search using Indexed Objects

The images used to test this method are collected randomly from different web pages. The objects within the collected images are fully indexed (Explicitly indexed); every object has many attributes extracted and computed using different Mathematical Equations explained in Section 4.3 of Chapter 4. The data for these attributes is indexed in different tables in the database (image database). This has been done to test the results obtained from the implicitly indexed image collection with an explicitly indexed image collection in order to evaluate the effectiveness of using ICIIR system.

Approximate image search using explicitly indexed objects is very similar to Approximate Image Search from Image Collection; the difference is that the images are retrieved from the database. Table 8.8 shows the number of retrieved images from the image database, the relevant and the irrelevant images and their percentage when using different objects.

Table 8.8 Retrieved Images from Image Database
Using Different Objects

Number and Percentage of Retrieved Images using Different Number of Objects					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Trees	120	115	95.83	05	04.17
Flowers	120	89	74.16	31	25.84
Cars	120	41	34.17	79	65.83

Example 1B:

Joint Objects Search

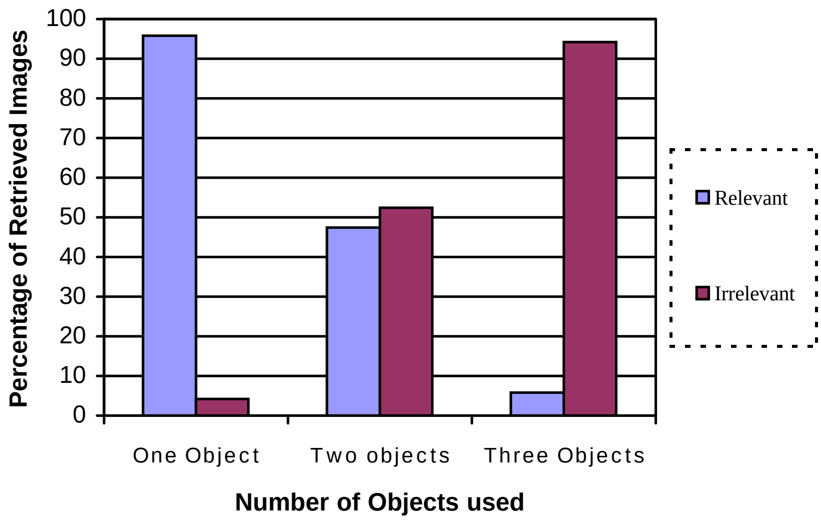
The procedure follows the same application as for the first two examples of Joint Objects Search, except that in this case the images are retrieved from an explicitly indexed database. The results obtained are presented in Table 8.9.

Table 8.9 Retrieved Images from Image Database
Using Joint Objects Search – Example 1B

Number and Percentage of Retrieved Images, Relevant and Irrelevant, using Different Number of Joint Objects					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Trees	120	115	95.83	05	04.17
Trees **AND** Flowers	120	57	47.50	63	52.50
Trees **AND** Flowers **AND** Cars	120	7	05.83	113	94.17

Figure 8.5 displays the information presented in Table 8.9 in a graphical form for the retrieved (relevant and irrelevant) images using different number of objects.

Figure 8.5 Joint Objects Search - Example 1B



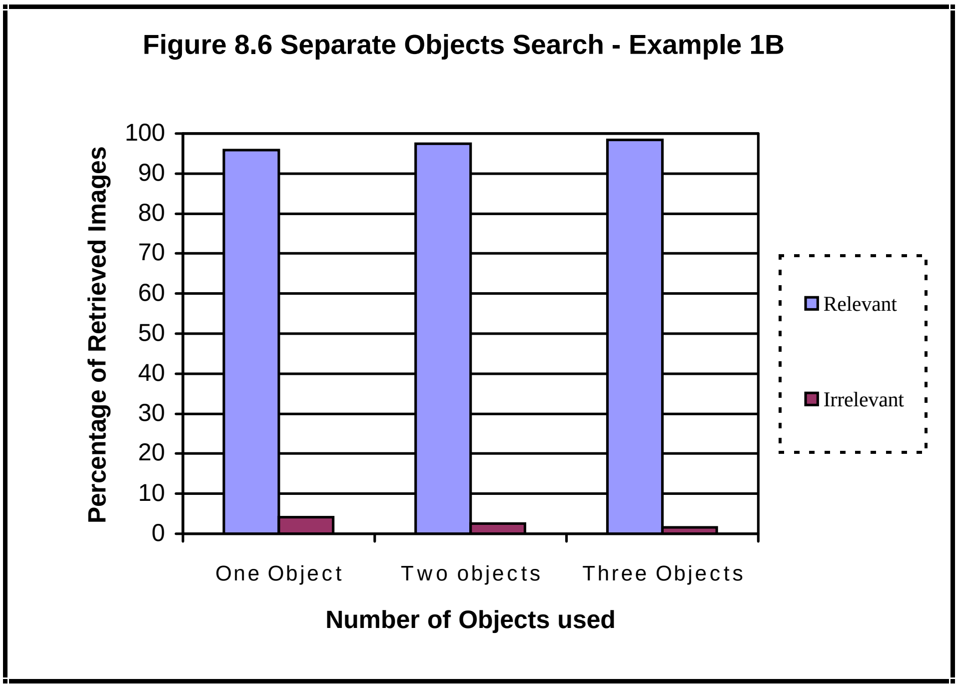
Separate Objects Search

The procedure follows the same application as for the first two examples of Separate Objects Search, except that in this case the images are retrieved from an explicitly indexed database. The results obtained are presented in Table 8.10.

Table 8.10 Retrieved Images from Image Database
Using Separate Objects Search - Example 1B

Number and Percentage of Retrieved Images, Relevant and Irrelevant, using Different Number of Separate Objects from Image Collection					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Trees	120	115	95.83	05	04.17
Trees **OR** Flowers	120	117	97.50	03	02.50
Trees **OR** Flowers **OR** Cars	120	118	98.33	02	01.67

Figure 8.6 displays the information presented in Table 8.10 in a graphical form for the retrieved (relevant and irrelevant) images using different number of objects.



Example 2B:

Below, another example with different objects is used for Approximate Image Search with fully indexed objects using the same technique as in all the methods used in the previous example. The results for this example are also reported in the same fashion. Table 8.11 displays the results for retrieved images (relevant and irrelevant) for the objects used in this example.

Table 8.11 Retrieved Images from Image Database
Using Different Number of Objects – Example 2B

Number and Percentage of Retrieved Images using Different Number of Objects					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Soccer Ball	120	105	87.5	15	12.50
Field	120	108	90.00	12	10.00
Goal	120	65	54.17	55	45.83

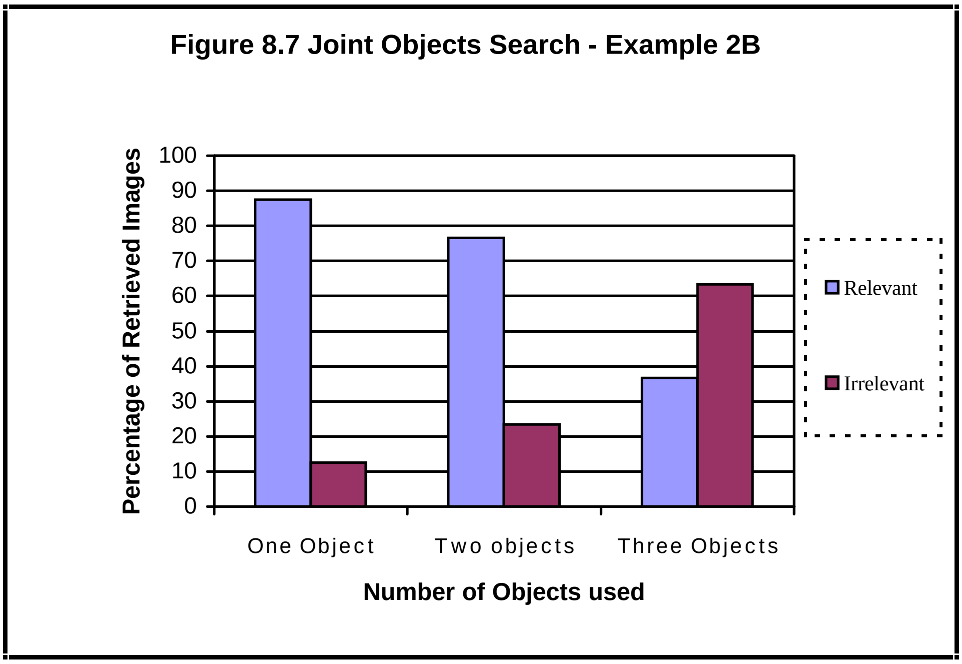
Joint Objects Search

The results for a varying number of entered objects are shown in Table 8.12 using the Joint Objects Search.

Table 8.12 Retrieved Images from Image Database
Using Joint Objects Search – Example 2B

Number and Percentage of Retrieved Images, Relevant and Irrelevant , using Different Number of Joint Objects from Image Database					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Soccer Ball	120	105	87.5	15	12.50
Soccer Ball **AND** Field	120	92	76.67	28	23.33
Soccer Ball **AND** Field **AND** Referee	120	44	36.67	76	63.33

Figure 8.7 displays the information presented in Table 8.12 in a graphical form for the retrieved (relevant and irrelevant) images using different number of objects.



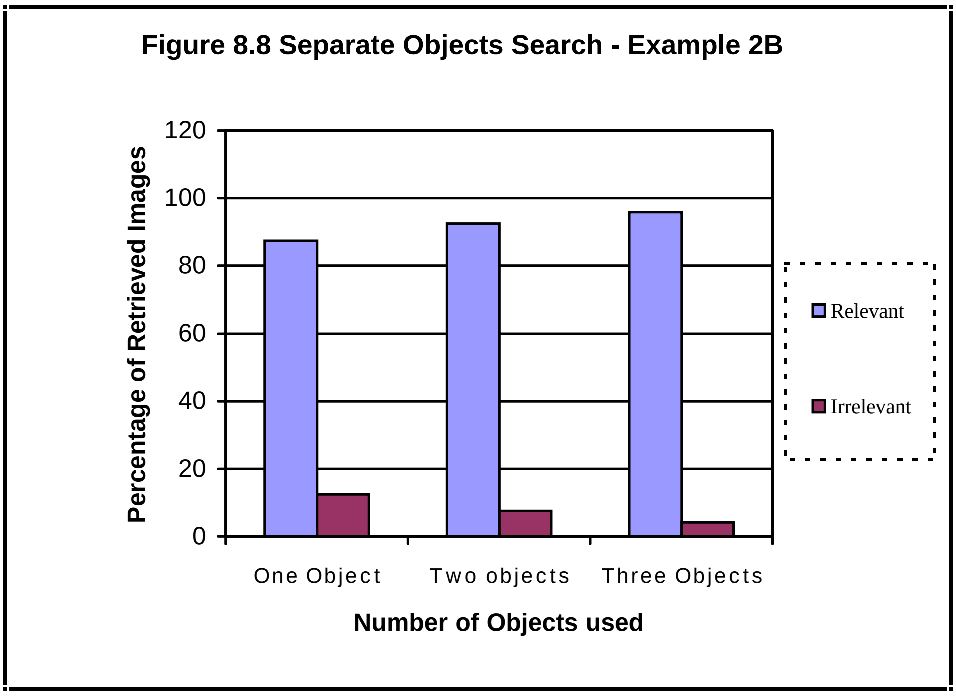
Separate Objects Search

The results for a varying number of entered objects are shown in table 8.6 using the Separate Objects Search.

Table 8.13 Retrieved Images from Image Database
Using Separate Objects Search - Example 2B

Number and Percentage of Retrieved Images, Relevant and Irrelevant, using Different Number of Separate Objects from Image Collection					
Object Name	# of retrieved images	Relevant images	Relevance %	Irrelevant images	Irrelevance %
Soccer Ball	120	105	87.5	15	12.50
Soccer Ball **OR** Field	120	111	92.50	9	07.50
Soccer Ball **OR** Field **OR** Referee	120	115	95.83	5	04.17

Figure 8.8 displays the information presented in Table 8.13 in a graphical form for the retrieved (relevant and irrelevant) images using different number of objects.



8.3 Interpretation of Experimental Results

Referring to our experiments and the result of the number of retrieved images using different aspect of image retrieval as shown in Sections 8.2 of this chapter, varying image numbers and relevant images are obtained for the different methods of the searches. Below, the results obtained for each method are explained.

8.3.1 Joint Objects Search

The number of retrieved images when using one object only is quite high. The number of retrieved images as well as the number of relevant images was reduced gradually as the number of objects is increased. The reason for that is: An increase in the number of objects means that the retrieved images have more conditions to follow; hence the relevant images retrieved from the image collection are narrowed down. As for relevant images being reduces with increasing number of used objects, this is due to the fact that not every image containing the first object will also contain the other objects. In the case where two objects are used, the images containing only one of the objects, if retrieved, will be considered irrelevant. Therefore the number of relevant images will be less as more objects are used.

8.3.2 Separate Objects Search

The number of retrieved images when using one object is also quite high. In contrast to Joint Objects Search, the number of retrieved images and relevant images were increased as the number of objects increased. The number of retrieved images increased because performing a Separate Objects Search using more than one object is like performing a single object search for each object separately and combining the retrieved images together, naturally excluding image duplication. Therefore adding extra objects would display the same images as well as possibly extra image categories. As for relevant images increasing with more objects, this is because adding objects could mean that images retrieved earlier not having the specified objects (irrelevant) may have the new object making it now relevant.

8.3.3 Backward Objects Search

The number of retrieved images in this type of search depends only on the entered object since only one object can be entered. When an object is entered, each one of its component objects which have a supporting factor value above a certain threshold will be displayed even if it is found in images of a different category in the database or image collection. As the images may be displayed from any category, the number of images displayed is usually high.

8.3.4 Performance of Implicitly Indexed Image Search

Referring to the results obtained from the implicitly indexed data, some relevant images are still being retrieved despite the objects used to retrieve these images not being indexed in the image collection. In addition, it is found that the results obtained from the implicitly indexed objects are still quite accurate in comparison to the results obtained from explicitly indexed objects. For example: Searching for the object, “Trees” using the Joint Objects Search. As the objects is not indexed the search is expected to retrieve no images, yet some images are retrieved as shown in Table 8.2, where the percentage number of relevant images is (60.83%). The percentage of relevant images value obtained for the object, “Trees” using the Join Objects Search from an indexed database is (95.83%). This ensures that our system produces excellent results. Even using two non-indexed objects, “Trees” and “Flowers” sees the retrieval of a fair amount of relevant images (30.00%) in comparison to (47.50%) obtained using indexed objects. The same applies when using three objects.

As for the Separate Objects Search and the Backwards Objects Search, the results for Implicitly Indexed Objects are also comparatively accurate in relation to Explicitly Indexed Objects.

8.4 Precision and Recall

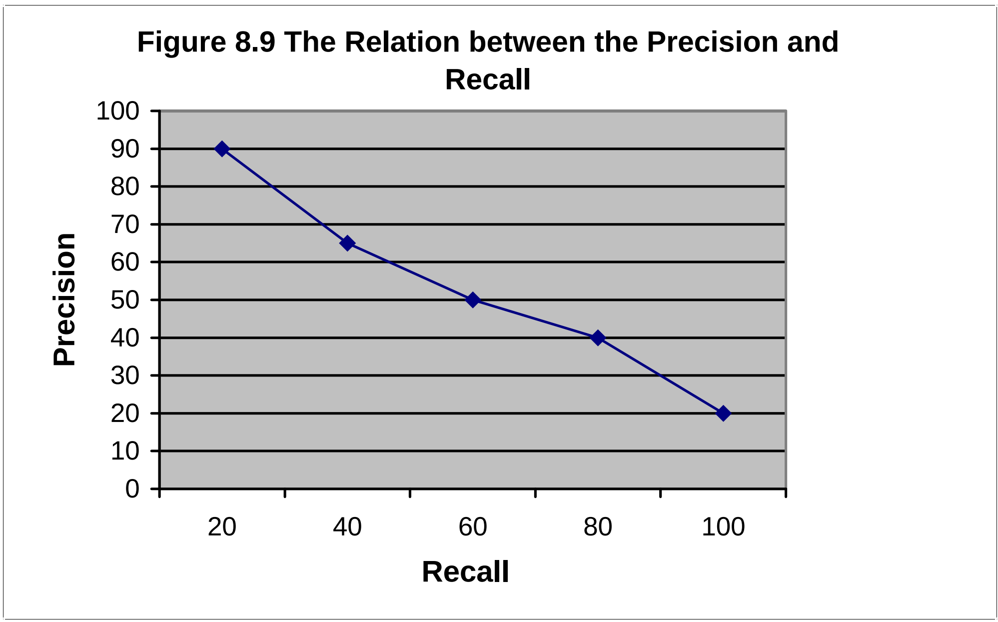
After completing image search, one must ask, whether the images retrieved are relevant.

Unfortunately in all image search engines, retrieving all relevant images, while avoiding irrelevant ones is difficult, if not impossible. However, it is possible to measure the performance of a search with respect to two main parameters: *Precision* (purity of retrieval) and *Recall* (completeness of retrieval). Empirical studies of retrieval performance have shown a tendency for precision to decline as Recall increases. Analysis of the relationships between Recall, the number of images retrieved, and Precision shows that there is a definable region for all reasonable retrieval results. For all cases of consistently better-than-random retrieval, Recall curves tend to follow an increasing curve rising from the origin, and a trade-off between Precision and Recall is inherent. More generally, a trade-off between Precision and Recall may not be involved in the case where as the total number of images retrieved increases, retrieval performance is equal to or better than overall retrieval performance thus far, a typical example is shown in Table 8.14.

Table 8.14 Relation between Precision and Recall

No of query	Precision %	Recall %
1	90	20
2	65	40
3	50	60
4	40	80
5	20	100

Figure 8.9 displays the information presented in table 8.14 is a graphical form.



Based on the discussion in this section it is obvious that Precision and Recall are the basic measures used in evaluating search strategies; the remaining experimental result is evaluated in term of precision and Recall defined as follows.

Precision consists of the proportion of target images that are retrieved up to the last correct one, which corresponds to 100% Recall. A high value of precision therefore indicates that the top-ranked hits all contain target images.

$$\Pr ecision = \frac{number\ of\ \ relevant\ \ retrieval\ \ images}{number\ of\ \ retrieval\ \ images} \cdots (A)$$

Recall consists of the proportion of target images (images from the same class as the query) that have been retrieved among all the relevant images in the database, given a certain value of weights (for example), this measure is clearly monotonically increasing with weight and attains 100% when weight includes the whole dataset.

$$\text{Recall} = \frac{number\ of\ relevant\ retrieval\ images}{number\ of\ relevant\ images} \cdots (B)$$

Suppose that R_r = Retrieval relevant images, R_n = Retrieval irrelevant images, and N_r = Not retrieval relevant images. Then:

The equations of Precision and Recall can take the following form:

$$P = \frac{R_r}{R_r + R_n} \times 100 \cdots\cdots\cdots\cdots\cdots\cdots (A)$$

$$R = \frac{R_r}{R_r + N_r} \times 100 \cdots\cdots\cdots\cdots\cdots\cdots (B)$$

Figure 8.10 illustrate the retrieved and relevant Images, while Figure 8.11 illustrate Relevant (retrieved and not retrieved) and retrieved irrelevant images.

