Content-based Image Retrieval Based On Emergence Index

A dissertation submitted to

The Department of Mathematics and Computing
Faculty of Sciences
The University of Southern Queensland

For the degree of
Doctor of Philosophy

By

Sagarmay Deb
Master of Science, India;
Master of Business Administration, USA

2003

Abstract

Emergence is a phenomenon where we study the implicit or hidden meaning of an image. We introduce this concept in image database access and retrieval of images using this as an index for retrieval. This would give an entirely different search outcome than ordinary search where emergence is not considered, as consideration of hidden meanings could change the index of search.

A feature of an image, which is not explicit would be emergent feature if it can be made explicit. There are three types of emergence: computational emergence, thermodynamic emergence and emergence relative to a model. In computational emergence, it is assumed computational interactions can generate different features or behaviors. This is one of the approaches in the field of artificial life. Thermodynamic emergence is of the view that new stable features or behaviors can arise from equilibrium through the use of thermodynamic theory. In emergence relative to a model, deviation of the behavior from the original model gives rise to emergence. We would use this latter view in our work.

Two classes of shape emergence have been identified: embedded shape emergence and illusory shape emergence. In embedded shape emergence all the emergent shapes can be identified by set theory procedures on the original shape under consideration. For example, in a set S= {a,b,c,d,e}, we can find subsets like S1={a,b,c}, S2={c,d,e}, S3={a,c,e} and so on. But in illusory shape emergence, where contours defining a shape are perceived even though no contours are physically present, this kind of set theory procedures are not enough and more effective procedures have to be applied to find these hidden shapes. These procedures could be based on geometrical, topological or dimensional studies of the original shape.

Content-based Image Retrieval (CBIR) techniques, so far developed, concentrated on only explicit meanings of an image. But more meanings could be extracted when we consider the implicit meanings of the same image. To find out the implicit meanings, we first destroy the shape of the original image which gives rise to unstructured image. Then we process the unstructured image to bring out the new emergent image.

We discuss emergence, calculation of emergence index and accessing multimedia databases using emergence index in this dissertation. To calculate emergence index in the access of multimedia databases, we take an input image and study the emergence phenomenon of it. Also we study the emergence phenomenon of the images of the database. Both input image and images of database would give rise to more meanings because of emergence as we explained earlier. Based on the new meanings, wherever there would be a match between input image and images of database, we would pick that record up for selection.

We defined emergence index as

$$EI = f(D,F,V,C,E)$$

where D stands for domain of the database, F for features of the image, V for various variables that define the image, C for constraints which represent the image and E for emergence phenomenon.

We calculate these five variables to get emergence index for each image of the database. Also we calculate these five variables for input image as well.

We talk about global aspects of features. It means features of the entire image. Examples are area, perimeter or rectangles, triangles. In some searches, to consider the global features could be advantageous in that a symmetry with the input image could be obtained on the basis of global features only. But as is clearly the case, to consider global features could overlook the individual objects that constitute the image as a whole. In the kind of searches we propose, we take into account the global features of the image of the database while considering in detail local features.

Various objects that lie within an image constitute local features. In our example, there are three objects in the image, namely, a lake and two houses. Studying the features of these three objects would add to studying the features of the image globally.

We took the example of a geographic location in the thesis and then showed how destruction of original image is done and further processing of the unstructured image gives new emergent image.

Partial implementation of this concept is also presented at the end. In implementation, we consider the retrieval of image globally. We do not consider break-up of image into multiple objects which is left for future research.

List of Publications

1.Deb, S., Zhang, Y. (2001). Concepts of Emergence Index in Image databases, *Distributed Multimedia Databases: Techniques and applications*, Idea Group Publishing, Hershey, PA 17033-1117, USA, pp. 73-88

2.Deb, S., Zhang, Y. (2001). Emergence Index Structure in Image Retrieval, *Tamkang Journal of Science and Engineering*, Tamkang University, Taipei, Taiwan, Vol.4, No.1, March 2001, pp. 59-69

3.Deb, S., Zhang, Y. (2001). Image retrieval in Multimedia with Emergence Index, *The Second IEEE Pacific-Rim Conference on Multimedia*, October 24-26, 2001, Beijing, China, pp. 351-358

4.Deb, S., Zhang, Y. (2002). Retrieval of Geographic Location with Emergence Index in Multimedia, 6th Joint Conference on Information Sciences, March 8-13, 2002, Research Triangle Park, North Carolina, USA, pp. 976-979

5.Deb, S., Zhang, Y. (2003). Emergence Index – An Approach to Content-based Image Retrieval, 9th International Conference on Multi-Media Modeling, January 8-10, 2003, Tamsui, Taiwan

6.Deb, S., Zhang, Y. (2003). Emergence Index and Content-based Image Retrieval, Information Resources Management Association (IRMA) International Conference 2003, Philadelphia, PA, USA, May 18-21, 2003

7.Deb, S. (2003). Multimedia Systems and Content-based Image Retrieval, *Multimedia Systems and Content-based Image Retrieval*, Idea Group Publishing, Hershey, PA 17033-1117, USA

8.Deb, S., Zhang, Y. (2003). Algorithm for the Retrieval of Image with Emergence Index in Multimedia, *Multimedia Systems and Content-based Image Retrieval*, Idea Group Publishing, Hershey, PA 17033-1117, USA

9.Deb, S., Zhang, Y. (2003). To Retrieve Images Based on Emergence Index, International Workshop on "Research Directions and Challenge Problems in Advanced Information Systems Engineering", Hanjo City, Akita Prefecture, Japan, Sept. 16-19, 2003

10.Worked as an editor for research publications for a book named *Multimedia Systems and Content-based Image Retrieval* for Idea Group Publishing, Inc., Pennsylvania, USA. The book had been published in October 2003.

11.Working as an editor for research publications for a book named *Video Data Management and Information Retrieval* for Idea Group Publishing, Inc., Pennsylvania, USA. The book would be published by the end of 2004.

Certification of Dissertation

I certify that the ideas, experimental work, results, analyses, software and conclusions reported in this dissertation are entirely my own effort, except where otherwise acknowledged. I also certify that the work is original and has not been previously submitted for any other award, except where otherwise acknowledged.

-- ------------------------------

Signature of Candidate Date

ENDORSEMENT

-- ______________________

Signature of Supervisor/s Date

-- ------------------------------

Acknowledgements

I extend my sincere thanks to my Supervisors Dr. Yanchun Zhang, Associate Professor, Department of Mathematics and Computing, University of Southern Queensland, Toowoomba, QLD 4350 and Dr. Hua Wang, Lecturer of the same university, for giving me valuable guidance during these efforts.

I extend my gratitude to The Department of Mathematics and Computing of The University of Southern Queensland, Toowoomba, QLD 4350, Australia for admitting me as a student of the doctoral program and offering financial assistance on four occasions to attend international conferences and to present my papers there.

Also I am very much grateful to my wife Ms. Clera Deb for giving me inspiration and support in this venture.

SAGARMAY DEB

Table of Contents

LIST OF FIGURES

Chapter 1 Introduction

In this chapter, we provide general background of our area of research. We also define the problems of the research done so far in this field. We then describe aims, scope and limitations of the research we are undertaking and mention the plan of the thesis. Lastly we describe concepts, definition and structure of emergence index.

1.1 General Background

Content-based Image Retrieval (CBIR)

Images are being generated at an ever-increasing rate by various sources. They include military purposes, aviation satellites, biomedical purposes, scientific experiments and home entertainment.

Previously there were two approaches to CBIR.

The first one is the attribute-based representation advocated by database researchers where image contents are modeled as a set of attributes extracted manually and managed within the framework of conventional database management systems. Queries are specified using these attributes. This entails a high-level of image abstraction (Chang, 1988; Gudivada and Raghavan, 1995).

The second approach propagated by image interpretation researchers depends on an integrated feature-extraction / object-recognition subsystem to overcome the limitations of attribute-based retrieval. This subsystem automates the feature-extraction and object-recognition task that occurs when the image is inserted into the database. These automated approaches to object recognition are computationally expensive, difficult and tend to be domain specific.

Recent CBIR research tries to combine both of these above mentioned approaches and has given rise to efficient image representations and data models, query-processing algorithms, intelligent query interfaces and domain-independent system architecture.

Image retrieval can be based on low-level visual features such as color (Pass et al, 1996; Smith et al, 1996; Swain and Ballard, 1991), texture (Manjunath, 1996; Sheikholeslami, 1997; Smith, 1994), shape (Safar, 2000; Shahabi and Safar, 1999; Tao and Grosky, 1999), high level semantics (Forsyth et al, 1996; Torratha and Oliva, 1999) or both (Zhao and Grosky, 2002).

There are two major categories of features. One is primitive which is concerned with extracting boundaries of the image and the other one is logical which defines the image at various levels of detail.

Regardless of which approach is used, the retrieval in CBIR is done by color, texture, sketch, shape, volume, spatial constraints, browsing, objective attributes, subjective attributes, motion, text and domain concepts (Gudivada and Raghavan, 1995). Out of these, color, shape and texture are the main characteristics or properties of images (Aslandogan and Yu, 1999; Gatzer et al, 2000; Smeulders et al, 2000; Vailaya et al, 2001). In order to use the access methods to index the features of images and accelerate the retrieval of images, many techniques for image comparison using color histograms (Ko et al, 2000; Pass et al, 1996; Pentland et al, 1996), shape (Esperanca and Samet, 1997; Gagandakis and Rosin, 2001; Tuytelaars and Gool, 1999) and texture (Gimel'farb and Jain, 1996; Tomita and Saburo, 1990; Zhou et al, 2001) have been proposed.

As we will discuss later, retrieval of images by using low-level features has proved very difficult. Automatic image retrieval by semantics also proved very difficult due to the problem in proper object recognition and image understanding.

Keyword indexing has been used by many picture libraries like Getty Images (Bjarnestam, 1998) which used over 10000 key words to index their collection of contemporary stock photographs. As the indexing by keywords is done manually, it suffers from problems. The first problem is it is labor-intensive and the second one is it is not reliable as the same image can be interpreted differently by different users (Eakins and Graham, 1999). There has to be some way to bridge the gap between low-level features and high-level semantics for efficient image retrieval. Many researchers are currently working on this issue but no universally accepted solution has been achieved yet.

1.2 Problems of Existing Retrieval Packages

Our study of the existing literature, which is presented in chapter 2, suggests, in recent times, there have been very many attempts to perform CBIR on efficient basis based on feature, color and texture. Quite a few models have been developed which address the problem of image retrieval from various angles. Some of the models are QBIC (Query by Image Content) , Virage, Pichunter, VisualSEEK, Chabot, Excalibur, Photobook, Jacob, UC Berkeley Digital Library Project.

One of the most well known packages is QBIC which was developed by IBM. It uses color, texture and shape to retrieve images and is used in art galleries and art museums (Seaborn, 1997). Based on the same kind of approach was developed Virage which made use of color, spatial color and shape and being applied in face recognition and in retrieval of ophthalmologic images (Seaborn, 1997). NEC Research Institute has developed Pichunter, which utilizes image properties like ratio of image dimensions, color percentages and global statistical and frequency properties. It's more applicable in database retrieval instead of feature detection and also being applied in the relevance feedback using Bayesian probability theory (Seaborn, 1997). VisualSEEK uses color percentage method in content-based retrieval. Using regional colors and their relative locations, the image is segmented and this is quite a bit similar to the way we perceive an image (Seaborn, 1997). Chabot mainly uses texts to retrieve images. It uses to some extent color percentages to retrieve images automatically otherwise all features are input manually (Seaborn, 1997). Excalibur is of the same type as QBIC and Virage and uses standard metrics, color, shape and texture and like Pichunter uses image ratio. In addition it extracts features like structure of brightness and color. It gives option to the users to indicate which features are dominant (Seaborn, 1997). MIT's Vision and Modeling Group developed Photobook, which uses color percentages, textures and the statistical analysis. The images are segmented here and then texts are made out of those segmentations using predefined templates and techniques. The images are retrieved based on these texts (Seaborn, 1997). Jacob uses combination of color, texture and motion as features to retrieve video clips as this package is developed for video databases (Seaborn, 1997). The Digital Library Project of Berkeley University applies color percentage method and feature of dots. The user can define the quantity of various colors in the image and also can define colors and sizes of dots to be there in the image (Seaborn, 1997).

These models have brought this area of CBIR from its infancy to a matured stage. They study the various features of the images, make statistical analysis of color distribution as well as shape and texture and retrieve images from the contents of the image. But none of these models did make any attempt to study the hidden or implicit meanings of an image. They are solely concentrated on explicit meanings of the images as we discussed above. An image cannot be analyzed in its contents correctly until and unless we study the various hidden meanings of it, which cannot be otherwise discernible as we will discuss later. For example, a square with single diagonal could be an image. But when we try to find out images lying hidden in it, we end up getting the images of two triangles. The retrieval will be correct and more efficient when this factor will be taken into account while retrieving images. This will give an entirely different set of results than the searches so far talked about in image retrieval. This is the issue of emergence. As we know in any image, emergence phenomena play a vital role in exactly defining an image. Although it could be that the image has some apparent meaning, but emergence could give rise to more meanings about the image. We attempt to study the problem of image query whereas query made would be searched through the database to pick up those records where a similar shape has been found. But in addition to that we pick up records based on the emergence phenomena where the query input may not have an apparent match in a particular image of the database, but emergence phenomena could give rise to a similar structure in the same image and as such this image should be selected as a query result. As we have mentioned, a square with single diagonal can be observed as two triangles. So whenever search intends to find a triangle this image which apparently is much different than triangle, would be selected because of emergence.

A practical example of how this study of implicit or hidden meanings is useful and essential is the case of image retrieval where the image is a geographic location. As we will show in chapter 6, roads in the map constitutes the shape of a bowl which is the implicit or emergent meaning found in the map. So if we want to retrieve that image of the map from a large volume of maps in the database, then we should put the image of a bowl as input and search the database. In that case when the input image of the bowl finds symmetry with the bowl shape which is implicit in the road structure of the map, that image of the map would be retrieved. This way studying the emergence phenomenon can help us retrieving an image from the voluminous database where there is no other way to identify and retrieve that image very quickly. But in search, where emergence phenomenon is not considered, it will take much longer time and more resources to retrieve this same image as there is no other way to specifically identify any particular feature of the image. That’s why it is very important that we undertake appropriate research to apply the concepts of emergence in CBIR.

1.3 Objectives

1.3.1 Aims

In order to rectify the problems of existing image retrieval systems of not taking into account the implicit or hidden meanings lying in an image and to make the retrieval systems more accurate and efficient, as discussed in the last section, we undertake this project of content-based image retrieval based on emergence index, where we take into account the explicit as well as implicit or hidden meanings of the image.

The basic idea is to have an input in the form of text or image and then based on that input, we search the images of the database. If a match is found between the input and a particular image of the database, then that image is selected for retrieval. But if no match is found between the input and a particular image, then we study the emergence phenomenon for that image to bring out implicit or hidden meanings. If now a match is found between the input and the image, the image is selected for retrieval. If even after studying the emergence phenomenon of the image, no match could be established between the input and the image, then that image is not selected for retrieval. In other words, the emergence phenomenon in images of the database is studied only when no match between the input and a particular image in its current form, is found. We presented a Flow-Chart of the plan in section 7.2.

The idea of indexing database is to facilitate searches so as to retrieve the desired records quickly without scanning the whole database. In a relational database, for example, which is quite widely being used these days, in a particular table, there could be many fields or data items. Out of these fields, one or two key fields are defined as indexed. For example, in a company maintaining employees' personal information, a table would be created where information regarding employees' name, address, age, sex, qualifications, date of joining the company, designation would be maintained. For the purpose of indexing, it is possible for the company to assign a unique employee number to each employee and that employee number would be the index through which search would be done. There could be more than one index like the surnames of the employees, which could be another index in addition to employee number. But surnames may not be unique as there could be more than one employee with the same surname and hence could be used as duplicate index.

Usually when search is done through a table with the index value, a binary search is performed where the table would be divided first into two halves and the part which contains the index value would be located. Then that part would again be divided into two halves and the part, which contains the index value, would be located. Then that part would be divided into two halves. This process goes on till the record containing the index value is found. This way we avoid scanning the complete table, which is very inefficient and time consuming.

As the volume of images generated is very high, the image database management systems have to be based on indexing access methods for the purpose of storing and retrieving the image, audio, video and textual data. There are many access methods developed for traditional data like B tree (Comer, 1979) and hash based indexing access methods (Garcia-Molina et al, 2000). Spatial data, such as regions of maps given by their multidimensional coordinates, have been managed by Spatial Access Methods which include tree-like structures such as R tree (Guttman, 1984), R+ tree (Sellis et al, 1987), R* tree (Beckmann et al, 1990), k d B Tree (Robinson, 1981), space filling curves (Kamel and Paloutsos, 1994), spatial hash files (Samet, 1995) and so on. An excellent survey on Spatial Access Methods is given in (Gaede and Gnther, 1998). Whenever the features extracted have a fixed number of attributes, this method can be applied to index such data. The majority of features extracted from images can be seen as multidimensional points in n dimensional space. This is the case of histograms, moment invariant, Fourier features, wavelet coefficients, principal component analysis values and so on. There are other access methods available which can deal with high-dimensional datasets and can be used to answer queries in such cases like X tree (Berchtold et al, 1996), TV tree (Lin et al, 1994) and the Hybrid tree (Chakrabarti and Mehrotra, 1999).

In our case, input would be, as mentioned earlier, in the form of text or image. We convert this input into parametric forms and get ready for accessing the image database.

To access the multimedia database, we convert each record of the images into parametric forms like input and then try to match it against the input. Whenever a match is found, we retrieve that record. In this way, we end up getting output based on the emergence phenomenon of images in the database, according to the input given.

1.3.2 The Scope of the Thesis

As we mentioned in the previous section, a query in a computer could be made in graphic or textual form. Before going to access the multimedia database, we have to convert the input into parametric forms so as to enable it to find appropriate symmetry with the image of the database. In order to convert the input into parametric forms, we first consider the domain or the class where the input belongs.

For example the input could be a geographical map. Then we calculate the features of the image like shape, color, texture or background and so on. After determining the features we use suitable variables to define those features. Using those variables like number of sides the shape has, angles various sides make with one another, we select some constraints, which restricts the pattern or behavior of the shape. Examples of constraints are, for the image of a square, the length of all four sides are equal, all the four angles the four sides make with one another, are of 90 degrees, the opposite sides are parallel to each other. Lastly we study the emergence phenomenon of the image to see if there is any hidden meaning lying in it. These measurements convert the input image into parametric form.

After converting input into parametric forms, we access the image database. We study those images alongwith emergence phenomenon and then convert these images into parametric forms in the same way we do for input image. When the input in its parametric form finds a suitable match with any record of the database, we select that record as output of the query. We presented a Flow-Chart of the plan in section 7.2.

We would consider images of objects with certain background as well as images of geographic location. Also we suggest using a multimedia database, where records containing images, videos, audio, texts or documents could be present.

1.4 Plan of the Thesis

We define the procedures to be followed in the thesis followed by the plan of the thesis in this section. We provide a definition of emergence index, then a model of calculating emergence index with examples. Also we describe accessing multimedia databases using emergence index by establishing symmetry between input and images of the database.

1.We start up by defining the Emergence Index. Since it is a new conception we are introducing in the field of image retrieval, it needs an explicit definition.

2.We proceed to define a model for image retrieval with emergence index. We define the components that would be involved in image retrieval of this kind. They are features, domain, variables, constraints, symmetry, indexing. We show a simple example of developing image parameters for indexing purpose by a square with a diagonal.

3. Next we give a simple example of indexing using input in the form of text.

4. We provide an approach to attack the problem of indexing by giving examples of lake and umbrella.

5. We define geometric, topological, dimensional or statistical methods we follow in generating emergence index.

6. We define the emergence index structure in retrieving images in a multi-dimensional and multi-media environment. Also we establish symmetry between input image and images of the multimedia database.

7. We take example of a geographic location and find symmetry between input and image of the database with and without studying emergence phenomenon.

8. We develop an algorithm for finding symmetry between input image and image of the database with and without studying emergence phenomenon.

9. We study the emergence phenomenon for a three dimensional figure.

10. We present some experiments and demonstrations.

This doctoral thesis is being organized into seven chapters. The first one covers general background, problems of existing works and aims, scope and limitations of the thesis. Also this chapter talks about definition, structure and construction of emergence index.

The second chapter describes the analyses of works done in this field so far and the research procedures being followed in this thesis. Then this chapter describes the approach to solve the problem of image retrieval with emergence.

The third chapter discusses semantic representation of images.

The fourth chapter covers calculation of emergence index using various parameters.

The fifth chapter presents accessing multimedia databases using emergence index. Also this chapter talks about finding symmetry in three-dimensional image.

The sixth chapter describes application of emergence index in geographic location.

The seventh chapter discusses implementation of the concepts talked about. Also this chapter covers algorithm for accessing databases with emergence index and analysis of experimental results. The conclusion of the thesis and direction of future research are also presented in this chapter.

1.5 Definition of Emergence Phenomenon

A feature of an image which is not explicit would be emergent feature if it can be made explicit. There are three types of emergence: computational emergence, thermodynamic emergence and emergence relative to a model (Cariani, 1992). In computational emergence, it is assumed computational interactions can generate different features or behaviors (Forrest, 1991; Langton, 1991). This is one of the approaches in the field of artificial life. Thermodynamic emergence is of the view that new stable features or behaviors can arise from equilibrium through the use of thermodynamic theory. In emergence relative to a model, deviation of the behavior from the original model gives rise to emergence. We will use this latter view in our work.

In computational emergence, new shapes or images develop but within certain limit as programmed by the computer programmers. No new shape can emerge beyond the logic of the program.

In thermodynamic emergence, emergence can be defined as emergence of order from noise. Stochastic processes at micro-level form discrete macro-level structures or behaviors. The example of this type of emergence is gas where stochastic movements of atoms or molecules within the gas create the ordered properties of temperature, pressure and volume at a higher level.

Example of emergence relative to a model is where changes in internal structure and consequently in its behavior occur and we as observers will need to change our model to track the device’s behavior in order to successfully continue to predict actions. The example of a square having two triangles hidden in it as given earlier is of this type.

Whenever we shift our focus on an existing shape in otherwords an image, new shape emerges. The representation of the new shape is based upon view of the original shape. The new shape emerges as we change our view of the original shape. This is the fundamentally most important idea of emergence.

Two classes of shape emergence have been identified: embedded shape emergence and illusory shape emergence. In embedded shape emergence all the emergent shapes can be identified by set theory kind of procedures on the original shape under consideration. For example, in a set *S = {a, b, c, d, e}*, we can find subsets like *S1 = {a, b, c}*, *S2 = {c, d, e}*, *S3 = {a, c, e}* and so on. But in illusory shape emergence, where contours defining a shape are perceived even though no contours are physically present, this kind of set theory procedures are not enough and more effective procedures have to be applied to find these hidden shapes (Gero and Maher, 1994), (Gero, Year Unknown). These procedures could be based on geometrical, topological or dimensional studies of the original shape.

1.5.1 Structure, Behavior and Function of Emergence

Structure of a shape is the physical definition of the shape. For example, a box could be rectangular in shape, its length, width and height as well as color, substance like wood, metal or hard paper would define the structure of the shape. Behavior of the box could be to contain certain stuffs in it and the function could be to carry stuff from one place to another using the box as a container, which is the purpose for which the box is used. Emergence of new structure, behavior or function takes place when these descriptions are interpreted in ways not anticipated in the original description (Gero and Maher, 1994).

1.5.2 Examples of Emergence

Shape emergence is associated with emergence of individual or multiple shapes. The following figures are examples of shape emergence.

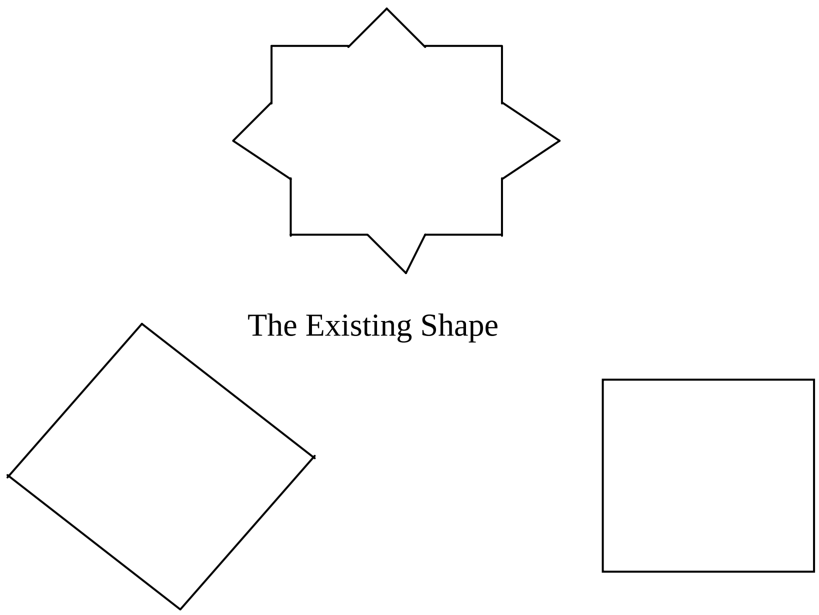


Figure 1.1 Two emergent shapes derived from the existing one (Gero, Year Unknown)

1.5.3 Definition of Emergence Index

Image retrieval where the hidden or emergence meanings of the images are studied and based on those hidden meanings as well as explicit meanings, an index of search is defined to retrieve images is called emergence index.

When images are retrieved based on textual information, then various parameters and descriptions might define the input and the images of the database. Whenever there would be symmetry of parameters and descriptions, the image could be retrieved. In CBIR, color, texture and shape are widely used as index to retrieve images. But in our studies, we can find the hidden meanings of the images and whenever those hidden meanings match with the input given, although the original image may not match at all with the input, we can retrieve that image.

To describe in detail, as we have mentioned, emergence is a phenomenon where we bring out the shapes, which are not explicit but implicit. The following figure shows a simple example of emergence where an apparently square shape has two triangles hidden in it as we discussed earlier.

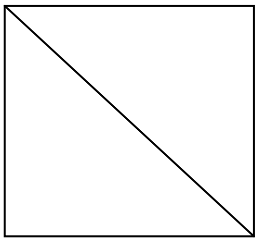


Figure 1.2 Example of emergence where diagonal generates two triangles

When an input would come in the form of an image, the image could be studied based on features, constraints, variables, domains and emergence and converted into parametric form. Then the image database would be accessed and each image would be interpreted considering the items mentioned above and converted into parametric form like the input image. Whenever there would be a match between parameters of the input and the images of the database, those records would be selected. In other words, indexing would be decided by the outcome of emergence which means more meaningful images could be found hidden in an image which would otherwise not be understood.

As we have mentioned earlier, many images of the database may not have any apparent similarities with the input, but emergence could bring out the hidden meaning of the image and could establish similarities with the input image. So emergence outcomes of the images would form the index structure of the search.

1.6 Structure of Emergence Index

1.6.1 Parameter Definitions

To make an effective query, the images in the query or database must be analyzed so that we know what we are looking for and where to look for. Features, domains, variables, constraints, similarities, indexing are the parameters, which play very important role in similarity searching. Hence they constitute the structure of the emergence index.

Features

Input, as we mentioned earlier, could come in the form of text. For example, the text may indicate we have to pick up all the images of a database that contains the image of a particular person or an object. In this case, nothing much could be done on input side in the sense that we cannot study the input's features etc. We have to go through the images and pick up the image records. But if the query comes in the form of an image of a person then we have to analyze it before accessing the database images. Features would tell us about a few important characteristics of the input image. To start up, our query could be an image where a particular person is sitting on a chair. Then obviously there are two important features in the query image - the particular person and the chair. We have to locate these two features in the database image while searching for similarities.

We know color plays a very important role in the definition of features. Quite often a query might mention an object with certain specified color to be picked up from the databases. Besides, color is being used extensively in various models as a tool in finding symmetry with the input image.

Sometimes the input may be in the form of a sketch. In that case, a similar kind of image should be selected from the image database. The selection should be made on the basis of few dominant characteristics of the input and image and finding similarities in those characteristics.

Texture is another part of the feature where the general alignment of the image like the background part of the image is considered where the image of a person or an object could be the dominant part.

There would be some global features of an image like area, perimeters and a set of rectangles or triangles that cover the entire shape and there would be some local features, which can be obtained from the image's local region.

When we retrieve images by browsing, then in most cases the user does not have clear idea about what he or she is looking for. In this case, there would not be input image and the search through browsing would be manual. The user would have vague idea about the features of the images to be retrieved like the picture of a particular person with certain specified background. In the objective features based queries, the retrieval is performed on an exact match of attribute values whereas in the subjective features, query is specified by features, which could be interpreted differently by different users.

Retrieval by motion facilitates retrieving spatio temporal image sequence depicting a domain phenomenon that varies in time or geographic space.

Again, sometimes images are retrieved, as we mentioned earlier, by text. In other words, text defines the features of the images to be selected. Then the database is searched based on those features defined in the input (Gudivada and Raghavan, 1995).

Domain

We now proceed to discuss the domain. Domain is a sort of classification of the images into certain categories. Domain is a way for a class of objects to present knowledge representing a certain concept held by objects (Yoshitaka et al, 1994).

We can make use of the various properties of the features of an image to define the domain in which the image concerned would lie. For example, from an image we can understand whether the image is that of a geographical location of a certain area of the earth or the image is that of a person or object with certain background like a screen or a landscape behind. This kind of classification at the initial stage of search should enable us to access the image database rapidly and more efficiently.

Also in a multimedia database, the database might contain various kinds of records like the images, the data in figures, the documents, audios, videos and so on. The concept of domain would classify them according to their categories.

The domain could also be formulated based on the length of the certain features of the image like finding the images of a particular person or object where the length is of certain range value. Color could also define a domain where the images having a particular color or a combination of colors lie in one domain. Domain could be defined on the basis of objects only. For example, we can pick up image where the images of a triangle and square would be present. We can make it more specific by mentioning color and length or size of the triangle and the rectangle to pick up images like where the triangle is red colored and sides of length, say, *(3, 3, 3)* and rectangle of color white and sides of length, say, *(4, 2).*

Variables

Since in our research we are considering a multimedia database where a particular record in the database could be an image or a data record or a document and so on, the definition of variables would vary depending upon the type of records we are considering.

In an image database, where we would consider two-dimensional pictures, the image of an object or a person have to be measured. If we try to measure it graphically, then the size could be measured in terms of x and y coordinates. Therefore, size would be a very important variable in our research.

Color could be another very important variable. There are, as we know, many colors available and for specific definition of the image, the color would play a very vital role. It is possible to define colors digitally in the sense that each color could be given a digital number like red =*1*, blue = *2* and so on.

Location would point towards where a particular object of interest is situated. If we were interested in finding the image of a person with a certain background at a particular point of the image, then location would tell us about where it is present. In a graphically defined image, the coordinate of the center of the image would give information about the location of the object.

Distance between two particular objects of interest is a variable to be considered. Sometime in an image depicting a geographic picture, the distance between two points is very important. Here also we should be able to measure the distance between two objects by applying graphical methods.

Motion of objects, like storm or cloud moving in a satellite picture, is also an important variable. We have to measure the distance traveled by the object in the picture and then convert it into kilometers and notice the time difference. From these we can measure the speed, velocity and so on of the object.

Constraints

It could be a very good idea to define an image in terms of various constraints in the sense that constraints help define the image more specifically.

In our case of multimedia database, where various kinds of data could be there, the concept of constraints is very important. For example, if the image is that of a rectangle, we know one of the constraints would be that the number of sides of the object is *4*. Then the second constraint would be opposite sides are parallel. The third one would be opposite sides are equal. If we include the emergence phenomenon, then if there is a diagonal drawn on it, this would give rise to two triangles. These constraints together could define the image successfully.

In an image of a geographical map of any part of the world, the concept of constraint would be effective in finding the location. If we are interested in finding a place, for example, an island with triangular shape, then obviously the constraint would be number of sides is *3*. If we have more information about sides like whether any two sides are same or all sides are same or all sides are of different size, then these kind of information should help us identify the object more accurately.

Similarities

In a database containing only data, the input may be a query with certain constraints like to pick up records from a SALARY database where salary is greater than, say, *30000*. In relational database, as we know, this can be accomplished by a SQL command with the following kind of statement:

SEL * FROM SALARY_DB WHERE SALARY > *30000*.

This would pick up all the records with salary > *30000*.

In our multimedia database system, this kind of queries could also be made and we can handle them with this kind of or more complicated SQL statements.

But when the input is in the form of image, then we have to find the similarities of the input in the image part of the database. The basic approach to the problem of similarity is to find certain parametric values as well as some coordinates of the input image. Then we find the same for various image of the image database and pick up records where some matching occurs. Of course, we study the emergence phenomenon in both input and images of an image database while calculating parameters. For example, if we want to find similarities involving a triangular figure as input, then some of the parametric values could be defined like, number of sides which is 3, length of each sides, color of the triangle. Based on these values and constraints, we can find out similarities in the image database. But in the image database, there could be figures like squares or rectangles with a diagonal drawn on them. Then obviously this diagonal gives rise to two triangles according to emergence. So we have to study these cases too, find out the parameters of these triangles to see whether they match our parameters from the input.