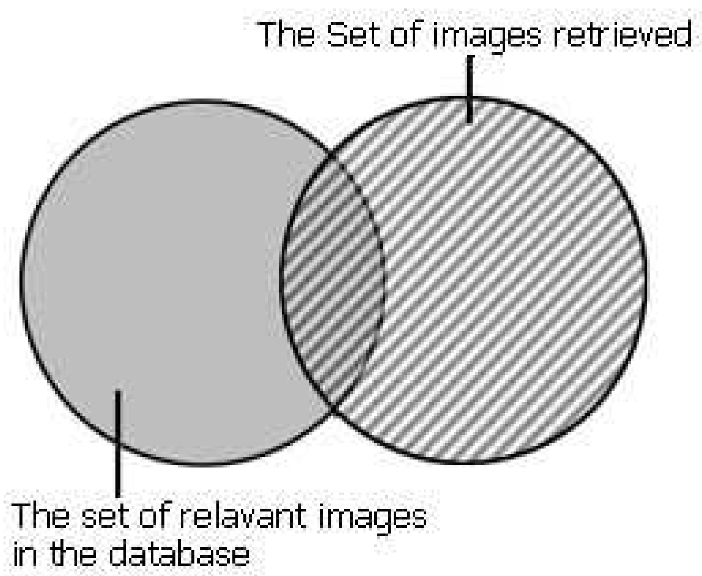


Figure 8.10 Retrieved and Relevant Images in the Database

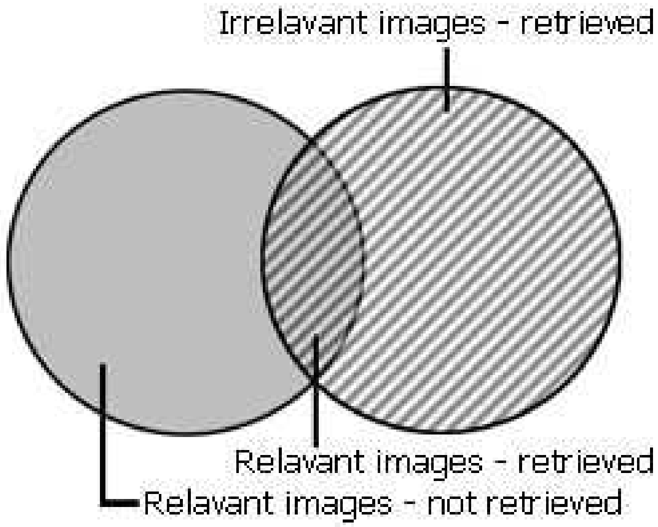


Figure 8.11 Relevant (Retrieved and Not Retrieved) and Retrieved Irrelevant Images

As mention earlier in this chapter (Section 8.2), one important issue covered in our experiments is to test the ratio of relevant retrieved images to the retrieved images (Precision). The ratio of relevant retrieved images to the relevant images (Recall) is not tested because in open collections it is not physically measuring the Recall. The result of relevance percentage of images of Example 1A in Section 8.2.1 and Example 1B in Section 8.2.2 in this chapter is reported in Table 8.15 and Table 8.16. This result gives a good way to investigate how relevant the retrieved images are to the user's desire.

Table 8.15 Percentage of Retrieved Images using Different Number of Objects "Joint Objects Search" (Examples 1A & 1B)

Percentage of Retrieved Images using Different Number of Joint Objects		
Object name	Relevance % (Explicit Indexed)	Relevance % (Implicit Indexed)
Trees	95.83	60.83
Trees **AND** Flowers	47.50	30.00
Trees **AND** Flowers **AND** Cars	05.83	01.66

Figure 8.12 displays the information presented in Table 8.15 in a graphical form.

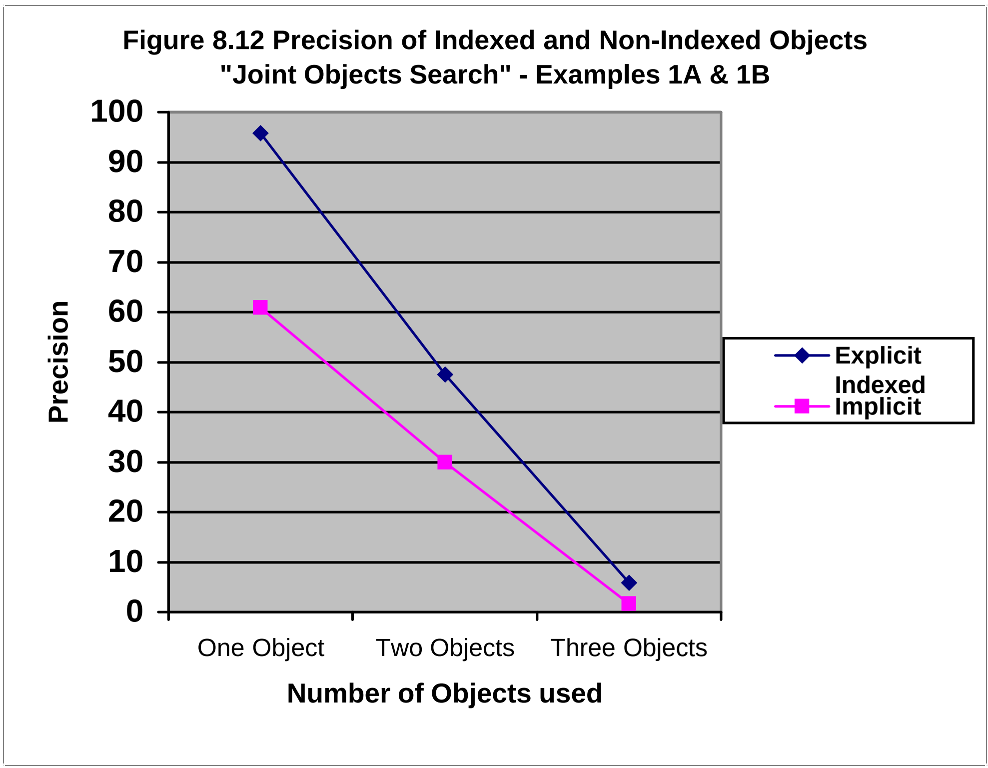
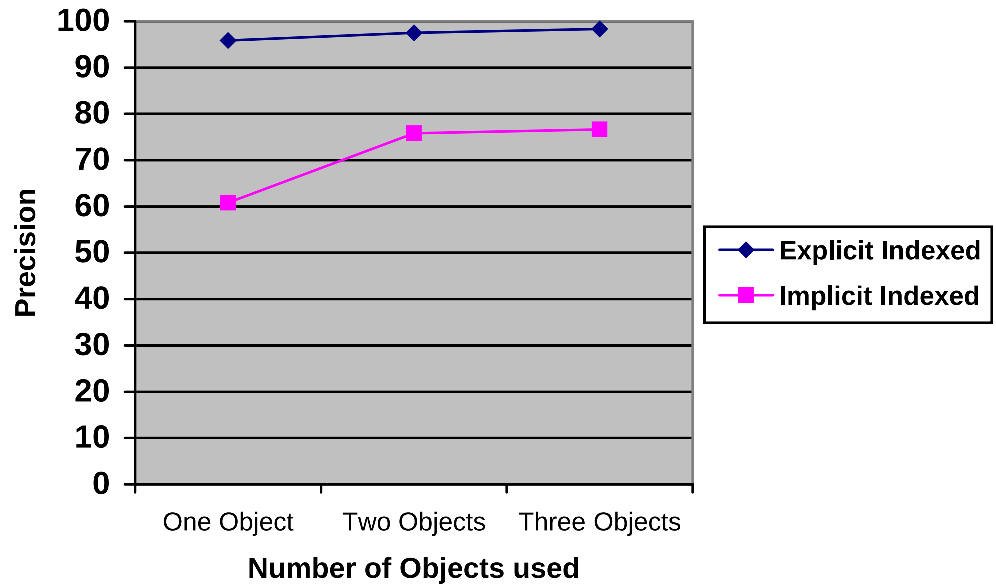


Table 8.16 Percentage of Retrieved Images using Different Number of Objects "Separate Objects Search" (Examples 1A & 1B)

Percentage of Retrieved Images using Different Number of Joint Objects		
Object name	Relevance % (Explicit Indexed)	Relevance % (Implicit Indexed)
Trees	95.83	60.83
Trees **OR** Flowers	97.50	75.83
Trees **OR** Flowers **OR** Cars	98.33	76.66

Figure 8.13 displays the information presented in Table 8.16 in a graphical form.

Figure 8.13 Precision of Indexed and Non-Indexed Objects "Separate Objects Search" - Examples 1A & 1B



The evaluation of our system using Precision shows that the results we obtained were very accurate. A reasonable number of desired images are retrieved and some of irrelevant images are displayed.

8.5 Summary

This chapter introduces the different aspects used in the ICIIR system. Experiments to validate different aspects of our research are provided. Experiments and results are listed in tables for different aspects, an explanation for every experiment is provided in detail. Furthermore, graphical representations of the tables are provided to explain the difference of retrieved images (relevant and irrelevant) when using various objects and different number of objects.

We applied the evaluation criteria to our system, all points are explained in detail, and we summarised them in a table.

The important features in ICIIR System are:

- The images stored in our system are not duplicated. In other words, some images may be placed in more than one category yet it only appears once in the database. This will assist in saving some space.
- As for objects in the images, every category has some essential objects that must be contained in the image. Also as in most images other non-essential objects could be present.

- Using conditions such as supporting factor values when retrieving images will result in a more precise search reducing ambiguities. The outcome will be the retrieval of more relevant and less irrelevant images.
- The results obtained from the experiments ensured quite accurate image searches to match the user's needs. Furthermore, the precision results are quite fine verifying quite good search accuracy.

Chapter 9

Conclusion

9.1 Summary of Contributions

Implicit Concept-based Image Indexing and Retrieval is one of the most important functions of Visual Information Systems. In this thesis, we developed an improved method for the image storage and retrieval. This method involved the construction of a new algorithm, which involved the development of techniques to enable components of an image to be categorised on the basis of their relative importance. Thus the storage of images involved an implicit, rather than an explicit, indexing scheme.

Retrieval of images will then be consequence by application of an algorithm based on this categorisation, which will allow relevant images to be identified quickly, and thus allow these images to be researched for required details.

Consequently, the search time involved in locating target images will be considerably reduced, providing a more responsive and more productive system. A secondary aim is to research existing methods of storing and retrieving images from a database, and to quantify the performance characteristics of the most effective existing methods, in order to provide a basis upon which to evaluate the performance of the algorithm developed in this thesis.

We summarized the major points and contributions of this thesis below:

Literature Review

There are two methods of modelling and retrieving images:

- Concept-based methods: Allows the search for images using keywords or form of natural language.
- Content-based methods: Allows the search for images by image contents using characteristics such as colour, shape or texture.

Both methods are indexed and retrieved in a different manner. There are problem with each method for both aspects of indexing and retrieval:

- Concept-based methods problems: Sometimes briefly include, descriptive difficulties of images, time consuming and incomplete text description in some cases.
- Content-based methods problems: Include errors such as, shape and colour complications and the method is also not rich in image semantics.

The concept-based method has been selected in our research due to its higher success rate in many important areas. It also provides a technique for incorporating domain knowledge into a user's query.

Forward and backward chaining are two storage and retrieval methods. Forward chaining involves beginning from the known data and proceeding forward with the data. Backward chaining, on the other hand, involves an expert system setting a goal with the inference engine attempting to find the evidence to establish it.

Image Organization

The image collection and the selection rules are presented. Our current system is used for categorizing images in collections, where images tend to fall into multiple categories in the real world.

The image collection contains multi-objects, and the rules control the image categories. There are certain relationships between objects in an image, which are; a Composite Relationship, Is-A Relationship and Aggregation Relationship.

Image indexing consists of many discrete object types, where there are three object types (Low-level object, Mid-level object and High-level object), which gives a good presentation. The object levels are built in a way that, some low-level objects constitute a mid-level object, and in turn, some mid-level objects constitute a high level object.

An image can be located in any category provided that it contains objects satisfying the category rules.

Measures of the Reliability of Facts

Every object is given multiple attributes; these attributes are extracted and computed in order to be stored in the database. The attributes will be used for retrieval of desired objects. These attributes are as follows:

- Objects' Number**:** The principal means of identifying objects within an object set.
- Objects' Name**: S**uitable description of an object.
- Objects' Image**:** The image address.
- Objects' Weight**: A** measure of the significance of an object in its set in an image.
- Objects' Availability**: A** measure of the expected probability of the object's appearance in an image.
- Objects' Supporting Factor**:** A measure of the dependence of the appearance of a high-level object on this object.
- Objects' Value**: C**omputed by multiplying together the object's weight, availability and supporting factor.
- Objects' Rule Significant Factor**: A** measure of the total appearance of all object components in a certain image divided by the total appearance of all objects for the same object's name in the database.
- Objects' Weight in Image**: A** measure of the importance of an object in the image.
- Object's Density**:** Equivalent to the area occupied by the object in the image.
- Object's Type: An object's type describes the object level in an image (low-level, mid-level, high-level objects, etc).

Mathematical Equations were created and used to compute the different object attribute values.

Image Indexing

The images must be indexed in a way that is easy to store and retrieve. Our indexing method is based on the fact that several atomic indices, which will be able to define intermediate indices and intermediate indices will be able to define higher indices. This approach involved developing algorithms for the storage and retrieval of image data from databases using a methodology that incorporates implicit image indexing. A higher level index can be built from lower level indices.

The basic steps of the indexing algorithm are as follows:

- Suitable images collected and stored in image folder
- Data extraction
- Compute the fact attributes' values
- Save the information

Image Retrieving

Our method concentrates on matching subparts of images, defined in a variety of ways, in order to find particular objects. It is obvious that the use of fuzzy logic and supporting factors lead to improvements of both precision and recall.

The retrieval algorithm follows four steps:

- User interface: User enters details about their specific search.

- Data processing: Details will be processed and checked.
- Output: Matched images are displayed along with extra information.
- Feedback: User states feedback to help improve search in the future.

There are two main searching aspects used, both having many methods involved. These aspects are:

- Approximate image search: When a user enters a certain object with a supporting factor value above a threshold, the entire category is displayed, including those images without the entered object. This also applies for multiple objects. A few methods can be applied for this search, which are; the Joint Objects Search, Separate Objects Search and Backward Objects Search.
- Exact Image Search: The user can enter an object name as well as many supporting objects to be components of the entered object. The user can also specify certain details about the supporting objects such as weight, density and availability. The result is an exact match of the user's entered data.

System Implementation

The programming language used to implement our system is Active Server Pages (ASP). As for the database to store our data, Oracle9i is used.

The architecture of the ICIIR system can be divided into a number of managers which are; User Interface (UI), Transaction Manager (TM), Storage/Retrieval Manager (SRM), Query Manager (QM), Display Manager (DM) and Interpretation Manager (IM). These are thoroughly explained in Chapter 7.

The ICIIR Engine Evaluation is assessed with various criteria including; System Scope, System Search Options, System Performance, System Presentation and System Support. These are used to evaluate our system and determine its efficiency.

Experimental Results

This chapter experiments the different methods of the Approximate Image Search. The data is entered in different tables and graphs, reporting the relevance and irrelevance of the images displayed, when different objects or different number of objects are used.

The data for implicit and explicit indexing is compared. Our research makes use of implicit indexing, which was found to be reasonably accurate despite the objects used to retrieve images not actually being indexed. We tested the precision to ICIIR, and the results were found to be quite accurate in retrieving images meeting the user's queries.

9.2 Limitations

The high-level abstractions and concepts can be easily expressed in concept-based methods, the query can be easily issued and the user interface is very simple, and the standard text retrieval techniques can be used for image searching, where the text retrieval techniques have been highly developed in the information retrieval system. There are many more advantages of the concept-based retrieval system, but it still suffers from a number of limitations, some of which are:

- The understanding of an image depends on different levels of knowledge and experience.
- Organising the description of a picture's subject matter in such a way that the picture can be most certainly and efficiently retrieved has to be considered carefully.
- It is tedious and time-consuming to associate keywords or text with each image.
- Some features are very difficult to describe with text, some special textures and complex shapes can't be clearly represented.
- Text description is sometimes incomplete. It is possible that some image features may not be mentioned in textual description, and this is time consuming.
- Text description is sometimes subjective. Different indexers or even the same indexer may describe the same features with different terms or different features with the same terms.
- Decomposing the image to its components is not easy and many experts can use different ideas, for example, some indexers may consider a table's objects to be primitive, while others may decompose the table to many objects.
- Determine the objects inside images is still suffering from many problems, this leads the indexer the do this job manually, which is another time consuming task
- The relationship between objects makes it difficult when making any changes and inserting any new images, which may lead to the need of changing some or all of the objects' values.
- The collection of images which satisfy our rules sometimes require that copyright fees are met, which will cause the loss of more funds.

9.3 Future Directions

Many theoretical and practical research methods may be possible in the future. Some of these are listed below:

- *Store the data in XML documents*: it is easy to edit data by making use of a text editor rather than a complicated database tool. XML files can be easily backed up, uploaded and downloaded by/to the clients. When a relational database is used, installing and supporting a separate server process is required that needs a database administrator. The knowledge of SQL queries is required for relational databases. Entity relationship schema or normalizing the tables is not required for XML. XML format is hierarchical rather than relational which is used in traditional relational databases. If one element contains another element then they are treated as parent and child elements and are designed in a structured manner rather than using a join table. XML becomes a stronger and programmer friendly database for storing, browsing and searching.
- *Objects recognition*: when using a system that is able to recognize the objects inside the image and then store these objects automatically, the objects' values of all its attributes will be automatically computed and stored in the database, this will save time.
- *Combine between content-based and concept-based methods*: combining our system with the content-based method could help describe the objects a little more.

Bibliography

[ABDE96] Mohamed Abdel-Mottaleb, Nevenka Dimitrova, Ranjit Desai and Jacquelyn Martino, "*Conivas: Content-based Image and Video Access System*", *ACM Multimedia*, pp 427-428, Nov. 1996.

[AZZA04] I. A. Azzam, C.H.C.Leung, J.F. Horwood, "Implicit Concept- based Image Indexing and Retrieval", *The 10th International Multimedia Modeling Conference MMM2004*", Jan-2004, Brisbane, pp 354-359.

[AZZA04a] I. A. Azzam, A. Charlapally, C.H.C.Leung, J.F. Horwood, "Content-based Image Indexing and Retrieval using XML Representation", *@004 International Symposium on Intelligent Multimedia, Video & Speech Processing*", October-2004, Hong Kong, pp 181-185.

[AZZA05] I. A. Azzam, C.H.C.Leung, J.F. Horwood, "A Fuzzy Expert System for Concept-based Image Indexing and Retrieval", The *11th International Multimedia Modeling Conference MMM2005*", Jan-2005, Melbourne, pp 452-457.

[BARB95] Barber, R. etal., "A Guided Tour of Multimedia Systems and Application: Query by Content for Large On-Line Image Collection", *IEEE Computer*, 1995, pp. 357-378.

[BARB94] Barber, R. etal., "MULTIMEDIA MANAGER: Query By Image Content and its Applications", *IEEE Spring Compcon '94*, 1994, pp. 424-429.

[BODL91] H. Bodlaender, T. Kloks, and T. Gonzalez. "Complexity aspects of map compression.", *Proc. Data Compression Conf., April 1991*, pp 287-296.

[BOUE98] M. Bouet and C. Djeraba, "Powerful Image Organization in Visual Retrieval Systems"' in ACM Multimedia'98, Bristol, UK, 1998.

[CARS99] C. Carson, M. Thomas, S. Belongie, and et al. Blobworld: "A system for region-based image indexing and retrieval", "*In Proc. Int. Conf Visual Inf. Systems.*" 1999, pp 509-516.

[CHAF00] Alex Chaffee, Using XML and JSP together, Java World, March 2000.

[CHAN01] Yung-Kuan Chan , Chin-Chen Chang, "Image Retrieval Based on Tolerable Difference of Direction", *Proceedings of the 15th International Conference on Information Networking*, January 31-February 02, 2001, p.387.

[CHAN87] S. K. Chang, Q. Y. Shi, C. W. Yan, "Iconic indexing by 2-D strings", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.9 n.3, May 1987, p.413-428.

[CHEN03] Yixin Chen, J. Z. Wang, and R. Krovetz, "An Unsupervised Learning Approach to Content-based Image Retrieval", "*IEEE Pro. Inter. Symposium on Signal Processing and its Applications*", 2003, pp 197-200.

[CHUA94] Tat-Seng Chua, Hung-Keng Pung, Guo-Jun Lu and Hee-sen Jong, "A Concept-based Image retrieval system", *IEEE computer*, Jan 1994, pp 590-598.

[CORR99] J. Corridoni, A. Del Bimbo, and P. Pala, "Image retrieval by colour semantics", *Multimedia Systems, (7)*, 1999, pp 175-183.

[COX94] Cox E. 1994 "The fuzzy systems handbook: A Practitioner's guide to building using, and maintaining Fuzzy Systems Academic Press", *Cambridge,* 1994.

[DENG04] Da Deng, "Content-based Comparison of Image Collections via Distance measuring of Self-organised Maps", "*10th International Multimedia Modeling Conference MMM2004*", Jan-2004, Brisbane, pp 354-359.

[DETE00] John Determan and Greg Becker "A prototype Generic Expert System Architecture for Data Review/Validation", *7th Non-destructive Assay Waste Characterization Conference,* Salt Lake City, Utah, May 22-26, 2000, pp. 261-270.

[DICK98] Dickinson S et al “Viewpoint-invariant indexing for content-based image retrieval” in *IEEE International Workshop on Content-based Access of Image and Video Databases (CAIVD’98)* , Bombay, India,1998, pp 20-30

[ENSE95] Enser, P.G.B. (1995) “Pictorial Information Retrieval” *Journal of documentation* 51(2): 126-170.