Indexing

In the early stage of data processing, there was no established conception of indexing. Most of the data files were accessed sequentially. This was pretty slow and inefficient particularly when the data file is big enough. To get rid of this problem, the concept of indexing came to the picture. At the initial stage, a number is used to be given against each record by the system in a file created on disk. We could specify these numbers to access any record randomly. Then came the concept of Indexed Sequential Access Method where instead of assigning separate number against each record, a field or a combination of fields were started being used as key. There could be two kinds of indexing, one where the key value in a particular file is unique and the other where the key value could be duplicate. The search method is called Binary Search where to find a particular key value, the whole file is divided into two halves and the part, which contains the particular key value we are searching, is taken and then divided into two halves again. The part here which contains the key value is again taken and divided into two halves. This process continues until it finds the match against the key value.

Nowadays an old file system is hardly used in maintaining computer records. Instead, a database system has been developed. The latest development in this field is Relational Database System, which contains tables to store data.

In our problem of dealing with multimedia databases, which would contain images, the concept of indexing is very important. We are trying to develop a more sophisticated method of indexing where there won't be any clear-cut definition of index against the images, but indexes would be defined based on our study of emergence phenomenon of each of the image.

Sometime to locate a particular spot in the geographic map of a part of the world, an input image would point to a particular part and that particular part in one or more than one image could be the outcome of emergence or it could be straight away present in the map without any emergence. In either case, input image refers to an index, which is nothing but that particular spot of the map.

1.6.2 Model of the Emergence Index

Emergence indexes can be defined out of five factors as discussed in section 1.6.1.

$$EI = f(D, F, V, C, E) \qquad (1)$$

Where *EI* stands for emergence index, *D* for domain where the image belongs, *F* for features, *V* for variables which can define the feature's constraints under which the features are defined, *C* for constraints and *E* for emergence characteristics of images.
We believe any image, static or in motion, could be expressed semantically in terms of the above mentioned five parameters.

Construction

We take the case of a square with a diagonal, as mentioned earlier, to build up an emergence index. If this is an image in the database, then firstly we have to put it under certain domain *D* for the ease of accessing. Since images are generated in enormous volume, we have to put them in various separate entities or tables according to certain classification rather than in one table which could be extremely time consuming to access. The table that would contain this square image record would define the domain of the image. We can term it as *TAB1*.

To define the second factor *F*, we find the number of maximum sides present would be *5*, where there are *4* regular sides and *1* diagonal.

The variables are *a, b, c, d, e* where first four define the perimeter of the square and *e* the diagonal.

The constraints *c* are $a = b = c = d$ since it is a square.

The emergence *E* is composed of two triangles with sides *a ,b, e* and c, d, e.

Hence Emergence Index

$$EI = \{TAB1;\ 5;\ a, b, c, d, e;\ a = b = c = d;\ (a, b, e \text{ and } c, d, e)\}$$

1.7 Conclusion

In this chapter, we provided the general background of our area of research. We defined the problems of the research done so far in this field. We then described aims and scope of the research we are undertaking and mentioned the plan of the thesis. Lastly we described concepts, definition and structure of emergence index.

Chapter 2 Relevant Research and Procedures Followed in the Thesis

In this chapter, we present the analyses of works done in the field of content-based image retrieval. Although plenty of research works have been done so far in the field, no universally accepted model has yet been developed. The research concentrated on image segmentation based on low-level features like color, shape, texture and spatial relations. Although image segmentation has been achieved based on low-level features, quite often they do not bear any proper meaning. Meaningful image segmentation has not yet been achieved except only in some limited cases. Also to find the semantic meanings or high-level meanings of an image like whether it is the image of human beings or a bus or a train and so on is still a problem. Attempts are being made to link low-level and high-level features. But it is proving difficult for the very simple reason that there remains a vast gap between human perception and computer perception. In addition, very little work has been done so far to apply the phenomenon of emergence in content-based image retrieval.

2.1 Previous Research in Content-based Image Retrieval

Attempts have been made to give rise to symbolic representation of shapes where a shape is defined as (Gero, 1992b)

$$S = \{N; \text{Constraints}\}$$

where N is the cardinality i.e. the number of infinite maximal lines constituting shape S and the constraints limit the behaviors or properties resulting from the infinite maximal lines, based upon which a particular shape is defined.

Lines have been defined as Ik, Ij and so on with their intersection as Ikj. Then topological, geometric and dimensional properties are defined.

Also symmetry has been found through the corresponding relevant positions of the lines and coordinates of one shape with that of the other and in the process emergence of the shapes are studied (Jun, 1994).

There is no direct approach to solve the problem of emergence index other than the one mentioned above. Only there is an indirect approach where this conception has been applied. In a model named Copycat involving computer programs, the program makes all possible sets of consistent combinations of pairings once all plausible pairings have been made. In other words, it gives rise to something explicit which were implicit earlier, which is the essential feature of emergence phenomenon. Also it studies symbolic system where the symbol used, as descriptions are explicitly defined (e.g., a single node in a semantic network represents the concept dog). In a sub-symbolic system, symbols are statistically emergent entities, represented by complex patterns of activation over large numbers of sub-symbols (Mitchell and Hofstadter, 1994).

There does not seem to be any other approaches to the problem of emergent index. Of course, there are extensive works done in accessing image databases, but all of them seem to be concentrated on accessing existing images without considering emergence phenomenon. Below we present the research works done so far starting with image retrieval then relevance feedback, existing gap between low-level and high-level features followed by relevant pattern analysis and data manipulation issues.

Image retrievals

As we pointed out, plenty of research works have been done in image retrieval based on contents of the image. One approach is where document images are accessed directly, using image and object attributes and the relative positions of objects within images, as well as indirectly, through associated document components. This is based on retrieving multimedia documents by pictorial content. Queries may address, directly or indirectly, one or more components. Indirect addressing involves references from associated components, e.g., an image caption is a text component referring to an image component and so is an in-text reference to an image. A symbolic image consists, in general, of objects, relations among objects and descriptions of object and image properties. The properties of objects and whole images are described by object and image attributes respectively (Constantopoulos et al, 1991).

Then there is another method of querying and content-based retrieval that considers audio or visual properties of multimedia data through the use of MORE (Multiple Objects Relationship). In MORE, every entity in an application domain is represented as an object. An object's behavior is presented with a method, which activates the object by receiving a certain message. Objects bearing the same characteristics are managed as a single class. Generally, a class contains structural definitions, methods, or values that the objects in the class commonly posses

(Yoshitaka et al, 1994). Attempts have been made to retrieve similar shape when shapes are measured by coordinate systems (Mehrotra and Gary, 1995).

A system named MARCO (denoting MAp Retrieval by COntent) that is used for the acquisition, storage, indexing and retrieval of map images is presented. The input to MARCO is raster images of separate map layers and raster images of map composites. A legend-driven map interpretation system converts map layer images from their physical representation to their logical representation. This logical representation is then used to automatically index both the composite and the layer images. Methods for incorporating logical and physical layer images as well as composite images into the framework of a relational database management system are described. Indices are constructed on both the contextual and the spatial data thereby enabling efficient retrieval of layer and composite images based on contextual as well as spatial specifications. Example queries and query processing strategies using these indices are described. The user interface is demonstrated via the execution of an example query. Results of an experimental study on a large amount of data are preserved. The system is evaluated in terms of accuracy and in terms of query execution time (Samet, 1996).

As content-based image retrieval is emerging as an important research area with application to digital libraries and multimedia databases, the focus, in this paper, is being put on the image processing aspects and in particular using texture information for browsing and retrieval of large image data. It proposed use of Gabor wavelet features for texture analysis and provided a comprehensive experimental evaluation. Comparisons with other multi-resolution texture features indicate that the Gabor features provide the best pattern retrieval accuracy. An application to browsing large air photos is also illustrated (Manjunath and Ma, 1996).

A method is developed for the content-based retrieval of multi-spectral satellite images using invariant representations. Since these images contain a wide variety of structures with different physical characteristics it is useful to exploit several classes of representations and algorithms. Working from a physical model for multi-band satellite image formation, existing algorithms for this application have been modified and integrated. The performance of the strategy has been demonstrated for image retrieval invariant to atmospheric and illumination auditions from a database of 166 multi-band images acquired at different times over areas of the US (Healey and Jain, 1996).

IMEDIA project which is related to image analysis, the bottleneck of multimedia indexing concerns about image analysis for feature space and probabilistic modelisation, statistics and information theory for interactive browsing, similarity measure and matching. To achieve these goals, research involves the following topics: image indexing, partial queries, interactive search, multimedia indexing (Boujemma et al, 2000).

In a project named Efficient Content-Based Image Retrieval, the focus is the development of a general, scalable architecture to support fast querying of very large image databases with user-specified distance measures. They have developed algorithms and data structures for efficient image retrieval from large databases with multiple distance measures. They are investigating methods for merging their general distance-measure independent method with other useful techniques that may be distance measure specific, such as keyword retrieval and relational indexing. They are developing both new methods for combining distance measures and a framework in which users can specify their queries without detailed knowledge of the underlying metrics. They have built a prototype system to test their methods and evaluated it on both a large general image database and a smaller controlled database (Shapiro et al, 2000).

An approach based on visual-based image retrieval method with respect to MPEG-7 still image description scheme is presented. A segmentation method based on a multivariate minimum cross entropy is used hierarchically for partitioning the color image in classes and regions. Local and global descriptors are defined in order to characterize the color feature of these regions. The retrieved images are presented in a description space which allows the user to better understand and interact with the results (Idrissi et al, 2001).

Another paper provides a state-of-the-art account of Visual Information Retrieval (VIR) systems and Content-Based Visual Information Retrieval (CBVIR) systems. It provides directions for future research by discussing major concepts, system design issues, research prototypes and currently available commercial solutions (Marques et al, 2002).

Based on user interaction, a search-by-similarity method is developed in which request and dissimilarity measure are updated. Complex queries are processed by combining relevant images scattered in the database. The authors proved the effectiveness of this method through quality assessment for a large and heterogeneous image database (Fournier and Cord, 2002).

A procedure for combining configurational and statistical approaches in image retrieval has been developed as configuration contains semantic descriptive power but does not posses the vector space structure which statistical feature-based representations contains (Yu and Grimson, 2002).

An object-based image retrieval procedure has been presented which allows user to specify and to search for certain regions of interest in images. The marked regions are represented by wavelet coefficients and searched in all image sections during runtime. All other image elements are ignored and a detailed search can be performed (Joubert and Kao, 2002).

Based on feature elements instead of traditional feature vectors, an image retrieval system has been developed by finding out whether image contains the feature elements of the demand set. It attempts to analyze synthetically the feedback data, retrieval history and existing result to find out the associated elements that potentially hit the retrieval target. Experimental result is reported to have shown inspiring future for the approach (Xu and Zhang, 2002).

Scene Structural Matrix (SSM) has been applied to retrieval of landscape images. The SSM takes into account the whole structural characteristics of the scene by indexing the geometric features of the image. A binary image tree (bintree) is used to partition the image and from which multi-resolution geometric structural descriptors of the image are derived. The authors showed that SSM is particularly effective in retrieving images with strong structural features like landscape photographs (Qiu and Sudirman, 2002).

The problem of retrieving images from a large database is addressed using an image as a query. The method is specifically aimed at databases that store images in JPEG format and works in the compressed domain to create index keys. A key is generated for each image in the database and is matched with the key generated for the query image. The keys are independent of the size of the image. Images that have similar keys are assumed to be similar, but there is no semantic meaning to the similarity

(Shneier and Abdel-Mottaleb, 1996). An approach to data-adaptive and user-adaptive image retrieval based on the idea of peer indexing has been presented which describes an image through semantically relevant peer images. Every image has been assigned two-level peer index, which models the data characteristics of the image and the user characteristics of individual users with respect to that image. From this two-level image peer indices, retrieval parameters like query vectors and similarity metric can be optimized for data and user characteristics by applying the pseudo feedback strategy (Yang, J. et al, 2003). A model of image retrieval is proposed by using keyblocks which are analogous to keywords in text document retrieval. An image can be represented as a list of keyblocks similar to a text document which can be considered as a list of keywords and various feature models can be constructed for supporting image retrieval (Zhu et al, 2002).

A histogram generation technique using HSV (Hue, Saturation and Value) color space has been proposed for image retrieval. The histogram retains a perceptually smooth color transition that makes it possible to do a window-based comparison of feature vectors for efficient image retrieval from very large databases. For the purpose of ordering of image feature vectors, a vector cosine distance measure is used (Sural et al, 2002). In an attempt to overcome the drawback of the histogram techniques of color image retrieval which consider only global properties and hence cannot effectively define an image, a scheme to capture local properties has been developed for more accurate retrieval. The original image is segmented into several subimage blocks and color histograms for every subimage block is generated. All these color histograms generated are then combined into a multidimensional vector to search database for similar images (Chang et al, 2002).

A new method of content-based image retrieval has been developed by using the color distribution of images and the method is called Metric Histogram. This method takes into account adjacent bins of histograms and thereby reduces the dimensionality of the feature vectors extracted from images. According to the authors, this is a faster and more flexible indexing and retrieval process (Traina et al, 2003). This method is built on the basis of metric distance function, which was developed from the work of Burkhard and Keller (Burkhard and keller, 1973) and has recently achieved an efficiency level sufficient to be used in practical problems. This is obtained using the triangle inequality property to prune blocks of objects, as the B-trees do using the ordering property of ordered domains. This is the case of the M-tree (Claccia et al, 1997), the Slim-tree (Traina Jr et al, 2000) and the Omni-family members (Santos et al, 2001).

In face detection in color images, a method has been used for integrating the well-known color models by using fuzzy set based concept. The shape analysis is performed by using RAMHD, an enhancement of the conventional Hausdorff Distance. Also an algorithm for updating the elliptical model has been developed (Srisuk and Kurutach, 2002). A methodology to find multiple persons in image has been developed by finding face-like regions through skin, motion and silhouette features. Attempts have been made to eliminate false faces based on face geometric and the Support Vector Machine (SVM) by developing an algorithm. To get rid of the effect of lighting changes, a method of color constancy compensation is applied. To track multiple persons, a face-status table is used. The authors claim the method is much robust and powerful than other traditional methods (Hsieh et al, 2002).

Also a video based face recognition system by support vector machines is presented. The authors used Stereovision to coarsely segment face area from its background and then multiple-related template matching method is used to locate and track the face area in the video to generate face samples of that particular person. Face recognition algorithms are based on Support Vector Machines of which both “1 vs. many” and “1 vs. 1” strategies are discussed (Zhuang et al, 2002).

An approach for semantic image annotation and retrieval has been proposed based on monotonic tree model (Song and Zhang, 2002). The branches of the monotonic tree of an image, termed as structural elements are classified and clustered based on their low- level features such as color, spatial location, coarseness and shape. Each cluster corresponds to some semantic feature. The category keywords indicating the semantic features are automatically annotated to the images. For computation of monotonic trees, a top-down algorithm is available. The detailed discussion of the algorithm and the properties of monotonic trees could be found in (Song, 2002). According to the authors, based on the semantic features extracted from images, high-level (semantic-based) querying and browsing of images can be achieved and semantic features can be effectively retrieved and located in image (Song et al, 2003). Also a system for the image indexing and retrieval using speech annotations based on a pre-defined structured syntax is presented where an introduction of N-best lists for index generation and a query expansion technique is explored to enhance the query terms and to improve effectiveness. Through addition of the most probable substitutions for the query terms, more relevant images are distinguished from the data collection (Jlayi et al, 2003).

Relevance feedback

In order to help the users retrieve the correct images they seek, relevance feedback techniques have been developed. This involves allowing users to make further selections from the initial lot of images, presented for a query. The users can keep on refining the search from the results of the previous search until they get the desired images or closest to what they desire. Issues regarding relevance feedback have been presented where the linear and kernel-based biased discriminant analysis, BiasMap is proposed to fit the unique nature of relevance feedback as a small sample biased classification problem. Also a word association via relevance feedback (WARF) formula is presented and tested for erasing the gap between low-level visual features and high-level semantic annotations during the process of relevance feedback (Zhou and Huang, 2002).

Gap between low-level and high-level features

There is work that addresses the issue of gap existing between low-level visual features addressing the more detailed perceptual aspects and high-level semantic features underlying the more general aspects of visual data. Although plenty of research works have been devoted to this problem so far, the gap still remains (Zhao and Grosky, 2002). To address this problem of large gap existing between low-level image features and high-level semantic meanings, attempts have been made to apply feedback techniques to refine the query or similarity measures in image retrieval process. Few relevance feedback algorithms have been presented alongwith semantic learning processes. Low-level features and keyword annotations are integrated in image retrieval and feedback processes in order to improve the performance. The methodology developed has been applied to web image search engine as well. The authors claim the experimental result shows this approach better than other CBIR and relevance feedback approaches (Zhang et al, 2003). Details of the algorithms can be found in the references (Chen et al, 2001; Jing et al, 2002; Lu et al, 2000; Su et al, 2001; Su et al, 2001).

Pattern analyses

Among the methods developed to capture shape of an object, Fourier descriptors achieve good representation and easy normalization. To overcome the drawbacks of Fourier descriptors to locate local shape features, a few methods are proposed which include short-time Fourier transform and wavelet transform. A comparison has been drawn in shape retrieval using Fourier descriptors and short-term Fourier descriptors and query data (Zhang and Lu, 2002). Based on Synergetic Neural Network (SNN) proposed by Hermann Haken, its associated discrete SNN has been presented and the recognition stability and the convergence of a generalized discrete SNN is analyzed. An algorithm of iterative step length refinement for synergetic recognition to ensure fast convergence and network stability for different kinds of input pattern has been developed. Also the concepts of SNN has been applied to trademark retrieval and its ability to support affine invariant retrieval of 2D patterns is studied (Zhao et al, 2002). Also to represent a 3D model by a set of characteristic views has been proposed and the model has been indexed by using this set of views. This approach is independent of the facetisation of 3D model but the authors seem to have problems in the choice of the number of view to characterize the 3D model (Mahmoudi and Daoudi, 2002).

In spatial relation model, a new representation model named ‘Two Dimension Begin-End boundary string’ (2D Be-string) is proposed. This model represents an icon by its MBR (Minimum Bounding Rectangle) boundaries and a number of dummy objects. Also an image similarity evaluation method based on modified Longest Common Sequences (LCS) is presented (Wang, 2002). Range searches in multi-dimensional space have been studied extensively and several excellent search structures have been devised. However, all of these require that the ranges in the different dimensions be specified independently. In other words, only rectangular regions can be specified and searched so far. Similarly, non-point objects can be indexed only in terms of their bounding rectangles. Polyhedral search of regions and polyhedral bounding rectangles can often provide a much greater selectivity in the search. How to use multi-attribute search structures for polyhedral regions, by mapping polyhedral regions into rectangular regions of a higher dimensions has been shown (Jagadish, 1990).

Fingerprint databases are characterized by their large size as well as noisy and distorted query images. Distortions are very common in fingerprint images due to elasticity of the skin. In this paper, a method of indexing large fingerprint image database is presented. The approach integrates a number of domain-specific high-level features such as pattern class and ridge-density at higher levels of the search. At the lowest level, it incorporates elastic structural feature-based matching for indexing the database. With a multilevel indexing approach, the search space is reduced. The search engine has also been implemented on Splash 2 - a field programmable gate array (FPGA)-based array processes to obtain near ASIC level speed of matching. This approach has been tested on a locally collected test data and on NIST-S, a large fingerprint database available in the public domain (Ratha et al, 1996).

Data manipulations

To assimilate data from disparate sources to provide a unified view to a user, there has been significant interest in XML and semantic web in recent times. The role of semantics in communication and computing has been studied and some ideas has been presented regarding how semantics could be an emergent process in experimental environments rather than an appendage (Jain, 2003). Also a method has been developed which creates high dimensional index structure that adapts to the data distribution and adjusts well with the database size. Marginal distribution of the data along each dimension is characterized using Gaussian mixture model and parameters are estimated using Expectation-Maximization method. Using marginal distribution information, each of the dimensions can be partitioned in such a way that each bin contains approximately equal number of objects (Wu et al, 2002).

2.2 Input in the Form of an Irregular Shape

Now an approach to use emergence index is presented by giving a very simple example of an image consisting of a lake and two houses. We show how emergence index can generate different retrievals from databases. In order to present an approach to use this concept, we have to consider four aspects of an image to define the image properly. They are domains, features, constraints and emergence. Once we can define the properties of input and images of the database to be retrieved, it would be easy to retrieve the image based on input given.

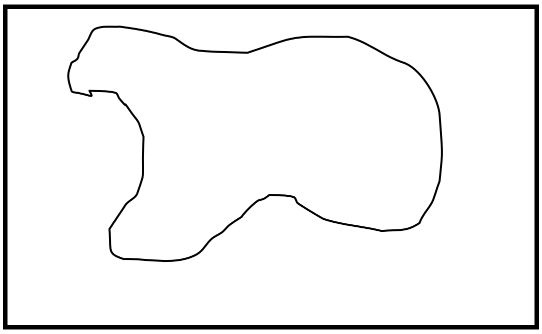


Figure 2.1 An irregular shape

We propose to study a very irregular shape.

First we would study the features of the image. If we consider the image including the rectangular perimeter, then we have the texture or background of the image. If we do not consider the perimeter of the image, then the irregular shape itself defines the image without any background. We call this the object.

In order to study the perimeter of the object we use straight lines to define its shapes which fit closest to its curvatures in the form of polyhedra. Defining the perimeter is a very important step in shape recognition.

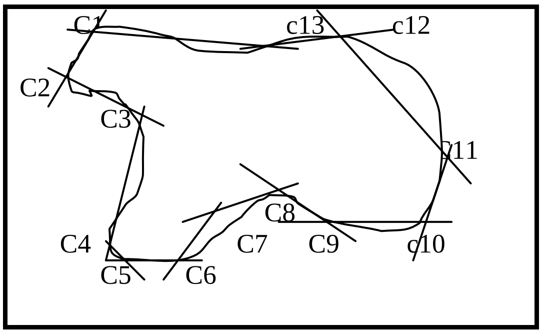


Figure 2.2 The irregular shape defined in 13 coordinate points based on tangential intersections

If we count the number of straight lines defining the object, it is *13*. Also the number of intersections of these straight lines is *13*. We define the coordinates of these intersections as *C1*, *C2*, *C3*,....., *C13*. These *13* coordinate points can quite effectively describe the irregular shape under consideration.

This way we could transform the image into the semantic form.

If we are looking to establish symmetry with another image, then wherever those *13* coordinate points match respectively would identify a similar image for retrieval.

If we define the whole video screen as the perimeter, then the perimeter of the image would be in the shape of the screen, which could be a square. In that case, the perimeter of the object also defines the internal perimeter of the texture or background. Not only that, there could be more objects lying in the background part and since now we are considering the whole video, they would also come under consideration.

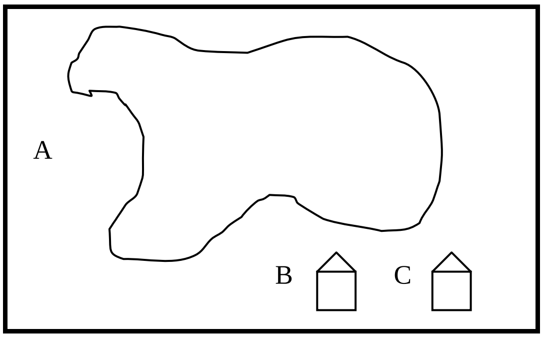


Figure 2.3 A lake and two houses

For example, if the irregular shape under consideration represented a lake, then the background could be residential area with houses built around. If there are two houses *B* and *C* in the background, then the distance between *B* and *C* could also express the features of the whole image under consideration. Also now there would be three objects under consideration --the lake and the two houses. We can measure the distances between each of these three objects.

If we divide the whole video screen in the coordinate system with x and y-axes, then the location of these three objects in this coordinate system could be understood.

If *C1, C2, ..., C13* continue to be our coordinate points for irregular shape A and if *C14, ..., C18* are the coordinate points for House B and *C19, ..., C23* are the coordinate points for House C, then the various distances between these three objects could be measured.

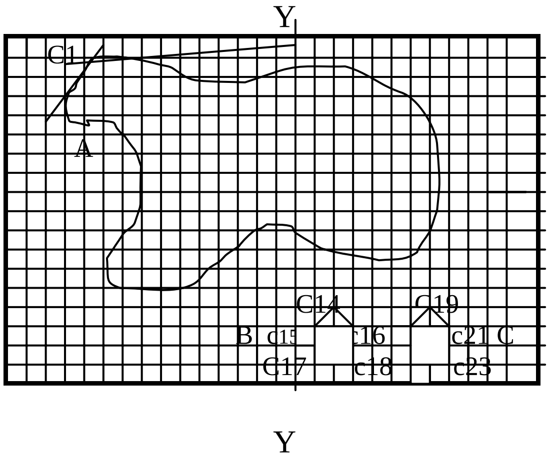


Figure 2.4 Lake and two houses in coordinate systems

Distance between the topmost parts of objects B and C would be *d1 = d(C19 ~ C14).* Similarly distance between topmost part of object A and topmost part of object B would be *d2 = d(C1 ~ C14)* and distance between topmost part of object A and topmost part of object C would be *d3 = d(C1 ~ C19).*

The structure of the above-described image consists of a lake and two houses. Function of the lake is to store water for whatever purposes. Similarly the function of the two houses is to accommodate people. Behavior of the lake is to supply water and houses for accommodation.

But the overall behavior changes when we consider this lake and two houses as giving rise to a residential area. This is the explanation of the image when emergence is considered.

So whenever an input would call for retrieving pictures of residential area, then this image could be retrieved although its components parts are different than what is asked in the input.

2.3 Input in the Form of Text

Here we describe approaches as to how to retrieve images from databases when the input comes in the form of a text. It is possible somebody could put text which says 'Find all umbrellas'. Then obviously we have to pick up all images where at least one image of an umbrella is existing. But we do not have any input parameters as from the above statement, we cannot deduce any parameters.

To handle this kind of query, one available option seems to be to maintain a table containing data dictionary which stores various item names like umbrella, chair, table, tree and so on and corresponding information about their shape. The information would be regarding some basic features of the item under consideration. For example, for umbrella, we have to define it as an item containing a handle, then a round shaped structure atop the handle, made up of cloth, with thin metallic support which keeps the cloth in shape. Of course all this descriptions would be in the form of parameters and stored in the database. So when the user wants all umbrellas to be displayed then Data Dictionary would find the corresponding features in parametric form and go to access the database. Whenever it finds a match with certain goodness of fit, that particular image would be selected.

Now to define the features of an umbrella, we have to find out some common features, which could be

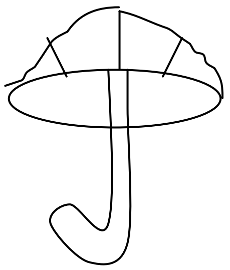


Figure 2.5 Image of an umbrella

applicable to any umbrella since the size of the umbrella could vary and so also the design. But certain basic characteristics would be common to all of them. We have to define those basic characteristics. We consider only umbrellas opened, say. We do not consider any folded umbrellas.

1. Each and every umbrella has an upper part, which is like a ball cut into halves. The size of the radius, of course, could vary.
2. From the topmost part, a handle rod comes out.
3. Also from the rod, in the upper half portion, few metallic structures emerge, bent shaped support the upper part to keep it in shape.

Basically these three features would be common to any umbrella. So wherever in the image database we find similar structure, we could pick that up for selection.

So Feature *F* = { Size of the circular upper part, Handle, Metallic structure }

= { *1/2* Sphere, Straight Rod, Curved rods}

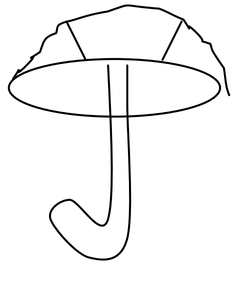


Figure 2.6 Studying the features of the umbrella

Basically these three features would be common to any umbrella. So wherever in the image database we find similar structure, we could pick that up for selection.

So Feature *F* = { Size of the circular upper part, Handle, Metallic structure }

= { *1/2* Sphere, Straight Rod, Curved rods}

Since the size of the pictures of umbrellas could vary, as we mentioned earlier, we cannot specify any particular value to these three feature items.

This particular shape could also be interpreted as the shape of an earring. Hence the emergence outcome of the shape of an umbrella could be the shape of an earring and vice versa. Hence, when we search for umbrellas, we could pick up shapes which resemble umbrellas but which are actually shapes of earrings.

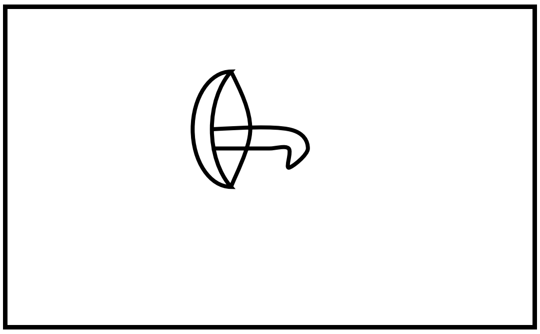


Figure 2.7 Image of an earring

So whenever a feature describing an umbrella is input, we could pick up an earring if the shape satisfies symmetry criteria.

Another available option is to have icons of various objects preserved in the database. So for umbrellas, there would be few icons, each icon defining umbrella in various styles like open, closed and so on. The word ‘umbrella’ itself could be used as index for the icons for umbrella. So when a query comes ‘Find all umbrellas’, we would access the database containing icons against index umbrella where umbrellas at various positions would be kept. We could pick up all of those icons and search the image databases for each one of them and pick up those where symmetry is obtained within a limit.

2.4 Conclusion

In this chapter, we presented the analyses of works done in the field of content-based image retrieval, where although plenty of research works have been done, still no universally accepted model has yet been developed. Also we presented approaches to use emergence index.

Chapter 3 Semantic Representation of Images

Here we discuss how emergence index is useful in image database search, symbolic representation of images through infinite maximal lines which are straight lines and constraints. Also we present various mathematical tools that would be used throughout the dissertation. Then we describe model of emergence by showing how destruction of an original image leaves an unstructured image and then processing this unstructured image could give rise to a new structured image which is different from the original image considered.

3.1 Relevance of Emergence Index in Image Databases

Emergence phenomenon can change the structure, function and behavior of the image in a very effective way. Because when we search and analyze the implicit meaning of an image, as we have mentioned earlier, we can end up discovering many other form of images in the particular image which can give rise to many more different meanings in the image. Hence if we access an image database based on emergence index, it would be altogether a different search and would be having more precision and recall than normal search where we consider only the explicit meaning of the image.

3.2 Symbolic Representation of Images

As we have mentioned earlier, the symbolic representation of shapes can be defined using infinite maximal lines as

$$I = \{N; constraints\} \qquad (2)$$

Where N is the number of infinite maximal lines which effectively constitutes image I and constraints are restrictions which define behaviors or properties that come out from the infinite maximal lines (Gero, 1992b).

Various mathematical tools

1. Geometric properties

There are four geometric properties involved in infinite maximal lines:

Two lines La and Lb are perpendicular, $La \perp Lb$

Two lines La and Lb are parallel, $La \,/\!/\, Lb$

Two lines La and Lb are skewed, $La \times Lb$

Two lines La and Lb are coincident, $La = Lb$

2. Topological properties

Intersection and segment are two properties of a set of infinite maximal lines.

If La and Lb are two infinite maximal lines then intersection Iab would be denoted by

$$La \times Lb => Iab$$

$$La \perp Lb => Iab$$

The first case of the above is the skewness whereas the second case is perpendicularity of the geometric property.

The intersection cannot occur if $La \mathbin{/\!/} Lb$ or $La = Lb$. In other words, parallel behavior of two infinite maximal lines and also coincidence do not generate any intersection.

Properties of intersection

a. Iab is same as Iba.

b. Iab and Ibc are called collinear intersection in Lb.

c. $Iabc$ exists if La, Lb and Lc are concurrent.

The segment generated by two intersections is denoted by (Iab, Ibc) and this segment lies obviously in infinite maximal line Lb.

There are three types of intersection groups: ordinary groups, adjacent groups and enclosed groups. These three groups indicate three kinds of topological structures which defines intersections and line segment in different ways.

Ordinary group could be expressed by a pair of '(' and ')' parentheses. In this case a line segment could be defined by two intersections. If La, Lb and Lc are three lines, then segment of line would be

$$(Iab,Icb)$$

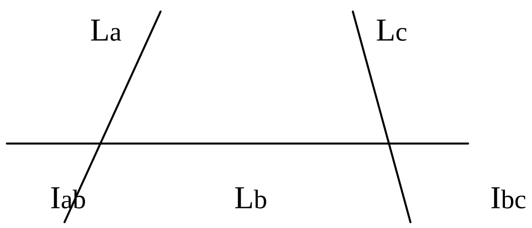


Figure 3.1 Line segment defined by two intersections of ordinary group

In otherwords, the line segment is the intersection between La and Lb which is (Iab) and Lb and Lc which is (Ibc).

The length of the segment (Iab,Ibc) is

$$d(Iab,Ibc)$$

The order of intersections in ordinary group does not matter.

An adjacent group is defined by a pair of angle ‘<’ and ‘>’ brackets. Here only two adjacent intersections can represent line segment and they are of the order of $<I_{ab}, I_{ac}>$ as shown below.

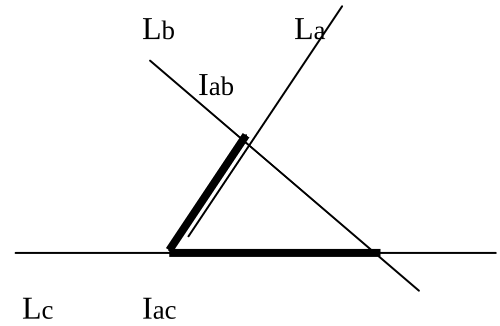


Figure 3.2 Line segment defined by intersections of adjacent group

The first and last intersection in an adjacent group is adjacent to each other. The order of intersections in an adjacent group is important. The same set of intersections defines different line segments when adjacent orders are different.

An enclosed group, defined by a pair of square ‘[’ and ‘]’ brackets represents a circuit of line segments, bounded polyline shape. It is a subcase of adjacent group

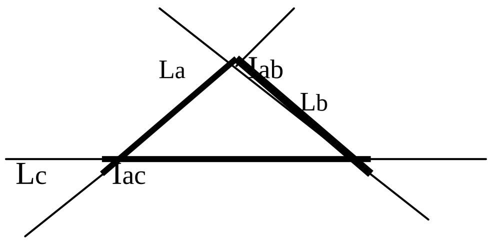


Figure 3.3 Line segment defined by intersections of enclosed group

and any two adjacent intersections must satisfy (I_{ab}, I_{ac}).

This could be represented by three line segments in the figure above as follows

$$[I_{ab}, I_{ac}, I_{bc}]$$

3. Dimensional property

The length of the segment of two intersections is the dimensional property and is denoted by $d(I_{ab}, I_{bc})$.

4. Statistical properties

a) Mean and standard deviation of distances between objects.
b) If $X=x_1, x_2, x_3,\ldots, x_n$ are components of features of an image, then we ascribe weights $w_1, w_2, w_3,\ldots, w_n$ respectively to define the complete features of the image.

c) Boolean =, <, >, *AND*, *OR* could be used in defining various features of image.

d) Summation could be applied when the various individual components of the features of an image multiplied by their respective weights are added to find the total value of the features in calculating the index value of a search.

3.3 Example of a Representation of Image and Symmetry

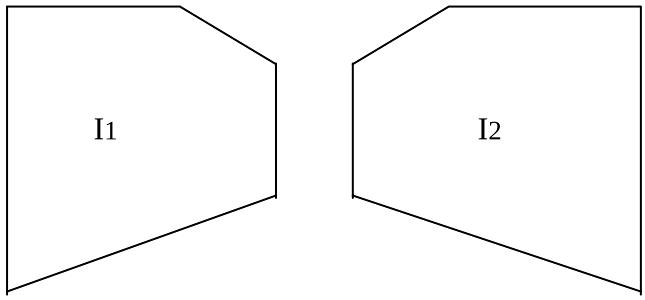


Figure 3.4 Original structures

We consider two images I_1 and I_2 as above.

The symbolic representation of these two images would be

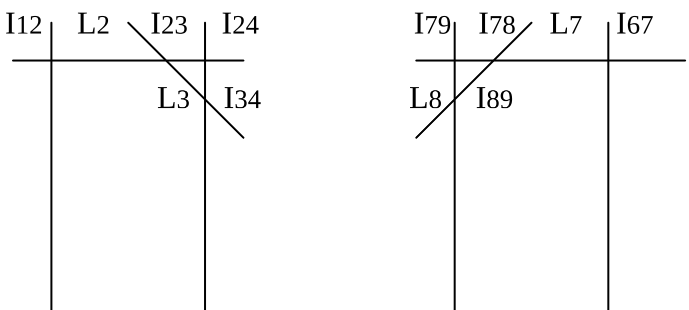
$$I_1 = \{5;[I_{12}, I_{23}, I_{34}, I_{45}, I_{15}]\}$$

$$I_2 = \{5;[I_{67}, I_{78}, I_{89}, I_{90}, I_{60}]\}$$

Where '[' and ']' are enclosed group.

To find the symmetry between the two images, we first find out the number of infinite maximal lines and number of intersections in the images. If the number of infinite maximal lines and the corresponding intersections match, we can succeed in finding symmetry between images.

In the original image, where no emergence is considered, symmetry could be established from the fact that number of straight lines is *5* in both I_1 and I_2. Also I_{12} corresponds to I_{67}, I_{23} to I_{78}, I_{34} to I_{89}, I_{45} to I_{90} and I_{15} to I_{60}. That establishes correspondent similarity between the images and hence symmetry is established.



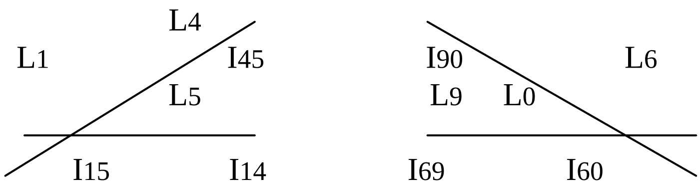


Figure 3.5 Emergent structures

Considering the emergence, the images I_1 and I_2 can be represented as

$$I_1 = \{4;(,I_{24}\ I_{12}, I_{23}, I_{34}, I_{45}, I_{14}, I_{15})\}$$

$$I_2 = \{4;(I_{67}, I_{78}, I_{79}, I_{89}, I_{90}, I_{69}, I_{60})\}$$

Where '(' and ')' represent ordinary group.

We notice the number of infinite maximal lines in both I_1 and I_2 after emergence becomes equal and 4 in this case. Also I_{12} corresponds to I_{67}, I_{23} to I_{78}, I_{24} to I_{79}, I_{34} to I_{89}, I_{45} to I_{90}, I_{14} to I_{69} and I_{15} to I_{60} respectively.

Hence the similarities of I_1 and I_2 after emergence is established and also it could be found that the new shape in each case is no longer the original shape, but rectangles.

3.3.1 Image Formed by Three Lines

According to our definition of an image as

$$I = \{N;\ \text{constraints}\}$$

the image formed by three lines have number of sides $N = 3$.

So $I = \{3\}$.

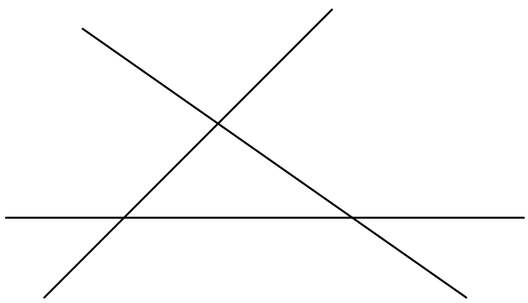
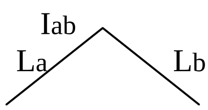


Figure 3.6 Image of a three line shape

Now if we consider the case of a triangle then

$$I = \{3;[I_{ab}, I_{ac}, I_{bc}]\}.$$

Here we use square brackets to define constraints as it comes under enclosed intersection group.



I_{ac} L_c I_{bc}

Figure 3.7 Image in enclosed intersection group

If we consider the case of an isosceles triangle where two sides are equal, $L_a = L_b$ then

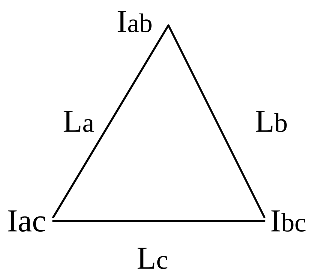


Figure 3.8 Studying the features of an isosceles triangle

$$I = \{3;[I_{ab}, I_{ac}, I_{bc}], L_a = L_b\}$$
$$= \{3; [I_{ab}, I_{ac}, I_{bc}], d(I_{ab}, I_{ac}) = d(I_{ab}, I_{bc})\}$$

$L_a \Rightarrow d(I_{ab}, I_{ac})$ and $L_b \Rightarrow d(I_{ab}, I_{bc})$ by the application of the dimensional property.

Now we go to the case of equilateral triangle where all sides are equal, $L_a = L_b = L_c$.

Hence

$$I = \{3;[I_{ab}, I_{ac}, I_{bc}], L_a = L_b = L_c\}$$
$$= \{3;[I_{ab}, I_{ac}, I_{bc}], d(I_{ab}, I_{ac}) = d(I_{ab}, I_{bc}) = d(I_{ac}, I_{bc})\}.$$

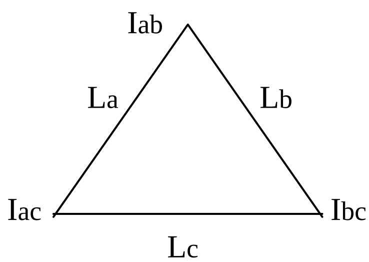
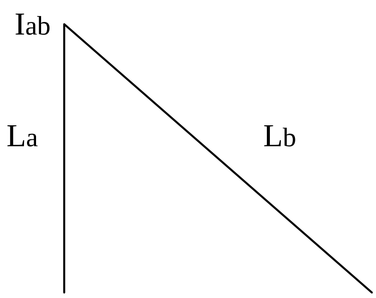


Figure 3.9 Studying the features of an equilateral triangle

Finally we consider a right triangle where one of the three sides is perpendicular to another one. Hence

$$I = \{3;[I_{ab}, I_{ac}, I_{bc}], L_a \perp L_c\}.$$



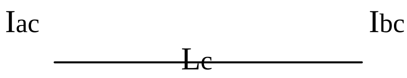


Figure 3.10 Studying the features of a right triangle

3.3.2 Image Formed by Four Lines

For four sided image

$$I = \{4\}$$

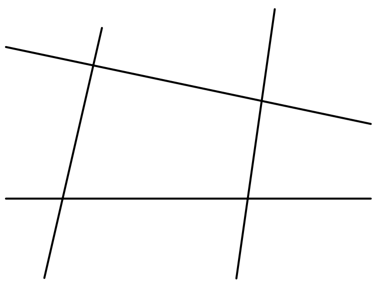


Figure 3.11 Image formed by four sides

When it is a parallelogram

$$I = \{4;[I_{ab}, I_{bc}, I_{cd}, I_{da}],\ L_a /\!/ L_c, L_b /\!/ L_d\}$$

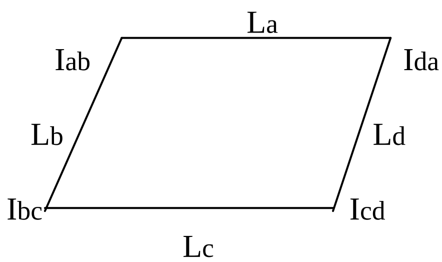


Figure 3.12 Studying the features of a parallelogram

When it is a rectangle

$$I = \{4;[I_{ab}, I_{bc}, I_{cd}, I_{da}], L_a = L_c, L_b = L_d, L_b \perp L_c, L_a \perp L_b, L_c \perp L_d, L_d \perp L_a, L_a /\!/ L_c, L_b /\!/ L_d\}$$

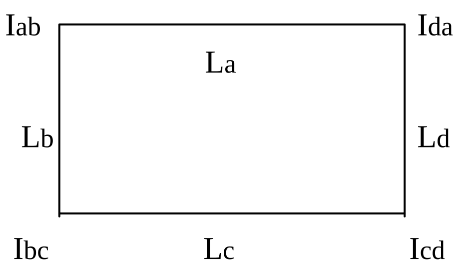


Figure 3.13 Studying the features of a rectangle

When it is a square

$I = \{4;[I_{ab}, I_{bc}, I_{cd}, I_{da}], L_a = L_b = L_c = L_d, L_a \parallel L_c, L_b \parallel L_d, L_b \perp L_c, L_c \perp L_d, L_d \perp L_a, L_a \perp L_b\}$

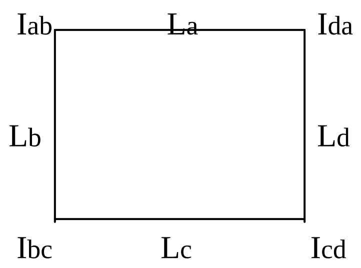


Figure 3.14 Studying the features of a square

Finally when it is a trapezoid

$I = \{4;[I_{ab}, I_{bc}, I_{cd}, I_{da}], L_a \parallel L_c\}$

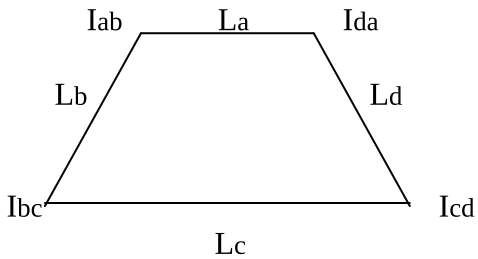


Figure 3.15 Studying the features of a trapezoid

We presented examples for images formed with three and four lines. But it could be extended to images formed with more than four lines.

3.4 Emergence in Images

3.4.1 Model of Emergence

When we go to analyze an image, first thing we have to do is to segregate the image from the background. For example in the following figure a triangle lies within a rectangle.

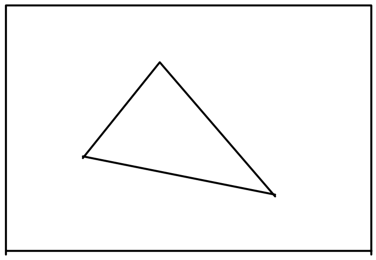


Figure 3.16 A triangle within a rectangle

So if we want to deduct the representation of the triangle or in other words define the constraints which gives rise to the triangle, then we have to identify and separate the triangle from the rest part of the rectangle.

Once we perform the segregation of the triangle from rectangle, then we have to define the constraints and parameters which represent the triangle.

In any image retrieval based on color, texture or shape, these two functions, namely, segregation of the image from the background and then defining the constraints and parameters of the images are needed.

When we go to study the emergence or hidden meaning of an image to make out more features out of it, we have to perform the above mentioned two functions.

Any existing image, whatever it is, we consider as structured image. By structured image we mean a completely defined image or in other words image has some shape in it.

For emergence, the first step is to destroy the structure of the image and make it unstructured. By unstructured we mean the image still has some shape defined in it, but the shape is different than the original or structured shape.

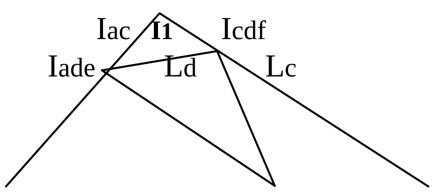
When we destroy the constraints of the original image that define the image, the explicit structure of the image becomes implicit. The purpose of this destruction of the constraints of the original image is to take the image out of any previous bonding that used to define it.

As we have discussed, a line segment is created when there are two separate intersections like I_{ab} and I_{bc} where I_{ab} is the intersection of lines L_a and L_b whereas I_{bc} is of lines L_b and L_c.

If intersections I_{ab} and I_{bc} belong to different groups of image, then no line segment could exist.

By destroying the structure of the image, we change adjacent groups of intersections into one ordinary group and it relaxes the topological constraints of the original image.

For example, in the following shape, image I is represented by three triangles I_1, I_2 and I_3



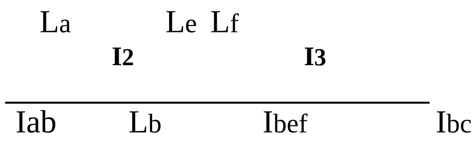


Figure 3.17 Three triangles I_1, I_2, I_3

$$I = \{6;[I_{ac}, I_{ade}, I_{cdf}], [I_{ade}, I_{ab}, I_{bef}], [I_{cdf}, I_{bef}, I_{bc}]\}$$

If we relax the topological constraints of the above representation, it becomes

$$= \{6;(I_{ac}, I_{ad}, I_{cd}, I_{ad}, I_{bf}, I_{ab}, I_{cf}, I_{bc}, I_{be})\}$$

This shows line segments (I_{ac}, I_{ab}),(I_{ac}, I_{bc}),(I_{ab}, I_{bc}) come into expression now. This changes the structure of the original image.

Processing unstructured image

By processing unstructured image, we discover the implicit meaning of an image.

The first step involved in this process is the definition of constraints and to derive new constraints based on the existing constraints. Rules for new constraints derivation, which are needed to process an unstructured image, would be:

1) $L_a // L_b \wedge L_b // L_c \Rightarrow L_a // L_c$
2) $L_a \perp L_b \wedge L_b \perp L_c \Rightarrow L_a // L_c$
3) $L_a \perp L_b \wedge L_b // L_c \Rightarrow L_a \perp L_c$
4) $L_a // L_b \wedge L_b \times L_c \Rightarrow L_a \times L_c$
5) $L_a \perp L_b \Leftrightarrow L_a \times L_b$
6) $L_a \times L_b \Leftrightarrow I_{ab}$
7) $L_a // L_b \wedge L_c // L_d \wedge L_a \times L_c \Leftrightarrow d(I_{ac}, I_{ad}) = d(I_{bc}, I_{bd}), d(I_{ac}, I_{bc}) = d(I_{ad}, I_{bd})$

In the above cases, the symbol $A \Rightarrow B$ means *IF A THEN B* and $A \Leftrightarrow B$ means vice versa. Also symbol ^ means AND. These above-mentioned rules could cover majority of the cases of processing unstructured images. New rules could be formulated as and when required depending upon the particular case under consideration.

The second step involved in the process of discovering implicit meaning has two ways. In the first case, which is hypothesis-driven search, an image is predefined and the particular unstructured image is searched to find a match of the predefined image. In the second case, which is data driven search, a feature or a combination of features are passed through the unstructured image until some rule is satisfied to stop the process. In such cases, the following rules are applied to go through the infinite maximal lines until an image is obtained.

8) $(I_{ab}) => \langle I_{ab}, I_{ac}\rangle; I_{ac} = I_{ab}$

9) $A \cup \langle I_{ab}, I_{ac}\rangle => A \cup \langle I_{ab}, I_{ac}, I_{cd}\rangle; I_{cd}$ does not belong to A

10) $\langle I_{cd}\rangle \cup A \cup \langle I_{ab}, I_{ac}\rangle => \langle I_{cd}\rangle \cup A \cup \langle I_{ab}, I_{ac}, I_{cd}\rangle; I_{cd}$ does not belong to A

11) $\langle I_{cd}, I_{ce}\rangle \cup A \cup \langle I_{ab}, I_{ac}\rangle => \langle I_{cd}, I_{ce}\rangle \cup A \cup \langle I_{ab}\ I_{ac}, I_{ce}\rangle; I_{ce}$ does not belong to A

12) $\langle I_{ab}, I_{bc}, \ldots, I_{pq}, I_{pr}, I_{ab}, I_{bc}\rangle => [I_{ab}, I_{bc}, \ldots, I_{pq}, I_{pr}]$

Rule 8 is the starting point of the search. It generates a line segment, which is part of the line L_a and it begins from the intersection I_{ab} of the unstructured image.

Rule 9 generates line segments where intersections are used which were not used before.

Rules 10,11,12 are the stopping rules which give rise to a structured emergent image.

Example of hypothesis-driven search

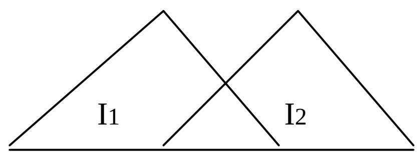
We consider an image I consisting of two triangles I_1 and I_2.

We would like to find all triangles hidden in these two triangles through emergence.

The representation of the image I is

$$I = \{5; [I_{ab}, I_{ac}, I_{bc}], [I_{de}, I_{cd}, I_{ce}]; L_a \ // \ L_d, L_b \ / \ /L_e\}.$$

When we make this original image unstructured through shape hiding, the representation now becomes



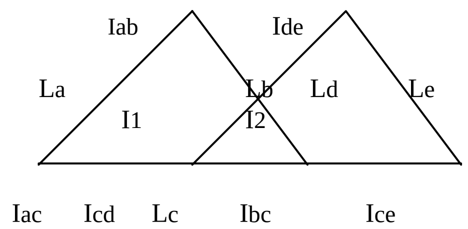


Figure 3.18 Image formed by two triangles I_1 and I_2

$I = \{5;(I_{ab}, I_{ac}, I_{bc}, I_{de}, I_{cd}, I_{ce});L_a // L_d, L_b // L_e\}.$

Now we introduce two new intersections I_{bd} and I_{ae} throughthe following reasoning:

$I_{ab} => L_a \times L_b$

$L_a \times L_b \wedge L_a // L_d => L_b \times L_d$

$L_b \times L_d => I_{bd}$

$L_d \times L_e \wedge L_a // L_d => L_a \times L_e$

$L_a \times L_e => I_{ae}$

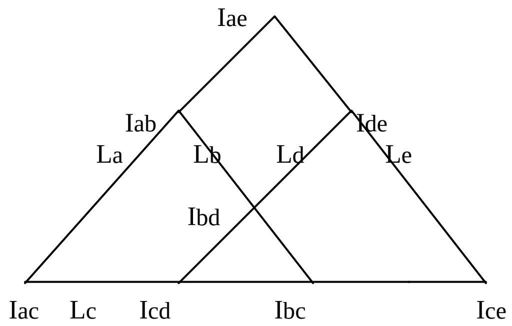


Figure 3.19 Image defined in 3.18 with two new intersections I_{bd} and I_{ae}

The above can be represented by

$I = \{5;(I_{ab}, I_{ac}, I_{bc}, I_{de}, I_{cd}, I_{ce}, I_{bd}, I_{ae});L_a // L_d, L_b //L_e\}.$

Our process of discovering emergence through hypothesis-driven search continues by trying different segments and intersections of the above figure until we can come to a point where it satisfies at least some of the definitions of triangle, isosceles triangle, equilateral triangle, quadrilateral right angle as we discussed earlier.

We come up with two new triangles as follows

$I_3 = \{3;[I_{bd}, I_{cd}, I_{bc}]\}$

$I_4 = \{3;[I_{ae}, I_{ac}, I_{ce}]\}$

Also we come up with at least two quadrilaterals

$I_5 = \{4;[I_{ae}, I_{ac}, I_{cd}, I_{de}]\}$

$I_6 = \{4;[I_{ae}, I_{ab}, I_{bc}, I_{ce}]\}$

Further we can derive few more quadrilaterals

$I_7 = \{4;[I_{ab}, I_{ac}, I_{cd}, I_{bd}]\}$

$I_8 = \{4;[I_{ae}, I_{ab}, I_{bd}, I_{de}]\}$

$I_9 = \{4;[I_{de}, I_{bd}, I_{bc}, I_{ce}]\}$

Example of data-driven search

We take an example of a sixteen side's image as the original image I_1, which is represented by eight infinite maximal lines as follows

$$I = \{8;[Iab, Iac, Ibc, Ibd, Icd, Ice, Ide, Idf, Ief, Ieg,, Ifg, Ifh, Igh, Iag, Iah, Ibh];$$
$$La // Le, Lc // Lg, Lb // Lf, Ld // Lh \}$$

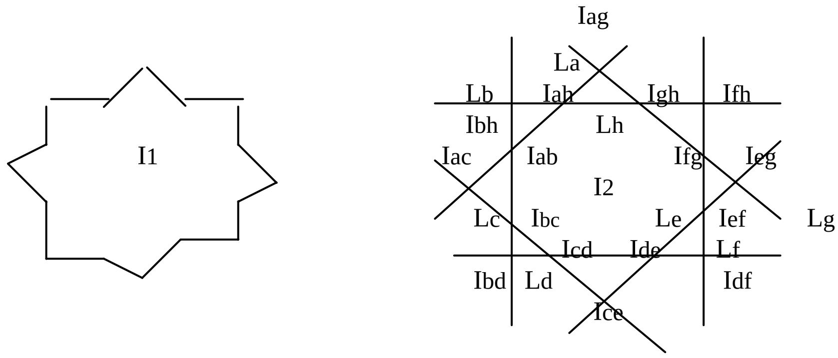


Figure 3.20 Sixteen sided image on the left gives rise to a square after destruction of original shape

When we destroy the structure of the image, it becomes

$$I = \{8;(Iab, Iac, Ibc, Ibd, Icd, Ice, Ide, Idf, Ief, Ieg, Ifg, Ifh, Igh, Iag, Iah, Ibh);$$
$$La // Le, Lc // Lg, Lb // Lf, Ld // Lh \}$$

Emergence is determined by the application of rules 8 to 12.

$(Iag) => \langle Iag, Iac\rangle;$ by 8

$\langle Iag, Iac\rangle => \langle Iag, Iac, Ice\rangle;$ by 9

$\langle Iag, Iac, Ice\rangle => \langle Iag, Iac, Ice, Ieg\rangle;$ by 9

$\langle Iag, Iac, Ice, Ieg\rangle => \langle Iag, Iac, Ice, Ieg, Iag\rangle;$ by 10

$\Rightarrow \langle Iag, Iac, Ice, Ieg\rangle$

This gives rise to a square as shown below

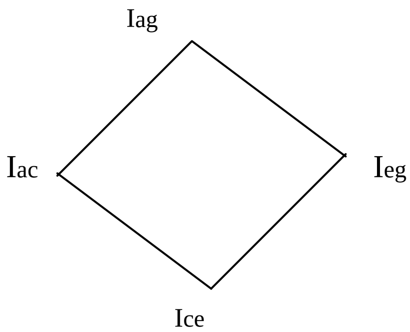


Figure 3.21 A square generated after emergence

This is how a data driven search starts from an open individual point without having any pre-defined format and keeps on searching for an image until it ends up getting a closed image like square above (Gero and Yan, 1994).

Model of emergence

Now we take the unstructured shape and find out the extra or implicit meaning out of it in addition to the original meaning and this process gives rise to emergent image with implicit meaning making explicit. This can be defined in a model as follows (Gero and Yan, 1994):

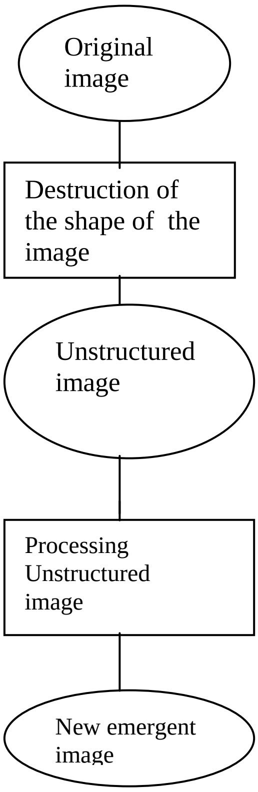


Figure 3.22 Model of emergence

3.4.2 Example of Emergence

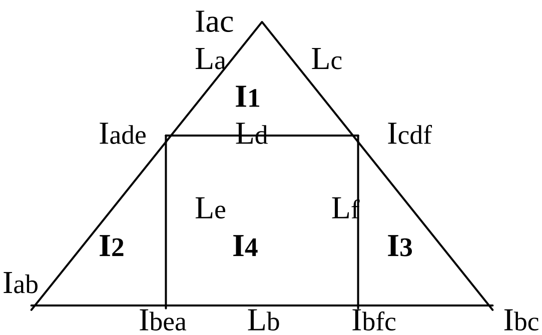


Figure 3.23 Image formed by three triangles *I1, I2, I3* and a square *I4*

Let us consider the figure above where a structure of image is constructed with three triangles I_1, I_2, I_3 and a square I_4. The intersections are four enclosed groups (three for three triangles and one for the square).

The symbolic representation of the image structure I is, considering enclosed groups only

$$I = \{6;[I_{ac}, I_{ade}, I_{cdf}],[I_{ade}, I_{ab}, I_{bea}],[I_{bfe}, I_{bc}, I_{cdf}],[I_{ade}, I_{bea}, I_{bfc}, I_{cdf}]\}$$

Now if we change the topological structure in the above equation and make it an ordinary group then

$$I = \{6;(I_{ac}, I_{ade}, I_{ab}, I_{bea}, I_{bfc}, I_{bc}, I_{cdf})\}$$
$$= \{3;(I_{ac}, I_{ab}, I_{bc})\}$$

The above expression is nothing but a triangle. In other words we destroyed the original structure containing three triangles and a square to give rise to only one

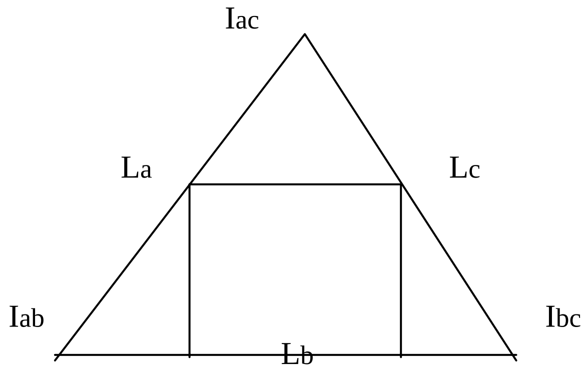


Figure 3.24 An unstructured image after destruction of image structure of Figure 3.23

triangle. Since this structure is totally different than the original one, which we call structured image, we call the transformed structure as unstructured image.

Now we consider the following image structure where we process the unstructured image of the last figure and consider all parts of the last image except lines L_e and L_f.

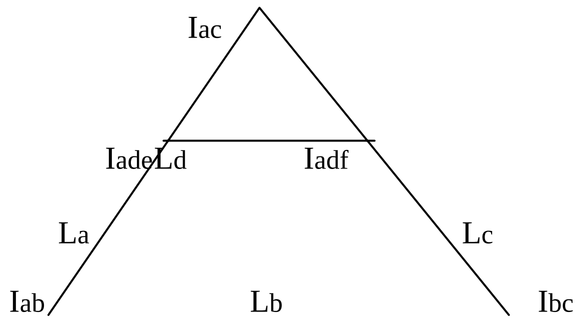


Figure 3.25 Image after emergence from original image in Figure 3.23

The representation of the triangle remains as it was in the last expression and closed group to be

$$I = \{3;[I_{ac}, I_{ab}, I_{bc}]\}$$

Now if we consider intersections $I_{ab}, I_{bc}, I_{adf}, I_{ade}$ of the structure, the representation would be

$I = \{4;[I_{ab}, I_{bc}, I_{adf}, I_{ade}], L_d \,//\, L_b, d(I_{ab}, I_{ade})$ not $= d(I_{bc}, I_{adf})$ not $= d(I_{ade}, I_{adf})$ not $= d(I_{ab}, I_{bc}), (I_{ab}, I_{ade})$ not $// (I_{bc}, I_{adf})]\}$

This satisfies all the conditions of a trapezoid which has 4 sides, length of each side is unequal to others, two sides are parallel, the remaining two sides are not parallel.

So the processing of unstructured image leads to the development of a new image which is trapezoid.

If we consider the original image, we find the structure of a trapezoid lying implicit in the images of the three triangles and a square, as we stated.

This proves our model of emergence.

3.5 Symmetry

Now we come to define symmetry. As we have pointed out, when an input comes, we search the image database to find a match of that input in images of the database. This match is established and that particular image is selected only when the search finds reasonable symmetry of the input image with that particular image of the database. That means symmetry is a very important component of the database search.

Symmetry derives from the Greek word symmetria. It can be obtained between the input image and any particular image of the database when

(1) geometrical, topological or dimensional properties are identical

(2) constraints are the same

(3) coordinate position of the corresponding points of the two images are identical.

Let us consider two original images I_1 and I_2.

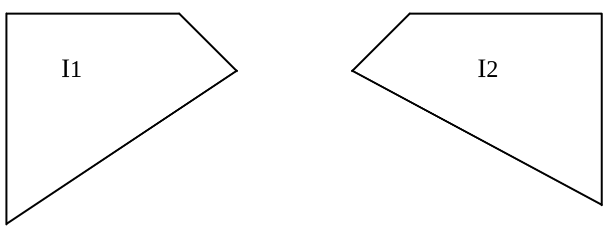


Figure 3.26 Original images

The representations of these two images are

$$I_1 = \{4;[I_{ab},I_{bc},I_{cd},I_{ad}]\}$$

$$I_2 = \{4;[I_{ef},I_{fg},I_{gh},I_{eh}]\}$$

In order to determine the structure correspondence, we see the number of infinite maximal lines and number of intersections and whether they match to each other.

Obviously the number of infinite maximal lines in both I_1 and I_2 are same and 4. Also both I_1 and I_2 have same number of intersections and 4. After that, we consider geometrical constraints of infinite maximal lines like parallel behavior, perpendicular behavior, skewness, coincidence to determine whether the two images are identical or not.

We now destroy the original structures as we have established equivalence between

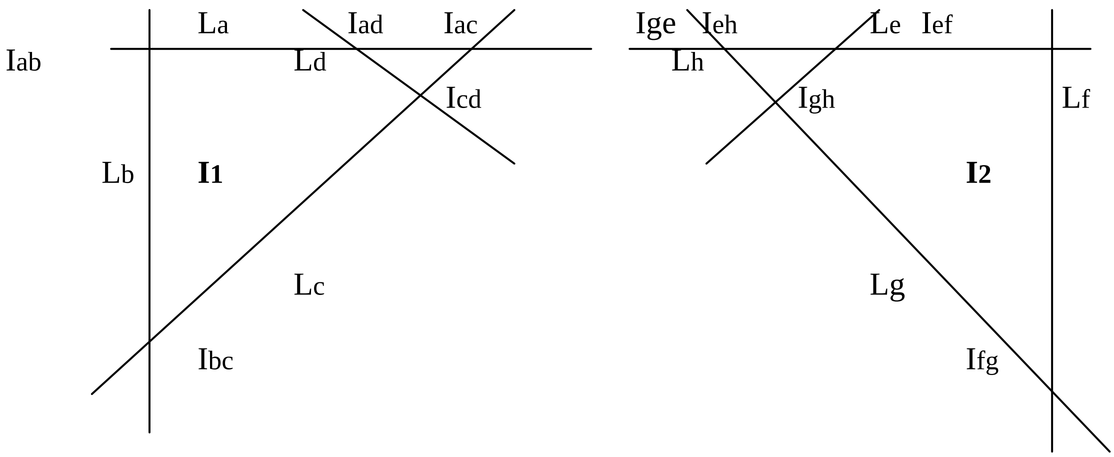


Figure 3.27 Emergent images

the original images. To get implicit image structure, we have to destroy the explicit image structure. The above figure shows emergent images after the destruction of original structure.