[FALO94] Faloutsos, C et al (1994) “Efficient and effective querying by image content” *Journal of Intelligent Information Systems* **3**, 231-262

[FLIC95] Flickner, Myron et.all., “Query by Image and Video Content: The QBIC System”, *IEEE Computer,* Vol. 28 Issue 9, September 1995, pp 23-32.

[FENG04] HuaMin Feng and Tat-Seng Chua, “A Learning-based Approach for Annotating Large On-Line Image Collection”, “*10th International Multimedia Modeling Conference MMM2004*”, Jan-2004, Brisbane, pp 249-256.

[FRAK88] Frakes, W.B. and Nejmeh, B.A., “Information System for Software Reuse”, in Tracz, W. (Ed), *IEEE Tutorial: Software Reuse: Emerging Technology*, IEEE Computer Society, 1998.

[FRAU01] Fraunhofer-IPSI, “MPEG-7 (Moving Picture Experts Group)”, Homepage, www.ipsi.fhg.de/ delite/ Projects/ MPEG7/index.html) August 22, 2001

[GEVE92] T. Gevers and A.W.M. Smeulders, “Sigma nigma: an image retrieval system”, *IEEE computer society press*, pp 697-700, Sep. 1992.

[GRAH99] Graham, Margaret E. "Description and Indexing of images: Report of a survey of ARLIS members, 1998/99." *IIDR Research Report*, no. 2. Institute for Image Data Research, 1999.

[GUDI97] Venkat N. Gudivada and Vijay V. Raghavan, “Modeling and Retrieving Images Content System”, *Information Processing and Management*, 1997, Vol.33, No.4, pp. 427-452.

[GUTT84] Antonin Guttman, R-trees: a dynamic index structure for spatial searching, *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, June 18-21, 1984, Boston, Massachusetts

[HAMP97] Hampapur et. Al. Virage Vidio. *Proc. Of SPIE, Storage and Retrieval for Image and Video Database V*, 3022, 1997, pp. 188-200

[HAN96] Kyung-Ah Han and Sung-Hyun Myaeng, “Image Organization and Retrieval with Automatically Constructed Feature Vectors”, *in ACM SINGER’96*, Zurich, Switzerland, pp. 157-165.

[HIRA00] K. Hirata and S. Mukherjea, “Integration of image matching and classification for multimedia navigation”, “*Multimedia Tools and Application*”, (11), 2000, pp 295-309.

[HOWA95] Besser, Howard. (1995) “Getting the picture on image database” *Database* April/May: 12-19.

[HUAN98] Chung-Lin Huang and Dai-Hwa Huang, “A Content-Based image Retrieval System”, *Image and Vision Computing 16*, pp 149-163, 1998.

[HUAN98a] J. Huang, S. Kumar, M. Mitra, and W. Zhu, ”Spetial colour indexing and application., “*In Proc. Of ICCV*”, 1998, pp 602-607.

[HWAR00] R.J. Hwarth and H. Buxton, “Conceptual-Description from Monitoring and Watching Image Sequences”, *Image and Vision Computing 18*, 2000, pp 105-135.

[IMAT04] IMatch - Image Organization with Albums and Dynamic Categories - November, 01. 2004

[JAIN93] Jain, R. C. "Storage and Retrieval for Image and Video Databases." *Proceedings-SPIE, the International Society for Optical Engineering in San Jose, Calif.*, 1993, pp. 198-218.

[JIN98] Jin, J S et al “Using browsing to improve content-based image retrieval” in *Multimedia Storage and Archiving Systems III* , Proc SPIE 3527,1998, pp 101-109

[JORG98] Corinne Jorgensen, "Attributes of Images in Describing Tasks", *Information Processing and Management*, Vol.34, No. 2/3, 1998, pp 161-174.

[KIM99] Bowon Kim , Kyhyun Um, 2D+ String: A Spatial Metadata to Reason Topological and Directional Relationships, Proceedings of the 11th International Conference on Scientific on Scientific and Statistical Database Management, July 1999, p.112.

[KONW02] Kishori M. Konwar, and Sankar K. Pal, "Fuzzy Decision Tree, Linguistic Rules and Fuzzy Knowledge-based Network: Generation and Evaluation", *IEEE Transaction on Systems*, Man, Cybernetics-Part C: application and Reviews, Vol. 32, N0 4, 2002, pp. 328-338.

[KURN97] Kurniawati, R et al "The SS+ tree: an improved index structure for similarity searches in high-dimensional feature space" in *Storage and Retrieval for Image and Video Databases V* (Sethi, I K and Jain, R C, eds), Proc SPIE 3022, 1997, pp 110-120

[LEUN92] Leung, C.H.C., D. Hibler and N. Mwara, "Image retrieval by Content Description", *Journal of Information Science*, Vol (18), 1992, pp. 111-119.

[LEUN95] Leung, C.H.C. and Zheng, Z.J., "Image Data Modeling for Efficient Content Indexing", *Proc IEEE International Workshop on Multi-Media Database Management Systems*, 1995, pp. 143-150.

[LEUN97a] Clement Leung "Guest Editor's Introduction: Visual Information Systems", *Journal of Visual Languages and Computing*, Vol. 8 No.3, Academic Press, June 1997, pp. 261-263.

[LEUN97b] C.H.C. Leung and W.W.S. So, "Characteristics and Architectural Components of Visual Information System", in Clement H.C. Leung (Ed.), Lecture Notes in Computer Science, Berlin Heidelberg: Springer-Verlag, Vol. 1306, 1997, pp. 1-12.

[LEUN99] C.H.C. Leung and W.W.S. So, "Visual Information System", in Borko Furht (Ed.), *Handbook of Internet & Multimedia Systems & Application*, Chapter 16, Boca Raton FL: CRC Press LLC, 1999, pp 361-379.

[LI95] Li, H and Gupta, M., "Fuzzy logic and intelligent Systems", *Kluwer Academic Publishers*, Boston, 1995.

[LIN94] Lin, K I etal., "The TV-tree: an index structure for high dimensional data" *Journal of Very Large Databases* **3**(4), 1991, pp 517-549

[LIN99] Hsin-Chih Lin, Ling-Ling Wang and Shi-Nine Yang, "Regular-Texture Image Retrieval based on Texture-Primitive Extraction", *Image and Vision Computing 17*, 1999, pp 51-63.

[LIPS97] P. Lipson, W. E. L. Grimson and P. Sinha, " Configuration based scene classification and image indexing", in IEEE Conference on Computer Vision and Pattern Recognition 1997.

[LIU05] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma, "Region-based Image Retrieval with High-Level Semantic Color Names", "*11th International Multimedia Modeling Conference MMM2005*", Jan-2005, Melbourne, pp 180-186.

[LONG01] H.Z.Long and W.K. Leow, "Perceptual texture pace improves perceptual consistency of computational features". *In Proc. IJCAI*, 2001.

[LONG01a] H.Z.Long and W.K. Leow, "Perceptual consistency improves images retrieval Performance" ACM, SIGIR01, Sep., 2001, pp 9-12.

[LU00] Y. Lu, C.Hu, X. Zhu, H.J. Zhang and Q. Yang " A unified framework foe semantics and feature based relevance feedback in image retrieval system" in Proc. ACM Multimedia, Los Angeles, CA, Oct. 30-Nov. 4, 2000. pp 31-38.

[MA98] Ma, W. & Manjanath, B. "Netra: A toolbox for navigating large image databases", *Proceedings of IEEE International Conference on Image Processing (ICIP97)*, 1, 1998, pp568-571.

[MARK92] T. Markas and J. Reif. "Quad tree structures for image compression applications", *Information Processing and Management*, 28(2), November 1992, pp707-721.

[MOJS01] A. Mojsilovic and B. Rogowitz, "Capturing Image Semantics with Low-Level Description", "*Proc. of ICIP*", 2001, pp 947-963.

[MPEG02] MPEG "Industry focus group," www.mpeg-7.org, November 2002.

[MUKE00] A. Mukerjee, K. Gupta, S. Nautal, and et al., "Concptual description of visual sences from linguistic models", *Image and Vision Computing*, (18), 2000, pp 173-187.

[OBEI01] M. Obeid, B. Jedynak and M. Daoudi, "Image Indexing and Retrieval Using Intermediate Features", *Proc. of ninth ACM (Association for Computing Machinery) International Conference on Multimedia*, Sep. 30 – Oct. 5, 2001, pp 531-533.

[ORLA00] Ratko Orlandic , Byunggu Yu, A Study of MBR-Based Spatial Access Methods: How Well They Perform in High-Dimensional Spaces, Proceedings of the 2000 International Symposium on Database Engineering and Applications, September 18-20, 2000, pp306-318.

[PASS99] G. Pass and R. Zabih, "Comparing images using joint histograms", "*Multimedia Systems*", 7(3), 1999, pp 234-240.

[PAYN99] J.S. Payne. L. Hepplewhite and T.J. Stonham "Perceptually based metrics for the evaluation of textual image retrieval methods". *In Proc. IEEE ICMCS'99*, pp 793-797.

[PENT96] A. Pentland, R.W. Picard, and S. Sclaroff. Photobook: Tools for Content-Base Mainpulation of Image Database, *Int. J. Comput. Vision*, 18(3), 1996, pp. 233-254.

[PRIC92] Price, R et al "Applying relevance feedback to a photo archival system" *Journal of Information Science* 18(3), 1992, 203-215

[PRIE87] Prieto-Diaz, R., and Freeman, P., "Classifying Software for Reusability", *IEEE Software*, v.4, no.1, January, 1987.

[QIN05] Tao QIN, Tie-Yan LIU, Xu-Dong ZHANG, Wei-Ying MA, and Hong-Jiang ZHANG, "Subspace Clustering and Label Propagation for Active Feedback in Image Retrieval", "*11th International Multimedia Modeling Conference MMM2005*", Jan-2005, Melbourne, pp 172-179.

[QUIN86] J. R. Quinlan. "Introduction on decision trees", Machine Learning, Vol 1, 1986, pp, 81-106.

[RABB89] Rabbitti, F and Stanchev, P "GRIM_DBMS: a graphical image database management system" in *Visual Database Systems* (Kunii, T, ed), Elsevier, Amsterdam, 1989, pp 415-430.

[RASM97] Rasmussen, E. M., “Indexing images”, “*Annual Review of information Science and Technology*”, (32), 1997, pp 167-196.

[ROBE94] S. E. Robertson, s. Walker, “Some Simple Effective Approximations to the 2-Poisson Meodel for Probabilistic weighted Retrieval”, “*Proceeding of the 17th ACM SIGIR Conference”*, 1994, pp. 232-241.

[RUI96] Y. Rui, C. Alfred, A.C. She, and T.S. Huang, “Modifies Fourier description for shape representation – a practical approach*”, “International Proc. of First International Workshop on Image Database and Multimedia Search*”, 1996.

[RUI97] Rui, Y et al “Relevance feedback techniques in interactive content-based image retrieval” in *Storage and Retrieval for Image and Video Databases VI* (Sethi, I K and Jain, R C, eds), Proc SPIE 3312, 1997, pp 25-36

[SAJJ97] A. Sajjanhar and G. Lu, " A grid based shape indexing and retrieval method", Special issue of Australian Computer Journal on Multimedia Storage and Archiving Systems, Vol. 29, No.4, November 1997, pp. 131-140.

[SALT83] Salton G. and McGill M., “Introduction to Modern Information Retrieval”, New York, McGraw-Hill, 1983.

[SAME90] Hanan Samet, “Applications of spatial data structures: Computer graphics, image processing, and GIS, Addison-Wesley Longman Publishing Co. Inc.”, Boston, MA, 1990

[SANT97] Santini, S and Jain, R C (1997) "The graphical specification of similarity queries" *Journal of Visual Languages and Computing* **7**, 1997, pp 403-421

[SANT98] S. Santini and R. Jain, "Beyond query by example.", "*In Proc. Int. Conf Visual Inf. Sys"*, 1998, pp 345-350.

[SELO90] G. A. Seloff, "Automated Access to the NASA-JSP Image Archives", *Library Trend, 38(4)*, Spring, 1990, pp 682-696.

[SETH95] Ishwar K. Sethi, "Image Computing for Visual Information Systems", *IEEE computer*, May 1995, pp 06-10.

[SHAH04] Biren Shah, Vijay Raghavan, and Praveen Dhatric, "Efficient and Feeective Content-based Image Retrieval using Space", "*10th International Multimedia Modeling Conference MMM2004*", Jan-2004, Brisbane, pp 279-284.

[SHAK96] Hussain Sabir Shakir and Makoto Nagao, "Context-Sensitive Processing of Semantic Queries in an Image Database System", *Information Processing and Management*, Vol.32, No.5, 1996, pp. 573-600.

[SHEI98] Gholamhosein Sheikholeslami, Wendy Chang and Aidong Zhang, "Semantic Clustering and Querying on Heterogeneous Features for Visual Data", ACM Multimedia, Sep. 1998, pp 03-12.

[SMEA96] Alan f. Smeaton and Ian Qigley. "Experiments on Using Sematic Distances Between Words in Image Caption Retrieval". *In Proceedings of ACM SIGIR Conference,* 1996.

[SMEU00] A. Smeulders, M. Worring, S. Santini, A. Gupta, and J. R. , "Content-based image retrieval at the end of the early years", "*IEEE Transaction on Pattern Analysis and Machine Intelligence"*, 22(21), 2000, , pp 1349-1380.

[SMIT03] John R. Smith, Milind Naphade and Apostol (Paul) Natsev," Multimedia Semantic Indexing Using Model Vectors", "*International Conference in Multimedia and Expo.*", 2003.

[SMIT96] J.R. Smith and S.F. Chang, "Automated binary texture feature sets for image retrieval", "*in Proceeding of IEEE International Conference Acoustic, Speech, Signal Processing"*, 1996.

[SMIT96a] J. Smith and S. Chang. "Visualseek: a fully automated content-based image query system", *"In Proc. Of ACM Multimedia"*, 1996, pp 87-98.

[SMIT97] Smith J R and Chang S F "Querying by color regions using the VisualSEEk content-based visual query system" *Intelligent Multimedia Information Retrieval* (Maybury, M T, ed). AAAI Press, Menlo Park, CA, 1997, pp 23-41.

[SMIT97a] Smith J R and Chang S F "An image and video search engine for the World-Wide Web" in *Storage and Retrieval for Image and Video Databases V* (Sethi, I K and Jain, R C, eds), Proc SPIE 3022, 1997, pp 84-95.

[SONG99] Ju-Won Song , Kyu-Young Whang , Young-Koo Lee , Sang-Wook Kim, The clustering Property of Corner Transformation for Spatial Database Applications, 23rd International Computer Software and Applications Conference, October 19-26, 1999, pp.28-35.

[SPAR81] Sparck-Jones, K., Ed., *Information Retrieval Experiment*, London, Butterworths, 1981.

[STRI95] M. Stricker and M. Orengo, "Similarity of colour image", "*in Proceedings of SPIE Storage and Retrieval for Image and Video Database*", 1995

[SUTA99] Dwi Sutano and C.H.C.Leung, "Automatic Index Expansion for Concept-based Image Query", *Proceedings of the International Conference on Visual Information Systems*, June 1999, pp 399-408.

[SRIH93] Rohini K. Srihari, "Intelligent document understanding: Understanding photographs with captions", *IEEE computer society press*, Oct 1993, pp 664-667.

[SUTA99] Dwi Sutano and C.H.C.Leung, "Automatic Index Expansion for Concept-based Image Query", *Proceedings of the International Conference on Visual Information Systems*, June 1999, pp 399-408.

[SWAI91] M. Swain and D. Ballard, "Colour indexing", "*International Journal of Computer Vision*", 7(1), 1991, pp 11-32.

[SZUM98] M. Szummer and R. W. Picard, "Indoor-outdoor image classification", in IEEE CVPR'98 Workshop on Content-Based Access of Image and Video Libraries, 1998.

[TAM01] A. Tam and C.H.C. Leung "Structured natural-Language content retrieval of visual materials", *Journal of American Society for information Science and Technology*, 2001, pp. 930-937.

[TANG96] A.K.L.Tang, "Content–based image query" Brigham Young University, Department of Computer Science, April 1996.

[TASI03] Technical Advisory Service for Images (TASI), University of Bristol, UK - February 2003

[TURK92] Turksen, I.B., Tain, Y. and Berg, M. "A Fuzzy expert system for servis centre of spare parts", "*Expert Systems with Applications*", 1992, 5, pp 447-464.

[VELL98] Vellaikal, A. and Kuo, C.C.J. “Hierarchical clustering techniques for image database organization and summarization” in *Multimedia Storage and Archiving Systems III* , Proc SPIE 3527, 1998, pp 68-79.

[WITT99] Ian H. Witten, Alistair Moffat, Timothy C.Bell “Managing Gigabyte? Evaluation retrieval effectiveness”, *Morgan Kaufman publishers Inc.*1999, pp 188-190.

[YANG97] Li Yang and Jiankang Wu, “Towards a Semantic Image Database System”, *Data and Knowledge Engineering 22*, 1997, pp 207-227.

[YONG99] Rui, Yong, Thomas S. Huang, and Shih-Fu Chang. "Image retrieval: Current techniques, promising directions and open issues." *Journal of Visual and Communication and Image Representation* 10, no. 1, 1999, pp 39-62

[ZHAN04] Danqing Zhang, Binh Pham, and Yuefeng Li, “Modelling Traditional Chinese Paintings for Content-Based Image Classification”, “*10th International Multimedia Modeling Conference MMM2004*”, Jan-2004, Brisbane, pp 258-264.

[ZHEN99] Zheng, M., “Metadata elements for object description and representation: A case report from a digitized historical fashion collection project”, *Journal of the American Society for Information Science*, 50(13), 1999, pp 1193-1208.

[ZHOU00] X. S. Zhpu and T. S. Huang, "CBIR: From Low-Level Features to High-Level Semantics", "*Proc. SPIE Image and Video Communication and Processing*", San Jose, Ca. Jan-2000.

[ZIMB98] Geraldo Zimbrao , Jano Moreira de Souza, "A Raster Approximation For Processing of Spatial Joins", *Proceedings of the 24th International Conference on Very Large Data Bases,* August 24-27, 1998 , p.558-569.

Appendix A: System Pages

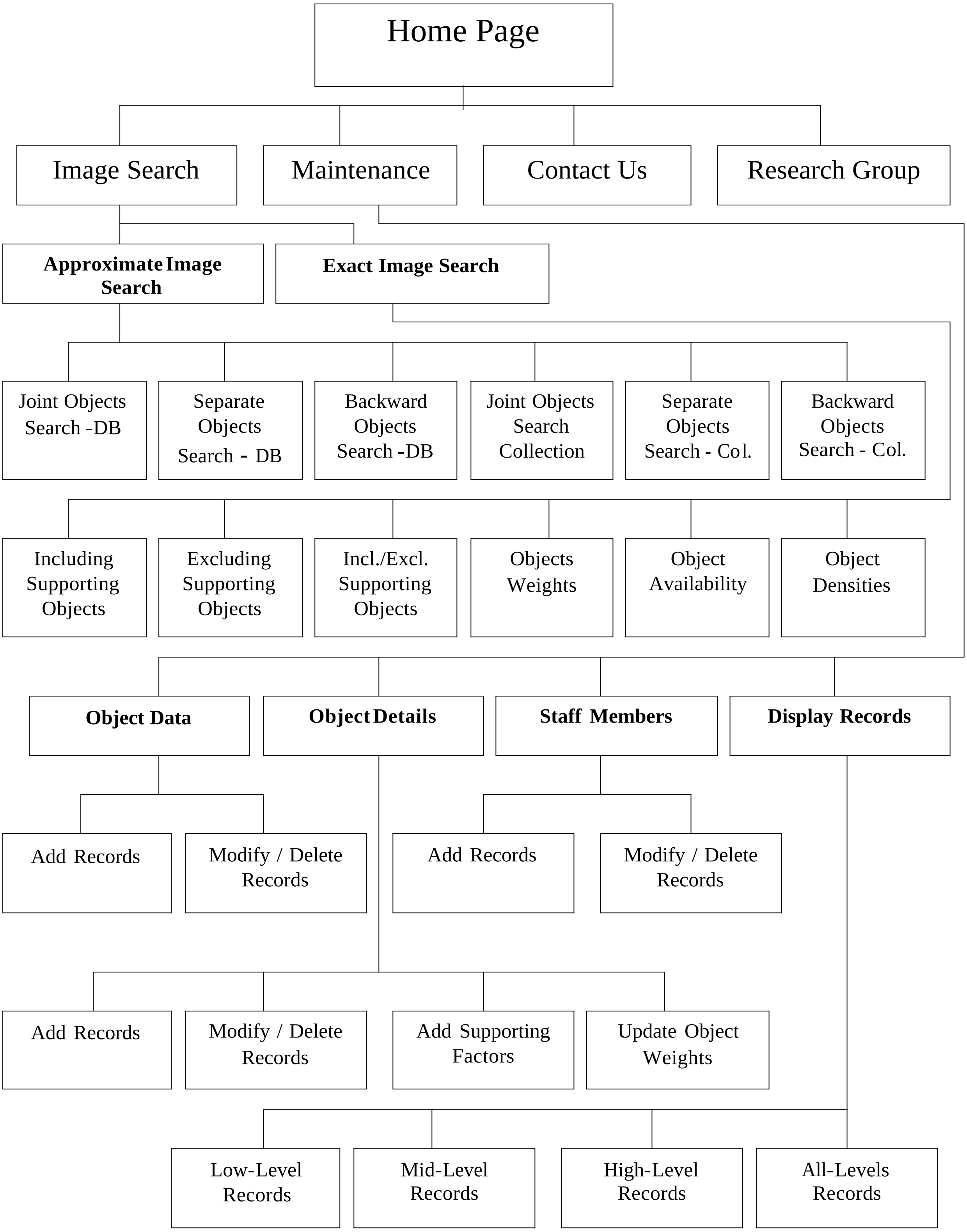


Figure A.1 The System's Pages Structure

The Main Page

As usual every project has a home page with all the other pages linked together, which can be accessible via this home page. The first page in this project is home.asp, its address appears in the browser address field as shown in Figure A.2.