The representation of these two images now changes to

$$I_1 = \{4;[I_{ab},I_{bc},I_{cd},I_{ac},I_{ad}]\}$$

$$I_2 = \{4;[I_{ef},I_{fg},I_{gh},I_{ge},I_{eh}]\}$$

As we have destroyed the original structures of the two images, we define the above two representations as ordinary with parentheses whereas the original structures were closed and were defined with brackets.

The above two representations now give rise to following images which are emergent outcomes after destruction of the original structures. Obviously these are

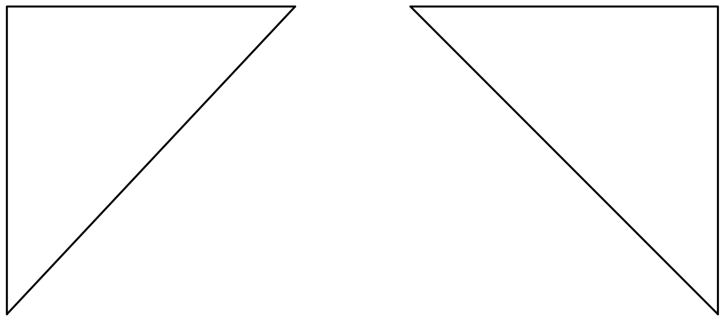


Figure 3.28 Emergent images after destruction of original images in Figure 3.26

two triangles which could not be discernible in the original images.

The corresponding equivalence of various segments is

$$L_a \Leftrightarrow L_e \wedge L_b \Leftrightarrow L_f \wedge L_c \Leftrightarrow L_g \wedge L_d \Leftrightarrow L_h$$

And the corresponding equivalence of various intersections are

$$I_{ab} \Leftrightarrow I_{ef} \wedge I_{bc} \Leftrightarrow I_{fg} \wedge I_{cd} \Leftrightarrow I_{gh} \wedge I_{ac} \Leftrightarrow I_{ge} \wedge I_{ad} \Leftrightarrow I_{eh}$$

To establish correspondence between two images

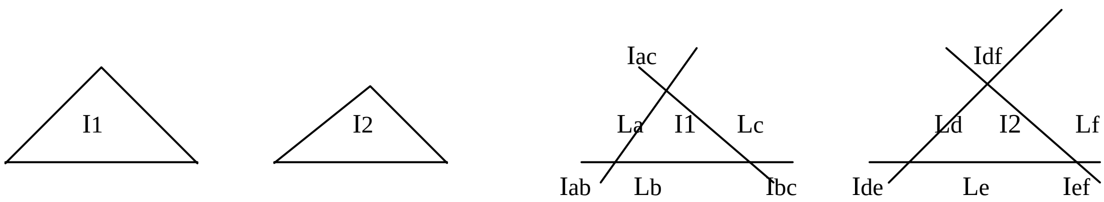


Figure 3.29 Two original images I_1 and I_2 on the left and in infinite maximal lines on the right

We have two original images I_1 and I_2. We would find corresponding infinite maximal lines and intersections between two images.

The representations of I_1 and I_2 are as follows:

$$I_1 = \{3;[I_{ab},I_{bc},I_{ac}]\}$$

$$I_2 = \{3;[I_{de},I_{ef},I_{df}]\}$$

Number of infinite maximal lines

Since number of infinite maximal lines are 3 for both I_1 and I_2, both images are equivalent to each other in this respect.

Number of intersections

Number of intersections in I_1 is 3. Also number of intersections in I_2 is 3. So our search for symmetry between I_1 and I_2 is still successful.

Geometric constraints of infinite maximal lines

I_1 has $I_{ab} <=> L_a \times L_b$

$I_{bc} <=> L_b \times L_c$

$I_{ac} <=> L_a \times L_c$

So $L_a \times L_b \times L_c$ is established since they intersect each other.

I_2 has $I_{de} <=> L_d \times L_e$

$I_{ef} <=> L_e \times L_f$

$I_{df} <=> L_d \times L_f$

So $L_d \times L_e \times L_f$ is established here as well as they intersect each other.

Ordinary group intersections in each image

$I_1 = \{3;[I_{ab},I_{bc},I_{ac}]\}$

$I_2 = \{3;[I_{de},I_{ef},I_{df}]\}$

Here also in ordinary group, number of intersections are same and 3.

Dimensional constraints of segments

Since the number of intersections in ordinary group in both I_1 and I_2 are same, we deduct

In I_1 $L_a \times L_b \wedge L_a \times L_c <=> (I_{ab}, I_{ac})$

$L_a \times L_b \wedge L_b \times L_c <=> (I_{ab}, I_{bc})$

$L_b \times L_c \wedge L_a \times L_c <=> (I_{bc}, I_{ac})$

The dimensional constraints in I_1 is

$d(I_{ab}, I_{ac}),d(I_{ab}, I_{bc}),d(I_{bc}, I_{ac})$

In I_2 $L_d \times L_e \wedge L_d \times L_f <=> (I_{de}, I_{df})$

$L_d \times L_e \wedge L_e \times L_f <=> (I_{de}, I_{ef})$

$L_e \times L_f \wedge L_d \times L_f <=> (I_{ef}, I_{df})$

The dimensional constraints in I_2 is

$d(I_{de}, I_{df}), d(I_{de}, I_{ef}), d(I_{ef}, I_{df})$

Dimension of segments between two images

We see whether dimension of segments between I_1 and I_2 are equal

For I_1 and I_2

$$d(I_{ab}, I_{ac}) = d(I_{de}, I_{df})$$

$$d(I_{ab}, I_{bc}) = d(I_{de}, I_{ef})$$

$$d(I_{bc}, I_{ac}) = d(I_{ef}, I_{df})$$

Corresponding infinite maximal lines from these

$$d(I_{ab}, I_{ac}) = d(I_{de}, I_{df}) => ((I_{ab}, I_{ac}) \in) L_a \leftrightarrow ((I_{de}, I_{df}) \in) L_d$$

$$d(I_{ab}, I_{bc}) = d(I_{de}, I_{ef}) => ((I_{ab}, I_{bc}) \in) L_b \leftrightarrow ((I_{de}, I_{ef}) \in) L_e$$

$$d(I_{bc}, I_{ac}) = d(I_{ef}, I_{df}) => ((I_{bc}, I_{ac}) \in) L_c \leftrightarrow ((I_{ef}, I_{df}) \in) L_f$$

symbol like $((I_{ab}, I_{ac}) \in) L_a$ means both intersections I_{ab} and I_{ac} lie in infinite maximal line L_a and so on.

Corresponding intersections

$$L_a \leftrightarrow L_d \wedge L_b \leftrightarrow L_e \wedge L_a \times L_b \wedge L_d \times L_e => I_{ab} \leftrightarrow I_{de}$$

$$L_a \leftrightarrow L_d \wedge L_c \leftrightarrow L_f \wedge L_a \times L_c \wedge L_d \times L_f => I_{ac} \leftrightarrow I_{df}$$

$$L_b \leftrightarrow L_e \wedge L_c \leftrightarrow L_f \wedge L_b \times L_c \wedge L_e \times L_f => I_{bc} \leftrightarrow I_{ef}$$

This establishes corresponding intersections between I_1 and I_2 are identical.

So from the above analysis, the symmetry between original images I_1 and I_2 are established.

To establish correspondence between emergent images

We take the concepts of correspondence of infinite maximal lines and intersections between two images as we discussed which are

$$L_a \leftrightarrow L_d,\ L_b \leftrightarrow L_e,\ L_c \leftrightarrow L_f$$

$$I_{ab} \leftrightarrow I_{de},\ I_{ac} \leftrightarrow I_{df},\ I_{bc} \leftrightarrow I_{ef}$$

We deduce three emergent segments as

$$I_{ab} \leftrightarrow I_{de} => (I_{ab}, I_{de})$$

$$I_{ac} \leftrightarrow I_{df} => (I_{ac}, I_{df})$$

$$I_{bc} \leftrightarrow I_{ef} => (I_{bc}, I_{ef})$$

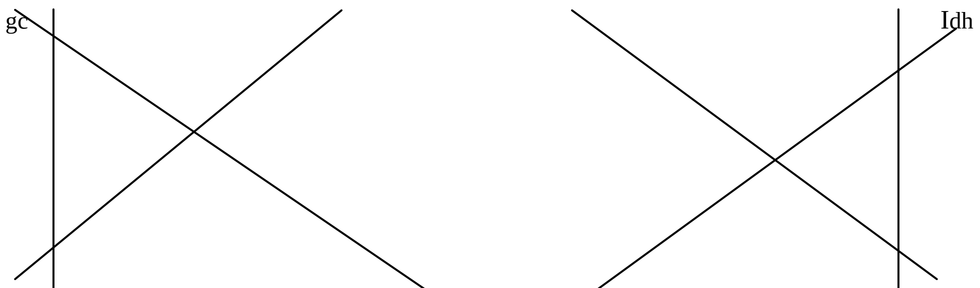
Midpoints and perpendicular bisectors of emergent segments

There are three midpoints of emergent segments

$$I(I_{ab}, I_{de})_m,\ I(I_{ac}, I_{df})_m \text{ and } I(I_{bc}, I_{ef})_m$$

Also there are three perpendicular bisectors of emergent segments

$$L_m(I_{ab}, I_{de}),\ L_m(I_{ac}, I_{df}) \text{ and } L_m(I_{bc}, I_{ef})$$



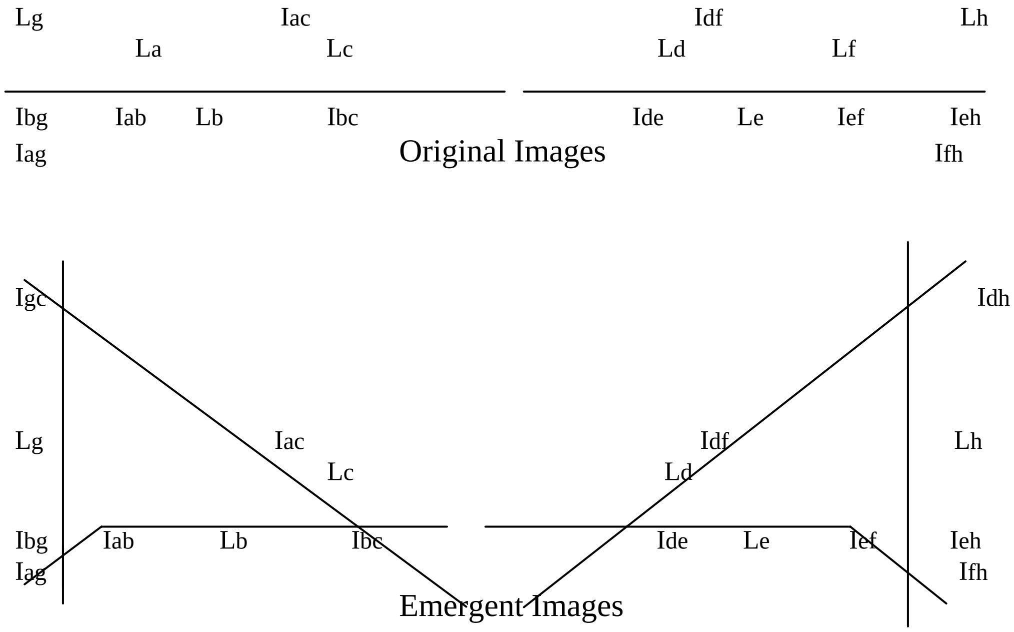


Figure 3.30 Original and emergent images

Original images and their corresponding emergent images are shown above checking coincidence, perpendicular bisectors and midpoints of emergent segments. Also we check the infinite maximal lines of new segments and geometric constraints.

3.6 Conclusion

Here we discussed how emergence index is useful in image database search. We also covered symbolic representation of images through infinite maximal lines, which are straight lines and constraints. Also we presented various mathematical tools that would be used throughout the dissertation. Then we described model of emergence by showing how destruction of original image leaves an unstructured image and then processing this unstructured image could give rise to a new structured image which is different from the original image considered.

Chapter 4 Calculation of Emergence Index

Now we move to calculate emergence index in the access of multimedia databases. For that we take an input image and study the emergence phenomenon of it. Also we study the emergence phenomenon of the images of the database. Both input image and images of database would give rise to more meanings because of emergence as we explained earlier. Based on the new meanings, wherever there would be a match between input image and images of database, we would pick that record up for selection.

In section 1.6.2, we defined emergence index as

$$EI = f(D, F, V, C, E).$$

We have to calculate these five variables to get emergence index for each image of the database. Also we have to calculate these five variables for input image as well.

In this chapter we deal with the calculation of index of the image of the database only. We would discuss about the input and finding the match while discussing the access of the database in next chapter.

4.1 Feature

First we concentrate on the calculation of features. To do that we take our earlier

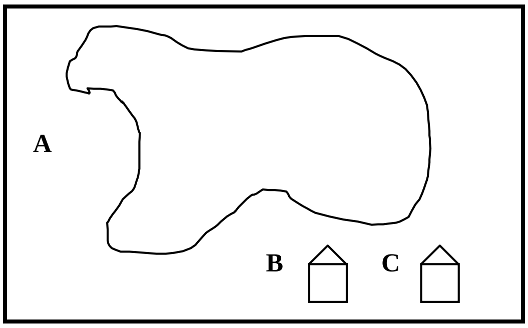


Figure 4.1 Image of a lake and two houses.

example in chapter 2 of a lake and two houses.

For any image, we have to consider two types of features: global and local.

Global

When we talk of global aspects of features, it means features of the entire image. Examples are area, perimeter or rectangles, triangles. In some searches, to consider the global features could be advantageous in that a symmetry with the input image could be obtained on the basis of global features only. But as is clearly the case, to consider global features could overlook the individual objects that constitute the image as a whole. In the kind of searches we propose, we would take into account the global features of the image of the database while considering in detail local features.

Local

Various objects that lie within an image constitute local features. In our example, there are three objects in the image, namely, a lake and two houses. Studying the features of these three objects would add to studying the features of the image globally.

Image analysis

One of the popular approaches so far followed is the text-based retrieval. In this method, each and every image of the database is manually and visually analyzed and the findings are written in textual form. Then the text against each image is written in a separate database in priory.

So when an input image is presented, this input is also manually analyzed like images of the database and whenever the parameters of the input matches parameters of the images of the database, corresponding image is selected.

This method has a problem. Since the volume of images generated is enormous, it is difficult to manually analyze each image.

To get rid of this problem of manually analyzing each and every image, content-based image retrieval techniques are being developed where images are analyzed based on color, texture, shape and spatial locations. This analysis and then segmentation of the images into various objects or sub-images are being done automatically. We mentioned image segmentation is beyond the scope of the thesis.

For our purpose, we would presume that images are already segmented. Then we would analyze the images and calculate the index. Based on these indices, the database would be searched for a given input image.

Analysis, calculation of index and search of the database would be done dynamically which means at the time of retrieval.

Calculation of features

While calculating the index, we have to consider all the objects within the image, their distances among them, angles they make with vertical and horizontal axes as the case may be, their perimeters, their relative position within the image. After calculating these values, we have to go further in our analysis to find hidden meanings, if any, lie in the image consisting of the objects and textures. As we pointed out earlier, these hidden meanings or emergence will also contribute to establishing symmetry between the input image and images of the database.

Two images could rarely be exactly same. Although same kind of image, the sizes may vary. Also the state of the objects in the image may vary. For example, a horse standing and not in motion would have different features than a horse in motion, particularly its leg parts.

Therefore, retrieval would be a generic search. In other words, we have to establish symmetry upto a certain point of similarity. When symmetry upto a certain point is established, we select that image or object within the image.

The design of appropriate image symmetry, distance functions, position functions, implicit meaning analysis are the key issues here.

There have been attempts to make approximate database search and indexing. Also there have been many approaches of image retrieval by contents based on features of objects within an image or relationship among objects. Our emphasis would be, as we pointed out, in establishing implicit meanings in the features or positions of objects within an image or textures, which could contribute to more precision and recall in search and symmetry between input and images of the database.

To define any image, we use coordinate system using x and y coordinate points.

As an image is considered to be composed of many objects, we have to define each object within the image in semantic form. Various objects could be

$$O_1, O_2, O_3, \ldots\ldots\ldots O_m$$

Where m is the number of objects in the image. Then each of these objects would have its representation in terms of feature components. Let the object O_1 be represented by features

$$F_1, F_2, F_3, \ldots\ldots\ldots, F_n$$

Where n is the number of features in object O_1. Similarly we can define features for objects $O_2, O_3, \ldots\ldots\ldots O_m$. Again F_1 could be expressed in terms of various coordinate points like

$$(x_1, y_1), (x_2, y_2), \ldots\ldots\ldots (x_k, y_k)$$

where k is the number of points in the coordinate system to define the feature F_1.

As the object O_1, as discussed in the last paragraph, could have many features $F_1, F_2, F_3, \ldots\ldots\ldots, F_n$, these features could be color, textures, shape, distances between various points and could be more depending upon various points and could be more depending upon cases. To represent these features in a proportionate way, we have to attach weights to each of these features. These weights would define their corresponding importance in the definition of the whole object under consideration.

Let $w_1, w_2, w_3, \ldots\ldots\ldots, w_n$ be the weights corresponding to features $F_1, F_2, F_3, \ldots\ldots\ldots, F_n$. So object O_1 of the image would be represented by

$$O_1 = f(w_1F_1 ; w_2F_2 ; w_3F_3 ; \ldots\ldots ; w_nF_n) \qquad (3)$$

Our next step is to calculate these features and their corresponding weights. This would give a good representation of the object O_1.

To maintain coherence, we would assign each feature a particular role althroughout. For instance, F_1 for any image of the database would speak about the color, may be F_2 for shape, F_3 for texture, F_4 for distances between various coordinate points within the object.

The color feature F_1 could have values like red $=1$, blue $= 2$, green $= 3$, yellow $= 4$, black $= 5$, violet $= 6$ and so on as we mentioned under variables in section 1.6.1. The weight corresponding to F_1 is w_1, which would have a value between 0 and 1 depending upon the part of the object that color occupies. For a red occupying 25% of the total object, the weight-feature factor would be

$$w_1F_1 = 0.25 * 1$$

Now we consider the shape, which is represented by F_2. If the shape of the object O_1 is circular with radius r then F_2 in terms of the shape of the image is

$$\pi r^2$$

and if this occupies 30% of the image then weight-feature factor would be

$$w_2F_2 = 0.3 * (\pi r^2)$$

Texture or background of an object in an image could be represented by color. This is possibly the best way to define texture. If the background is land mass covered with green grass, then F_3, the feature which stands for texture alongwith weight w_3, which in this case, say *50%* of the size of the total image could be expressed as

$$w_3 F_3 = 0.5 * 3$$

where *3* stands for green color.

If F_4 stands for distances between various points of the object O_1 and if $d_1, d_2, d_3, \ldots, d_n$ are the distances between various important points then F_4 and w_4 could be defined as

$$\text{Perimeter} = \sum_{1}^{n} di \text{ which is } 2\pi r \text{ in this case being a circle}$$

$$W_4 F_4 = 1.0 * (2\pi r)$$

Where $2\pi r$ is the perimeter of the circle under consideration. The value of w_4 in this case is *1* since it covers the total perimeter. If the object has some open or undefined perimeter, then $w4$ would be percentage of covered perimeter in relation to total perimeter. In such case, we have to extrapolate the undefined part of the perimeter to get complete perimeter.

Therefore, $O1$ could be represented as follows:

$$O_1 = f(w_1F_1; w_2F_2; w_3F_3; w_4 F_4)$$

$$= \{0.25 * 1; 0.3 * (\pi r^2); 0.5 * 3; 1.0 * (2\pi r)\}$$

This would be our semantic representation of object O_1. In the same way, we can define $O_2, O_3, \ldots\ldots Om$. The combination of the representations of $O_1, O_2, O_3, \ldots\ldots Om$ would define image I including all objects and textures in the semantic form. Also the distances between objects would also contribute to the definition of the contents of the image. The index structure of image I would , therefore, be

$$I = f(O_1, O_2, O_3, \ldots\ldots O_m) + \text{distances between objects.}$$

The RHS could be calculated as we have done for $O1$.

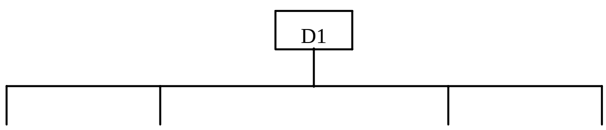
4.2 Domain

As we mentioned earlier, whenever we carry out a search of the image database we would do it dynamically in the sense that all processing and indexing of the images of the database would be done at the time of the search. No priory feature extraction would be performed.

The process of dynamically extracting features of the images for indexing purposes could be very time consuming and expensive particularly when the database contains huge number of images as could be the case quite often.

To reduce the time taken to perform the search and to use less computer resources, we have to conduct the search through the domains. In section 1.6.1 we have discussed domain to some extent. We need to classify records in the database according to some common criteria. In multimedia database where records based on image, audio, video, document would be there, some grouping of items has to be done. This grouping of items based on classification has to be done before any search is carried out.

Classification has to be done based on features of the record of the multimedia database. For images of the database, if the images are of geographical locations, then certain locations could be assigned to certain domains. That way when search involves a particular geographical location, we need to access the database of that particular location ignoring the rest. This would reduce the search time and also reduce the computer resources needed to carry out the search. Similarly when some particular person, say the President of the USA, is to be found out only in his appearances in that country, we have to search the domain where USA pictures are kept. If the President of the USA is to be found out in his appearances in some other country, then that particular country's domain is to be searched. We can also classify images based on certain time frame from a certain date upto a certain date. This also could reduce the search considerably. We have to use hierarchical indexing structure whereby at each level one could go for more limited search domain until we reach the target domain, which is reasonably small. Since volume of images generated is enormous, this classification of images into domains is very important. The hierarchical domain structure would be as given below.



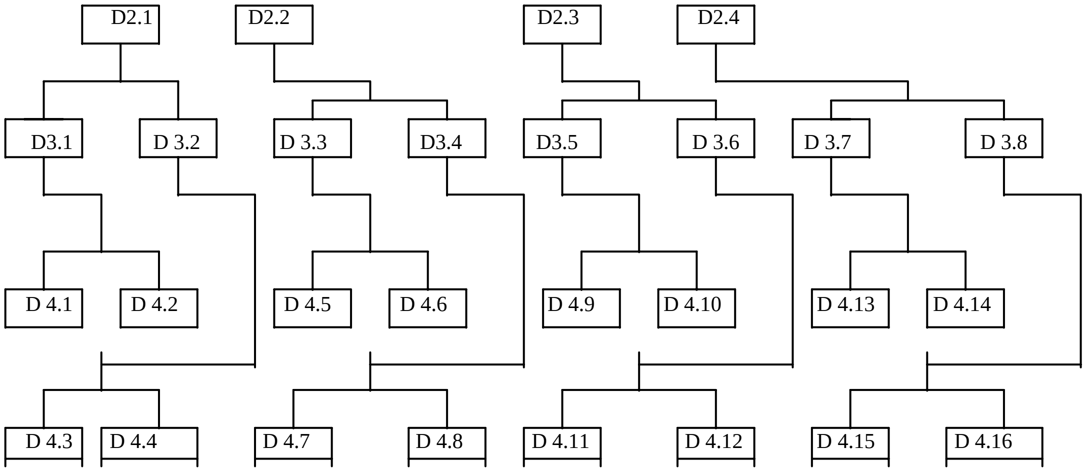


Figure 4.2 Hierarchical domain structure

Since we carry out the search dynamically, we have to have sequential search all throughout. For upper levels of domain structure, the search would be less than extensive as the items which depict various groups would be not too many. The more we move down to the lower levels, the search would be more and more extensive.

For example, if we want to find the pictures of the incumbent President of the USA taken in the city of New York in the state of New York during the period January 2000 till February 2000, then first level D_1 would be selected as it contains pictures of US President. In other words D1 is the President-USA database. $D_{2.1}$ could be selected next which could be the domain of pictures taken in the time frame mentioned above in the USA. $D_{3.1}$ could be selected as this contains the pictures of US President taken in New York State during the same period. $D_{4.1}$ would be selected next if it contains pictures of US President taken in New York City during that period. This way search efforts could be reduced to minimum using the domain hierarchy.

Accessing more than one databases

If a query comes in the form of text like ‘Find all the images of US President with spouses of current members of Senate’, then obviously it involves three databases. First we have to access the Senate Database which contains all the images of the Senators. We pick up one Senator at a time. Then access the Senator-Spouse database which would have Senators alongwith their spouses. We find the corresponding image of the spouse of the Senator of the Senator database. With this we go to access US President database to locate the image of US President with the spouse of the Senator. After this we go back to Senator database again, pick up the next Senator and repeat the process till all Senators are covered.

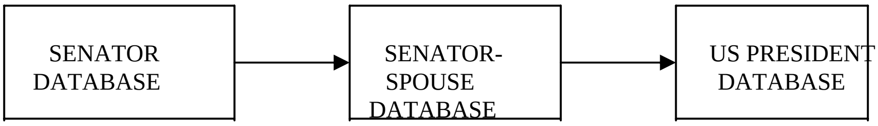


Figure 4.3 Accessing three databases

4.3 Example of Calculation of Emergence Index

Calculation of feature

Now we go back to our example of an image consisting of three objects, namely, a lake (O_1) and two houses (O_2 and O_3) and calculate the emergence index.

In our example of lake and two houses, we first define the features of the lake, which is a very irregular shape. As is very clear, to define an irregular shape is much more difficult than defining a regular shape.

We define some particular points at the boundary of the lake, which could be used for object matching. The feature of the lake would be a collection of few adjacent particular points. The feature represents a boundary segment and the shape is an ordered set of boundary segments.

If we follow a coordinate system to define our image then it would be like the following figure.

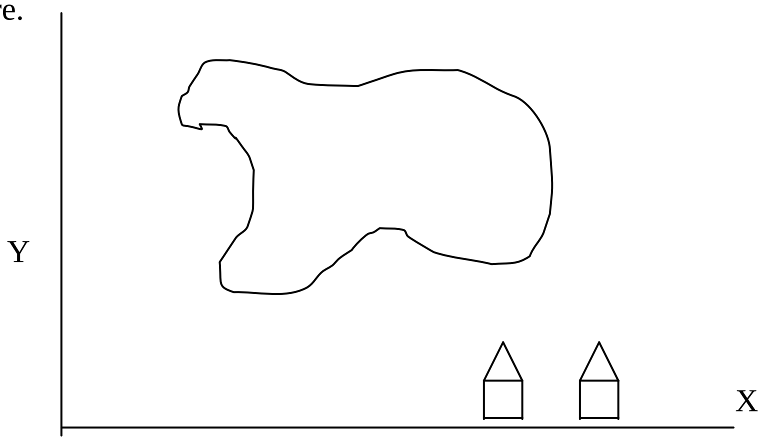


Figure 4.4 A lake and two houses in coordinate system

If the feature of the lake has n particular points, then using the conception of a pair of basic vector as discussed earlier the feature would be defined by the set

(x_1, y_1), (x_2, y_2),……. (x_n, y_n) and so on where (x_1, y_1) is the coordinate of the first particular point, the (x_2, y_2) is the coordinate of the second particular point and so on.

In chapter 2, we defined feature of the lake by *13* particular points.

To represent O_1, we use our earlier made expression

$$O_1 = f(w_1F_1; w_2F_2; w_3F_3 ;\ldots; w_nF_n).$$

Since lake is supposed to contain water, the color of the lake can be presumed to be blue and it occupies the whole area of the lake that is *100%*. Hence if F_1 is the standard color feature as we defined earlier then *100%* in decimal is *1.0* and we calculate

$$w1F1 = 1.0 * 2$$

Since O_1 would be composed of *13* coordinate points and if this object O_1 occupies *30%* of total image then the weight-feature factor would be

$$w_2F_2 = 0.3 * f\{(x_1, y_1), (x_2, y_2),\ldots\ldots, (x_{13}, y_{13})\}$$

where $(x_1, y_1), (x_2, y_2),\ldots\ldots (x_{13}, y_{13})$ represents the shape of the object O_1.

If the background is grass which is green and occupies *50%* of the image, then F_3 which represents texture and w_3 which represents area covered, in combined way would be

$$w_3F_3 = 0.5 * 3$$

where *3* stands for green color.

If $d_1, d_2, d_3 ,\ldots, d_n$ are the distances between various coordinate points of the object O_1, then

$$\text{Perimeter of } O_1 = \sum_{1}^{13} d_i$$

Since this covers the whole perimeter where there is no undefined or open part, the w_4 would be *1.0*. Hence weight-feature factor becomes in this case

$$w_4F_4 = 1.0 * \sum_{1}^{13} d_i$$

Therefore, O_1 could be represented in this case as

$$O_1=f(w_1F_1 ;w_2F_2; w_3F_3 ; w_4 F_4)$$

$$= [1.0 * 2; 0.3 * f\{(x_1,y_1), (x_2,y_2),\ldots\ldots, (x_{13},y_{13})\}; 0.5 * 3; 1.0 * \sum_{1}^{13} d_i]$$

The object O_2 which is a house can be defined as below:

If the house is composed of three different colors, say red, blue and yellow with covering *30%, 20%* and *50%* of the house respectively, then weight-feature factor would be

$$w_1F_1 = \{(0.3 * 1),(0.2 * 2), (0.5 * 4)\}$$

Since *O2* is composed of five coordinate points and if this object occupies 10% of the total image then

$$w_2F_2 = 0.1 * f\{(x_{14},y_{14}), (x_{15},y_{15}),\ldots, (x_{18},y_{18})\}$$

Where $f\{(x_{14},y_{14}), (x_{15},y_{15}),\ldots, (x_{18},y_{18})\}$ represents the shape of the object O_2.

The texture or background which comprises of *50%* of the image would be having weight-feature factor as

$$w_3F_3 = 0.5 * 3$$

where *3* stands for green color.

If $d_{14}, d_{15}, d_{16}, d_{17}, d_{18}$ are the distances between coordinate points of the object O_2, then

$$\text{Perimeter of } O_2 = \sum_{14}^{18} d_i$$

Here also as there is no unidentified or open part, the weight w_4 would be *1.0*. Hence

$$W_4F_4 = 1.0 * \sum_{14}^{18} d_i$$

So, O_2 could be represented by

$$O_2 = f(w_1F_1; w_2F_2; w_3F_3 ; w_4 F_4)$$

$$= [(0.3 * 1), (0.2 * 2), (0.5 * 4); 0.1 * f\{(x_{14}, y_{14}),(x_{15}, y_{15}),\ldots,(x_{18}, y_{18})\};$$

$$(0.5 * 3); (1.0 * \sum_{14}^{18} d_i)]$$

If the object O_3 which is another house is made up of same color as O_2 and the coordinate points are $(x_{19}, y_{19}),(x_{20}, y_{20}),(x_{21}, y_{21}),(x_{22}, y_{22}),(x_{23}, y_{23})$ then like O_2, it would be similarly represented as below

$$O_3 = f(w_1F_1; w_2F_2; w_3F_3 ; w_4 F_4)$$

$$= [(0.3 * 1), (0.2 * 2), (0.5 * 4); 0.1 * f\{(x_{19}, y_{19}),(x_{20}, y_{20}),\ldots,(x_{23}, y_{23})\};$$

$$(0.5 * 3); (1.0 * \sum_{19}^{23} d_i)]$$

Also the distances between objects as we calculated in section 2.2 are

Between O_2 and O_3 $d_1 = d(C_{19} \sim C_{14})$

Between O_1 and O_2 $d_2 = d(C_1 \sim C_{14})$

Between O_1 and O_3 $d_3 = d(C_1 \sim C_{19})$

Calculation of domain

In the hierarchical domain structure of 6.2, if our image lies in $D_{4.3}$, then clearly the domain path is $D_1/D_{2.1}/D_{3.2}/D_{4.3}$. As we discussed this would effectively reduce the searching time when the volume of data is enormous.

Variables

In the example image, we have three objects O_1, O_2, O_3. We already defined the variables for these three.

For O_1, they are $(x_1, y_1), (x_2, y_2),\ldots\ldots (x_{13}, y_{13})$.

For O_2, they are $(x_{14}, y_{14}),(x_{15}, y_{15}),\ldots,(x_{18}, y_{18})$. The six sides are L_a, L_b, L_c, L_d, L_e and L_f.

For O_3, they are $(x_{19}, y_{19}),(x_{20}, y_{20}),\ldots,(x_{23}, y_{23})$. The six sides are L_g, L_h, L_i, L_j, L_k and L_l.

Constraints

According to our definition of symbolic representation of shape in equation (2) in section 3.2

$$I = \{N; \text{constraints}\}$$

For O_1, $I = \{13; C_1, C_2, C_3,\ldots, C_{13}\}$

For O_2, the number of sides is 6 and the distance between (x_{15}, y_{15}) and (x_{17}, y_{17}) is same as the distance between (x_{16}, y_{16}) and (x_{18}, y_{18}). In other words,

$$d\{(x_{15}, y_{15}),(x_{17}, y_{17})\} = d\{(x_{16}, y_{16}),(x_{18}, y_{18})\}.$$

$$d(I_{acd},I_{df}) = d(I_{bce}, I_{ef})$$

Also distance between (x_{14}, y_{14}) and (x_{15}, y_{15}) is same as the distance between (x_{14}, y_{14}) and (x_{16}, y_{16}). In other words,

$$d\{(x_{14}, y_{14}),(x_{15}, y_{15})\} = d\{(x_{14}, y_{14}),(x_{16}, y_{16})\}.$$

$$d(I_{ab}, I_{acd}) = d(I_{ab}, I_{bce})$$

And distance between (x_{15}, y_{15}) and (x_{16}, y_{16}) is same as the distance between (x_{17}, y_{17}) and (x_{18}, y_{18}). In other words,

$$d\{(x_{15}, y_{15}),(x_{16}, y_{16})\} = d\{(x_{17}, y_{17}),(x_{18}, y_{18})\}.$$

$d(I_{acd}, I_{bce}) = d(I_{df}, I_{ef})$

Then $L_d // L_e, L_c // L_f$

For O_3, the number of sides is 6 and the distance between (x_{20}, y_{20}) and (x_{22}, y_{22}) is same as the distance between (x_{21}, y_{21}) and (x_{23}, y_{23}). In other words,

$d\{(x_{20}, y_{20}),(x_{22}, y_{22})\} = d\{(x_{21}, y_{21}),(x_{23}, y_{23})\}$.

$d(I_{gij}, I_{jl}) = d(I_{hik}\ I_{kl})$

Also distance between (x_{19}, y_{19}) and (x_{20}, y_{20}) is same as the distance between (x_{19}, y_{19}) and (x_{21}, y_{21}). In other words,

$d\{(x_{19}, y_{19}),(x_{20}, y_{20})\} = d\{(x_{19}, y_{19}),(x_{21}, y_{21})\}$.

$d(I_{gh}, I_{gij}) = d(I_{gh}, I_{hik})$

And distance between (x_{20}, y_{20}) and (x_{21}, y_{21}) is same as the distance between (x_{22}, y_{22}) and (x_{23}, y_{23}). In other words,

$d\{(x_{20}, y_{20}),(x_{21}, y_{21})\} = d\{(x_{22}, y_{22}),(x_{23}, y_{23})\}$.

$d(I_{gij}, I_{hik}) = d(I_{jl}, I_{kl})$

Then $L_j // L_k, L_i // L_l$

Emergence

In section 2.2, we have discussed about distances among O_1, O_2 and O_3 and also that individually O_1, O_2, O_3 have meanings of a lake and two houses. But when we think of the implicit meaning of the whole image, it becomes a residential area which is the emergence output of the image and whenever an input would call for a residential area, this image could be selected although the input may not have any apparent match with O_1, O_2 and O_3 independently.

4.4 Conclusion

In this chapter, we calculated emergence index in the access of multimedia databases. For that we took an input image and studied the emergence phenomenon of it. Also we studied the emergence phenomenon of the images of the database. Both input image and images of database would give rise to more meanings because of emergence. Based on the new meanings, wherever there would be a match between input image and images of database, we can pick that record up for selection.

Chapter 5 Accessing Multimedia Databases With Emergence Index

In this chapter we discuss the methodology we follow in accessing multimedia databases. First we establish symmetry between input image and images of database. Then accessing database using the emergence index calculated would be discussed.

5.1 Symmetry Between Input and Images of Regular Shape

5.1.1 Calculation of Index of Input

Let us suppose our input image is of the kind shown below.

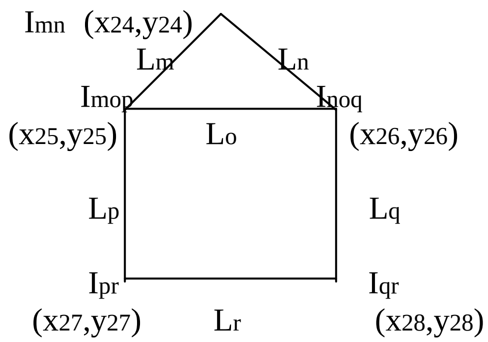


Figure 5.1 Input image of a house

This is the image of a house we have discussed in chapter 2. Our example image of the database remains the same, a lake and two houses as shown in section 4.1.

In the input image, coordinates are obviously

$(x_{24}, y_{24}), (x_{25}, y_{25}),\ldots\ldots, (x_{28}, y_{28})$.

We notice if L_m, L_n, L_o, L_p, L_q and L_r are the six lines constituting the house, then

$L_p // L_q$, $L_o // L_r$.

Also the number of sides is *6*.

The distance between (x_{25}, y_{25}) and (x_{27}, y_{27}) is same as the distance between (x_{26}, y_{26}) and (x_{28}, y_{28}).

$d\{(x_{25}, y_{25}), (x_{27}, y_{27})\} = d\{(x_{26}, y_{26}), (x_{28}, y_{28})\}$

$d(I_{mop}, I_{pr}) = d(I_{noq}, I_{qr})$

Again the distance between (x_{24}, y_{24}) and (x_{25}, y_{25}) is same as the distance between (x_{24}, y_{24}) and (x_{26}, y_{26})

$d\{(x_{24},, y_{24}), (x_{25}, y_{25})\} = d\{(x_{24}, y_{24}), (x_{26},, y_{26})\}$

$d(I_{mn}, I_{mop}) = d(I_{mn}, I_{noq})$

Also the distance between (x_{25}, y_{25}) and (x_{26}, y_{26}) is same as the distance between (x_{27}, y_{27}) and (x_{28}, y_{28})

$$d\{(x_{25}, y_{25}), (x_{26}, y_{26})\} = d\{(x_{27}, y_{27}), (x_{28}, y_{28})\}$$

$$d(I_{mop}, I_{noq}) = d(I_{pr}, I_{qr})$$

5.1.2 Correspondence Between Input and Image

Ordinary group intersections in each image

Now if we compare the representation of this input image with that of one of the two houses we considered in our index calculation in section 4.3, then we find

Input = $\{6;[I_{mn}, I_{mop}, I_{noq}, I_{pr}, I_{qr}]\}$

$O_2 = \{6;[I_{ab}, I_{acd}, I_{bce}, I_{df}, I_{ef}]\}$

Number of infinite maximal lines

We notice number of infinite maximal lines in each case is *6*.

Corresponding equivalence

According to the corresponding equivalence as discussed in 3.5, the corresponding equivalence of various segments is

$$L_a \Leftrightarrow L_m \wedge L_b \Leftrightarrow L_n \wedge L_c \Leftrightarrow L_o \wedge L_d \Leftrightarrow L_p \wedge L_e \Leftrightarrow L_q \wedge L_f \Leftrightarrow L_r$$

And the corresponding equivalence of various intersections are

$$I_{ab} \Leftrightarrow I_{mn} \wedge I_{acd} \Leftrightarrow L_{mop} \wedge I_{bce} \Leftrightarrow I_{noq} \wedge I_{df} \Leftrightarrow I_{pr} \wedge I_{ef} \Leftrightarrow I_{qr}$$

Number of intersections

The number of intersections in each case is *5*.

Geometric constraints of infinite maximal lines

Input has $I_{mn} \Leftrightarrow L_m \times L_n$

$I_{mo} \Leftrightarrow L_m \times L_o$

$I_{no} \Leftrightarrow L_n \times L_o$

So $L_m \times L_n \times L_o$ is established since they intersect each other.

O_2 has $I_{ab} \Leftrightarrow L_a \times L_b$

$I_{bc} \Leftrightarrow L_b \times L_c$

$I_{ac} \Leftrightarrow L_a \times L_c$

So $L_a \times L_b \times L_c$ is established since they intersect each other.

Dimensional constraints of segments

Since the number of intersections in input and O_2 is same, we make the following deductions:

In input $L_m \times L_o \wedge L_m \times L_n \Leftrightarrow (I_{mo}, I_{mn})$

$L_n \times L_o \wedge L_n \times L_m \Leftrightarrow (I_{no}, I_{mn})$

$L_o \times L_m \wedge L_o \times L_n \Leftrightarrow (I_{om}, I_{on})$

$L_p \times L_o \wedge L_p \times L_r \Leftrightarrow (I_{po}, I_{pr})$

$L_q \times L_o \wedge L_q \times L_r \Leftrightarrow (I_{qo}, I_{qr})$

$L_p \times L_r \wedge L_q \times L_r \Leftrightarrow (I_{pr}, I_{qr})$

So dimensional constraints in input is

$d(I_{mo}, I_{mn}), d(I_{no}, I_{mn}), d(I_{om}, I_{on}), d(I_{po}, I_{pr}), d(I_{qo}, I_{qr}), d(I_{pr}, I_{qr})$

In O_2 $L_a \times L_c \wedge L_a \times L_b \Leftrightarrow (I_{ac}, I_{ab})$

$L_b \times L_c \wedge L_b \times L_a \Leftrightarrow (I_{bc}, I_{ab})$

$L_c \times L_a \wedge L_c \times L_b \Leftrightarrow (I_{ca}, I_{cb})$

$L_d \times L_c \wedge L_d \times L_f \Leftrightarrow (I_{dc}, I_{df})$

$L_e \times L_c \wedge L_e \times L_f \Leftrightarrow (I_{ec}, I_{ef})$

$L_d \times L_f \wedge L_e \times L_f \Leftrightarrow (I_{df}, I_{ef})$

So dimensional constraints of O_2 is

$d(I_{ac}, I_{ab}), d(I_{bc}, I_{ab}), d(I_{ca}, I_{cb}), d(I_{dc}, I_{df}), d(I_{ec}, I_{ef}), d(I_{df}, I_{ef})$

Corresponding intersections

$L_m \leftrightarrow L_a \wedge L_n \leftrightarrow L_b \wedge L_m \times L_n \wedge L_a \times L_b => I_{mn} \leftrightarrow I_{ab}$

$L_m \leftrightarrow L_a \wedge L_p \leftrightarrow L_d \wedge L_m \times L_p \wedge L_a \times L_d => I_{mp} \leftrightarrow I_{ad}$

$L_n \leftrightarrow L_b \wedge L_q \leftrightarrow L_e \wedge L_n \times L_q \wedge L_b \times L_e => I_{nq} \leftrightarrow I_{be}$

$L_p \leftrightarrow L_d \wedge L_r \leftrightarrow L_f \wedge L_p \times L_r \wedge L_d \times L_f => I_{pr} \leftrightarrow I_{df}$

$L_q \leftrightarrow L_e \wedge L_r \leftrightarrow L_f \wedge L_q \times L_r \wedge L_e \times L_f => I_{qr} \leftrightarrow I_{ef}$

So we see the symmetry between the input and image O_2 although they are not of the same size.

5.1.3 Calculation of Index of Input As Triangle

If the input comes in the form of a triangle below, then we have

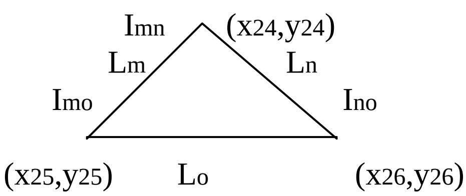


Figure 5.2 Input image in the form of a triangle

The coordinates are (x_{24}, y_{24}),(x_{25}, y_{25}), (x_{26}, y_{26}). The sides are L_m, L_n, L_o.

Also the number of sides is *3*.

The distance between (x_{24}, y_{24}) and (x_{25}, y_{25}) is same as the distance between (x_{24}, y_{24}) and (x_{26}, y_{26}).

$$d\{(x_{24}, y_{24}), (x_{25}, y_{25})\} = d\{(x_{24}, y_{24}), (x_{26}, y_{26})\}$$

$$d(I_{mn}, I_{mo}) = d(I_{mn}, I_{no})$$

5.1.4 Correspondence Between the Input (triangle) and Image

Ordinary group intersections in each image

Here also we compare the representation of the input with that of one of the two houses.

Input = $\{3; [I_{mn}, I_{mo}, I_{no}]\}$

If we consider only the upper part of object O_2, then

$$O_2 = \{3; [I_{ab}, I_{ac}, I_{bc}]\}$$

Number of infinite maximal lines

We notice number of infinite maximal lines in each case is *3*.

Corresponding equivalence

The corresponding equivalence of various segments are

$$L_a \Leftrightarrow L_m \wedge L_b \Leftrightarrow L_n \wedge L_c \Leftrightarrow L_o$$

Also the corresponding equivalence of various intersections are

$$I_{ab} \Leftrightarrow I_{mn} \wedge I_{ac} \Leftrightarrow L_{mo} \wedge I_{bc} \Leftrightarrow I_{no}$$

Number of intersections

The number of intersections in each case is *3*.

Geometric constraints of infinite maximal lines

Input has $I_{mn} \Leftrightarrow L_m \times L_n$

$I_{mo} \Leftrightarrow L_m \times L_o$

$I_{no} \Leftrightarrow L_n \times L_o$

So $L_m \times L_n \times L_o$ is established since they intersect each other.

O_2 has $I_{ab} \Leftrightarrow L_a \times L_b$

$I_{bc} \Leftrightarrow L_b \times L_c$

$I_{ac} \Leftrightarrow L_a \times L_c$

So $L_a \times L_b \times L_c$ is established since they intersect each other.

Dimensional constraints of segments

Since the number of intersections in input and O_2 is same, we deduct

In input $\quad L_m \times L_o \wedge L_m \times L_n \Leftrightarrow (I_{mo}, I_{mn})$

$L_n \times L_o \wedge L_n \times L_m \Leftrightarrow (I_{no}, I_{mn})$

$L_o \times L_m \wedge L_o \times L_n \Leftrightarrow (I_{om}, I_{on})$

So dimensional constraints in input is

$d(I_{mo}, I_{mn}), d(I_{no}, I_{mn}), d(I_{om}, I_{on})$

In O_2 $\quad L_a \times L_c \wedge L_a \times L_b \Leftrightarrow (I_{ac}, I_{ab})$

$L_b \times L_c \wedge L_b \times L_a \Leftrightarrow (I_{bc}, I_{ab})$

$L_c \times L_a \wedge L_c \times L_b \Leftrightarrow (I_{ca}, I_{cb})$

So dimensional constraints of O_2 is

$d(I_{ac}, I_{ab}), d(I_{bc}, I_{ab}), d(I_{ca}, I_{cb})$

Corresponding intersections

$L_m \leftrightarrow L_a \wedge L_n \leftrightarrow L_b \wedge L_m \times L_n \wedge L_a \times L_b \Rightarrow I_{mn} \leftrightarrow I_{ab}$

$L_m \leftrightarrow L_a \wedge L_o \leftrightarrow L_c \wedge L_m \times L_o \wedge L_a \times L_c \Rightarrow I_{mo} \leftrightarrow I_{ac}$

$L_n \leftrightarrow L_b \wedge L_o \leftrightarrow L_c \wedge L_n \times L_o \wedge L_b \times L_c \Rightarrow I_{no} \leftrightarrow I_{bc}$

Therefore, the input in the form of a triangle and upper part of image O_2 is symmetrical. Although the image O_2 is that of the image of a house, still it would be selected in the search as a result of emergence.

5.1.5 Calculation of Index of Input As Rectangle

If the input comes in the form of a rectangle below, then we have

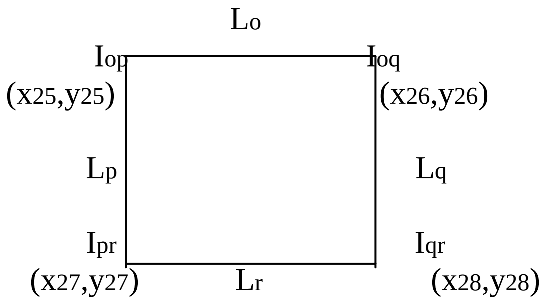


Figure 5.3 Input image in the form of a rectangle

The coordinates which define the rectangle are (x_{25}, y_{25}), (x_{26}, y_{26}), (x_{27}, y_{27}) and (x_{28}, x_{28}). The sides are L_o, L_p, L_q, L_r and $L_o \mathbin{/\!/} L_r$, $L_p \mathbin{/\!/} L_q$.

Also the number of sides is 4.

The distance between (x_{25}, y_{25}) and (x_{26}, y_{26}) is same as the distance between (x_{27}, y_{27}) and (x_{28}, y_{28}).

$$d\{(x_{25}, y_{25}), (x_{26}, y_{26})\} = d\{(x_{27}, y_{27}), (x_{28}, y_{28})\}$$

$$d(I_{op}, I_{oq}) = d(I_{pr}, I_{qr})$$

Also the distance between (x_{25}, y_{25}) and (x_{27}, y_{27}) is same as the distance between (x_{26}, y_{26}) and (x_{28}, y_{28}).

$$d\{(x_{25}, y_{25}), (x_{27}, y_{27})\} = d\{(x_{26}, y_{26}), (x_{28}, y_{28})\}$$

$$d(I_{op}, Ipr) = d(I_{oq}, I_{qr})$$

5.1.6 Correspondence Between the Input (rectangle) and Image

Ordinary group intersections in each image

Here also we compare the representation of the input with that of one of the two houses.

Input = $\{4;[I_{op}, I_{oq}, I_{qr}, I_{pr}]\}$

If we consider only the lower part of object O_2, then

$$O_2 = \{4;[I_{cd}, I_{ce}, I_{ef}, I_{df}]\}$$

Number of infinite maximal lines

We notice number of infinite maximal lines in each case is *4*.

Corresponding equivalence

The corresponding equivalence of various segments is

$$L_o \Leftrightarrow L_c \wedge L_p \Leftrightarrow L_d \wedge L_q \Leftrightarrow L_e \wedge L_r \Leftrightarrow L_f$$

Also the corresponding equivalence of various intersections are

$$I_{op} \Leftrightarrow I_{cd} \wedge I_{oq} \Leftrightarrow L_{ce} \wedge I_{qr} \Leftrightarrow I_{ef} \wedge I_{pr} \Leftrightarrow I_{df}$$

Number of intersections

The number of intersections in each case is *4*.

Geometric constraints of infinite maximal lines

Input has $I_{op} \Leftrightarrow L_o \times L_p$

$I_{oq} \Leftrightarrow L_o \times L_q$

$I_{qr} \Leftrightarrow L_q \times L_r$

$I_{pr} \Leftrightarrow L_p \times L_r$

O_2 has $I_{cd} \Leftrightarrow L_c \times L_d$

$I_{ce} \Leftrightarrow L_c \times L_e$

$I_{ef} \Leftrightarrow L_e \times L_f$

$$I_{df} \Leftrightarrow L_d \times L_f$$

Dimensional constraints of segments

Since the number of intersections in input and O_2 is same, we deduct

In input $L_o \times L_p \wedge L_o \times L_q \Leftrightarrow (I_{op}, I_{oq})$

$L_o \times L_q \wedge L_q \times L_r \Leftrightarrow (I_{oq}, I_{qr})$

$L_q \times L_r \wedge L_p \times L_r \Leftrightarrow (I_{qr}, I_{pr})$

$L_p \times L_r \wedge L_o \times L_p \Leftrightarrow (I_{pr}, I_{op})$

So dimensional constraints in input are

$$d(I_{op}, I_{oq}), d(I_{oq}, I_{qr}), d(I_{qr}, I_{pr}), d(I_{pr}, I_{op})$$

In O_2 $L_c \times L_d \wedge L_c \times L_e \Leftrightarrow (I_{cd}, I_{ce})$

$L_c \times L_e \wedge L_e \times L_f \Leftrightarrow (I_{ce}, I_{ef})$

$L_e \times L_f \wedge L_d \times L_f \Leftrightarrow (I_{ef}, I_{df})$

$L_d \times L_f \wedge L_c \times L_d \Leftrightarrow (I_{df}, I_{cd})$

So dimensional constraints in input is

$$d(I_{cd}, I_{ce}), d(I_{ce}, I_{ef}), d(I_{ef}, I_{df}), d(I_{df}, I_{cd})$$

Corresponding intersections

$$L_o \leftrightarrow L_c \wedge L_p \leftrightarrow L_d \wedge L_o \times L_p \wedge L_c \times L_d => I_{op} \leftrightarrow I_{cd}$$

$$L_o \leftrightarrow L_c \wedge L_q \leftrightarrow L_e \wedge L_o \times L_q \wedge L_c \times L_e => I_{oq} \leftrightarrow I_{ce}$$

$$L_q \leftrightarrow L_e \wedge L_r \leftrightarrow L_f \wedge L_q \times L_r \wedge L_e \times L_f => I_{qr} \leftrightarrow I_{ef}$$

$$L_p \leftrightarrow L_d \wedge L_r \leftrightarrow L_f \wedge L_p \times L_r \wedge L_d \times L_f => I_{pr} \leftrightarrow I_{df}$$

Therefore, the input in the form of a rectangle and lower part of image O_2 is symmetrical. Although the image O_2 is that of the image of a house, still it would be selected in the search as a result of emergence.

5.2 Symmetry Between Input and Images of Irregular Shape

5.2.1 Calculation of Index of Input As Irregular Shape

If the input comes in the form of an irregular shape then we define it in terms of coordinate points. First we define it with the help of *15* straight lines which can express the irregular shape. Then t he intersections of these straight lines into *15*

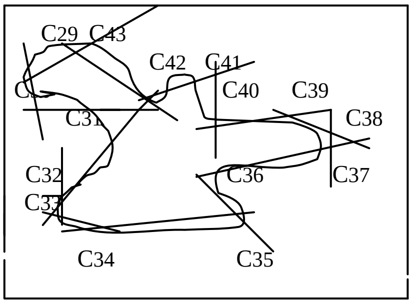


Figure 5.4 Input image in the form of an irregular shape

intersections define the *15* coordinate points, namely, C_{29}, C_{30}, ……., C_{43}. So the image according to our definition of symbolic representation of shape is

$I = \{N;$ constraints$\}$

$= \{15;\ C_{29}, C_{30}, \ldots\ldots, C_{43}\}$

5.2.2 Correspondence Between Input and Image

Input $= \{15; [\ C_{29}, C_{30}, \ldots\ldots, C_{43}]\}$

$O_1 = \{13; [\ C_1, C_2, \ldots\ldots, C_{13}]\}$

Number of intersections

The number of intersections in input is *15* whereas in O_1 is *13*. Hence they do not match.

Correspondence

If we start considering input from C_{29} and go down and start considering O_1 from C_2 and also go down, then

$C_{29} \approx (x_{29}, y_{29}) \Leftrightarrow C_2 \approx (x_2, y_2)$

$C_{30} \approx (x_{30}, y_{30})$ no correspondence

$C_{31} \approx (x_{31}, y_{31}) \Leftrightarrow C_3 \approx (x_3, y_3)$

$C_{32} \approx (x_{32}, y_{32})$ no correspondence

$C_{33} \approx (x_{33}, y_{33}) \Leftrightarrow C_4 \approx (x_4, y_4)$

$C_{34} \approx (x_{34}, y_{34}) \Leftrightarrow C_6 \approx (x_6, y_6)$

$C_{35} \approx (x_{35}, y_{35})$ no correspondence

$C_{36} \approx (x_{36}, y_{36}) \Leftrightarrow C_8 \approx (x_8, y_8)$

$C_{37} \approx (x_{37}, y_{37}) \Leftrightarrow C_{11} \approx (x_{11}, y_{11})$

$C_{38} \approx (x_{38}, y_{38})$ no correspondence

$C_{39} \approx (x_{39}, y_{39})$ no correspondence

$C_{40} \approx (x_{40}, y_{40})$ no correspondence

$C_{41} \approx (x_{41}, y_{41})$ no correspondence

$C_{42} \approx (x_{42}, y_{42})$ no correspondence

$C_{43} \approx (x_{43}, y_{43})$ no correspondence

This shows only 6 coordinate points of input match with *6* coordinate points of O_1. Since there are *15* coordinate points in input, we can ascribe equal weight to each individual matching. Since the total match is *100%*, each individual match would have weight *100/15 = 6.67*. We assume symmetry is established when at least *90%* match occurs. Since there are *6* matches, the total percentage match would be *6.67 * 6 = 40.02%*. Since this is far below *90%*, we reject symmetry. Hence correspondence could not be established and they are not identical.

Emergence outcome

But if the color of both input and O_1 is same and blue, then it would indicate both as lakes whose structure, behavior and function as described in chapter 2 would be same. Hence if we consider the hidden or emergent meanings of the two objects, they are same and correspondence is established. So object O_1 would be selected although there is no apparent symmetry between them. But if emergence is not considered then since there is no symmetry established, object *O1* would not be selected.

5.2.3 Calculation of Index of Input As Irregular Shape

As in 5.2.1, if the input again comes in the form of an irregular shape then we

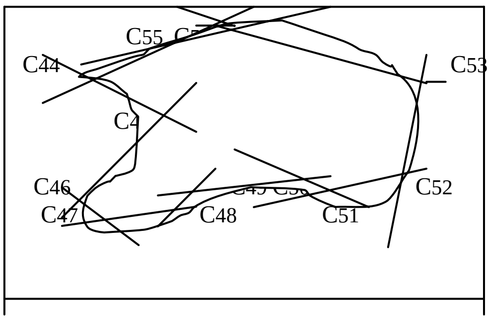


Figure 5.5 Input image in the form of an irregular shape

define it this time with *12* intersections of straight lines, namely, C_{44}, C_{45}, ……., C_{55}. So the representation in this case is

$$I = \{12; [C_{44}, C_{45}, \ldots\ldots, C_{55}]\}$$

5.2.4 Correspondence Between Input and Image

Input $= \{12; [\ C_{44}, C_{45}, \ldots\ldots, C_{55}]\}$

$O_1 = \{13; [\ C_1, C_2, \ldots\ldots, C_{13}]\}$

Number of intersections

The number of intersections in input is *12* whereas in O_1 is *13*. Hence they do not match.

Correspondence

If we start considering input from C_{44} and go down and start considering O_1 from C_1 and also go down, then

$C_{44} \approx (x_{44}, y_{44}) \Leftrightarrow C_2 \approx (x_2, y_2)$

$C_{45} \approx (x_{45}, y_{45}) \Leftrightarrow C_3 \approx (x_3, y_3)$

$C_{46} \approx (x_{46}, y_{46}) \Leftrightarrow C_4 \approx (x_4, y_4)$

$C_{47} \approx (x_{47}, y_{47}) \Leftrightarrow C_5 \approx (x_5, y_5)$

$C_{48} \approx (x_{48}, y_{48}) \Leftrightarrow C_6 \approx (x_6, y_6)$

$C_{49} \approx (x_{49}, y_{49}) \Leftrightarrow C_7 \approx (x_7, y_7)$

$C_{50} \approx (x_{50}, y_{50}) \Leftrightarrow C_8 \approx (x_8, y_8)$

$C_{51} \approx (x_{51}, y_{51}) \Leftrightarrow C_9 \approx (x_9, y_9)$

$C_{52} \approx (x_{52}, y_{52}) \Leftrightarrow C_{10} \approx (x_{10}, y_{10})$

$C_{53} \approx (x_{53}, y_{53}) \Leftrightarrow C_{11} \approx (x_{11}, y_{11})$

$C_{54} \approx (x_{54}, y_{54}) \Leftrightarrow C_{12} \approx (x_{12}, y_{12})$

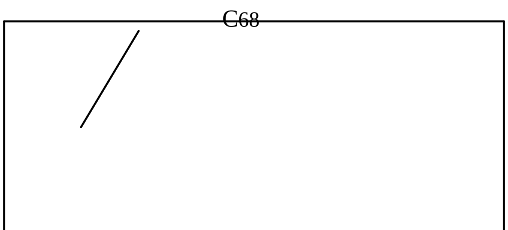
$C_{55} \approx (x_{55}, y_{55}) \Leftrightarrow C_{13} \approx (x_{13}, y_{13})$

This shows only *12* coordinate points of input match with *12* coordinate points of O_1. In this case, each individual match would have weight $100/13 = 7.69$. If *90%* match indicates symmetry, then total percentage match for *12* would be $7.69 * 12 = 92.28\%$.

Since this is more than *90%*, we accept symmetry and the input is identical to image O_1.

5.2.5 Calculation of Index of Input Smaller in Size

We assume input here also comes in the form of an irregular shape with *13*



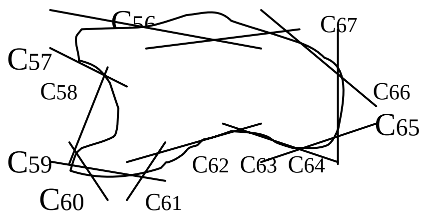


Figure 5.6 Input image in the form of an irregular shape, which is smaller than O_1

intersections of straight lines which define the shape. The shape here is smaller than O_1. Let these intersections be $C_{56}, C_{57}, \ldots\ldots, C_{68}$. So the representation in this case is

$$I = \{13;\ [C_{56}, C_{57}, \ldots\ldots, C_{68}]\}.$$

5.2.6 Correspondence Between Input and Image

Input $= \{13; [C_{56}, C_{57}, \ldots\ldots, C_{68}]\}$

$O_1 = \{13; [C_1, C_2, \ldots\ldots, C_{13}]\}$

Number of intersections

The number of intersections in input is 13 and in O_1 also it is 13. Hence they match.

Correspondence

Since the size of input is smaller than O_1, we cannot establish correspondence straightaway.

Let $\theta_1, \theta_2, \theta_3, \ldots\ldots, \theta_{13}$ be the angles at the points of intersection in O_1 and let $\theta_{56}, \theta_{57}, \theta_{58}, \ldots\ldots, \theta_{68}$ be the angles at the points of intersection in input.

Now if $\theta_1 \Leftrightarrow \theta_{56}$

$\theta_2 \Leftrightarrow \theta_{57}$

$\theta_3 \Leftrightarrow \theta_{58}$

$\theta_4 \Leftrightarrow \theta_{59}$

$\theta_5 \Leftrightarrow \theta_{60}$

$\theta_6 \Leftrightarrow \theta_{61}$

$\theta_7 \Leftrightarrow \theta_{62}$

$\theta_8 \Leftrightarrow \theta_{63}$

$\theta_9 \Leftrightarrow \theta_{64}$

$\theta_{10} \Leftrightarrow \theta_{65}$

$\theta_{11} \Leftrightarrow \theta_{66}$

$\theta_{12} \Leftrightarrow \theta_{67}$

$\theta_{13} \Leftrightarrow \theta_{68}$

then correspondence between input and O_1 is established although input is smaller in size compared to O_1.

So if the number of intersections are same and angles at those intersections correspond to each other in input and O_1, then symmetry is established although input and O_1 are not of same size.

5.2.7 Calculation of Symmetry Where Image Rotated

If the input is exactly same as in 5.2.5 but image is rotated from the vertical axis clockwise, then we find the symmetry in the following way.

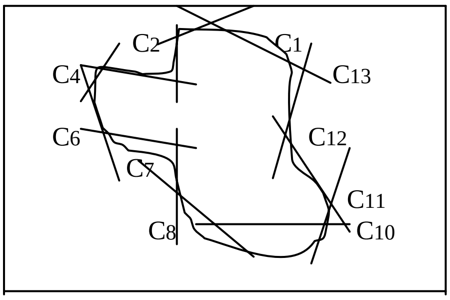


Figure 5.7 Image of the database when rotated

The correspondence between input and image would be same as in section 5.2.6 as this image has *13* intersections as C_1, C_2, C_3,…., C_{13} and angles at the points of intersections are θ_1, θ_2, θ_3, ……..., θ_{13}. Input has *13* intersections C_{56}, C_{57}, ……..., C_{68} and angles at the point of intersections are θ_{56}, θ_{57}, θ_{58},……..., θ_{68}.

Let D_1, D_2, D_3,…., D_{13} be the infinite maximal lines defining the image and θ_1', θ_2', θ_3', ……..., θ_{13}' are the angles D_1, D_2, D_3,…., D_{13} make with the vertical axis respectively. Let D_{56}, D_{57}, D_{58},…., D_{68} are the infinite maximal lines defining the input and θ_{56}', θ_{57}', θ_{58}', ……..., θ_{68}' are the angles D_{56}, D_{57}, D_{58},…., D_{68} make with the vertical axis respectively.

Now if $\theta_1' = \theta_{56}' + \theta$

$\theta_2' = \theta_{57}' + \theta$

$\theta_3' = \theta_{58}' + \theta$

$\theta_4' = \theta_{59}' + \theta$

$\theta_5' = \theta_{60}' + \theta$

$\theta_6' = \theta_{61}' + \theta$

$\theta_{7}' = \theta_{62}' + \theta$

$\theta_{8}' = \theta_{63}' + \theta$

$\theta_{9}' = \theta_{64}' + \theta$

$\theta_{10}' = \theta_{65}' + \theta$

$\theta_{11}' = \theta_{66}' + \theta$

$\theta_{12}' = \theta_{67}' + \theta$

$\theta_{13}' = \theta_{68}' + \theta$

then it is obvious that image of the database makes an angle of θ clockwise compared to the input object as image has for each and every infinite maximal line an angle orientation which is greater by constant value θ than the corresponding infinite maximal line of the input.

Since they correspond to each other as was shown in section 5.2.6, symmetry is established although the image of the database is rotated by an angle θ.

5.2.8 Calculation of Symmetry of Input As Lake and House

If our input comes in the form of a lake and a single house, then we could compute *Dist(I,O)* using L_p metric.