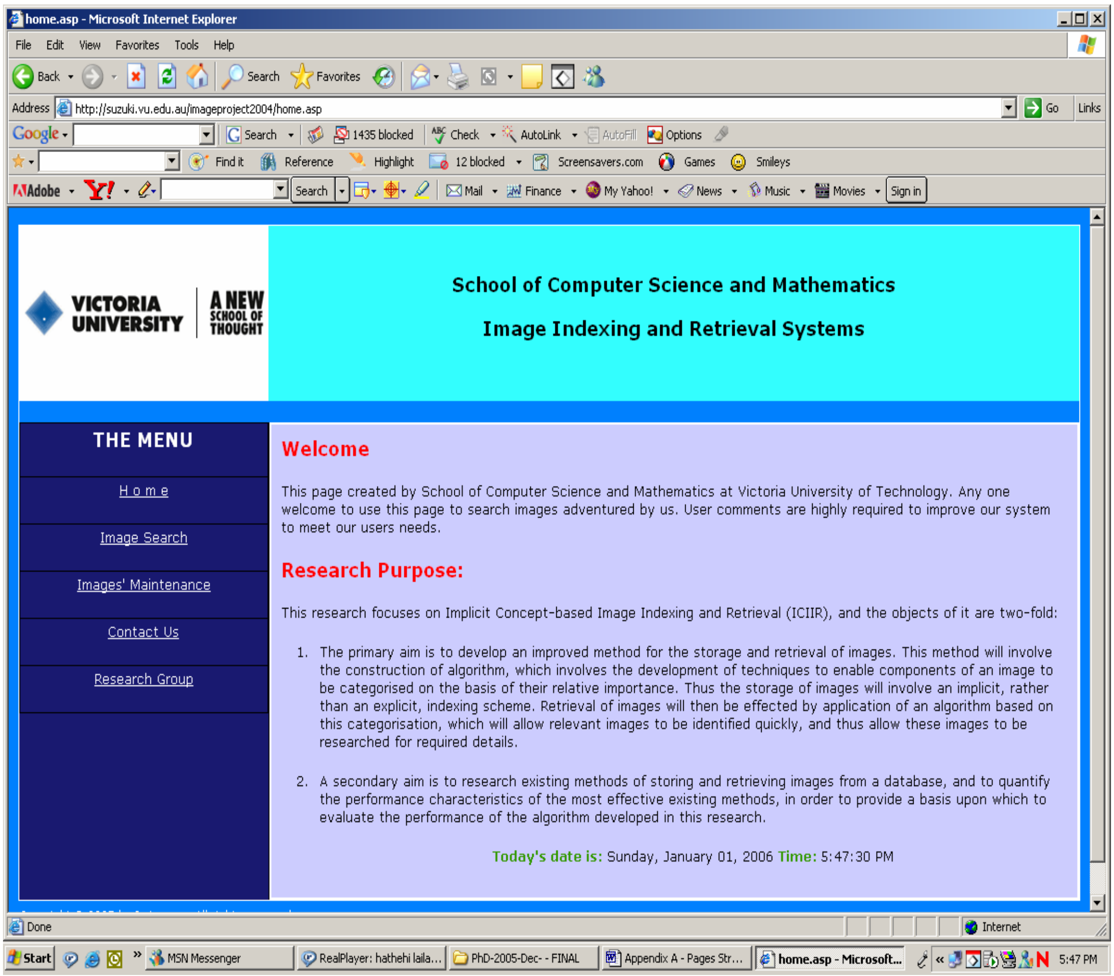


Figure A.2 Project’s Main Page

The home page “Project’s Main Page” contains many options including:

- *Home Page:* the starting point.
- *Image Search:* To search for and display desired images from the image collections. This option contains many sub-options of image searching, with every image incorporated with some information.
- *Image Maintenance:* This option is used by the Staff/Administrators and allows them to add, modify or delete records and give them the power to display all records in tables from the database.
- *Contact us:* Display the methods of contacting the Staff/Administrators.
- *Research Group:* Display the research group and related information (e-mail address, wed site, etc).

The following sections explains the uses of all options and sub-options of our system and examines their purposes, how to use them and includes the image of different pages used for every option.

Image Search

The user selects the ‘Image Search’ option; the main page of Image Search appears containing two main options as shown in Figure A.3.

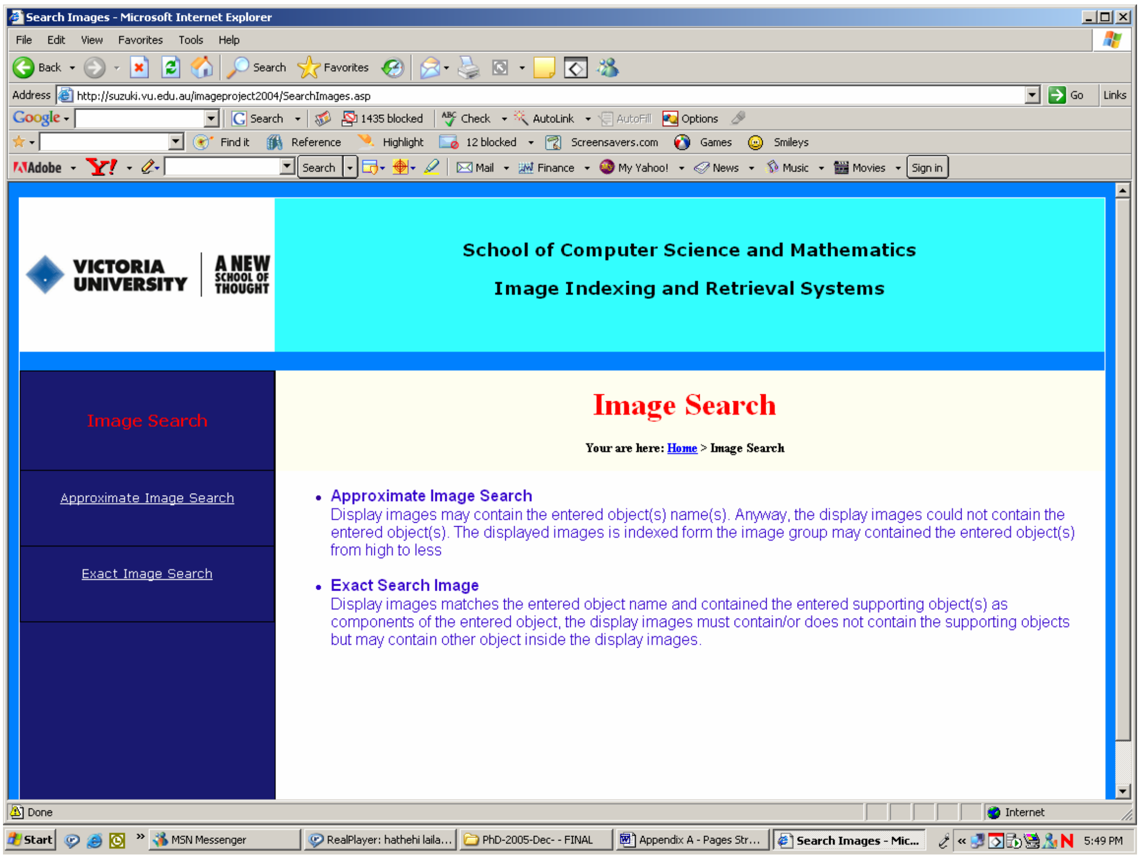


Figure A.3 Image Search Main Page

Approximate Image Search

When the user selects the first option of Image Search, an Approximate Image Search page appears with six sub-options under this option, three of them used for different search objects from our database and the remaining three used for search objects from specific image collection, these sub-options are shown in Figure A.4. and are:

- DB Joint Objects Search.
- DB Separate Objects Search.
- DB Backward Search.

- Collection Joint Objects Search.
- Collection Separate Objects Search.
- Collection Backward Search.

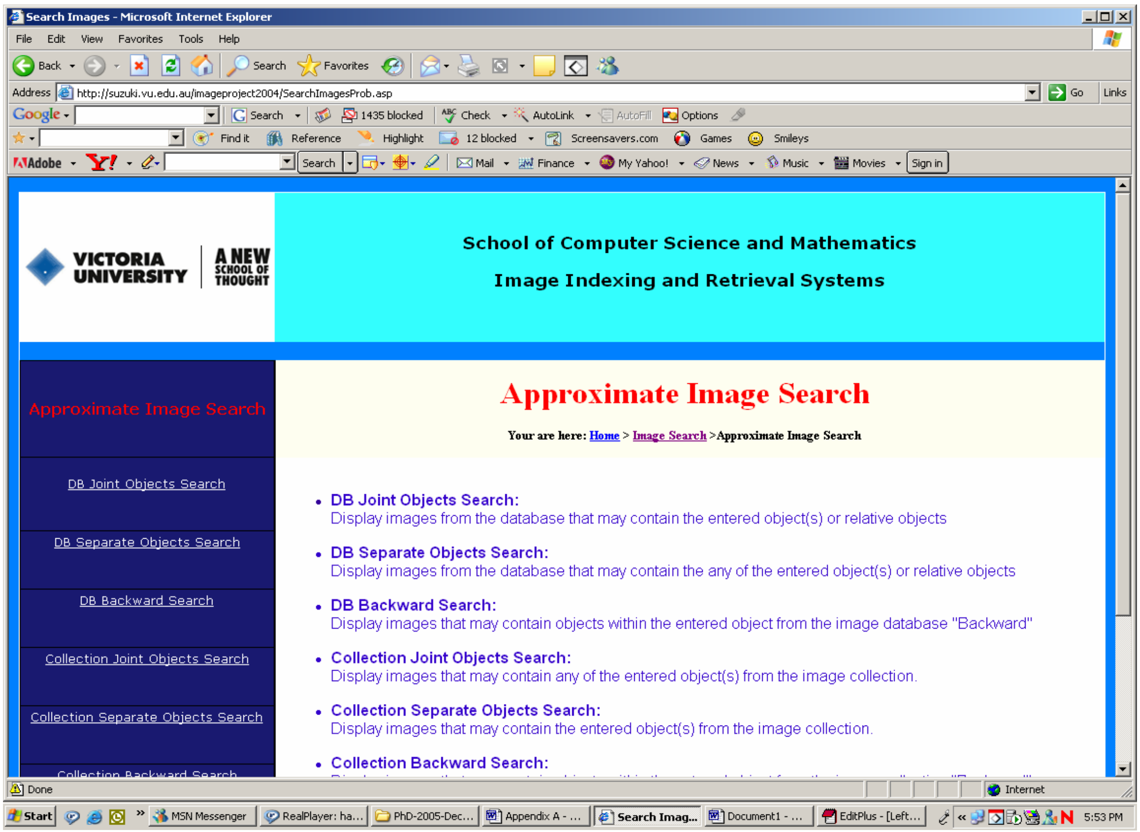


Figure A.4 Approximate Image Search Main Page

DB Joint Objects Search.

By selecting the first sub-option of the Approximate Image search, a “Joint Object Search” page appears, the user enters the objects names that may be found in some image categories and then submits the search, a list of images is displayed, these images are displayed in pages as shown in Figure A.5, with every page containing 20 images. The number of pages may contains relevant images is also displayed, giving the user the ability to a specific page.

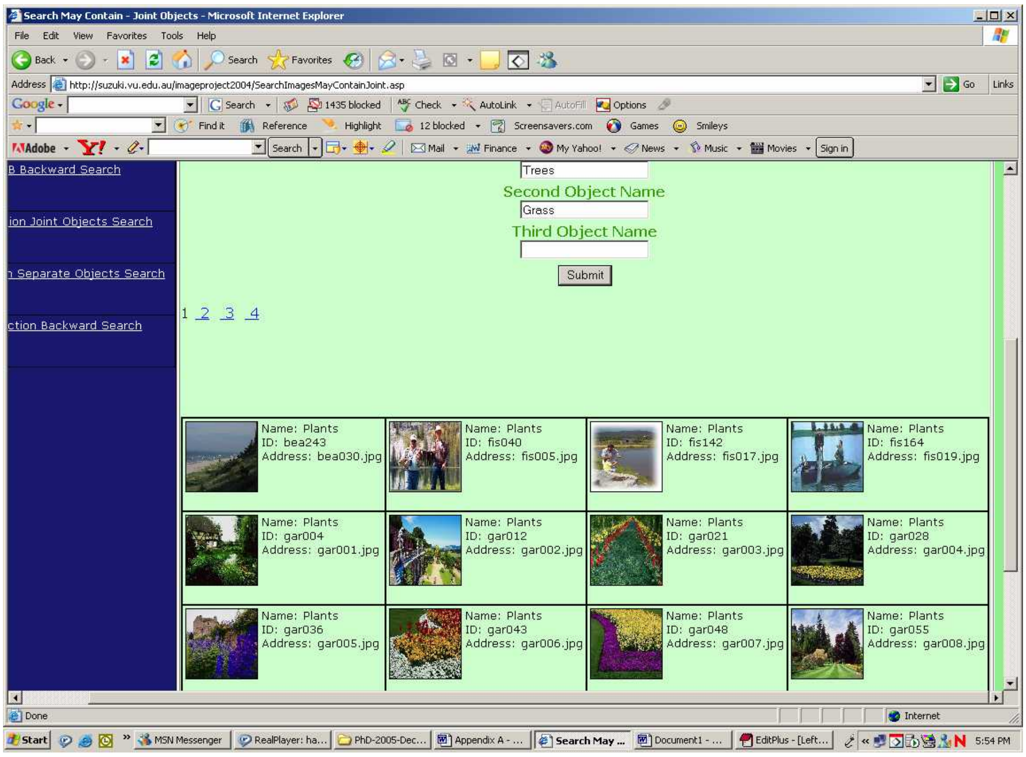


Figure A.5 DB Joint Objects Search

The user can enlarge any image of the displayed images by clicking on it; the image enlargement is shown in Figure A.6.

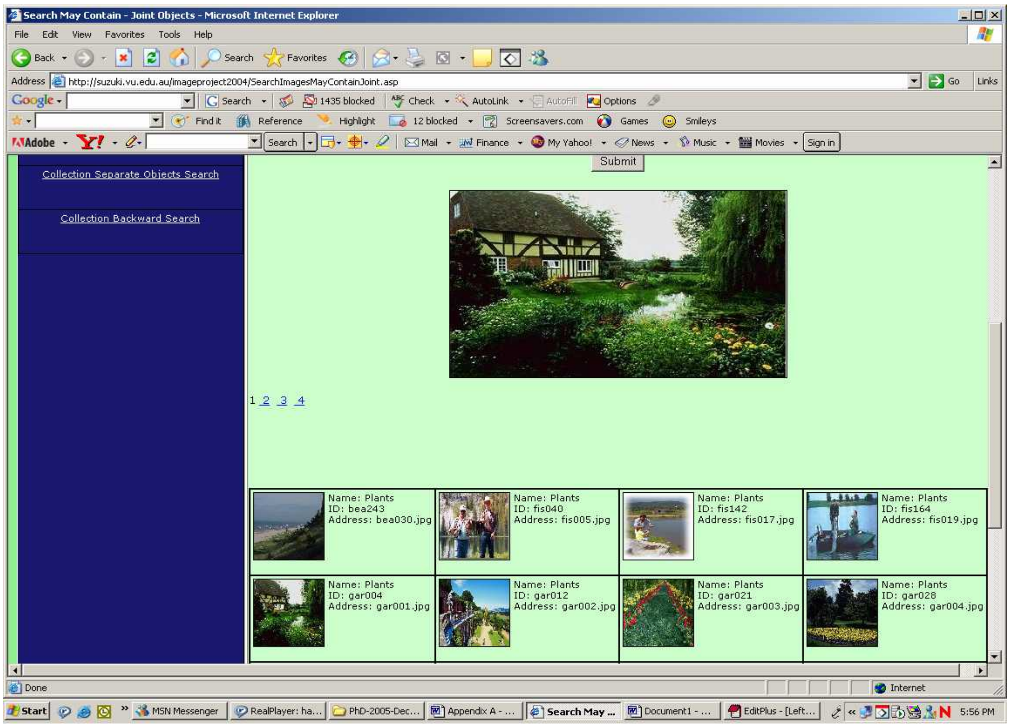


Figure A.6 Image Enlargement

DB Separate Objects Search.

This sub-option uses the same image retrieval method as "Joint Objects Search" with the query varying as the "Separate Objects Search" searches for any image that may contains the entered objects, commonly resulting in a greater amount of retrieved images such as figure A.7.

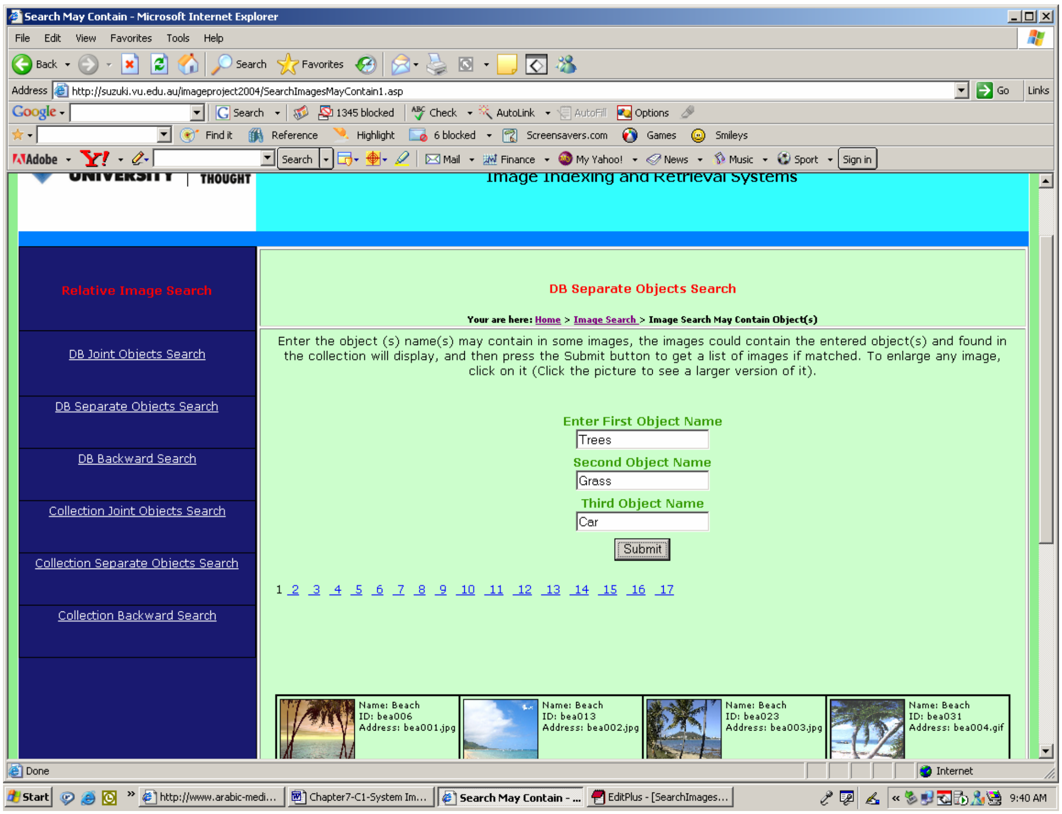


Figure A.7 DB Separate Objects Search

DB Backward Search

The third sub-option of Approximate Image Search is used to search objects which may be included in a specific object; this option is called "Backward Objects

Search". Figure A.8 show the displayed images when searching for the objects that may contain in the object of "Wrestling".

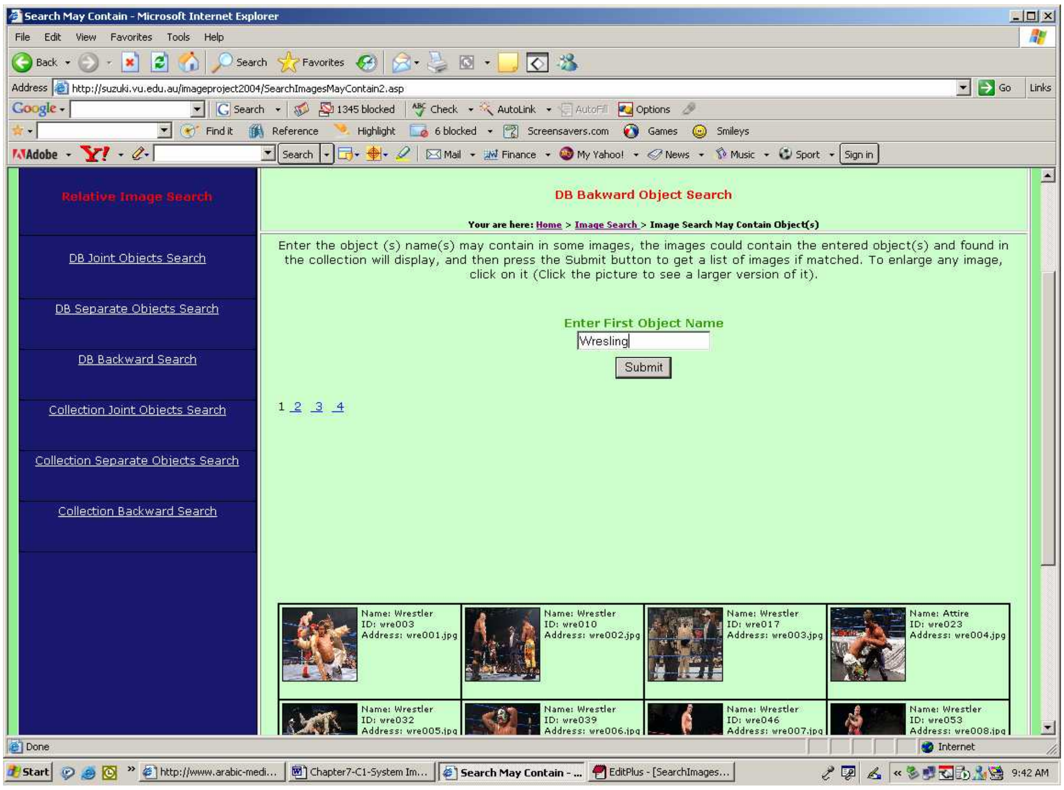


Figure A.8 DB Backward Objects Search

The last three sub-options of the Approximate Image Search are similar to the first three sub-options. The first three sub-options retrieve images from the database where as the last three sub-options retrieve objects from a specific image collection.

Exact Image Search

When the user selects the second option of 'Image Search Home Page, the 'Exact Image Search' page appears containing many different sub-options explained in details in this section. These sub-options are listed below and shown in Figure A.9:

- Search Images Included Specific Supporting Object(s).
- Search Images Excluded Specific Supporting Objects.
- Search Images Included and Excluded Specific Supporting Objects.
- Search Images using different weights of Supporting Objects.
- Search Images using Images Availabilities.
- Search Images using different densities of Supporting Objects.

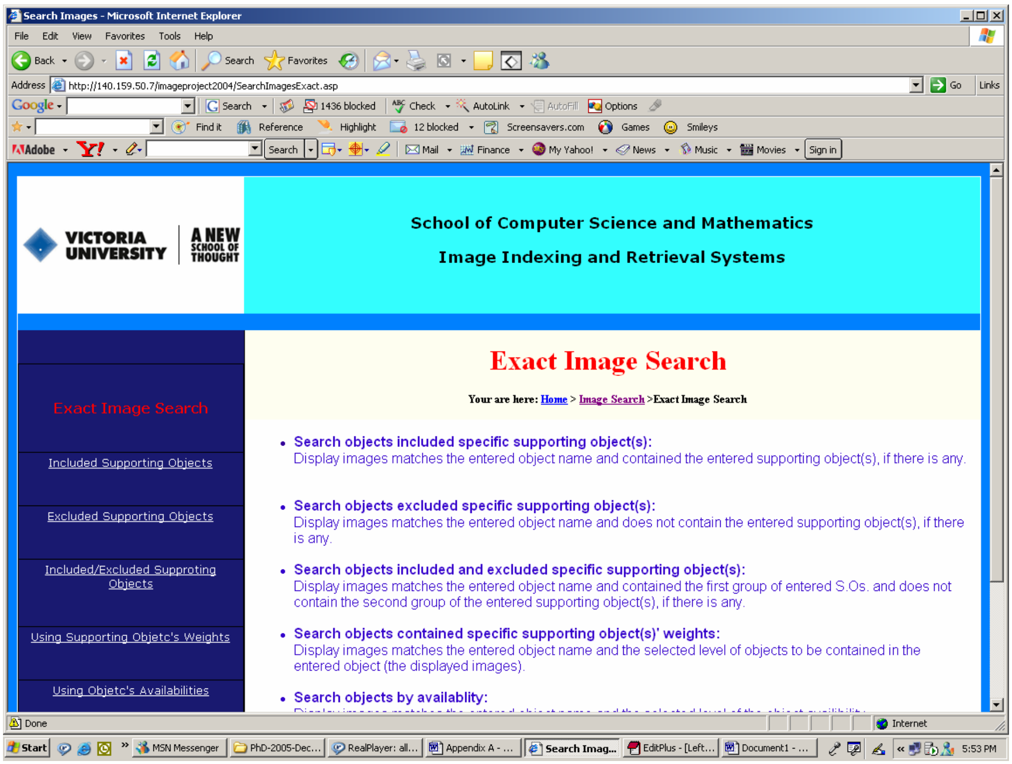


Figure A.9 Exact Image Search Main Page

Image *Search Including Specific Supporting Objects*

When the user selects the first sub-option of the Exact Image Search, a new page will appear. The user enters the desired object name in the first field. In the next fields, the user can enter up to six supporting object names to be components in the desired images, which are optional and then submit the query. The system checks the object name, if not found it will check the thesaurus for the entered name and when found, the system will look for all images having the name found in the thesaurus. The same procedure used to check the object name is used for the entered supporting objects' names, if there are any. The images that match the user's search will be listed on the screen as shown in Figure A.10.

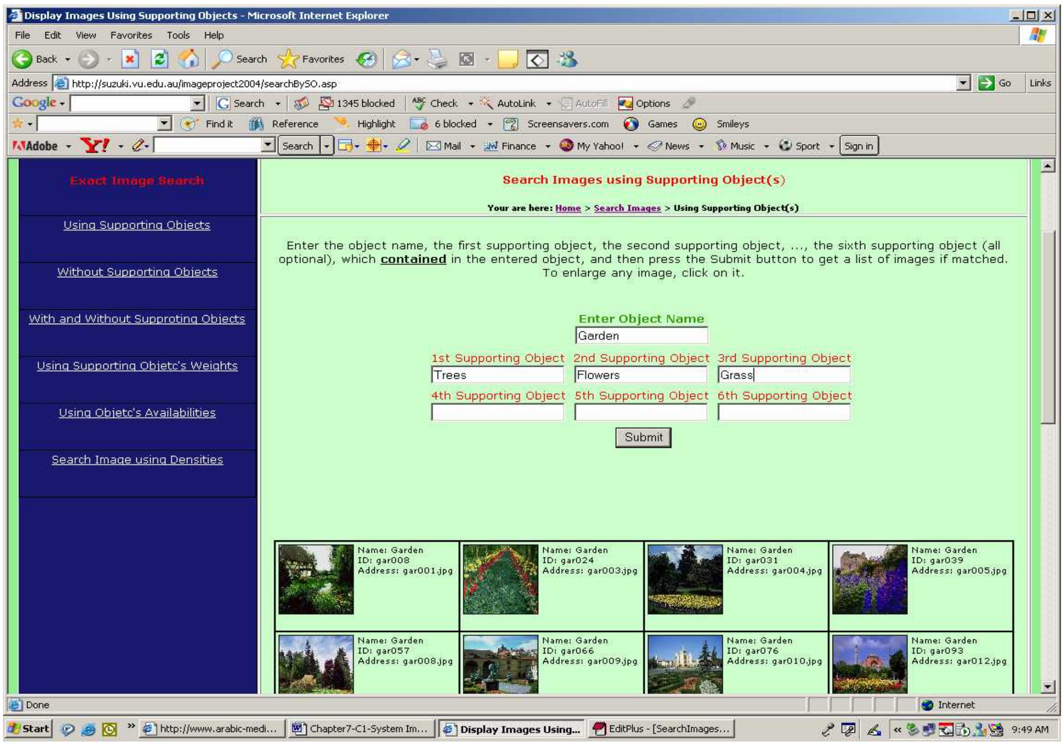


Figure A.10 Image Search using Supporting Object(s)

Image Search Excluding Specific Supporting Objects

When the user selects the second sub-option of Exact Image Search Page, a new page will appear. The user enters the desired object name in the first field. On the other fields, the user can enter the supporting objects names that are not contained in the desired images, and then press the Submit button. The images that match the user's search will be listed on the screen as shown in Figure A.11.

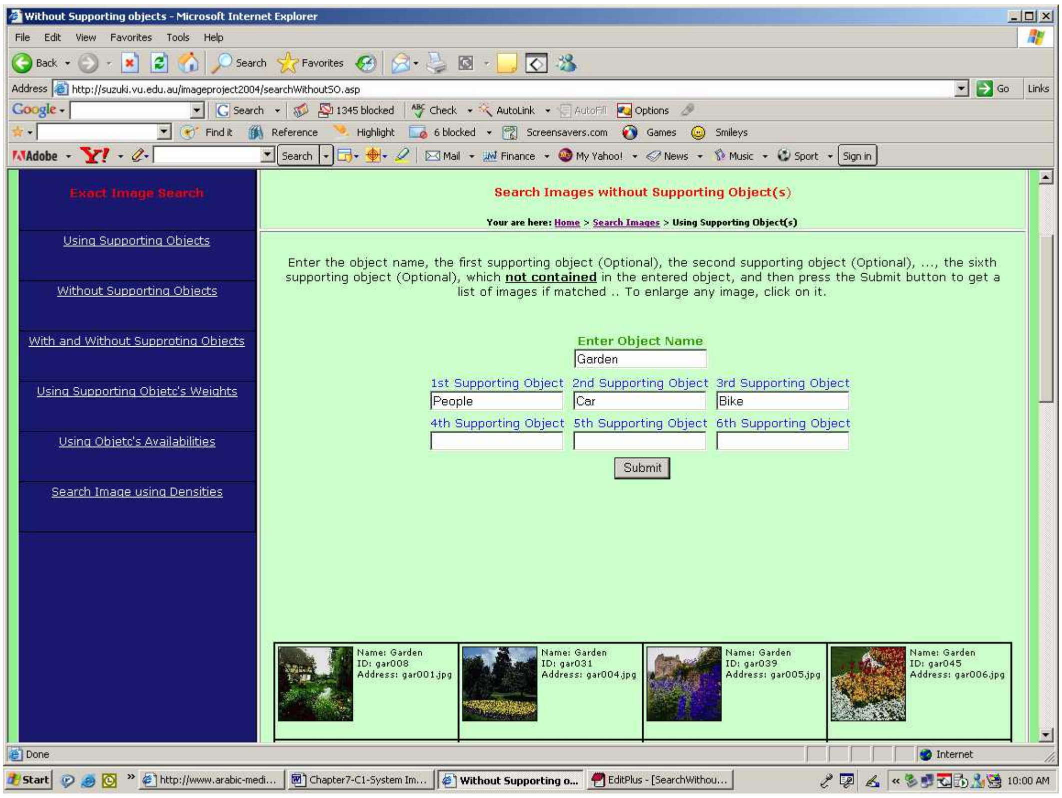


Figure A.11 Image Search without Supporting Objects

Image Search Including/Excluding Specific Supporting Objects

The third sub-option is to display images which containing specific objects and exclude some other specific objects. The user enters the object name and then enters up to three supporting objects to be represents in the display images and also enters up to three supporting objects that are to be excluded in the display images. The procedure is similar to any sub-option of searching images which contain supporting objects or searching images which do not contain specific supporting objects, this sub-option shown in Figure A.12.

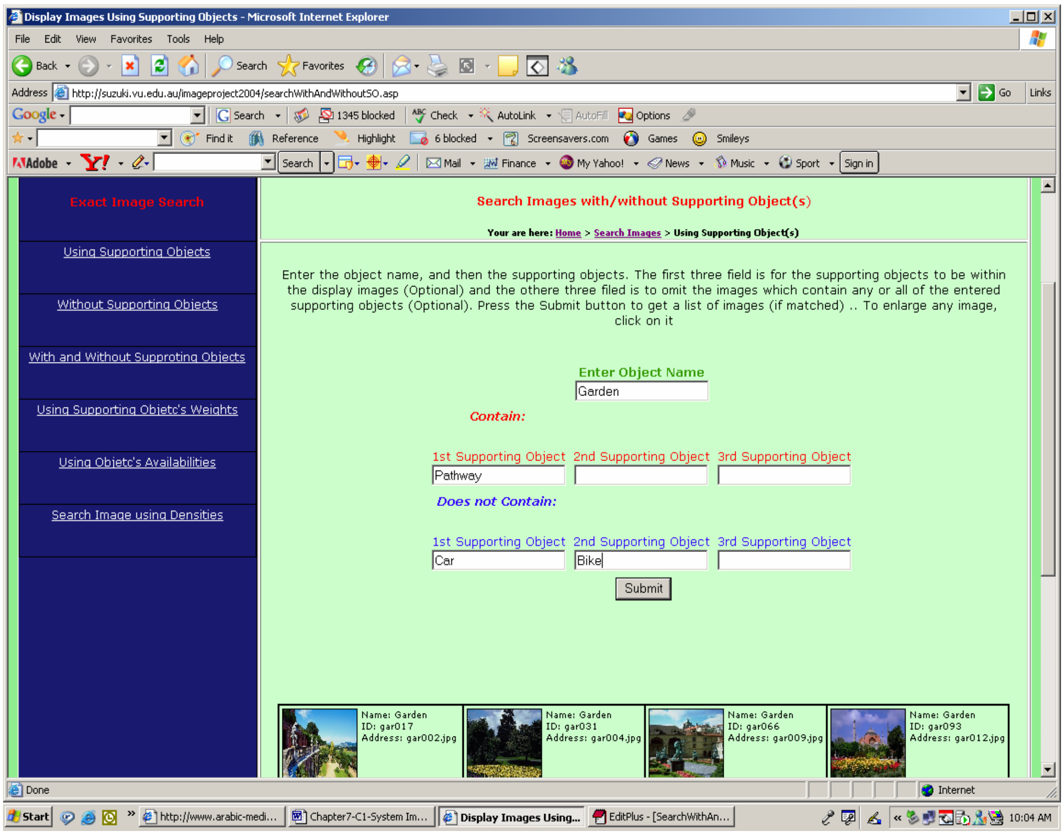


Figure A.12 Image Search with/and without Supporting Object(s)

Image Search using Weights of Supporting Objects

When the user selects the fourth sub-option of the Exact Image Search Page, a new page will appear with two variables (fields). The user enters the desired object name, and then selects the supporting objects’ weights levels to be included in the displayed images from the drop-down menu and then submit the search. The system checks the entered object name and the selected weight level, when found, a list of images is displayed as shown in Figure A.13.

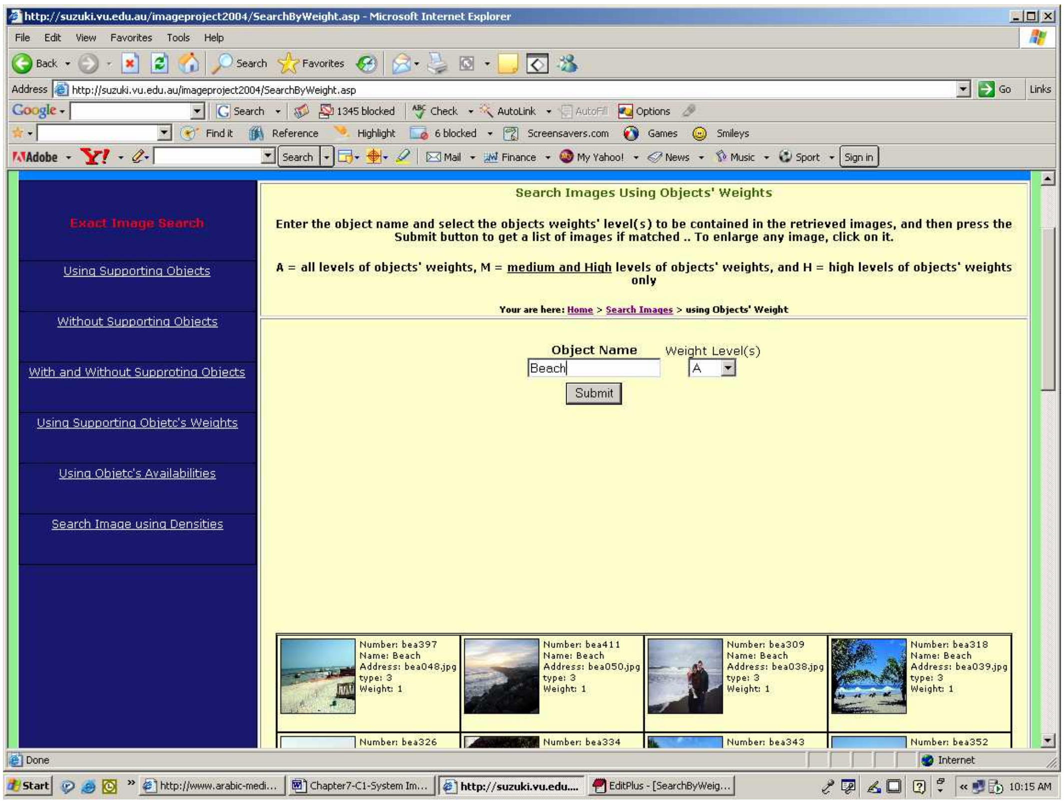


Figure A.13 Image Search using Objects Weights

Image Search using Availabilities

The user selects the fifth sub-option of the Exact Image Search Page; the Search Images using Availability sub-option is displayed. The user enters the object name and selects the object availability levels from the drop-down menu then submit the search, the system checks the entered object name and the selected availability level, when found, a list of images is displayed as shown in Figure A.14.