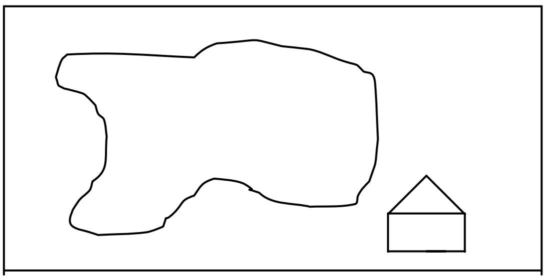


Figure 5.8 Input image in the form of a lake and a house

Let $I_1, I_2, \ldots, I_k$ be the vector of feature values derived from input *I* by considering feature values of *I* of all objects, namely lake and house and their distance etc. Similarly let $O_1, O_2, \ldots, O_k$ be the vector of feature values derived from output *O* by considering feature values of O of all objects, namely lake and two houses and their distances etc. Then distance between *I* and *O* is

$$Dist_p(I,O) = \left[\sum_{1}^{k} \left| I_i - O_i \right|^p \right]^{1/p}$$

where *p* is the order of the metric.

Suppose the perimeter of the input lake is, say, *200* units and house is *15* and distance between their center points is *60*. Also let us assume the perimeter of the image of the lake is *210*, one of the house has perimeter of *18* and distance between them is *55*.

For $p = 1$, we obtain Manhattan distance between I and O as

$$Dist\ (I, O) = (210 - 200) + (18 - 15) + (60 - 55)$$
$$= 18$$

If we set a limit that all images within distance t has to be retrieved then

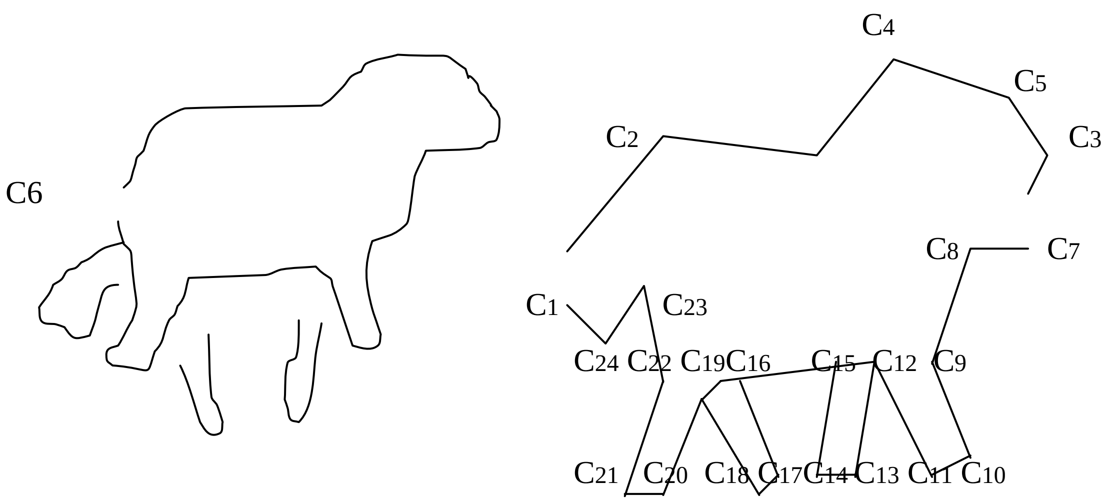
$Dist\ (I,O) \leq t$ is to be satisfied for retrieval.

If we have $t = 20$, then obviously the image containing a lake and two houses would be selected for this input. We could use Euclidean distance for $p=2$. This method can be applied for any L_p metric.

5.2.9 To Find Symmetry When an Object Is in Motion

5.2.9.1 Calculation of Index of Input As Irregular Shape

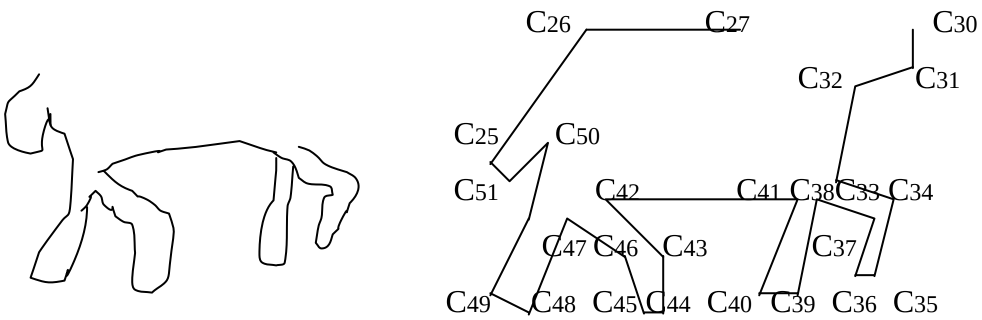
If the input comes in the shape of a four-footed, we convert the original shape (a) in the figure below in terms of infinite maximal lines in (b) with coordinates C_1, C_2, …., C_{24}.



(a) (b)

Figure 5.9 Input images in the form of a four-footed. (a) is the original shape and (b) is the shape after drawing infinite maximal lines tangential to the original shape

Now if the image from the database comes with the object in motion as shown below,



(a) (b)

Figure 5.10 Images of the object in motion. (a) is the original shape and (b) is the shape after drawing infinite maximal lines tangential to the original shape

then we could define it in terms of coordinates $C_{25}, C_{26}, \ldots, C_{51}$.

5.2.9.2 Correspondence Between Input and Image

Input $I = \{24;[C_1, C_2, \ldots\ldots, C_{24}]\}$

Object $O = \{27; [C_{25}, C_{26}, \ldots\ldots, C_{51}]\}$

Number of intersections

The number of intersections in input I is 24 whereas in Object O is 27. Hence they do not match.

Correspondence

If we start considering I from C_1 and go down and start considering O from C_{25} and also go down, then

$C_1 \approx (x_1, y_1) \Leftrightarrow C_{25} \approx (x_{25}, y_{25})$

$C_2 \approx (x_2, y_2) \Leftrightarrow C_{26} \approx (x_{26}, y_{26})$

$C_3 \approx (x_3, y_3) \Leftrightarrow C_{27} \approx (x_{27}, y_{27})$

$C_4 \approx (x_4, y_4) \Leftrightarrow C_{28} \approx (x_{28}, y_{28})$

$C_5 \approx (x_5, y_5) \Leftrightarrow C_{29} \approx (x_{29}, y_{29})$

$C_6 \approx (x_6, y_6) \Leftrightarrow C_{30} \approx (x_{30}, y_{30})$

$C_7 \approx (x_7, y_7) \Leftrightarrow C_{31} \approx (x_{31}, y_{31})$

$C_8 \approx (x_8, y_8) \Leftrightarrow C_{32} \approx (x_{32}, y_{32})$

$C_9 \approx (x_9, y_9) \Leftrightarrow C_{33} \approx (x_{33}, y_{33})$

No match $\Leftrightarrow C_{34} \approx (x_{34}, y_{34})$

$C_{10} \approx (x_{10}, y_{10}) \Leftrightarrow C_{35} \approx (x_{35}, y_{35})$

$C_{11} \approx (x_{11}, y_{11}) \Leftrightarrow C_{36} \approx (x_{36}, y_{36})$

No match $\Leftrightarrow C_{37} \approx (x_{37}, y_{37})$

$C_{12} \approx (x_{12}, y_{12}) \Leftrightarrow C_{38} \approx (x_{38}, y_{38})$

$C_{13} \approx (x_{13}, y_{13}) \Leftrightarrow C_{39} \approx (x_{39}, y_{39})$

$C_{14} \approx (x_{14}, y_{14}) \Leftrightarrow C_{40} \approx (x_{40}, y_{40})$

$C_{15} \approx (x_{15},y_{15}) \Leftrightarrow C_{41} \approx (x_{41}, y_{41})$

$C_{16} \approx (x_{16}, y_{16}) \Leftrightarrow C_{42} \approx (x_{42}, y_{42})$

No match $\Leftrightarrow C_{43} \approx (x_{43}, y_{43})$

$C_{17} \approx (x_{17}, y_{17}) \Leftrightarrow C_{44} \approx (x_{44}, y_{44})$

$C_{18} \approx (x_{18}, y_{18}) \Leftrightarrow C_{45} \approx (x_{45}, y_{45})$

No match $\Leftrightarrow C_{46} \approx (x_{46}, y_{46})$

$C_{19} \approx (x_{19}, y_{19}) \Leftrightarrow C_{47} \approx (x_{47}, y_{47})$

$C_{20} \approx (x_{20}, y_{20}) \Leftrightarrow C_{48} \approx (x_{48}, y_{48})$

$C_{21} \approx (x_{21}, y_{21}) \Leftrightarrow C_{49} \approx (x_{49}, y_{49})$

$C_{22} \approx (x_{22}, y_{22}) \Leftrightarrow$ No match

$C_{23} \approx (x_{23}, y_{23}) \Leftrightarrow C_{50} \approx (x_{50}, y_{50})$

$C_{24} \approx (x_{24}, y_{24}) \Leftrightarrow C_{51} \approx (x_{51}, y_{51})$

This shows only *23* coordinate points out of *24* of input match with *23* coordinate points out of *27* of *O*.

In this case, each individual match would have weight *100/27 = 3.70*. If *85%* match indicates symmetry, then total percentage match for *23* would be *3.70 * 23 = 85.10%*.

Since this is more than *85%*, we accept symmetry and the input *I* is identical to image *O*.

5.3 Accessing Database Using the Emergence Index Calculated

The ultimate goal of multimedia database is to have all information types digitized and computerized. But this has not yet been achieved.

The multimedia database could contain, as we know, text, audio, still images, digital video and graphic objects.

Multimedia database is a highly efficient database that supports multimedia data types, alphanumeric types and handle very large volumes of information. The multimedia database management system consists of three capabilities, which are conventional database system, hierarchical storage system support and information retrieval capacity. In hierarchical storage system, a multimedia database can support multimedia objects on-line, near-line and off-line. Each level has a different level of performance and capacity. The highest performance can be obtained from random access memory (RAM). This has the highest cost and the smallest capacity and little permanence. The next level of storage is magnetic Winchester hard drive. This is reasonable as far as cost, capacity and performance go. The next level is optical storage which can serve both as on-line and near-line. Performance is good in this case when it is on-line and slow when off-line. It has good capacity and acceptable cost. The third level is off-line which means optical media which is stored in cabinets and on shelves. In this case capacity could be very large but performance is extremely slow and it is cheap. The objects that are frequently accessed would be put in the upper level and objects that are no longer accessed are automatically put to lower level and slower media.

In multimedia databases queries are more often based on content of the objects. As we have shown, images can be accessed based on their contents and spatial relationships among various objects within the image. For voice data the index could be based upon speaker of the voice information. For color image color histograms could be used as index for search and access. For video images it could be scenes. For text data the search attributes could be nonstop words. Once we define the attributes and features of multimedia data, they could be used as index in searching multimedia database.

Images take lot of space on our hard disks. A multimedia database can help us wrangle these files. They allow us to view an image, play a movie or hear a sound even if you do not have the program that created it. This could be time saver. We do not need to launch several applications to look through a variety of files. We can drill down and search through results of a previous search to find an image, then drag it into a page layout program (Sandsmark, 1997).

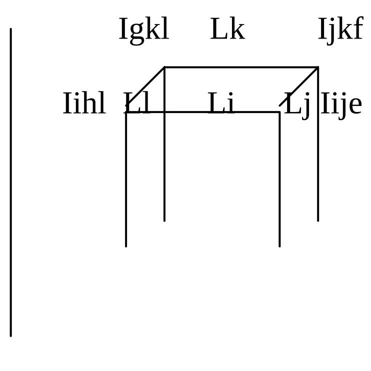
From a commercial point of view, the multimedia database is nothing but glorified file management system. In terms of research it means to do more in terms of finding semantic meanings of image (Swiderski, 1998).

Multimedia file types. Text files could be ASCII file containing text documents. Still image formats could be DCX,BMP,PCX,TIFF. For digitized voice it could be WAV files in Windows which can record and play back digitized voice files. For the file types of digitized video, it could be Microsoft AVI or Apple Quicktime. Multimedia database could be based on a conventional relational database with added software and hardware components. Hardware components needed could be video cameras, video boards, sound boards, high resolution monitors. Software components could be CAD/CAM, pattern recognition system, optical character recognition (OCR), hierarchical storage system as discussed earlier. Multimedia environments can use PC-based systems and also PCs with audiovisual capabilities and UNIX workstations with video conferencing facilities. Since we are dealing only with image databases, we will use content-based image retrieval system where we would define our index, based on contents of the image studying also the emergence phenomenon in the process for more precision and recall.

5.4 Calculation of Index of Input in the Form of a Cube

In this section, we consider a three dimensional image in the form of a house and an input in the form of a cube. We calculate index for both input and image and establish correspondence between input and lower section of the house after considering emergence phenomenon for the image using the same methodology described in earlier chapters.

If the input comes in the form of a cube then we define it in the following way.



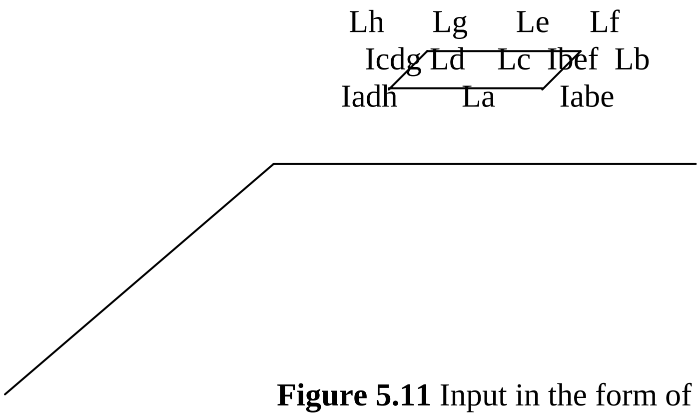


Figure 5.11 Input in the form of a cube

The *12* sides of the cube are L_a, L_b, L_c, L_d, L_e, L_f, L_g, L_h, L_i, L_j, L_k, L_l and there are *8* intersections I_{adh}, I_{abe}, I_{bcf}, I_{cdg}, I_{ije}, I_{jkf}, I_{gkl}, I_{ihl}.

Since all sides of the cube are of equal length, we have

$L_a = L_b = L_c = L_d = L_e = L_f = L_g = L_h = L_i = L_j = L_k = L_l$.

Also $L_a \,/\!/\, L_c$, $L_b \,/\!/\, L_d$, $L_e \,/\!/\, L_g$, $L_f \,/\!/\, L_h$, $L_i \,/\!/\, L_k$, $L_j \,/\!/\, L_l$, $L_a \,/\!/\, L_i$, $L_b \,/\!/\, L_j$, $L_c \,/\!/\, L_k$, $L_d \,/\!/\, L_l$.

The distance between I_{adh} and I_{abe} is same as the distance between I_{cdg} and I_{bcf}.

$$d(I_{adh}, I_{abe}) = d(I_{bcf}, I_{cdg})$$

The distance between I_{abe} and I_{bcf} is same as the distance between I_{adh} and I_{cdg}.

$$d(I_{abe}, I_{bcf}) = d(I_{adh}, I_{cdg})$$

The distance between I_{ihl} and I_{ije} is same as the distance between I_{gkl} and I_{jkf}.

$$d(I_{ihl}, I_{ije}) = d(I_{gkl}, I_{jkf})$$

The distance between I_{ihl} and I_{gkl} is same as the distance between I_{ije} and I_{jkf}.

$$d(I_{ihl}, I_{gkl}) = d(I_{ije}, I_{jkf})$$

The distance between I_{ihl} and I_{adh} is same as the distance between I_{ije} and I_{abe}.

$$d(I_{ihl}, I_{adh}) = d(I_{ije}, I_{abe})$$

The distance between I_{gkl} and I_{cdg} is same as the distance between I_{jkf} and I_{bcf}.

$$d(I_{gkl}, I_{cdg}) = d(I_{jkf}, I_{bcf})$$

The distance between I_{ihl} and I_{adh} is same as the distance between I_{gkl} and I_{cdg}.

$$d(I_{ihl}, I_{adh}) = d(I_{gkl}, I_{cdg})$$

The distance between I_{ije} and I_{abe} is same as the distance between I_{jkf} and I_{bcf}.

$$d(I_{ije}, I_{abe}) = d(I_{jkf}, I_{bcf}).$$

5.5 Calculation of Index of Image in the Form of a House

If the image comes in the form of a house, then we calculate the index in the following way.

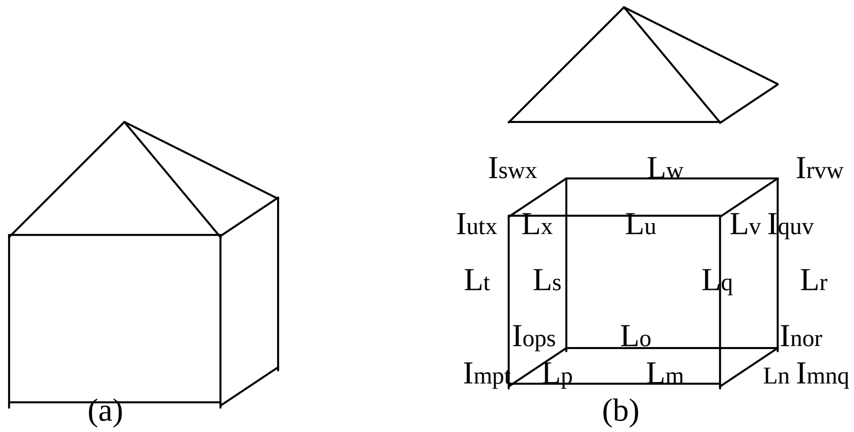


Figure 5.12 (a) House (b) Lower section and upper section shown Separately

The *12* sides of the lower section are L_m, L_n, L_o, L_p, L_q, L_r, L_s, L_t, L_u, L_v, L_w, L_x and there are *8* intersections I_{mpt}, I_{mnq}, I_{nor}, I_{ops}, I_{quv}, I_{rvw}, I_{swx}, I_{utx}.

Since all sides are of equal length, we have

$L_m = L_n = L_o = L_p = L_q = L_r = L_s = L_t = L_u = L_v = L_w = L_x$.

Also $L_m // L_o$, $L_n // L_p$, $L_q // L_s$, $L_r // L_t$, $L_u // L_w$, $L_v // L_x$, $L_m // L_u$, $L_n // L_v$, $L_o // L_w$, $L_p // L_x$.

The distance between I_{mpt} and I_{mnq} is same as the distance between I_{ops} and I_{nor}.

$$d(I_{mpt}, I_{mnq}) = d(I_{ops}, I_{nor})$$

The distance between I_{mnq} and I_{nor} is same as the distance between I_{mpt} and I_{ops}.

$$d(I_{mnq}, I_{nor}) = d(I_{mpt}, I_{ops})$$

The distance between I_{utx} and I_{quv} is same as the distance between I_{swx} and I_{rvw}.

$$d(I_{utx}, I_{quv}) = d(I_{swx}, I_{rvw})$$

The distance between I_{utx} and I_{swx} is same as the distance between I_{quv} and I_{rvw}.

$$d(I_{utx}, I_{swx}) = d(I_{quv}, I_{rvw})$$

The distance between I_{utx} and I_{mpt} is same as the distance between I_{quv} and I_{mnq}.

$$d(I_{utx}, I_{mpt}) = d(I_{quv}, I_{mnq})$$

The distance between I_{swx} and I_{ops} is same as the distance between I_{rvw} and I_{nor}.

$$d(I_{swx}, I_{ops}) = d(I_{rvw}, I_{nor})$$

The distance between I_{utx} and I_{mpt} is same as the distance between I_{swx} and I_{ops}.

$$d(I_{utx}, I_{mpt}) = d(I_{swx}, I_{ops})$$

The distance between I_{quv} and I_{mnq} is same as the distance between I_{rvw} and I_{nor}.

$$d(I_{quv}, I_{mnq}) = d(I_{rvw}, I_{nor}).$$

5.6 Correspondence Between Input and Lower Section of Image

Ordinary group intersections in each image

We compare the representation of the input with that of the image.

Input = $\{12; [\, I_{adh}, I_{abe}, I_{bcf}, I_{cdg}, I_{ije}, I_{jkf}, I_{gkl}, I_{ihl}]\}$

where 12 is the number of infinite maximal lines and I_{adh}, I_{abe}, …. are 8 intersections.

If we consider only the lower section of the image of the house, then

$O = \{12; [\, I_{mpt}, I_{mnq}, I_{nor}, I_{ops}, I_{quv}, I_{rvw}, I_{swx}, I_{utx}]\}$

where 12 is the number of infinite maximal lines and I_{mpt}, I_{mnq}, …. are 8 intersections.

Number of infinite maximal lines

We notice number of infinite maximal lines in each case is 12.

Corresponding equivalence

The corresponding equivalence of various segments is

$L_a \Leftrightarrow L_m \wedge L_b \Leftrightarrow L_n \wedge L_c \Leftrightarrow L_o \wedge L_d \Leftrightarrow L_p \wedge L_e \Leftrightarrow L_q \wedge L_f \Leftrightarrow L_r \wedge L_g \Leftrightarrow L_s \wedge$
$L_h \Leftrightarrow L_t \wedge L_i \Leftrightarrow L_u \wedge L_j \Leftrightarrow L_v \wedge L_k \Leftrightarrow L_w \wedge L_l \Leftrightarrow L_x$

Also the corresponding equivalence of various intersections are

$I_{adh} \Leftrightarrow I_{mpt} \wedge I_{abe} \Leftrightarrow L_{mnq} \wedge I_{bcf} \Leftrightarrow I_{nor} \wedge I_{cdg} \Leftrightarrow I_{ops} \wedge I_{ije} \Leftrightarrow I_{quv} \wedge I_{jkf} \Leftrightarrow I_{rvw} \wedge$
$I_{gkl} \Leftrightarrow I_{swx} \wedge I_{ihl} \Leftrightarrow I_{utx}$

Number of intersections

The number of intersections in each case is 8.

Geometric constraints of infinite maximal lines

Input has $I_{adh} \Leftrightarrow L_a \times L_d \times L_h$

$I_{abe} \Leftrightarrow L_a \times L_b \times L_e$

$I_{bcf} \Leftrightarrow L_b \times L_c \times L_f$

$I_{cdg} \Leftrightarrow L_c \times L_d \times L_g$

$I_{ije} \Leftrightarrow L_i \times L_j \times L_e$

$I_{jkf} \Leftrightarrow L_j \times L_k \times L_f$

$I_{gkl} \Leftrightarrow L_g \times L_k \times L_l$

$I_{ihl} \Leftrightarrow L_i \times L_h \times L_l$

O has $I_{mpt} \Leftrightarrow L_m \times L_p \times L_t$

$I_{mnq} \Leftrightarrow L_m \times L_n \times L_q$

$I_{nor} \Leftrightarrow L_n \times L_o \times L_r$

$I_{ops} \Leftrightarrow L_o \times L_p \times L_s$

$I_{quv} \Leftrightarrow L_q \times L_u \times L_v$

$I_{rvw} \Leftrightarrow L_r \times L_v \times L_w$

$I_{swx} \Leftrightarrow L_s \times L_w \times L_x$

$I_{utx} \Leftrightarrow L_u \times L_t \times L_x$

Dimensional constraints of segments

Since the number of intersections in input and *O* is same, we make the following deduction:

In input $L_a \times L_d \times L_h \wedge L_a \times L_b \times L_e \Leftrightarrow (I_{adh}, I_{abe})$

$L_a \times L_b \times L_e \wedge L_b \times L_c \times L_f \Leftrightarrow (I_{abe}, I_{bcf})$

$L_b \times L_c \times L_f \wedge L_c \times L_d \times L_g \Leftrightarrow (I_{bcf}, I_{cdg})$

$L_c \times L_d \times L_g \wedge L_a \times L_d \times L_h \Leftrightarrow (I_{cdg}, I_{adh})$

$L_i \times L_h \times L_l \wedge L_i \times L_j \times L_e \Leftrightarrow (I_{ihl}, I_{ije})$

$L_i \times L_j \times L_e \wedge L_j \times L_k \times L_f \Leftrightarrow (I_{ije}, I_{jkf})$

$L_j \times L_k \times L_f \wedge L_g \times L_k \times L_l \Leftrightarrow (I_{jkf}, I_{gkl})$

$L_g \times L_k \times L_l \wedge L_i \times L_h \times L_l \Leftrightarrow (I_{gkl}, I_{ihl})$

$L_a \times L_d \times L_h \wedge L_i \times L_h \times L_l \Leftrightarrow (I_{adh}, I_{ihl})$

$L_a \times L_b \times L_e \wedge L_i \times L_j \times L_e \Leftrightarrow (I_{abe}, I_{ije})$

$L_b \times L_c \times L_f \wedge L_j \times L_k \times L_f \Leftrightarrow (I_{bcf}, I_{jkf})$

$L_c \times L_d \times L_g \wedge L_g \times L_k \times L_l \Leftrightarrow (I_{cdg}, I_{gkl})$

So dimensional constraints in input are

$d(I_{adh}, I_{abe})$, $d(I_{abe}, I_{bcf})$, $d(I_{bcf}, I_{cdg})$, $d(I_{cdg}, I_{adh})$, $d(I_{ihl}, I_{ije})$, $d(I_{ije}, I_{jkf})$, $d(I_{jkf}, I_{gkl})$, $d(I_{gkl}, I_{ihl})$, $d(I_{adh}, I_{ihl})$, $d(I_{abc}, I_{ije})$, $d(I_{bcf}, I_{jkf})$, $d(I_{cdg}, I_{gkl})$.

In *O* $L_m \times L_p \times L_t \wedge L_m \times L_n \times L_q \Leftrightarrow (I_{mpt}, I_{mnq})$

$L_m \times L_n \times L_q \wedge L_n \times L_o \times L_r \Leftrightarrow (I_{mnq}, I_{nor})$

$L_n \times L_o \times L_r \wedge L_o \times L_p \times L_s \Leftrightarrow (I_{nor}, I_{ops})$

$L_o \times L_p \times L_s \wedge L_m \times L_p \times L_t \Leftrightarrow (I_{ops}, I_{mpt})$

$$L_u \times L_t \times L_x \wedge L_q \times L_u \times L_v \Leftrightarrow (I_{utx}, I_{quv})$$

$$L_q \times L_u \times L_v \wedge L_r \times L_v \times L_w \Leftrightarrow (I_{quv}, I_{rvw})$$

$$L_r \times L_v \times L_w \wedge L_s \times L_w \times L_x \Leftrightarrow (I_{rvw}, I_{swx})$$

$$L_s \times L_w \times L_x \wedge L_u \times L_t \times L_x \Leftrightarrow (I_{swx}, I_{utx})$$

$$L_m \times L_p \times L_t \wedge L_u \times L_t \times L_x \Leftrightarrow (I_{mpt}, I_{utx})$$

$$L_m \times L_n \times L_q \wedge L_q \times L_u \times L_v \Leftrightarrow (I_{mnq}, I_{quv})$$

$$L_n \times L_o \times L_r \wedge L_r \times L_v \times L_w \Leftrightarrow (I_{nor}, I_{rvw})$$

$$L_o \times L_p \times L_s \wedge L_s \times L_w \times L_x \Leftrightarrow (I_{ops}, I_{swx})$$

So dimensional constraints of O is

(I_{mpt}, I_{mnq}), (I_{mnq}, I_{nor}), (I_{nor}, I_{ops}), (I_{ops}, I_{mpt}), (I_{utx}, I_{quv}), (I_{quv}, I_{rvw}), (I_{rvw}, I_{swx}),

(Iswx, Iutx), (Impt, Iutx), (Imnq, Iuvq), (Inor, Irvw), (Iops, Iswx).

Corresponding intersections

$$L_a \leftrightarrow L_m \wedge L_d \leftrightarrow L_p \wedge L_h \leftrightarrow L_t \wedge L_a \times L_d \times L_h \wedge L_m \times L_p \times L_t => I_{adh} \leftrightarrow I_{mpt}$$

$$L_a \leftrightarrow L_m \wedge L_b \leftrightarrow L_n \wedge L_e \leftrightarrow L_q \wedge L_a \times L_b \times L_e \wedge L_m \times L_n \times L_q => I_{abe} \leftrightarrow I_{mnq}$$

$$L_b \leftrightarrow L_n \wedge L_c \leftrightarrow L_o \wedge L_f \leftrightarrow L_r \wedge L_b \times L_c \times L_f \wedge L_n \times L_o \times L_r => I_{bcf} \leftrightarrow I_{nor}$$

$$L_c \leftrightarrow L_o \wedge L_d \leftrightarrow L_p \wedge L_g \leftrightarrow L_s \wedge L_c \times L_d \times L_g \wedge L_o \times L_p \times L_s => I_{cdg} \leftrightarrow I_{ops}$$

$$L_i \leftrightarrow L_q \wedge L_j \leftrightarrow L_u \wedge L_e \leftrightarrow L_v \wedge L_i \times L_j \times L_e \wedge L_q \times L_u \times L_v => I_{ije} \leftrightarrow I_{quv}$$

$$L_j \leftrightarrow L_r \wedge L_k \leftrightarrow L_v \wedge L_f \leftrightarrow L_w \wedge L_j \times L_k \times L_f \wedge L_r \times L_v \times L_w => I_{jkf} \leftrightarrow I_{rvw}$$

$$L_g \leftrightarrow L_s \wedge L_l \leftrightarrow L_w \wedge L_k \leftrightarrow L_x \wedge L_g \times L_l \times L_k \wedge L_s \times L_w \times L_x => I_{gkl} \leftrightarrow I_{swx}$$

$$L_i \leftrightarrow L_u \wedge L_h \leftrightarrow L_t \wedge L_l \leftrightarrow L_x \wedge L_i \times L_h \times L_l \wedge L_u \times L_t \times L_x => I_{ihl} \leftrightarrow I_{utx}$$

Therefore, the input in the form of a cube and lower part of image O are symmetrical. Although the image O is that of the image of a house, still it would be selected in the search as a result of emergence.

5.7 Conclusion

In this chapter we discussed the methodologies we follow in accessing multimedia databases. First we established symmetry between input image and images of database. Then accessing database using the emergence index calculated had been discussed.

Chapter 6 Application of Emergence Index in Geographic Location

In this chapter we find out the index of geographic location and then establish symmetry through correspondence between input and image of the geographic location. Also we find out the emergent shape of the image by destroying the original shape, which gives rise to unstructured shape and then process the unstructured shape to generate emergent shape. We establish symmetry between input and emergent shape through correspondence between input and image.

6.1 Introduction

If we have a map of a geographic location like the following, then we find there are three streets, namely, STREET1, STREET2, STREET3. There is a park between STREET1 and STREET2 and HOUSE1, HOUSE2, HOUSE3, HOUSE4 are four houses on STREET2.

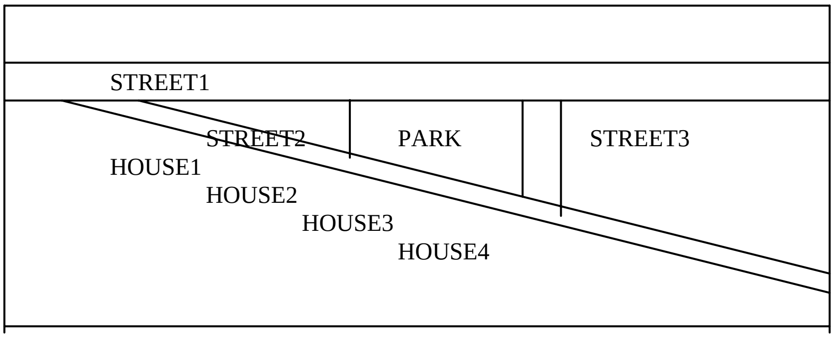


Figure 6.1 Geographic location

We also notice that STREET1, STREET2 and STREET3 form a triangle surrounding PARK. In normal map interpretation this may not surface. But when hidden shape is searched we get a triangle. This is the emergence outcome of the search. This would help us to locate the places more accurately by referring to the triangle in the map. Also if there is an input in the form of a triangle, then this image, although a map, would be selected because of emergence.

Next we consider the following map.

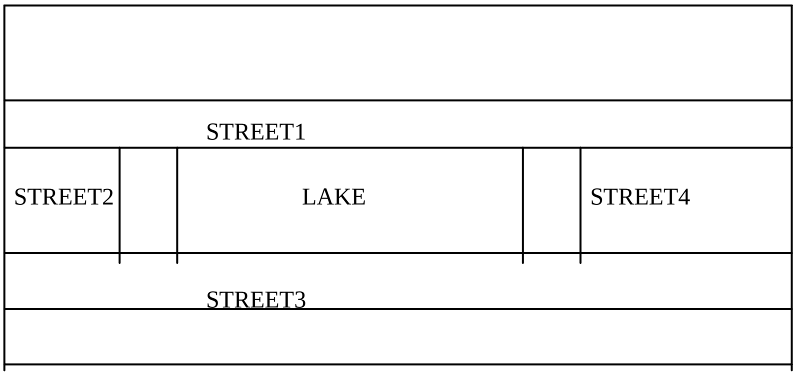


Figure 6.2 Geographic location

STREET1, STREET2, STREET3 and STREET4 surround a lake. But also these four streets give rise to a rectangle where opposite sides are parallel and equal. Obviously this is the emergence outcome and this image would be selected if an input comes in the form of a rectangle.

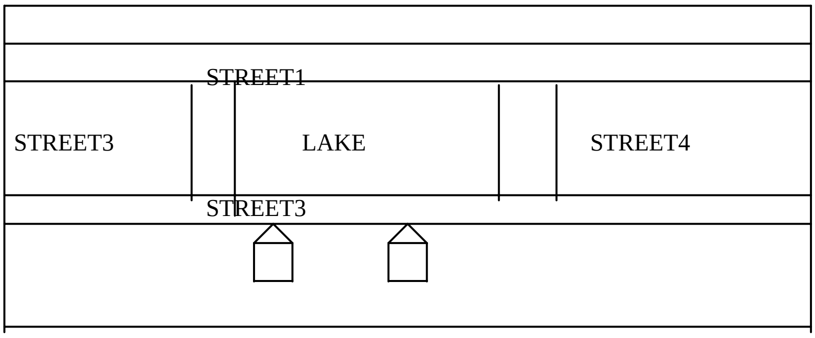


Figure 6.3 Geographic location

The above image is similar to the earlier one except that it has in addition two houses on STREET3.

So if an input image comes in the form of a lake, this image would be selected. If the input is a rectangle, then also this image would be selected. If the input comes in the form of a house then this image would be picked up. Lastly, if the input comes in the form of a locality, then also this image would be selected. None of the component parts or objects of the image is a locality, namely, streets, lake and houses. But combined together they generate a locality. This is the hidden meaning of the image based on which emergence index is formed.

6.2 Symmetry Between Input and Geographic Location

6.2.1 Calculation of Emergence Index Using Geographic Location

In order to calculate the emergence index, we use the equation (1) defined in 1.6.2.

Calculation of feature

In this example of a map, we find a park, a lake, roads and residential areas, which cover the rest part. This map is similar to any other map that could be drawn of a part of township. But if we look carefully into the picture of the map, we see roads which

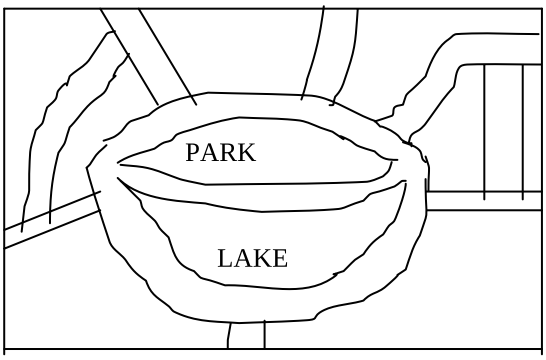


Figure 6.4 Geographic location

surround the park and the lake, form the shape of a bowl.

We take the shape of the bowl in coordinate system to analyze it.

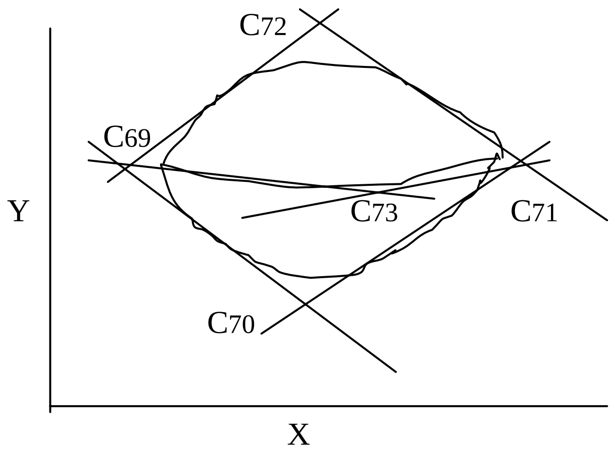
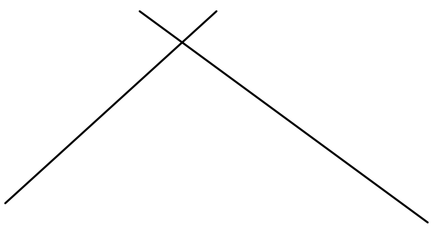


Figure 6.5 Coordinate system of shape



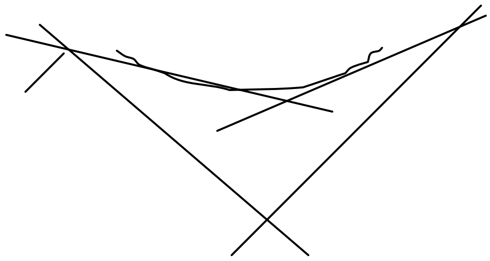


Figure 6.6 Destruction of original image

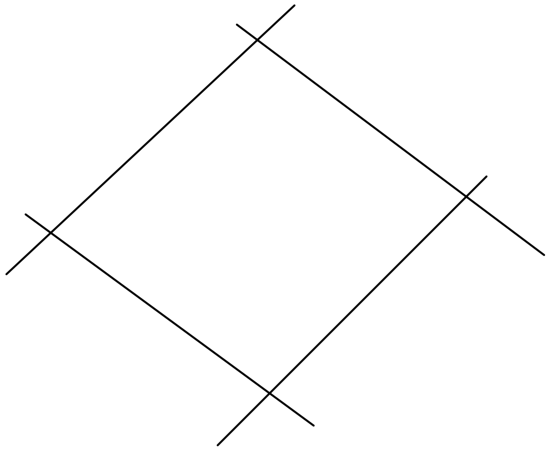


Figure 6.7 Emergent shape

Let C_{69}, C_{70}, C_{71}, C_{72} *and* C_{73} be the coordinates defining basic vectors where C_{69} corresponds to (x_{69}, y_{69}), C_{70} corresponds (x_{70}, y_{70}) and so on.

It is clear C_{69}, C_{70}, C_{71} *and* C_{72} form a parallelogram. This parallelogram is obtained from the shape of the bowl by first destroying the original shape, then processing the unstructured shape giving rise to new emergent shape of parallelogram as discussed in section 3.4.1 Model of Emergence.

To represent the object of bowl, we use equation (3) of 4.1 Feature

$$O_4 = f(w_1F_1;\ w_2F_2;\ w_3F_3\ ;\ldots..;\ w_nF_n).$$

The bowl is composed of a lake, a park and surrounding streets. Since lake is supposed to contain water, the color of the lake can be presumed to be blue and it occupies, say, around *40%* of the whole area of the bowl and color of water would be blue and so it is denoted by *2*. We assume park would occupy another *40%* of the area and the color would be green which is denoted by *3*. Let us assume streets occupy the rest *20%* of the bowl with black color which is denoted by, say, *5*. Hence if F_1 is the standard color feature as we defined earlier then we calculate

$$w_1F_1 = (0.4 * 2, 0.4 * 3, 0.2 * 5)$$

Since O_4 would be composed of *5* coordinate points and if this object O_4 occupies *40%* of total image then the weight-feature factor would be

$$w_2F_2 = 0.4 * f\{(x_{69}, y_{69}), (x_{70}, y_{70}), \ldots\ldots (x_{73}, y_{73})\}$$

where $(x_{69}, y_{69}), (x_{70}, y_{70}), \ldots\ldots (x_{73}, y_{73})$ represent the shape of the object O_4.

If the background is grass which is green and occupies *50%* of the image and streets which is black occupy the rest *10%*, then F_3 which represent background feature and w_3 which represent area covered, in combined way would be

$$w_3F_3 = (0.5 * 3, 0.1 * 5)$$

where *3* stands for green color and *5* for black.

If d_1, d_2, d_3, d_4, d_5 and d_6 are the distances between various coordinate points of the object O_4, then

$$\text{Perimeter of } O_4 = \sum_{1}^{6} d_i$$

Since this covers the whole perimeter where there is no undefined or open part, the w_4 would be *1.0*. Hence weight-feature factor becomes in this case

$$w_4F_4 = 1.0 * \sum_{1}^{6} d_i$$

Therefore, O_4 could be represented in this case as

$$O_4 = f(w_1F_1;\ w_2F_2;\ w_3F_3;\ w_4F_4)$$

$$= [(0.4 * 2, 0.4 * 3, 0.2 * 5);\ 0.4 * f\{(x_{69}, y_{69}), (x_{70}, y_{70}), \ldots\ldots x_{73}, y_{73})\};\ 0.5 * 3,$$

$$0.1 * 5);\ 1.0 * \sum_{1}^{6} d_i]$$

Calculation of domain

In the hierarchical domain structure of 4.2, if our image lies in *D4.7*, then clearly the domain path is $D_1 / D_{2.2} / D_{3.4} / D_{4.7}$.

Variables

In the example image, we have a lake, a park and streets that form a bowl O_4. We already defined the variables for this. They are $(x_{69}, y_{69}), (x_{70}, y_{70}), \ldots\ldots (x_{73}, y_{73})$.

Constraints

According to our definition of symbolic representation of shape in equation (2) in 3.2

$$I = \{N;\ \text{constraints}\}$$

For O_4, $I = \{5; [C_{69}, C_{70}, C_{71}, \ldots, C_{73}]\}$

Emergence

Although object O_4 is basically composed of a lake, a park and streets, it gives rise to the shape of a bowl as we pointed earlier. This is the emergence output of the image. This is the example of embedded shape emergence where emergence is a set of the whole image as we discussed in section 1.5.

6.2.2 Input Smaller and Rotated

The input here comes in the form of a bowl with *5* intersections of straight lines that define the shape. The shape here is smaller than O_4 and is rotated. Let these intersections be $C_{74}, C_{75}, C_{76}, \ldots, C_{78}$. So the representation in this case is

$$I = \{5;\ [C_{74}, C_{75}, C_{76}, \ldots, C_{78}]\}$$

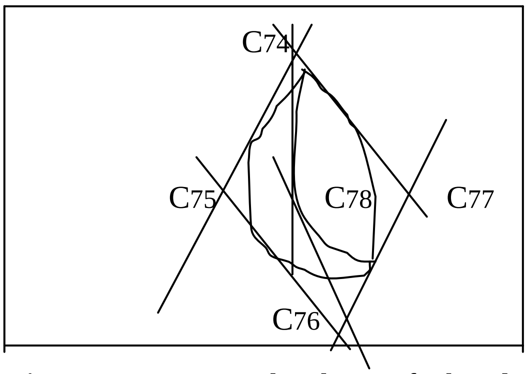


Figure 6.8 Input in the shape of a bowl

Correspondence between image and input

Input $= \{5; [C_{74}, C_{75}, \ldots, C_{78}]\}$

$O4 = \{5; [C_{69}, C_{70}, \ldots, C_{73}]\}$

Number of intersections

The number of intersections in input is *5* and in O_4 also it is *5*. Hence they match.

Correspondence

Since the size of input is smaller than O_4, we cannot establish correspondence straightaway.

Let $\theta_{69}, \theta_{70}, \theta_{71}, \ldots, \theta_{73}$ be the angles at the points of intersection in O_4 and let $\theta_{74}, \theta_{75}, \theta_{76}, \ldots, \theta_{78}$ be the angles at the points of intersection in input.

Now if $\theta_{69} \Leftrightarrow \theta_{74}$

$\theta_{70} \Leftrightarrow \theta_{75}$

$\theta_{71} \Leftrightarrow \theta_{76}$

$\theta_{72} \Leftrightarrow \theta_{77}$

$\theta_{73} \Leftrightarrow \theta_{78}$

then correspondence between input and O_1 is established although input is smaller in size compared to O_4.

So if the number of intersections are same and angles at those intersections correspond to each other in input and O_4, then symmetry is established although input and O_4 are not of same size.

Since the input is rotated from the vertical axis clockwise, we can obtain further symmetry in the following way.

Let $D_1, D_2, D_3,\ldots, D_6$ are the infinite maximal lines defining the image and θ_1, θ_2, θ_3, ……..., θ_6 are the angles $D_1, D_2, D_3,\ldots, D_6$ make with the vertical axis respectively. Let $D_7, D_8, D_9,\ldots, D_{12}$ are the infinite maximal lines defining the input and θ_7, θ_8, θ_9, ……...,θ_{12} are the angles $D_7, D_8, D_9,\ldots, D_{12}$ make with the vertical axis respectively.

Now if $\theta_7 = \theta_1 + \theta$

$\theta_8 = \theta_2 + \theta$

$\theta_9 = \theta_3 + \theta$

$\theta_{10} = \theta_4 + \theta$

$\theta_{11} = \theta_5 + \theta$

$\theta_{12} = \theta_6 + \theta$

then it is obvious that input makes a constant angle θ clockwise compared to the object of the image of the database for each and every infinite maximal line.

This establishes further symmetry although the image of the database is rotated by an angle θ.

When we want to locate a particular geographical area and we do not know where to look for but if we know there are streets in the location that make a bowl, then we can quite easily find that location by having input as image of a bowl, which can find the match with the image of emergent bowl of the map. This would be the advantage of applying this concept of emergence in image retrieval in practice.

6.3 Symmetry Between Input and Geographic Location After Emergence

6.3.1 Calculation of Emergence Index of Emergent Shape

We have seen how emergence has given rise to the shape of a parallelogram from a bowl-shaped geographic region in 6.2.1. Now we go to calculate emergence index of this parallelogram. This is the outcome of illusory shape emergence as discussed in section 1.5 where contours defining the shape are perceived though no contours are physically present.

In order to calculate the emergence index, we use the equation (1) defined in section 1.6.2 here as well.

Calculation of feature

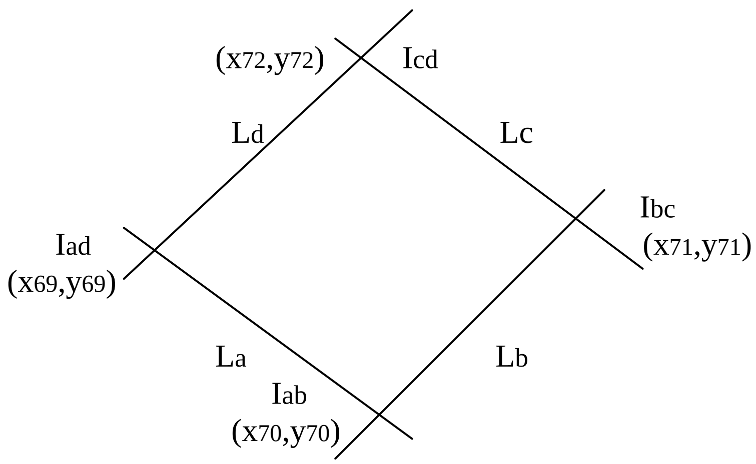


Figure 6.9 Emergent shape

Let C_{69}, C_{70}, C_{71} *and* C_{72} be the coordinates defining basic vectors where C_{69} corresponds to (x_{69}, y_{69}), C_{70} corresponds (x_{70}, y_{70}) and so on.

To represent the object of parallelogram, we use equation (3) of 4.1 Feature

$$O_5 = f(w_1F_1; w_2F_2; w_3F_3;\ldots; w_nF_n).$$

The parallelogram is composed of a lake, a park, surrounding streets and little bit of residential areas. If lake occupies, say, around *35%* of the whole area of the parallelogram and color of water is blue then it is denoted by *2*. We assume park would occupy another *35%* of the area and the color would be green which is denoted by *3*. Let us assume streets occupy the rest *20%* of the bowl with black color which is denoted by, say, *5*. Also if residential areas occupy the rest *10%* which is green then it would be denoted by *3*. So if F_1 is the standard color feature as we defined earlier then we calculate

$$w_1F_1 = (0.35 * 2, 0.45 * 3, 0.2 * 5)$$

Since O_5 would be composed of 4 coordinate points and if this object O_5 occupies *45%* of total image then the weight-feature factor would be

$$w_2F_2 = 0.45 * f\{(x_{69}, y_{69}), (x_{70}, y_{70}),\ldots\ldots (x_{72}, y_{72})\}$$

where $(x_{69}, y_{69}), (x_{70}, y_{70}),\ldots\ldots, (x_{72}, y_{72})$ represent the shape of the object O_5.

If the background is grass which is green and occupies *45%* of the image and streets which is black occupy the rest *10%*, then F_3 which represent background feature and w_3 which represent area covered, in combined way would be

$$w_3F_3 = (0.45 * 3, 0.1 * 5)$$

where *3* stands for green color and *5* for black.

If d_1, d_2, d_3 and d_4 are the distances between various coordinate points of the object O_5, then

$$\text{Perimeter of } O_5 = \sum_{1}^{4} d_i$$

Since this covers the whole perimeter where there is no undefined or open part, the w_4 would be *1.0*. Hence weight-feature factor becomes in this case

$$w_4F_4 = 1.0 * \sum_{1}^{4} d_i$$

Therefore, O_5 could be represented in this case as

$$O_5 = f(w_1F_1; w_2F_2; w_3F_3; w_4F_4)$$

$$= [(0.35 * 2, 0.45 * 3, 0.2 * 5); 0.45 * f\{(x_{69}, y_{69}), (x_{70}, y_{70}),\ldots\ldots, (x_{72}, y_{72})\};$$

$$(0.45 * 3, 0.1 * 5); 1.0 * \sum_{1}^{4} d_i]$$

Calculation of domain

In the hierarchical domain structure of figure 4.2, our image lies in $D_{4.7}$ as we mentioned in section 6.2.1, then clearly the domain path remains the same and is $D_1 / D_{2.2} / D_{3.4} / D_{4.7}$.

Variables

For object O_5, the variables are obviously $(x_{69}, y_{69}), (x_{70}, y_{70}),\ldots, (x_{72}, y_{72})$. The four sides are L_a, L_b, L_c and L_d.

Constraints

According to our definition of symbolic representation of shape in equation (2) in section 3.2

$$I = \{N; \text{constraints}\}$$

For O_5, the number of sides is 4 and the distances between (x_{69}, y_{69}), (x_{70}, y_{70}), (x_{71}, y_{71}) and (x_{72}, y_{72}) are same. In other words,

$d\{(x_{69}, y_{69}), (x_{70}, y_{70})\} = d\{(x_{70}, y_{70}), (x_{71}, y_{71})\} = d\{(x_{71}, y_{71}), (x_{72}, y_{72})\} = d\{(x_{72}, y_{72}), (x_{69}, y_{69})\}$.

$d(I_{ad}, I_{ab}) = d(I_{ab}, I_{bc}) = d(I_{bc}, I_{cd}) = d(I_{cd}, I_{ad})$.

Then $L_a // L_c$, $L_b // L_d$ since opposite sides are parallel.

Since θ_{69}, θ_{70}, θ_{71}, θ_{72} are the angles at the points of intersection in O_5 and it is a parallelogram, hence $\theta_{69} = \theta_{71}$ and $\theta_{70} = \theta_{72}$.

Emergence

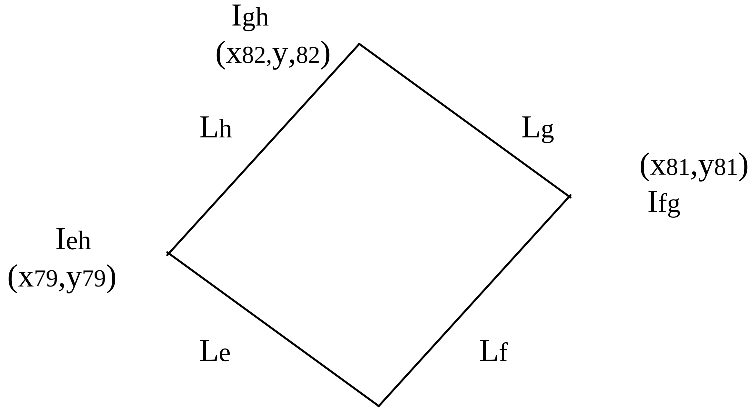
The shape of the parallelogram out of a bowl is the outcome of illusory emergence as we mentioned earlier.

This completes the calculation of emergence index of equation (1) in this case.

6.3.2 Calculation of Index of Input As Parallelogram

If the input comes in the form of a parallelogram below, then we have the coordinates which define the parallelogram are (x_{79}, y_{79}), (x_{80}, y_{80}), (x_{81}, y_{81}) and (x_{82}, x_{82}) representing C_{79}, C_{80}, C_{81} and C_{82} respectively. The sides are L_e, L_f, L_g, L_h.

Also the number of sides is 4 and the distances between (x_{79}, y_{79}), (x_{80}, y_{80}), (x_{81}, y_{81}), and (x_{82}, y_{82}) are same. In other words,



(x_{80}, y_{80})
I_{ef}

Figure 6.10 Input as a parallelogram

$d\{(x_{79}, y_{79}), (x_{80}, y_{80})\} = d\{(x_{80}, y_{80}), (x_{81}, y_{81})\} = d\{(x_{81}, y_{81}), (x_{82}, y_{82})\} = d\{(x_{82}, y_{82}), (x_{79}, y_{79})\}$.

$d(I_{eh}, I_{ef}) = d(I_{ef}, I_{fg}) = d(I_{fg}, I_{gh}) = d(I_{gh}, I_{eh})$.

Then $L_e \,//\, L_g$, $L_f \,//\, L_h$ since opposite sides are parallel.

If θ_{79}, θ_{80}, θ_{81}, θ_{82} are the angles at the points of intersection and since it is a parallelogram, hence $\theta_{79} = \theta_{81}$ *and* $\theta_{80} = \theta_{82}$.

6.3.3 Correspondence Between Input (parallelogram) and Image

Ordinary group intersections in each image

Here we compare the representation of the input with O_5 in section 6.3.1.

Input = $\{4; [I_{eh}, I_{ef}, I_{fg}, I_{gh}]\}$

$O_5 = \{4; [I_{ad}, I_{ab}, I_{bc}, I_{cd}]\}$

Number of infinite maximal lines

We notice number of infinite maximal lines in each case is *4*.

Corresponding equivalence

The corresponding equivalence of various segments is

$L_e \Leftrightarrow L_a \wedge L_f \Leftrightarrow L_b \wedge L_g \Leftrightarrow L_c \wedge L_h \Leftrightarrow L_d$

Also the corresponding equivalence of various intersections are

$I_{eh} \Leftrightarrow I_{ad} \wedge I_{ef} \Leftrightarrow L_{ab} \wedge I_{fg} \Leftrightarrow I_{bc} \wedge I_{gh} \Leftrightarrow I_{cd}$

Number of intersections

The number of intersections in each case is *4*.

Geometric constraints of infinite maximal lines

Input has $I_{eh} \Leftrightarrow L_e \times L_h$

$I_{ef} \Leftrightarrow L_e \times L_f$

$I_{fg} \Leftrightarrow L_f \times L_g$

$I_{gh} \Leftrightarrow L_g \times L_h$

O_2 has $I_{ad} \Leftrightarrow L_a \times L_d$

$I_{ab} \Leftrightarrow L_a \times L_b$

$I_{bc} \Leftrightarrow L_b \times L_c$

$I_{cd} \Leftrightarrow L_c \times L_d$

Dimensional constraints of segments

Since the number of intersections in input and O_5 is same, we deduct

In input $L_e \times L_h \wedge L_e \times L_f \Leftrightarrow (I_{eh}, I_{ef})$

$L_e \times L_f \wedge L_f \times L_g \Leftrightarrow (I_{ef}, I_{fg})$

$L_f \times L_g \wedge L_g \times L_h \Leftrightarrow (I_{fg}, I_{gh})$

$L_g \times L_h \wedge L_e \times L_h \Leftrightarrow (I_{gh}, I_{eh})$

So dimensional constraints in input is

$d(I_{eh}, I_{ef}), d(I_{ef}, I_{fg}), d(I_{fg}, I_{gh}), d(I_{gh}, I_{eh})$

In O_5 $L_a \times L_d \wedge L_a \times L_b \Leftrightarrow (I_{ad}, I_{ab})$

$L_a \times L_b \wedge L_b \times L_c \Leftrightarrow (I_{ab}, I_{bc})$

$L_b \times L_c \wedge L_c \times L_d \Leftrightarrow (I_{bc}, I_{cd})$

$L_c \times L_d \wedge L_a \times L_d \Leftrightarrow (I_{cd}, I_{ad})$

So dimensional constraints of O_5 is

$d(I_{ad}, I_{ab}), d(I_{ab}, I_{bc}), d(I_{bc}, I_{cd}), d(I_{cd}, I_{ad})$

Corresponding intersections

$L_e \leftrightarrow L_a \wedge L_h \leftrightarrow L_d \wedge L_e \times L_h \wedge L_a \times L_d => I_{eh} \leftrightarrow I_{ad}$

$L_e \leftrightarrow L_a \wedge L_f \leftrightarrow L_b \wedge L_e \times L_f \wedge L_a \times L_b => I_{ef} \leftrightarrow I_{ab}$

$L_f \leftrightarrow L_b \wedge L_g \leftrightarrow L_c \wedge L_f \times L_g \wedge L_b \times L_c => I_{fg} \leftrightarrow I_{bc}$

$L_g \leftrightarrow L_c \wedge L_h \leftrightarrow L_d \wedge L_g \times L_h \wedge L_c \times L_d => I_{gh} \leftrightarrow I_{cd}$

Corresponding angles

If $\theta_{69} = \theta_{79}, \theta_{70} = \theta_{80}, \theta_{71} = \theta_{81}, \theta_{72} = \theta_{82}$ then corresponding angles are also same. Therefore, the input in the form of a parallelogram and image O_5 are symmetrical. Although the image O_5 is the emergence outcome of the bowl O_4, still this would be selected when the input is a parallelogram as a result of emergence.

6.4 Conclusion

In this chapter, we found out the index of geographic location and then established symmetry through correspondence between input and image of the geographic location. Also we found out the emergent shape of the image, first by destroying the original shape, which gave rise to an unstructured shape and then processed the unstructured shape to generate emergent shape. We established symmetry between input and emergent shape through correspondence between input and image.

Chapter 7 Implementation of the System and Analysis of Experimental Results

The state-of-the-art technology is presented first in this chapter where we summarize the progresses so far made in the area of content-based image retrieval research. Then we produce some implementation and analysis of experimental results.

7.1 Current Technology

Before implementing, we have to consider the state-of-the-art technology in image retrieval. The developments in this field have been defined in three levels (Eakins and Graham, 1999).

Level one is the primitive level where low-level features like color, texture, shape and spatial locations are used to segment images in image database and then find symmetry based on these segmentations with the input image. Plenty of researches were being done during the last decade. Many software packages have been developed for efficient image retrieval. Most of them have used combination of text-based and content-based retrieval. In those packages, images are segmented manually first, then texts are generated based on those manual segmentations and finally retrievals are conducted based on the similarity of texts between the input and the images of the database. But since the volume of images generated could be enormous in fields like satellite picturing, this method of manual part processing is time-consuming and expensive. Few automatic retrievals without human intervention have been developed like QBIC, Virage, Excalibur which are now commercially being used in addition to packages developed which are not yet commercial. But they have limited applications in areas like trademark registration, identification of drawings in a design archive or color matching of fashion accessories based on input image. No universally accepted retrieval technique has yet been developed. Also the retrieval techniques developed without human intervention are far from perfect. Segmentation has been done in some packages based on color where the segmented parts taken individually do not contribute to any meaningful identification (Ma et al, Year Unknown). They generate a vague symmetry between input objects and objects in image database. This level still needs to be developed further to have universal applications.

Level two deals with bringing out semantic meanings of an image of the database. This problem was discussed in chapter 2 with umbrellas. One of the best known works in this field is of Forsyth et al (1996) by successfully identifying human beings within images and this technique had been applied for other objects like horses and trees.

Also for example, a beach can be identified if search is based on color and texture matching and color selected is wide blue with yellow texture below.

But this level also needs lots more developments to achieve universally accepted technique to bring out semantic meanings out of the image.

Level three attempts retrievals with abstract attributes. This level of retrieval can be divided into two groups. One is a particular event like ‘Find pictures of Australia playing cricket against another particular country’. Second one could be ‘Find a picture which is a residential area’. We discussed this in chapter 2.

To interpret an image after segmentations and analyzing it efficiently require very complex reasoning. This also requires retrieval technique of level two to get semantic meanings of various objects. It is obvious this retrieval technique is far from being developed with modern technology available in the field.

7.2 Aims of the Experiment

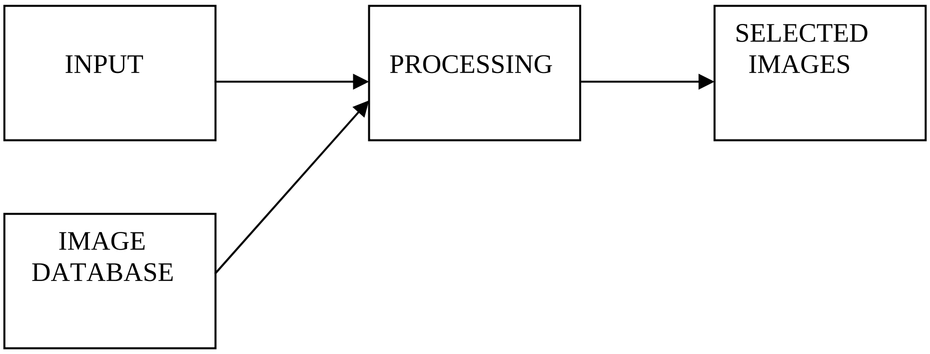
To rectify the problems of existing image retrieval systems, which does not take into account the hidden meanings of an image and to make the retrieval systems more accurate and efficient, we undertake this experiment on content-based image retrieval based on emergence index. Here we take into account the explicit as well as implicit or hidden meanings of the image.

To access the multimedia database, we convert each record of the images into parametric forms and then try to match it against the input, which is also in parametric form. Whenever a match is found, we retrieve that record. In this way, we end up getting output based on the emergence phenomenon of images in the database, according to the input given.

7.3 How to Attain the Objectives

In this section we present a plan in the form of a flow-chart for accessing database with emergence index. Then we would describe algorithms for the retrieval of geographic image without and with emergence.

Basically we have input and images of database and then we process input and images to select relevant images. In the processing stage, we match input against each image of the database to find symmetry. If symmetry is established, then this particular image is selected. If no symmetry is found, then we study the emergence phenomenon and check if symmetry could be established with the input. If symmetry is found, the image is selected otherwise we reject this image and go to the next image of the database.



PROCESSING

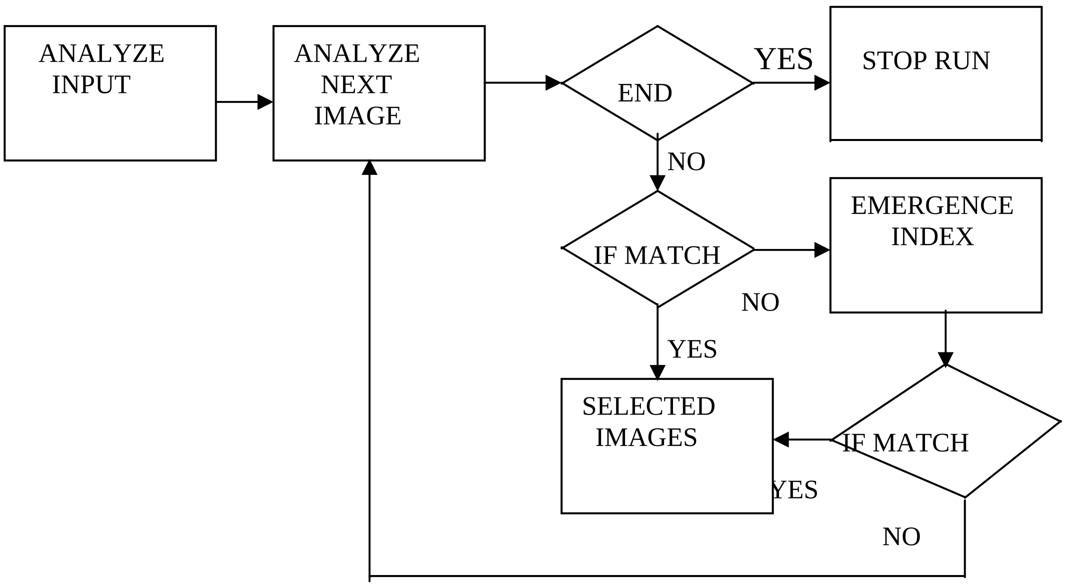


Figure 7.1 Flow chart

7.4 Algorithm for Retrieval Without Emergence

If the image comes in the form of a map of section 6.2.1 and input as in section 6.2.2 then we follow the steps below.

Inputs are the map of section 6.2.2 and image map of section 6.2.1.

Outputs are selected images.

Step 1. Analysis of input

Begin

Number of tangents = 6

Number of infinite maximal lines = 6 and they are $D_7, D_8, D_9, \ldots, D_{12}$.

Number of coordinate points Ni = 5 and they are $C_{74}, C_{75}, C_{76}, C_{77}, C_{78}$.

Angles infinite maximal lines make with the next one are $\theta_{74}, \theta_{75}, \theta_{76}, \ldots, \theta_{78}$.

Angles infinite maximal lines make with the vertical axis are

$\theta_7, \theta_8, \theta_9, \theta_{10}, \theta_{11}, \theta_{12}$.

For color, weight-feature factor $L_i = w_1F_1 = 1.0 * 0 = 0$ *as for the whole input is color independent.*

For shape, $S_i = w_2F_2 = 0.3 * f\{(x_{74}, y_{74}), (x_{75}, y_{75}), \ldots, (x_{78}, y_{78})\}$

if the bowl occupies 30% of the image.

For background, $B_i = w_3F_3 = 0.7 * 0 = 0$ *since background occupies 70% of the rest*

space and is color independent.

For distances between various coordinate points

$$P_i = w_4F_4 = 1.0 * \sum_{7}^{12} d_i$$

So feature $F = f(w_1F_1; w_2F_2; w_3F_3; w_4F_4)$

$$= [0; 0.3 * f\{(x_{74}, y_{74}), (x_{75}, y_{75}), \ldots, (x_{78}, y_{78})\}; 0; 1.0 * \sum_{7}^{12} d_i]$$

Since this is an input image, there is no domain. Hence D = 0.