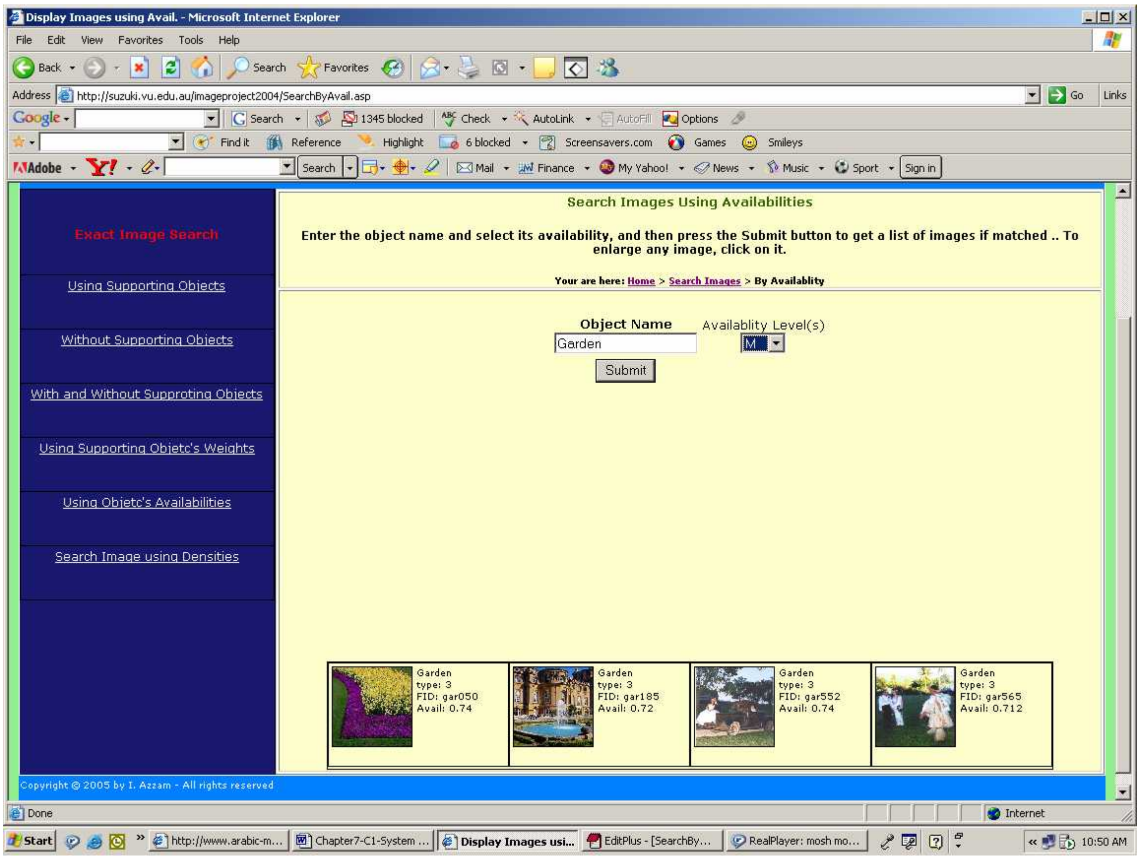


Figure A.14 Image Search using Availabilities

Image Search using Densities

This sub-option of image searching uses density functions in the same manner as search images using supporting objects; it uses an extra condition for every supporting object, which is the level of its density to be selected from the drop-down menu. The user enters the image name, the supporting objects and selects the density levels of the entered supporting objects and then submits the search in order to get a list of desired images. The system checks the entered object name, the supporting objects names and the density level of every entered supporting object, and when found, a list of images displayed as shown in Figure A.15.

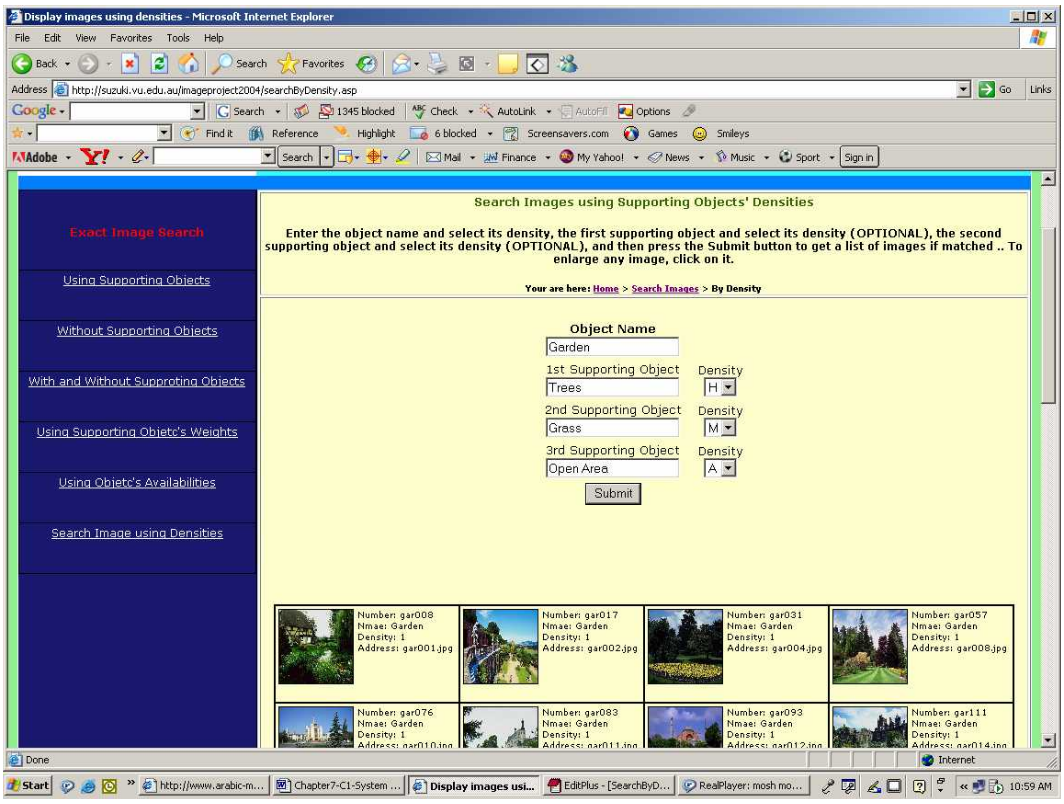


Figure A.15 Search Images using Supporting Objects' Densities

Image Maintenance

The Image Maintenance option is the second option of the Home page navigation menu. This page consists of many sub-options, and each sub-option has its specific role. The sub-options of Image Maintenance start when staff/administrators clicks the Image Maintenance option from the Home page, a Login page appears as shown in Figure A.16.

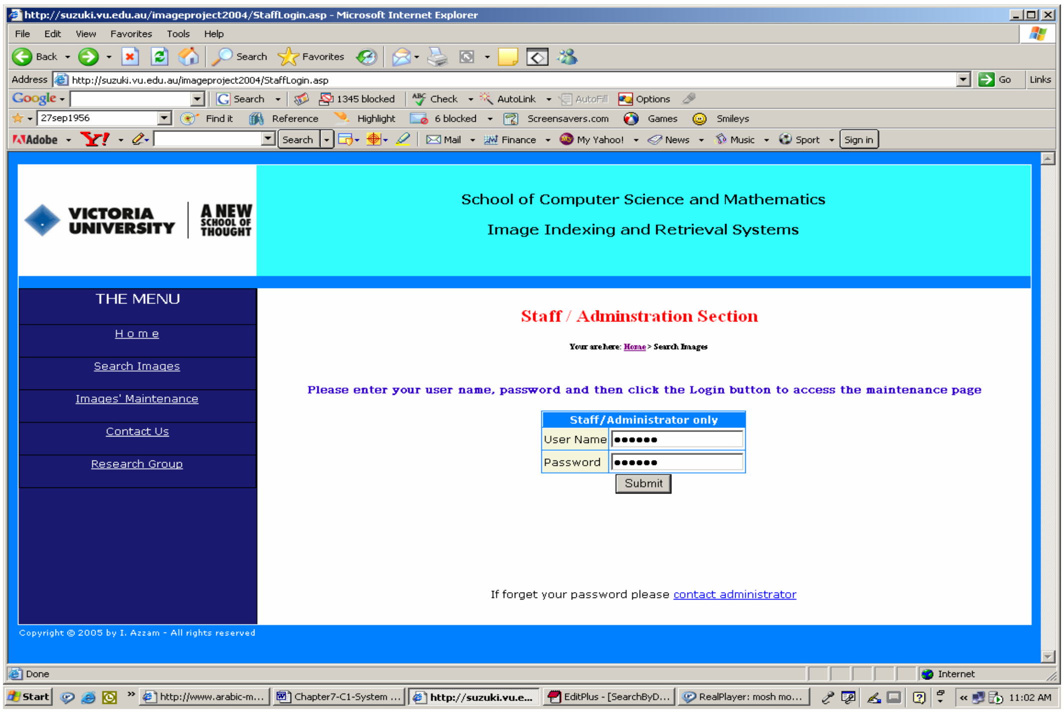


Figure A.16 Staff/Administration Main Page

The staff/administrator has to enter user name and password, if an eligible user name and password are entered, the next page displayed will contain the different sub-options of the Image Maintenance page as shown in Figure A.17, if not successful, they must either try again or check their user names and passwords with the administrators.

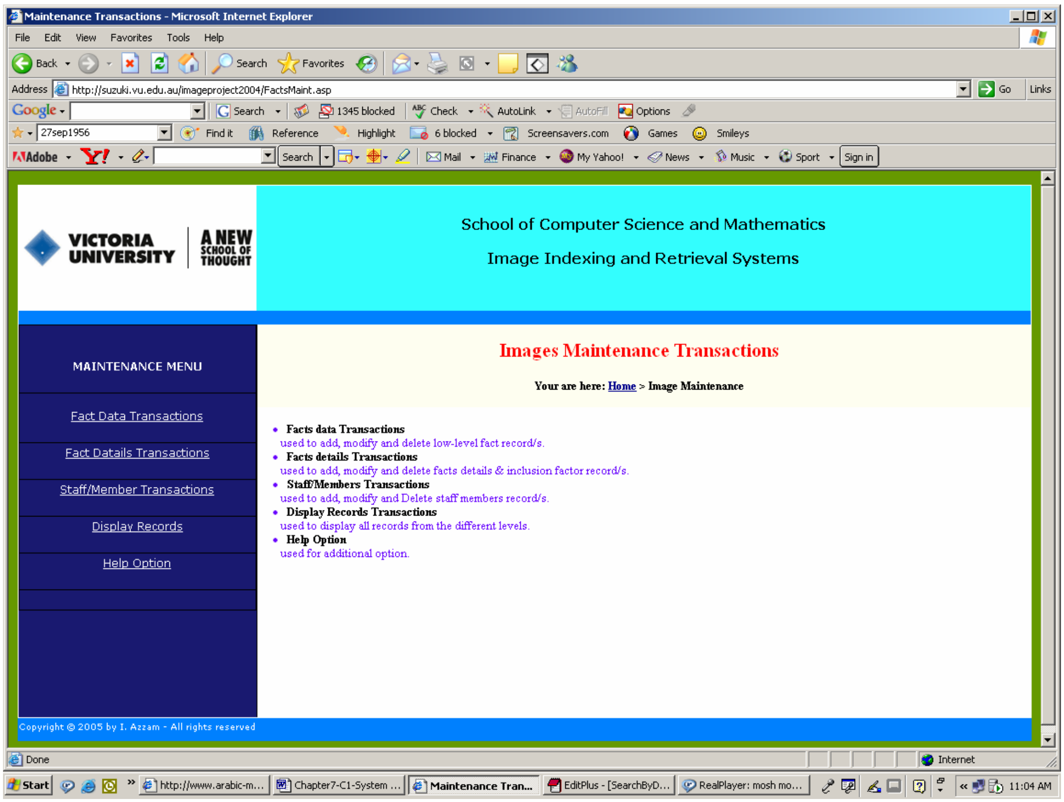


Figure A.17 Image Maintenance Transaction Main Page

The staff/administrator maintains the database from time to time; they create new records, modify and delete records as well as display records of the different object levels of objects in database. The maintenance transactions sub-options are illustrated in this section.

Object Data Transactions

When the staff/administrator selects the Object Data Transaction option, the Object Data Transaction page will appear as shown in Figure A.18. This page has two sub-options; the first sub-option is used to add a new record and the second sub-option is to modify and delete a record from an object data file.

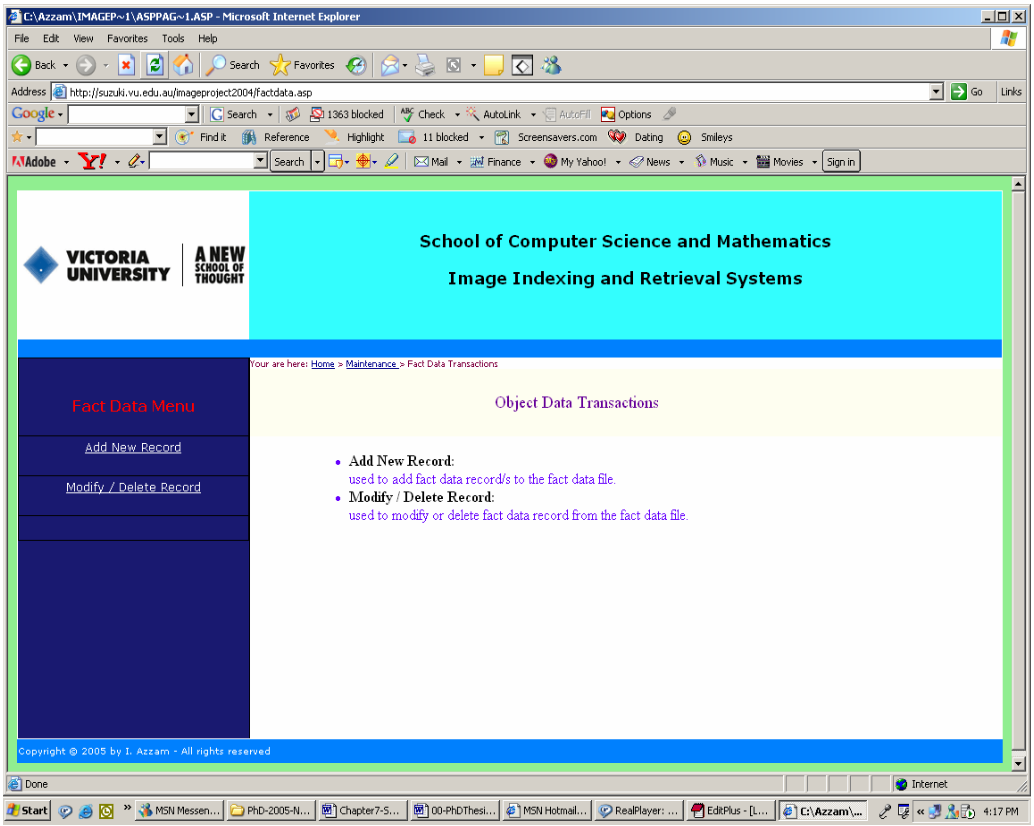


Figure A.18 Object Data Transaction Page

- *Add a New Record*

If the staff/administrator selects the first option, Add a New Record Page appear, as shown in Figure A.19. The staff/administrator fills the form and then presses the Submit addition button, the record will be saved in object data file; here the staff/administrator can enter as many records as needed.

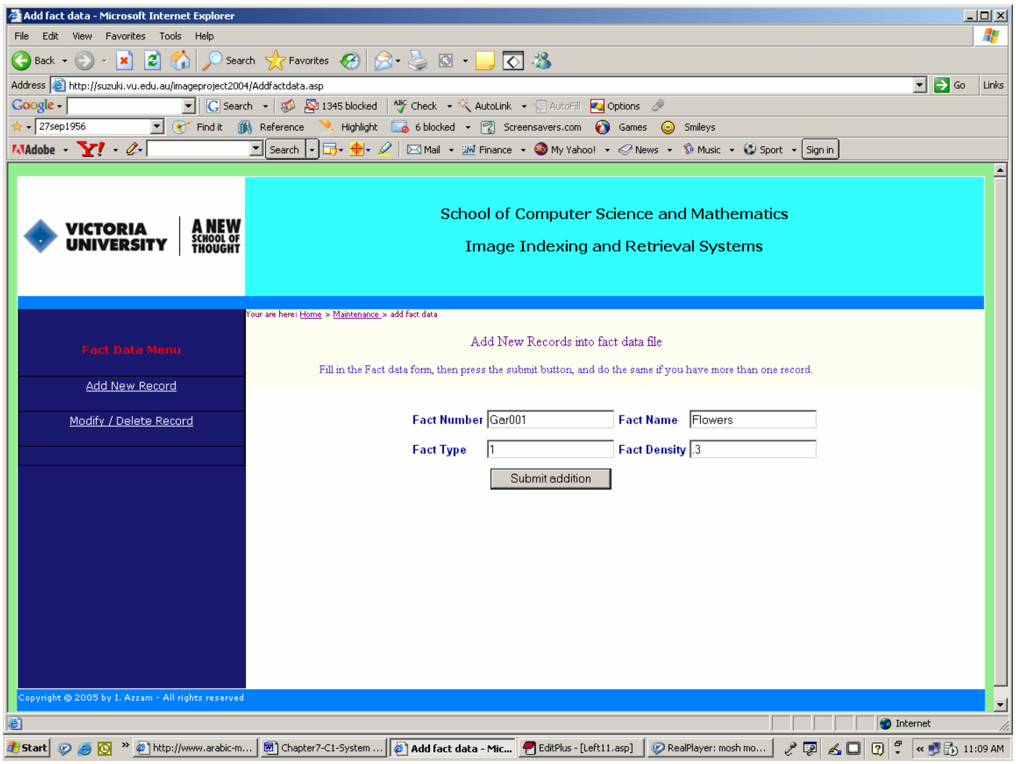


Figure A.19 Add New Record “Object Data”

- *Modify or Delete Record*

The next page is used to modify and/or to delete a record. When the staff/administrators choose the second sub-option of Object Data Transaction Page, a Modify/Delete Record Page will appear as shown in Figure A.20; the staff/administrator enters the object number in which they wish to modify or delete, and then submit their query.

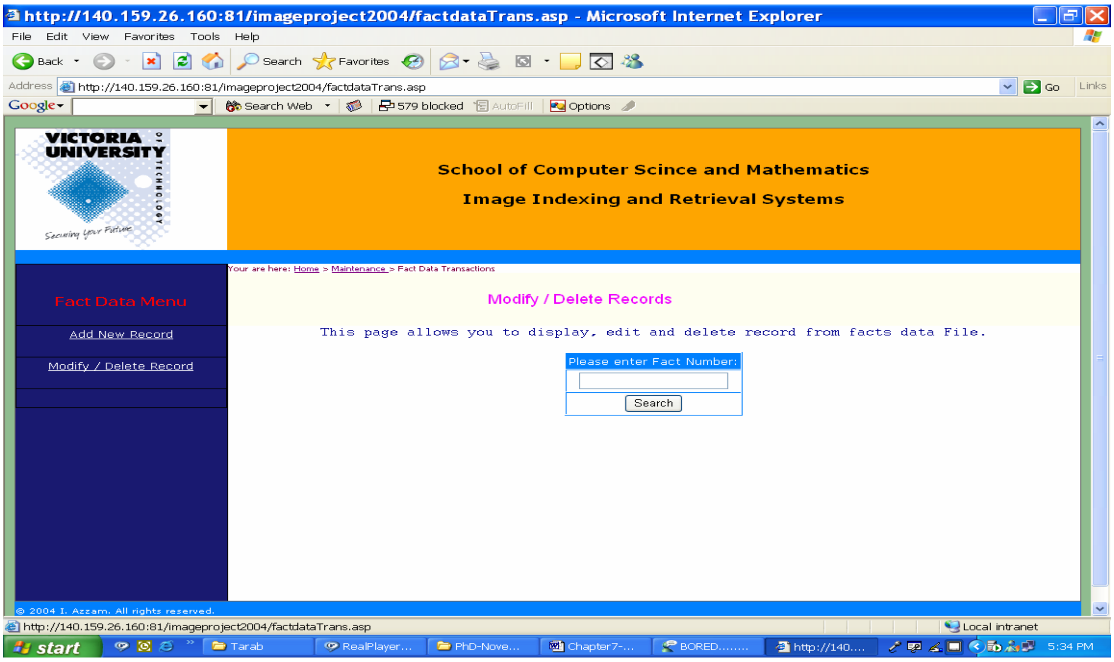


Figure A.20 Modify or Delete Record “Enter Object Data Number“

The system checks the entered number, if found, a new page will display as shown in Figure A.21, containing the desired record, at the end of the appeared record there are a links to Modify or Delete the record.

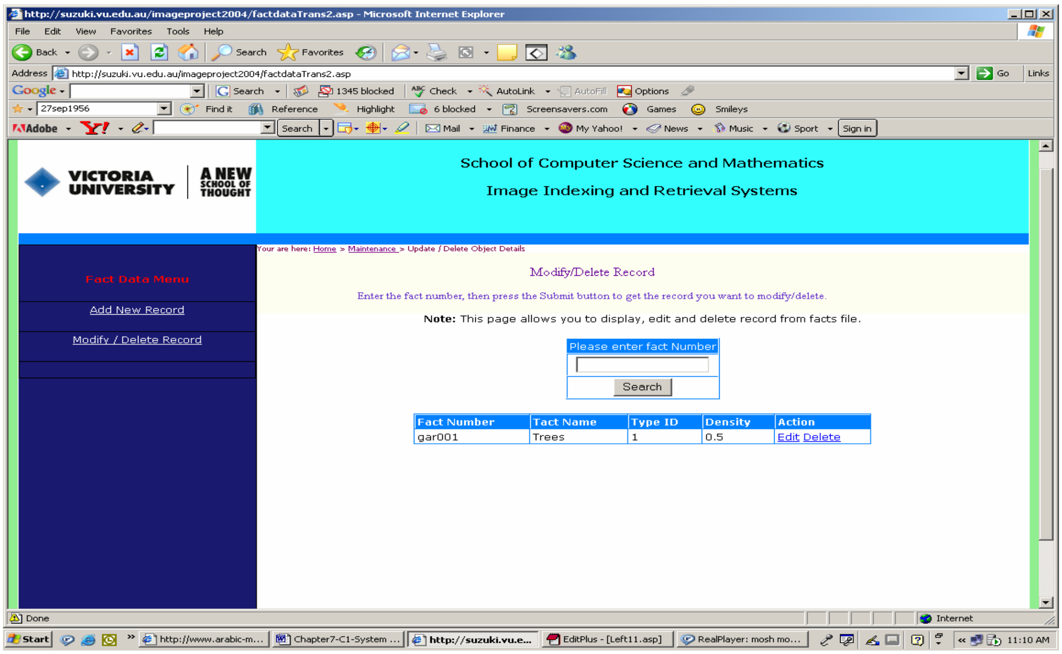


Figure A.21 Modify /Delete Record “Display Object Data Record “

If the staff/administrator selects Modify link, a new page will appear containing all the information about the desired object as shown in Figure A.22, the staff/administrator can make the required changes and click the Update button to save the changes.

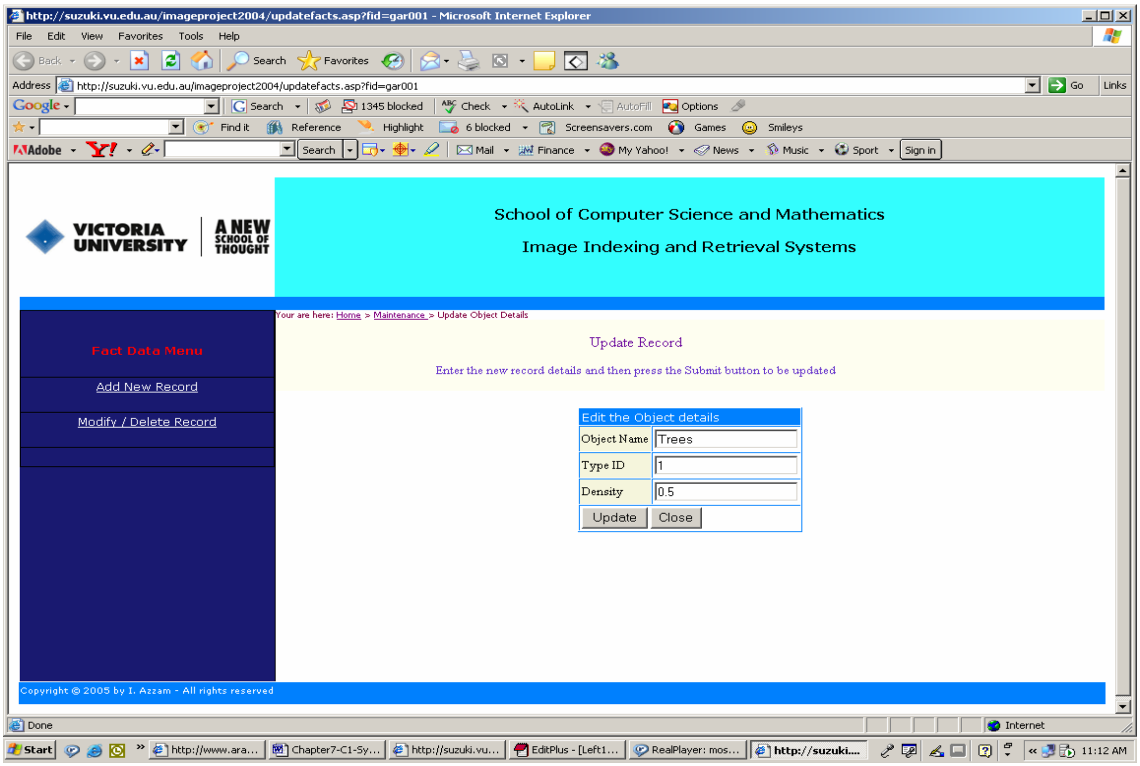


Figure A.22 Modify Record of Object Data

When the staff/administrator selects the Delete link, a warning message will displayed, if the staff/administrator selects the delete option, the record will be deleted from the database and a message will appear showing that the record has been deleted successfully.

Object Details Transactions

When the staff/administrator selects the second option of Image Maintenance Main Page, the Object Details Transaction option page appear, the Object Details Transaction page will appear as shown in Figure A.23. This page contains four sub-options as follow:

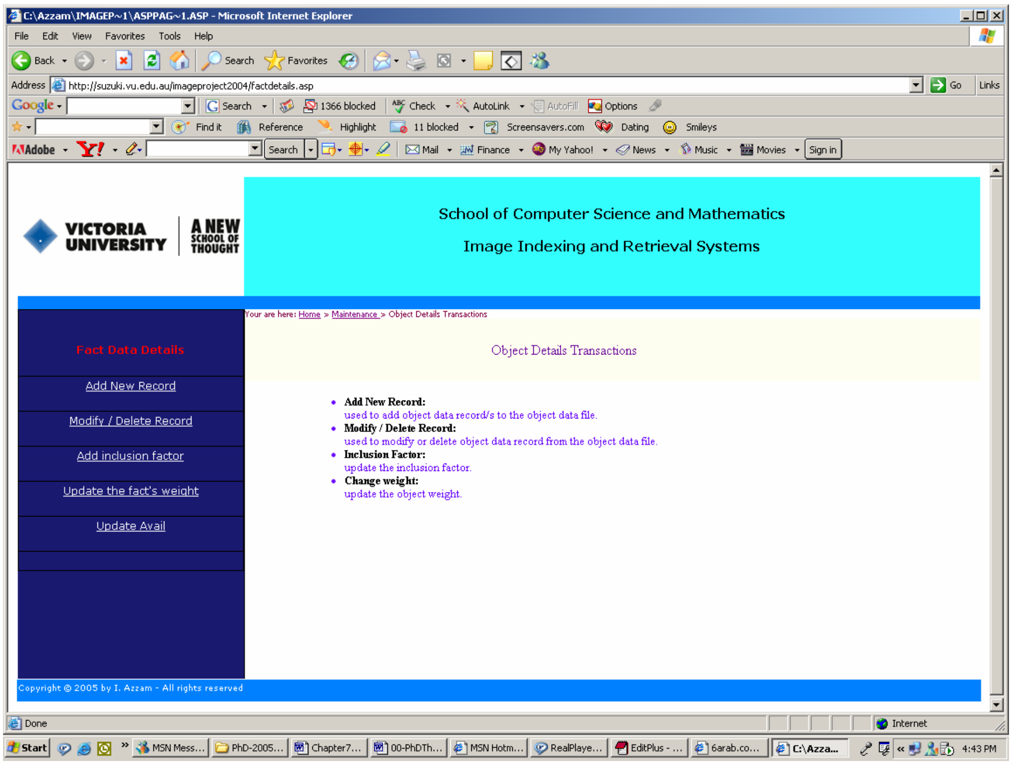


Figure A.23 Object Details Transaction Page

- *Add a New Record*

The first sub-option of the "Object Details Transaction" is the "Add New Record" option used to add new record.

The procedures of adding a record to the object details file follow the same procedure as in Object Data Transaction explained previously in this section. The Add Object Details record transaction is shown in Figure A.24.

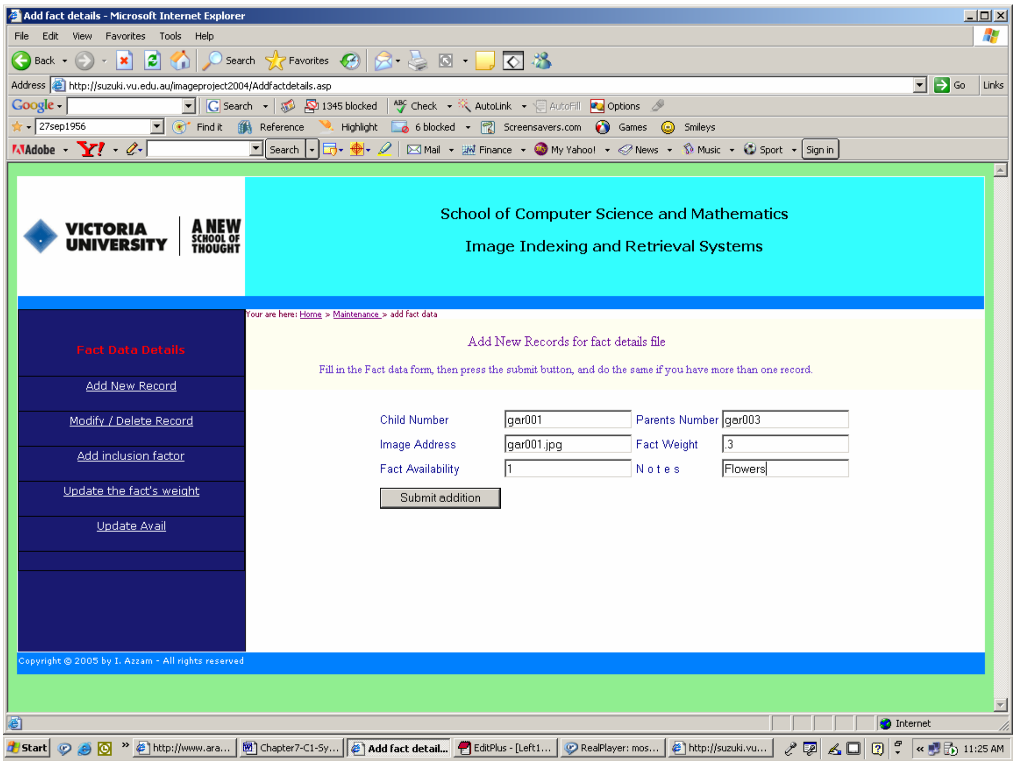


Figure A.24 Add Object Details Record

The Modify/Delete Record from an Object Details file option is follow the same procedure used to Modify/Delete Record from an Object Data file which was explained earlier in this section.

- *Add Supporting Factor*

The third sub-option of the Object Details Page is used to add the supporting factors, which build the relationship between different objects (the higher-level object and its components of the lower-level objects). When the administrator selects the third sub-option, the Add Supporting Factor page will appear as shown in Figure A.25.

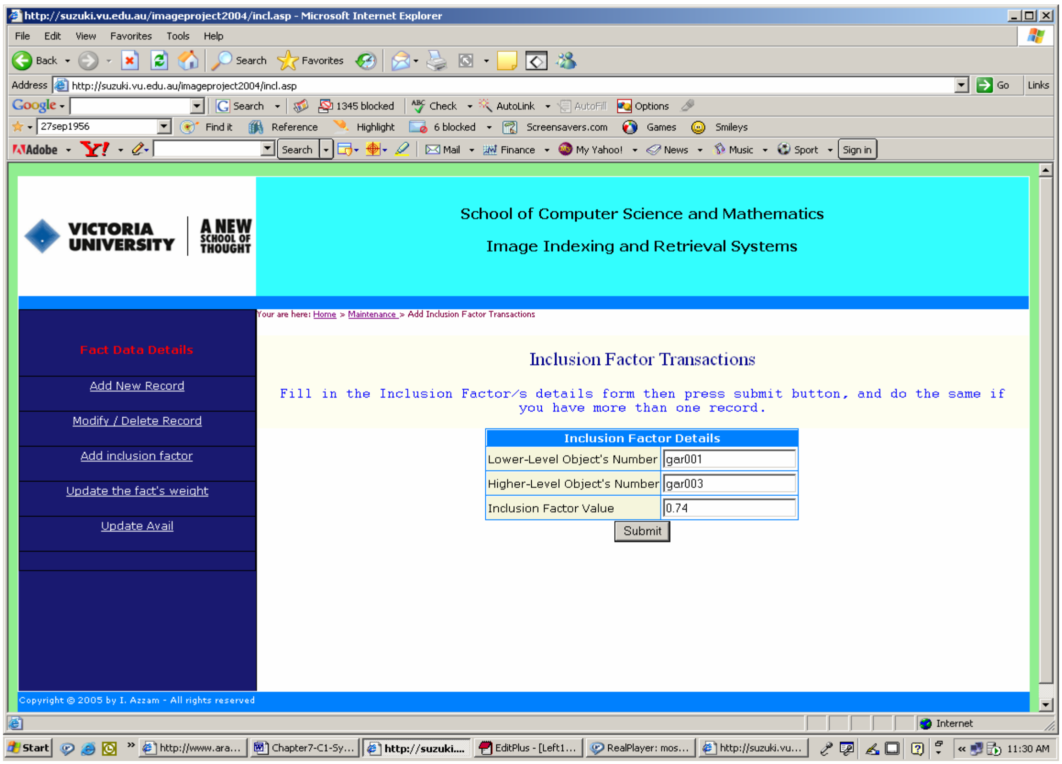


Figure A.25 Add Supporting Objector

The administrator fills the form and then submits the transaction, the supporting factor value will be saved in the object details file; and then some other computed value will occur.

- *Update Objects' Weight(s)*

The fourth sub-option is used to update the objects' weight. When the administrator selects the fourth sub-option of the Object Details Page, the Update Objects' Weight page will appear as shown in Figure A.26. The administrator fills the form, which contains the new objects' weight and then submits the changes, the entered objects' weight will be saved, and the objects' value will be recomputed based on the new weight. The new computed value of the lower-level object will replace the object's old value in the higher-level availability field, which it belongs to.

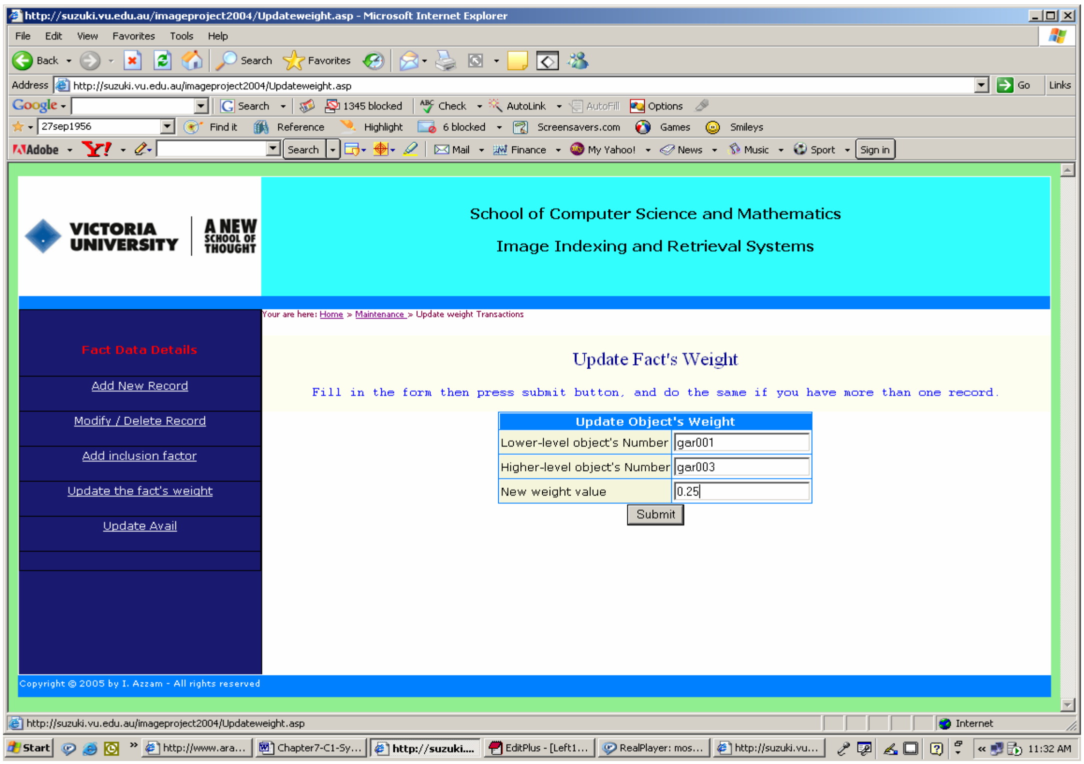


Figure A.26 Update Object's Weights

- *Update Objects' Availabilities*

The user can update the Object's availability, which is normally used for the non-primitive objects by selecting this sub-option, the user enters the object's number and the object's parent number and then the rule significant value of the object, when the user presses the Submit button, the object's availability will be updated, Figure A.27 shows the page used for updating the availabilities.

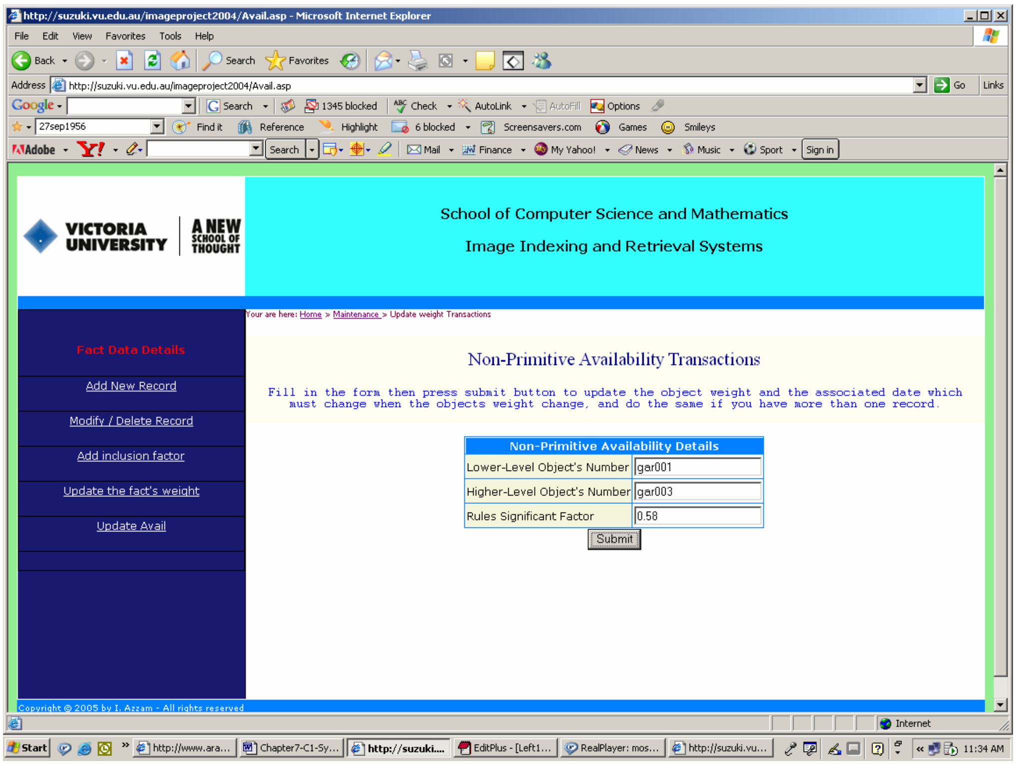


Figure A.27 Update Object's Availabilities

Staff/Members Transactions

When the administrators select the Staff/Members Transaction option, the Staff/Members Transaction page will appear as shown in Figure A.28. This page has two options:

These are the 'Add new record' and 'Modify/delete record' and are used by the administrator to add or change data from the administrator's/staff's file.

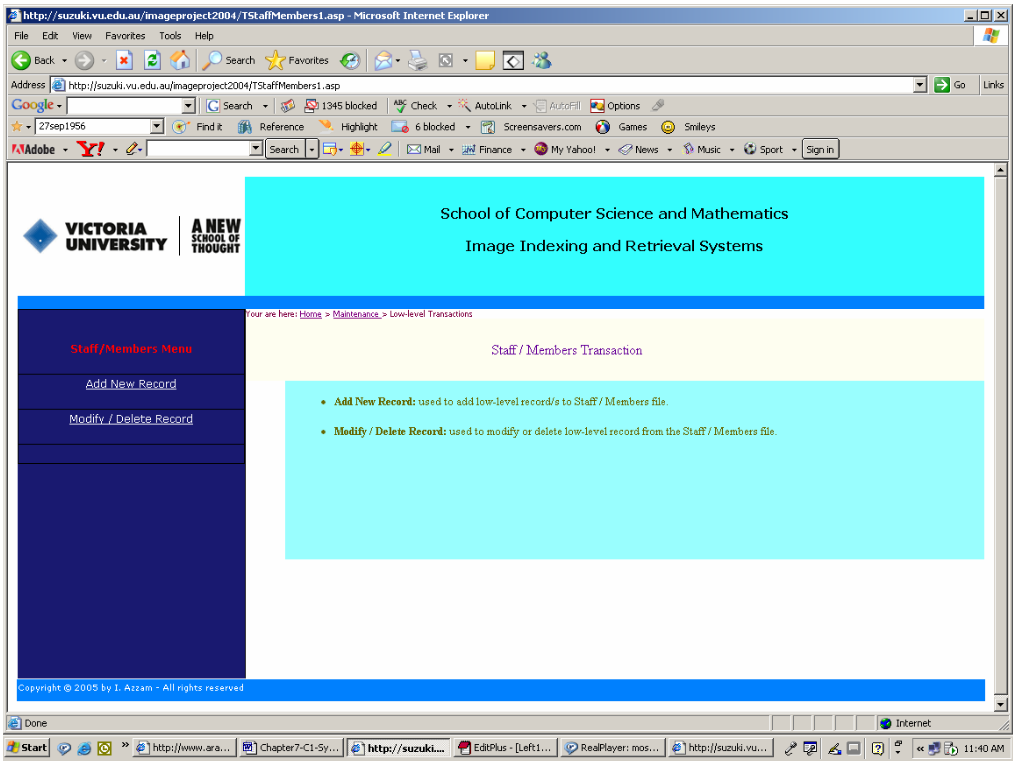


Figure A.28 Staff/Administrators Main Page

- *Add New Record*

If the administrator wants to add a new record, the administrator must choose the first option; an 'Add Record' page will appear as shown in Figure A.29. The administrator fills the form and then submit the transaction, the record will be saved in the staff/members file; here the administrator can enter as many records as needed.

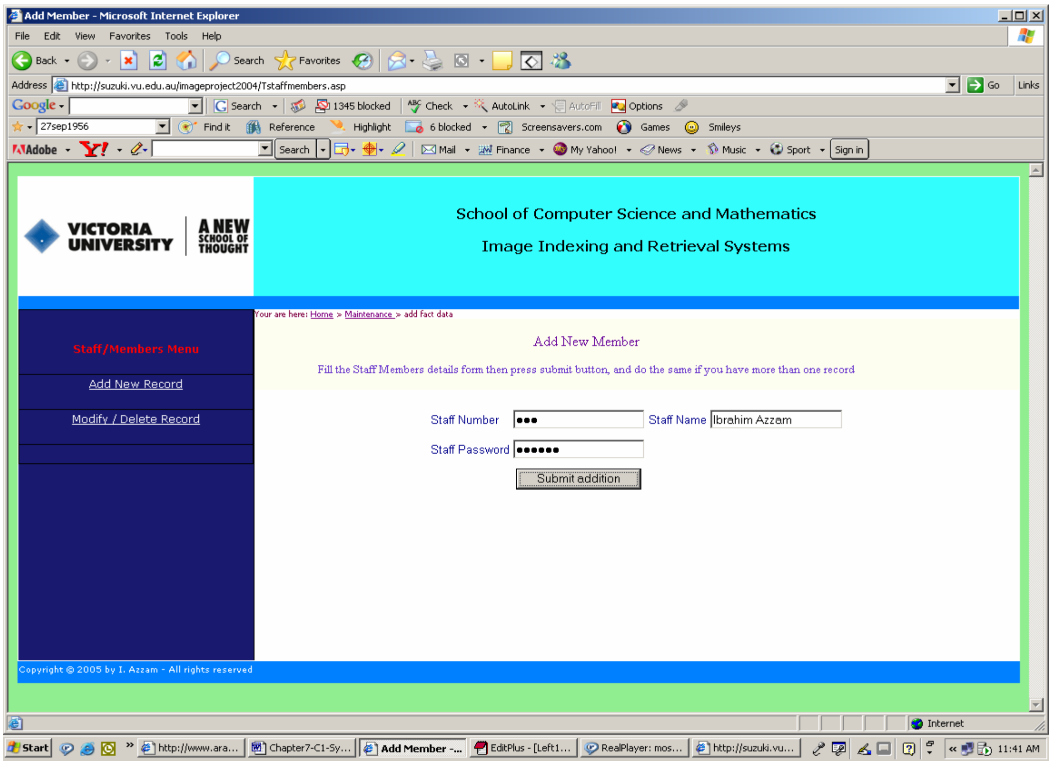


Figure A.29 Add New Member

- *Modify or Delete Record*

When the administrator chooses the second option, a Modify/Delete Record page will appear; the administrators need to enter the staff/member number, and to submit the transaction. The system will check the entered details and if found will display the record. At the end of the displayed record there is a link for Modify and Delete. If the Administrator wants to modify the record, then they will select the modify link, a new page will appear containing all the information about the object, the administrator will make the required changes and submit the changes to be saved. This procedure is the same as we did for the Object Data and Object Details files.

Display Records Transaction

When the Administrator selects the Display Records Transaction sub-option, the Displayed Records Transaction page is displayed as in Figure A.30, this option contains four sub-options: Display low-level records, display mid-level records, display high-level records, and display all records.

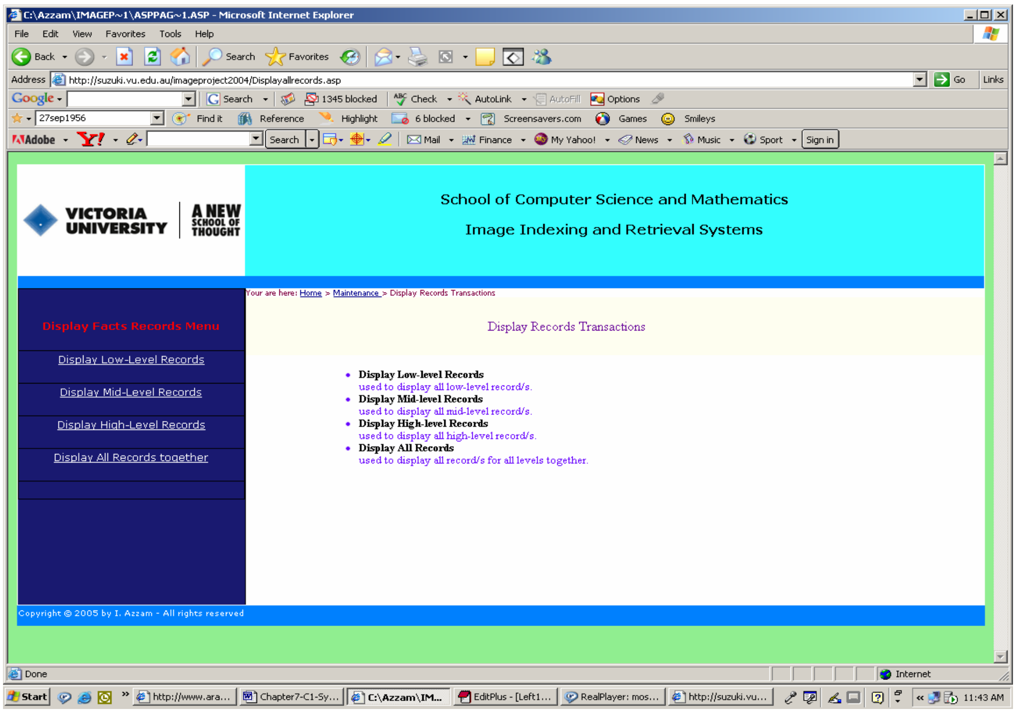


Figure A.30 Display Records Transactions Main Page

If the administrator selects the first sub-option (Display low-level records), all records of low-level objects in the database will be displayed; this is clarified in the table shown in Figure A.31. The object types are all equal to 1 meaning they're all low-level objects. To display all records of mid-level objects and high-level objects, the administrators have to select the second or third sub-option; it will follow the same technique of displaying all records as the low-level objects.

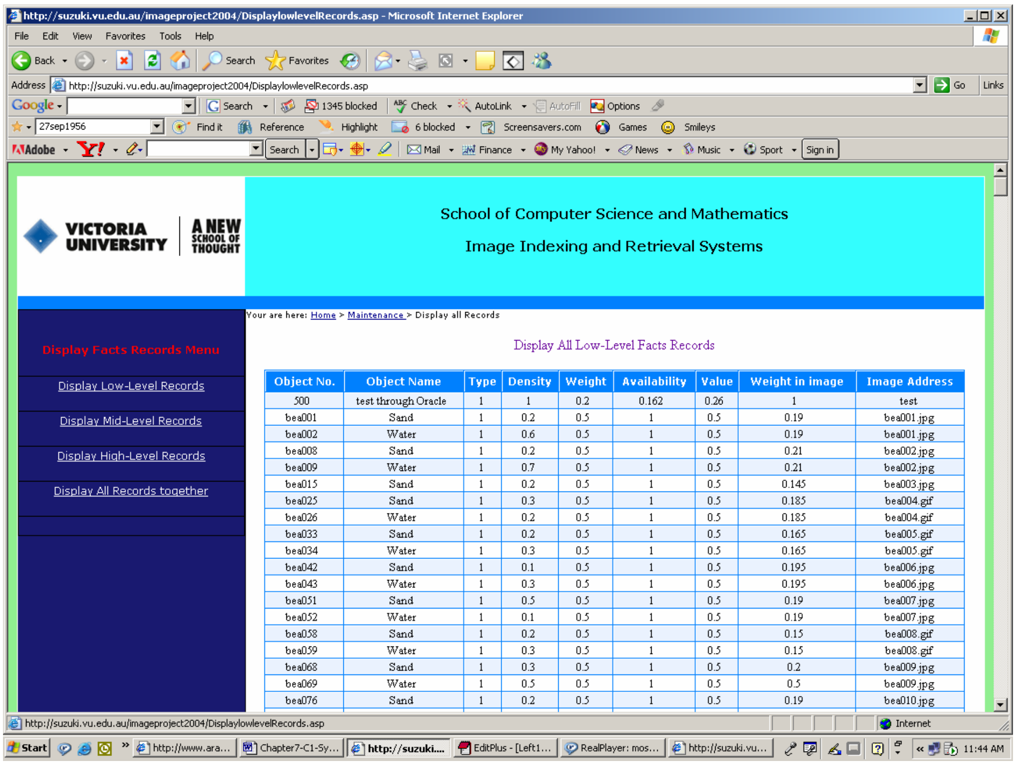
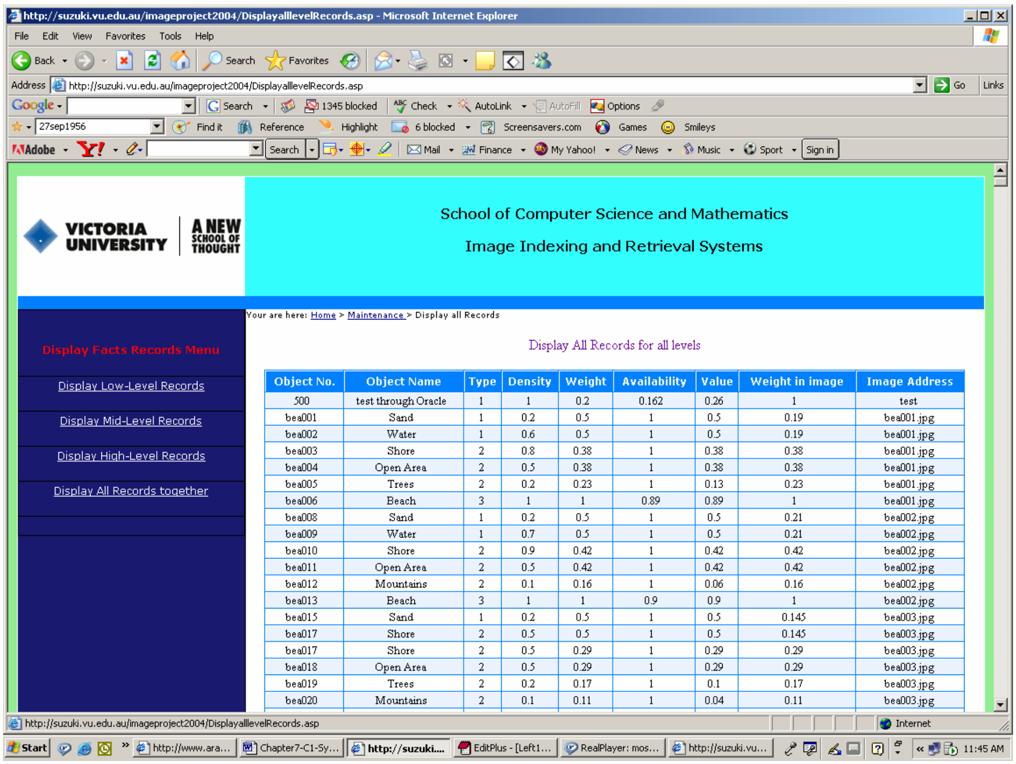


Figure A.31 Displays Low-Level Object Records

If the administrator selects the last option (Display all-levels of records), all records of different levels of objects in the database will be displayed, this is shown in Figure A.32, the object types of all records vary, the number of each type relates to the object level, 1 = low-level objects, 2 = mid-level objects, 3 = high-level objects.



Object No.	Object Name	Type	Density	Weight	Availability	Value	Weight in image	Image Address
500	test through Oracle	1	1	0.2	0.162	0.26	1	test
bea001	Sand	1	0.2	0.5	1	0.5	0.19	bea001.jpg
bea002	Water	1	0.6	0.5	1	0.5	0.19	bea001.jpg
bea003	Shore	2	0.8	0.38	1	0.38	0.38	bea001.jpg
bea004	Open Area	2	0.5	0.38	1	0.38	0.38	bea001.jpg
bea005	Trees	2	0.2	0.23	1	0.13	0.23	bea001.jpg
bea006	Beach	3	1	1	0.89	0.89	1	bea001.jpg
bea008	Sand	1	0.2	0.5	1	0.5	0.21	bea002.jpg
bea009	Water	1	0.7	0.5	1	0.5	0.21	bea002.jpg
bea010	Shore	2	0.9	0.42	1	0.42	0.42	bea002.jpg
bea011	Open Area	2	0.5	0.42	1	0.42	0.42	bea002.jpg
bea012	Mountains	2	0.1	0.16	1	0.06	0.16	bea002.jpg
bea013	Beach	3	1	1	0.9	0.9	1	bea002.jpg
bea015	Sand	1	0.2	0.5	1	0.5	0.145	bea003.jpg
bea017	Shore	2	0.5	0.5	1	0.5	0.145	bea003.jpg
bea017	Shore	2	0.5	0.29	1	0.29	0.29	bea003.jpg
bea018	Open Area	2	0.5	0.29	1	0.29	0.29	bea003.jpg
bea019	Trees	2	0.2	0.17	1	0.1	0.17	bea003.jpg
bea020	Mountains	2	0.1	0.11	1	0.04	0.11	bea003.jpg

Figure A.32 Display All Records

A.5.4 Other Options

The user may want to contact the administrators for any further inquiries. The user may select one of the remaining options (Contact Us or Research Group).

Contact Us

When users selects the option “Contact Us”. The next page will appear as shown in Figure A.33. If the user wants to send a message to the staff/administrators, they can use the link “Send Email”.

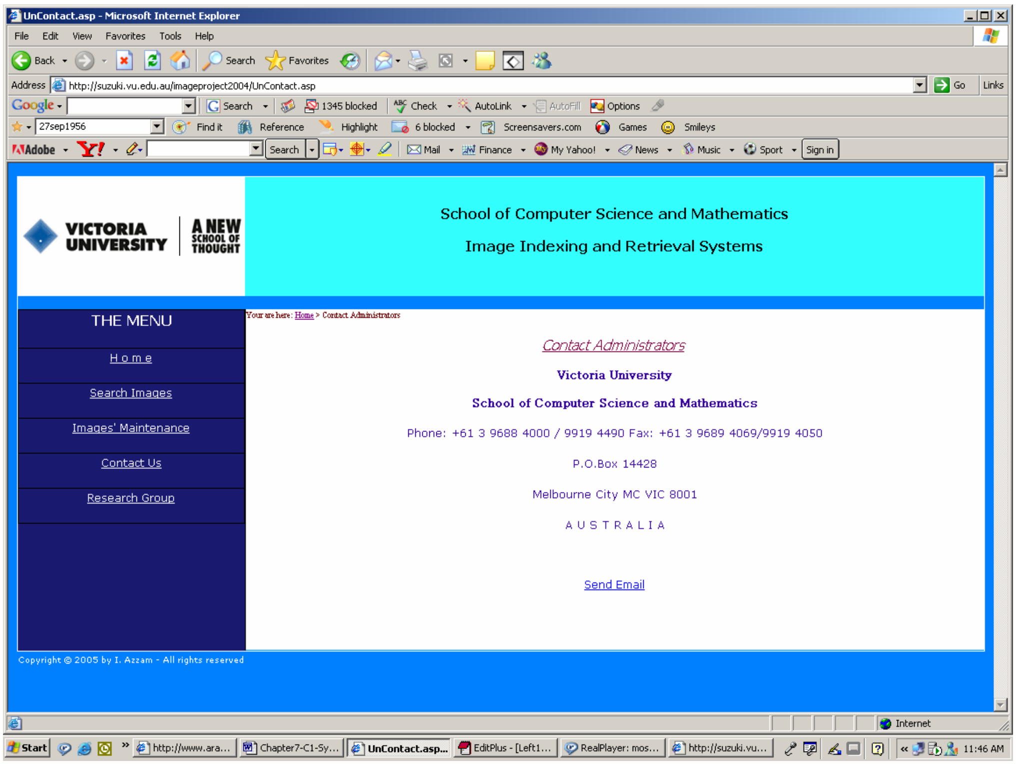


Figure A.33 Contact Administrators

Research Group

To contact the research group for further inquiries, a user can also select the 'Research Group' option, The Research Group page displayed shown in Figure A.34.

If the user wants to send a message to the research group, he/she will press the link "Email Contact" which is located under the particular member's details. The user can also enter the contact's web page if one is available.

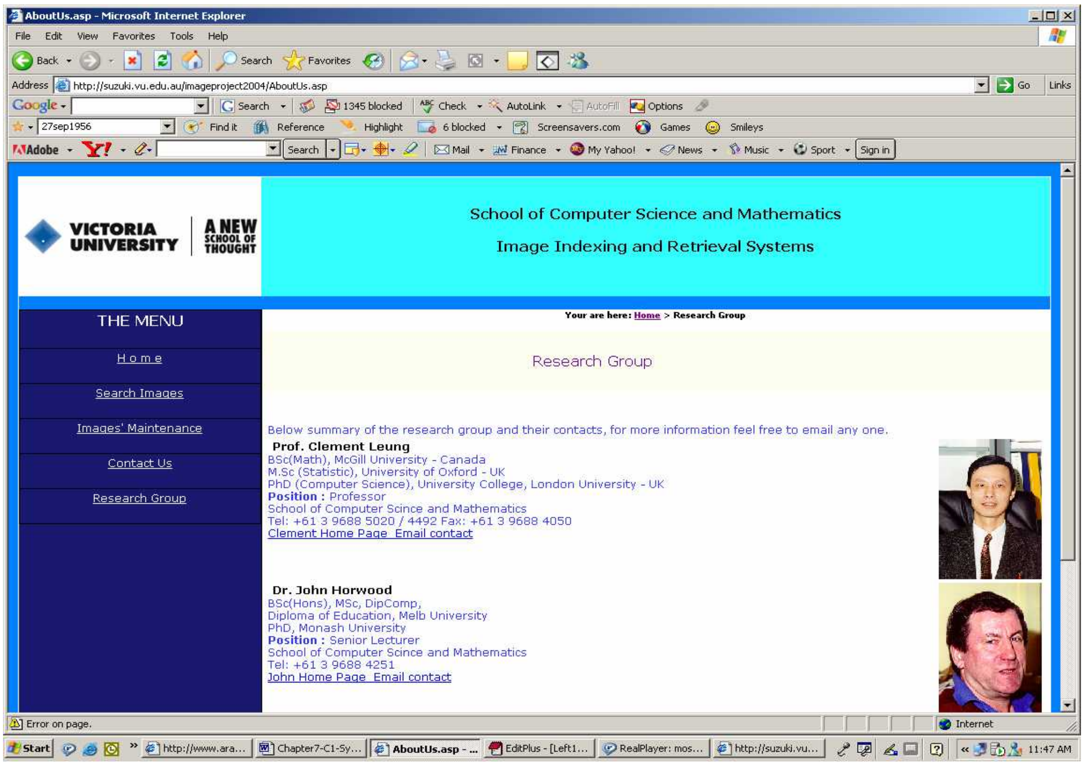


Figure A.34 Research Group