Variables are $V = (x_{74}, y_{74}), (x_{75}, y_{75}), \ldots, (x_{78}, y_{78})$.

Constraints are $C = \{5; (C_{74}, C_{75}, \ldots, C_{78})\}$.

As this is an input of a bowl, there is no emergence. Hence E = 0.

So emergence index

$EI = f(D, F, V, C, E)$

$$= [0; [0; 0.3 * f\{(x_{74}, y_{74}), (x_{75}, y_{75}), \ldots, (x_{78}, y_{78})\}; 0; 1.0 * \sum_{7}^{12} d_i];$$

$$(x_{74}, y_{74}), (x_{75}, y_{75}), \ldots, (x_{78}, y_{78}); \{5; (C_{74}, C_{75}, \ldots, C_{78})\}; 0]$$

End

Step 2. Analysis of next image

Begin

If end of file, stop run otherwise next sentence.

Number of tangents = 6

Number of infinite maximal lines = 6 and they are $D_1, D_2, D_3, \ldots, D_6$.

Number of coordinate points $No = 5$ *and they are* $C_{69}, C_{70}, C_{71}, C_{72}, C_{73}$.

Angles infinite maximal lines make with the next one are $\theta_{69}, \theta_{70}, \theta_{71}, \ldots, \theta_{73}$.

Angles infinite maximal lines make with the vertical axis are

$\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$.

For color, weight-feature factor $Lo = w_1F_1 = (0.4 * 2, 0.4 * 3, 0.2 * 5)$.

For shape, $So = w_2F_2 = 0.4 * f\{(x_{69}, y_{69}), (x_{70}, y_{70}), \ldots, (x_{73}, y_{73})\}$.

For background, $Bo = w_3F_3 = (0.5 * 3, 0.1 * 5)$.

For distances between various coordinate points

$$Po = w_4F_4 = 1.0 * \sum_{1}^{6} d_i$$

So feature $F = f(w_1F_1; w_2F_2; w_3F_3; w_4F_4)$

$= [(0.4 * 2, 0.4 * 3, 0.2 * 5);\ 0.4 * f\{(x_{69}, y_{69}),(x_{70}, y_{70}), \ldots, (x_{73}, y_{73})\}; (0.5 * 3, 0.1 * 5); 1.0 * \sum_{1}^{6} d_i]$

The domain path $D = D_1 / D_{2.2} / D_{3.4} / D_{4.7}$.

Variables are $(x_{69}, y_{69}),(x_{70}, y_{70}), \ldots, (x_{73}, y_{73})$.

Constraints are $C = \{5;\ (C_{69}, C_{70}, \ldots, C_{73})\}$.

Emergence E is the shape of a bowl

$= f((x_{69}, y_{69}), (x_{70}, y_{70}), \ldots, (x_{73}, y_{73}))$.

Hence emergence index

$EI = f(D, F, V, C, E)$

$= [(D_1 / D_{2.2} / D_{3.4} / D_{4.7}); [(0.4 * 2, 0.4 * 3, 0.2 * 5);\ 0.4 * f\{(x_{69}, y_{69}),(x_{70}, y_{70}), \ldots, (x_{73}, y_{73})\};\ (0.5 * 3, 0.1 * 5); 1.0 * \sum_{1}^{6} d_i];\ (x_{69}, y_{69}),(x_{70}, y_{70}), \ldots, (x_{73}, y_{73}); \{5; (C_{69}, C_{70}, \ldots, C_{73})\}; f(x_{69}, y_{69}), (x_{70}, y_{70}), \ldots, (x_{73}, y_{73})]$.

End

Step 3. Find Correspondence

Begin

Compare the features F calculated for input and image, then

Color $Li \neq Lo$

Shape $Si \neq So$

when we compare the percentage of image occupied by the object,

but

$Ni = No$*, the number of coordinates match, in each case being 5.*

Background $Bi \neq Bo$.

Distances between various coordinate points $Pi = Po$.

Although color, shape and background of the input and image do not match, the other factors match.

Number of intersections in input and image is 5 and so they match.

If $\theta_{69} \Leftrightarrow \theta_{74}$

$\theta_{70} \Leftrightarrow \theta_{75}$

$\theta_{71} \Leftrightarrow \theta_{76}$

$\theta_{72} \Leftrightarrow \theta_{77}$

$\theta_{73} \Leftrightarrow \theta_{78}$

then $Si \Leftrightarrow So$ *although input is smaller in size.*

Since number of intersections are same and angles correspond to each other, symmetry is established.

Also if $\theta_7 = \theta_1 + \theta$

$\theta_8 = \theta_2 + \theta$

$\theta_9 = \theta_3 + \theta$

$$\theta_{10} = \theta_4 + \theta$$

$$\theta_{11} = \theta_5 + \theta$$

$$\theta_{12} = \theta_6 + \theta$$

that proves input is rotated by a constant angle θ and further symmetry is established.

End

Step 4. *Since they match, this image is selected and go to* ***step 2.***

7.5 Algorithm for Retrieval With Emergence

If the image comes in the form of a map of 6.2.1 and input as in 6.3.2 in the form of a parallelogram, then inputs are map of section 6.3.2 and images of section 6.2.1. Outputs are selected images.

Step 1. Analysis of input

Begin

Number of infinite maximal lines = 4.

Number of coordinate points Ni = 4 and they are C_{79}, C_{80}, C_{81}, C_{82}.

Four sides are L_e, L_f, L_g, L_h.

Four intersections are I_{eh}, I_{ef}, I_{fg}, I_{gh}.

Distances between four intersections are same. In other words,

$d(I_{eh}, I_{ef}) = d(I_{ef}, I_{fg}) = d(I_{fg}, I_{gh}) = d(I_{gh}, I_{eh})$.

For color, weight-feature factor $Li = w_1F_1 = 1.0 * 0 = 0$ as for the whole input is color independent.

For shape, $Si = w_2F_2 = 0.45 * f\{(x_{79}, y_{79}), (x_{80}, y_{80}),\ldots\ldots,(x_{82}, y_{82})\}$

if the bowl occupies 45% of the image.

For background, $Bi = w_3F_3 = 0.55 * 0 = 0$ since background occupies 55% of the rest space and is color independent.

For distances between various coordinate points

$$Pi = w_4F_4 = 1.0 * \sum_{5}^{8} d_i$$

So feature $F = f(w_1F_1; w_2F_2; w_3F_3 ; w_4F_4)$

$= [0;\ 0.45 * f\{(x_{79}, y_{79}), (x_{80}, y_{80}),\ldots\ldots, (x_{82}, y_{82})\};\ 0; 1.0 * \sum_{5}^{8} d_i].$

Since this is an input image, there is no domain. Hence D = 0.

Variables are $V = (x_{79}, y_{79}), (x_{80}, y_{80}),\ldots\ldots, (x_{82}, y_{82})$.

Constraints are $C = \{5;\ (C_{79}, C_{80}, \ldots., C_{82})\}$.

As this is an input of a parallelogram, there is no emergence. E= 0.

So emergence index

EI = f(D, F,V, C, E)

$[0; [\ 0;\ 0.45 * f\{(x_{79}, y_{79}), (x_{80}, y_{80}),\ldots\ldots,(x_{82}, y_{82})\};\ 0; 1.0 * \sum_{5}^{8} d_i];$

$(x_{79}, y_{79}), (x_{80}, y_{80}),\ldots\ldots,(x_{82}, y_{82}); \{4; (C_{79}, C_{80}, \ldots., C_{82})\};\ 0]$

Step 2. Analysis of image

Begin

If end of file, stop run otherwise next sentence.

Since the image is the same one we used in section 7.3, analysis of image remains same as in step 2 of section 7.3.

End

Step 3. Find Correspondence

Begin

Compare the features F calculated for input and image, then

Color $Li \neq Lo$.

Shape $Si \neq So$

when we compare the percentage of image occupied by the object,

also

$Ni \neq No$ *number of coordinates does not match, in input it is 4 whereas in image, it is 5.*

Background $Bi \neq Bo$.

Distances between various coordinate points $Pi \neq Po$.

Hence the input and the image do not match.

End

Step 4. *Since they do not match, this image is not selected and goes to* ***step 5.***

Step 5. *We study the emergence phenomenon in the image to see whether any symmetry could be established with the input parallelogram.*

Begin

1. *Original image – a bowl.*
2. *To study the emergence, we have two options: Hypothesis-driven search and data-driven search as we discussed in section 3.4.1 Model of Emergence. We select hypothesis-driven search because the image is predefined and the particular unstructured image is to be searched to find a match of the input image.*
3. *Destruction of original image*
   a. *Since we are considering the image of a map of section 6.2.1, it has 5 coordinate points C_{69}, C_{70}, C_{71}, C_{72}, C_{73}. The image is represented by 6 infinite maximal lines which are D_1, D_2, D_3, D_4, D_5, D_6.*
   b. *We destroy the outer curve of the bowl and come up with the figure shown in 8.6. It has still 6 infinite maximal lines and inside curves.*
   c. *This image now becomes an unstructured image.*
4. *Processing of unstructured image*

   We further destroy the structure by removing inside curves and 2 infinite maximal lines D_5 and D_6.
5. *New emergent image*

   The image becomes emergent shape with infinite maximal lines D_1, D_2, D_3 and D_4 in figure 8.7.

This has been shown in section 6.2.1. We draw the analysis of image from section 6.3.1.

Number of infinite maximal lines = 4.

Number of coordinate points No = 4 and they are C_{69}, C_{70}, C_{71}, C_{72}.

Four sides are L_a, L_b, L_c, L_d.

Four intersections are I_{ad}, I_{ab}, I_{bc}, I_{cd}.

Distances between four intersections are same. In other words,

$d(I_{ad}, I_{ab}) = d(I_{ab}, I_{bc}) = d(I_{bc}, I_{cd}) = d(I_{cd}, I_{ad})$.

For color, weight-feature factor is

$Lo = w_1F_1 = (0.35 * 2, 0.45 * 3, 0.2 * 5)$

combination of blue, green and black.

For shape, $So = w_2F_2 = 0.45 * f((x_{69}, y_{69}), (x_{70}, y_{70}),\ldots,(x_{72}, y_{72}))$

if the bowl occupies 45% of the image.

For background, $Bo = w_3F_3 = (0.45 * 3, 0.1 * 5)$

combination of green and black.

For distances between various coordinate points

$$Po = w_4F_4 = 1.0 * \sum_{1}^{4} d_i$$

So feature $F = f(w_1F_1; w_2F_2; w_3F_3; w_4F_4)$

$= [(0.35 * 2, 0.45 * 3, 0.2 * 5); 0.45 * f\{(x_{69}, y_{69}), (x_{70}, y_{70}),\ldots, (x_{72}, y_{72})\};$

$(0.45 * 3, 0.1 * 5); 1.0 * \sum_{1}^{4} d_i]$

Domain path remains the same and is $D = D_1 / D_{2.2} / D_{3.4} / D_{4.7}$.

Variables are $V = (x_{69}, y_{69}), (x_{70}, y_{70}),\ldots, (x_{72}, y_{72})$.

Constraints are $C = \{4; (C_{69}, C_{70}, \ldots, C_{72})\}$.

Emergent shape is that of a parallelogram.

So emergence index

$EI = f(D, F, V, C, E)$

$= [(D_1 / D_{2.2} / D_{3.4} / D_{4.7});$

$[(0.35 * 2, 0.45 * 3, 0.2 * 5); 0.45 * f\{(x_{69}, y_{69}), (x_{70}, y_{70}),\ldots, (x_{72}, y_{72})\};$

$(0.45 * 3, 0.1 * 5); (1.0 * \sum_{1}^{4} d_i)]; (x_{69}, y_{69}), (x_{70}, y_{70}),\ldots, (x_{72}, y_{72});$

$\{5; (C_{69}, C_{70}, \ldots, C_{72})\}; f(x_{69}, y_{69}), (x_{70}, y_{70}),\ldots, (x_{72}, y_{72})]$

End

Step 6. Find Correspondence

Begin

Compare the features F calculated for input and emergent image, then

Color $Li \neq Lo$.

Shape Si = So

when we compare the percentage of image occupied by

the object,

also

Ni = No number of coordinates matches, in each case being 4.

Background Bi ≠ Bo.

Distances between various coordinate points Pi = Po.

Although color and background of the input and image do not match, the other factors match.

Number of intersections in input and image is 4 and so they match.

Now we consider the following as shown in section 6.3.3

Ordinary group intersections in each image

Number of infinite maximal lines

Corresponding equivalence

Number of intersections

Geometric constraints of infinite maximal lines

Dimensional constraints of segments

Corresponding intersections

Corresponding angles

And we find the input and emergent shape match and they are both parallelogram.

End

Step 7. *Since they match, this image is selected and go to* **step 2.**

7.6 Experiments and Results

Attrasoft Image Finder (Attrasoft, 2001) has developed an image retrieval technique where input images would be stored in various files. Also images would be kept in directory files. There is an interface screen where users can provide the file name containing the input image and also can put various parameters like focus, background, symmetry, rotation type, reduction type and so on. The images from the directory would then be retrieved based on these inputs. The images in the directory are defined containing the sample segments or translated segments, rotated segments, scaled segments, rotated and scaled segments, brighter or darker segments. The advantage of this method is it goes to some extent in bringing out semantic meanings in an image in the sense that the user can specify an input image semantically, then corresponding input image is retrieved and based on that input image, image database is searched to find symmetry. The disadvantage is it fails to consider the implicit or hidden meanings of an image (Attrasoft, 2001).

There is another image retrieval system called CIRES (Content-based Image retrieval system) based upon a combination of higher-level and lower-level vision principles. Higher-level analysis uses perceptual organization, inference and grouping principles to extract semantic information describing the structural content of an image. Lower-level analysis employs a channel energy model to describe image

texture and utilizes color histogram techniques. Gabor filters are used to extract fractional energies in various spatial-frequency channels. The system is able to serve queries ranging from scenes of purely natural objects like vegetation, trees, sky, etc. to images containing conspicuous structural objects such as buildings, towers, bridges, etc. Also it does not confine itself to selecting only similar images, but extends to include images, which bear same semantic meaning as the input image

(Computer Vision Homepage, 2001).

Next we discuss SIMPLIcity (Semantic-sensitive Integrated Matching for Picture Libraries). This uses semantic classification methods, a wavelet-based approach for feature extraction and integrated region matching based upon image segmentation. Here an image is represented by a set of regions, roughly corresponding to objects, which are characterized by color, texture, shape and location. The system classifies images into semantic categories such as textured-non-textured, graph-photograph. The categorization seems to enhance retrieval by permitting semantic searching methods and narrowing down the searching range in a database. A measure for the overall similarity between images is developed using a region-matching scheme that integrates properties of all the regions in the images. Compared with retrieval based on individual regions, the overall similarity approach reduces the adverse effect of inaccurate segmentation. The disadvantage is it fails to take due note of the semantics of a particular region and thus could give unpredictable results (Simplicity, 2001).

Now we present The Digital Library Project of University of California, Berkeley, USA, which applies statistical model that has been built from the data. The data can be queried with probabilistic consideration regarding which images have high probability of selection based on any combination of words and image features. Its advantages are (1) it provides access to large image datasets through browsing and search and (2) it uses large image collections as data for object recognition. Its disadvantage could be the slow response time and the inaccuracy involved in locating a desired image from the database as there is no way to specifically pinpoint the particular image to be retrieved (Computer Vision Meets Digital Libraries, 2003).

There is another searching method called Shape Queries Using Image Databases (SQUID) which was developed by The department of Electronic and Electrical Engineering, University of Surrey, UK. The advantage of this method is it allows users to submit shapes as query objects. Every image is first processed to recover the boundary contour. Then it measures curvatures, which are local measures of how fast a planar contour is turning. Based on measured curvature values, the system tries to establish symmetry with the objects of the image database by measuring their curvature values. The disadvantage of this method is that it is not applicable to establish symmetry based on the whole image consisting of more than one shapes where in addition to shape of each individual object, texture, color and spatial locations have to be considered (Search for similar shapes in the SQUID system: Shape Queries Using Image Databases, 2002).

In the next sections, we perform our experiments with these image retrieval systems.

We present the input images and retrieved images in appendices only for the convenience of organization of the thesis.

7.6.1 Expected Results

We have input in the form of tree, flower and leaves. This input would search the database and find out relevant records. We expect the search to select not only perfectly identical records if these exist, but also records, which are not identical but bear the same intricate meanings as the input. This will show how emergence index works.

7.6.2 Approach Without Emergence Index

We present approaches where emergence index, which we talked about earlier, is not used.

First we show image retrieval using Attrasoft..

Input

Here input is the image of a stamp as shown in figure 7.2.

Image database

Figures 7.3 and 7.4 give examples of images of various stamps in the image database, which we access to retrieve images.

Search results

Figure 7.5 shows the results after search is carried out, based on query image of stamp of figure 7.2. Four similar stamps are retrieved.

The results, after the search, contain images identical with the original image in the stamp in figure 7.2. These are the image retrieval results obtained without considering emergence index, as input and outputs are identical.

We now show image retrieval using University of California Digital Library Project.

Input

Here input is the photo of landscape-habitat as shown in figure 7.6.

Search results

Figures 7.7 and 7.8 show the results after search is carried out, based on query image of landscape of figure 7.6. 12 landscape-habitat images are retrieved.

The effectiveness of this system is that input is selected through text and based on that text relevant images are selected. But it does not consider hidden or implicit meanings of the image.

We now show image retrieval using SIMPLIcity.

Input

Here input is the image of a red flower as shown in figure 7.10.

Search results

Figure 7.11 shows the results after search is carried out, based on query image of the flower of figure 7.10. 24 images are retrieved. But we notice the results show at least 5 images which are in no way related to the input image of the flower. These unpredictable results make rooms for further improvement of this system by the developers.

Next we show image retrieval using SQUID, UK system.

Input

Here input is the shape of a fish as shown in figure 7.12 (top left hand).

Search results

Figure 7.13 shows the results after search is carried out, based on query image of fish from figure 7.12. Images of fish are retrieved.

The effectiveness of this system is, as we mentioned earlier, that user can input the shape of an object. But this also does not consider hidden or implicit meanings of the image.

7.6.3 Approach With Emergence Index

Now we experiment using CIRES.

Input

We select query image from a file through a menu as shown in figure 7.14. Figure 7.14 contains the interface screen that is used to retrieve a query image. The query image is shown in figure 7.15 on the upper side where there is only one image. This is the image of a section of a flower tree with different colors of flowers and a colorful butterfly.

Search results

The retrieved images, after the search is carried out, are shown in lower side of figure 7.15. There are 20 images selected.

7.6.4 Analysis of Experimental Results

We presented the results of five approaches above, namely, retrieval with and without emergence index. In section 7.5.1, all retrieved images in figure 7.5 were similar to query image of a stamp in figure 7.2. This is the current trend in content-based image retrieval where images from the image database are retrieved only when there is a reasonable symmetry with the query image. If there is no reasonable symmetry exists between query image and a particular image of the database, that image would not be selected for retrieval. This is the problem of the existing trend in image retrieval. Same is true for retrieved images in figures 7.7, 7.8, 7.11, 7.13.

But in section 7.5.2, there are 20 retrieved images in lower side of figure 7.15. But none of them is exactly same as the query image in the upper side of figure 7.15. But although retrieved images are not identical with the query image, still they are selected based on texture and color. This speaks for the emergence phenomenon because these images, although not identical, are same in the sense that they contain leaves, flowers like the way input image contains. So they bear the same meaning as the query image which is flower tree and hence selected. These are retrieval based on emergence index. So in this case, while obtaining symmetry, we not only consider the explicit features of the images and the query image, we consider the implicit meanings also. So even when no explicit symmetry could be established between query image and a particular image of the database, still that image could be selected because of identical implicit meanings between query image and a particular image of the database.

Since while considering retrieval based on symmetry, considering the implicit meanings of query image and image of the database and retrieving images based on these inner meanings is more accurate, retrieval considering emergence gives better results than retrieval without considering emergence.

7.6.5 Comparison of Expected and Experimental Results

As we mentioned in the last section, there are 20 retrieved images in lower side of figure 7.15. But none of them is exactly same as the query image in the upper side of figure 7.15. But although retrieved images are not identical with the query image, still they are selected based on texture and color. This explains the emergence phenomenon because these images, although not identical, are same in the sense that they contain leaves, flowers like the way input image contains. So they bear the same meaning as the query image which is flower tree and hence selected.

Hence the expected and experimental results tally.

4.1 Stamp Recognition
4.1.1 Translation Symmetry
4.1.2 Scaling Symmetry
4.1.3 Rotation Symmetry
4.1.4 Rotation and Scaling Symmetry

4.1 Stamp Recognition

The story line is:

1. The stamp: A stamp collector wants this stamp. He is about to search a stamp database. A keyword search is just impossible, in this case, because the stamp images are not organized.

2. The images: This database contains the stamp images. We can search this database using our photograph(s) or sketch(s) of the stamp.

3. Search results: This is the list of images, which have matched up with the sketch(s), or photo(s) of the stamp.

Now let us go through this process:

1. The stamp: A stamp collector wants this stamp. He is about to search a stamp database. A keyword search is just impossible in this case.

Figure 4-1. Key images.

056tc3.jpg	22k
057tc3.jpg	52k
059_256.jpg	19k
063_can2.jpg	32k
063p256.jpg	40k
064b_256.jpg	23k

Figure 7.2 Input of Attrasoft

niagra_falls150.jpg 25k
niagra_falls200.jpg 62k
niagra_falls50.jpg 6k
niagra_falls60.jpg 7k
niagra_falls70.jpg 9k
niagra_falls80.jpg 11k
niagra_falls90.jpg 13k
niagra_fallsa.jpg 11k
niagra_fallsa1.jpg 12k
niagra_fallsa2.jpg 12k
niagra_fallsa3.jpg 12k
niagra_fallsb.jpg 10k
niagra_fallsc.jpg 11k
niagra_fallsd.jpg 11k

The first a few images are given below:

Figure 7.3 Images of Attrasoft

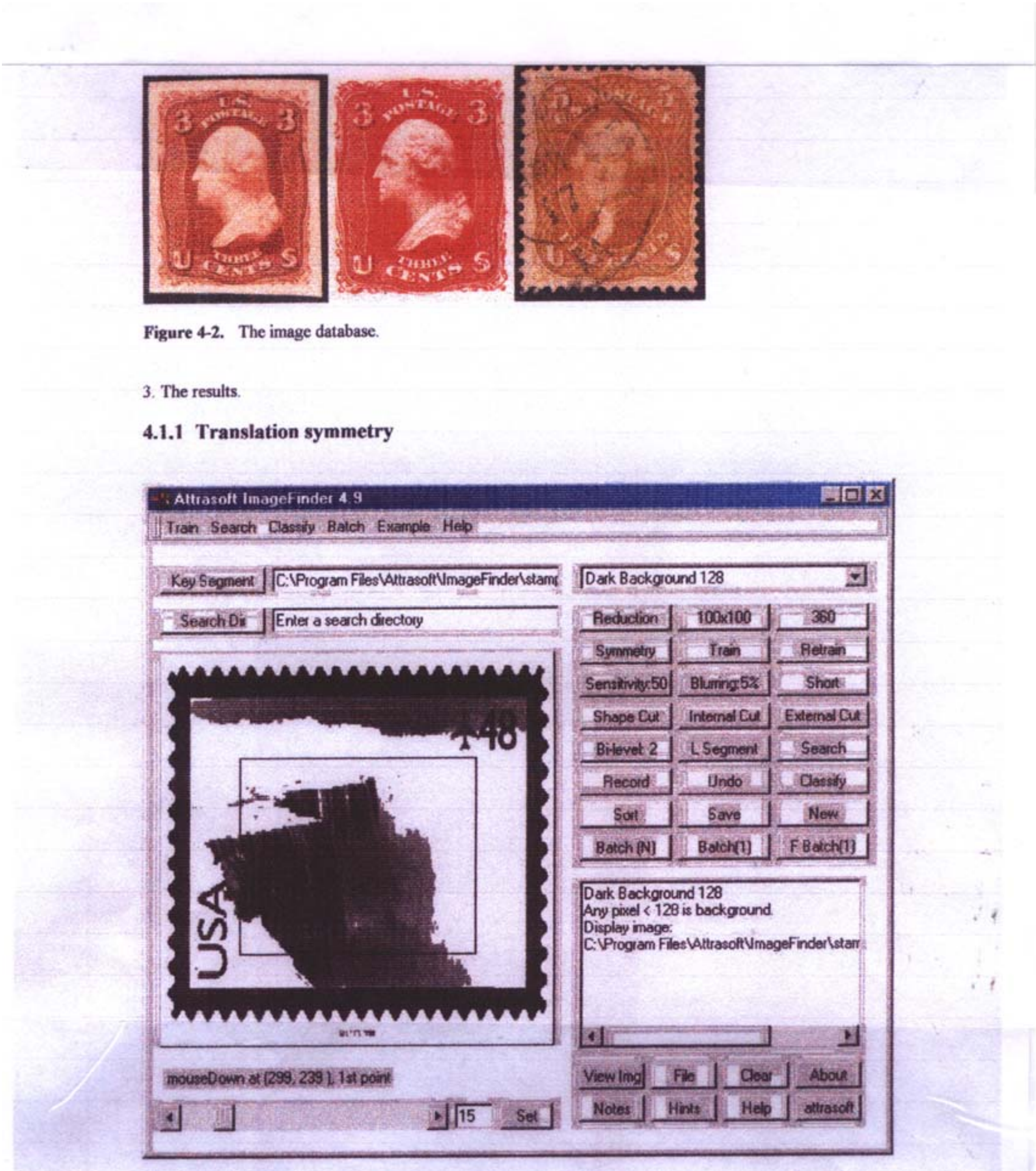


Figure 7.4 Images of Attrasoft

Figure 7.5 Outputs of Attrasoft

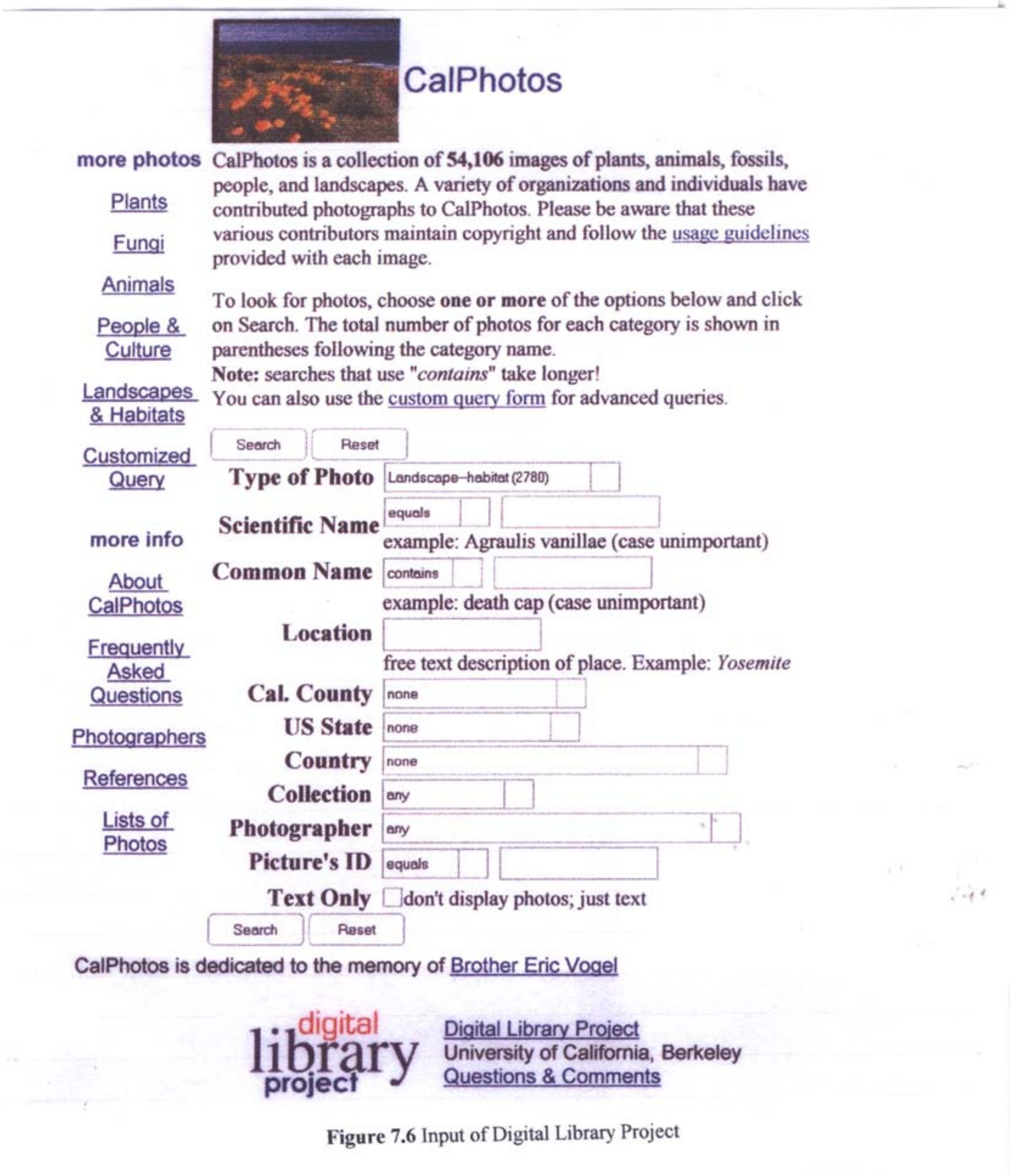


Figure 7.6 Input of Digital Library Project

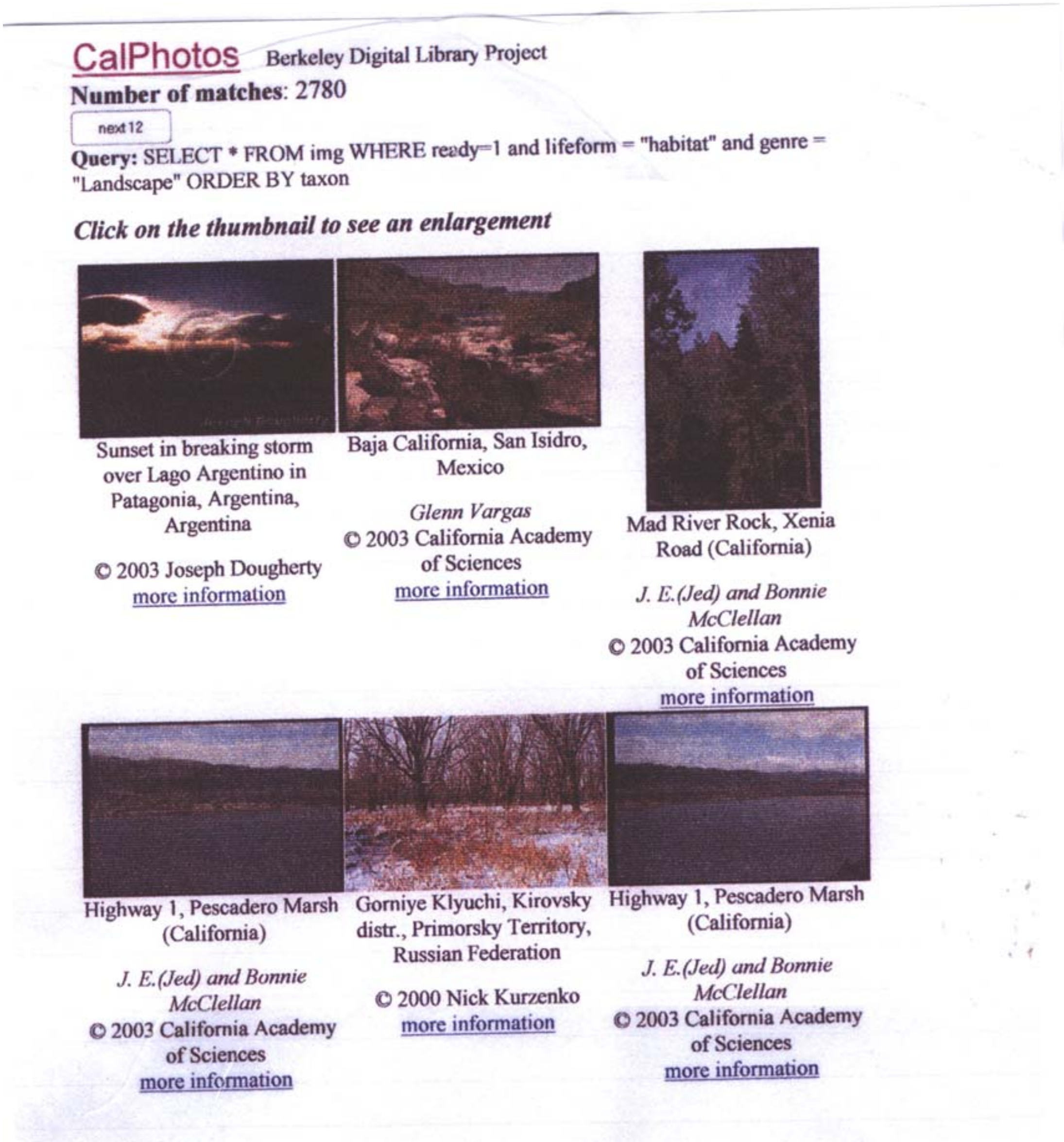


Figure 7.7 Outputs of Digital Library Project

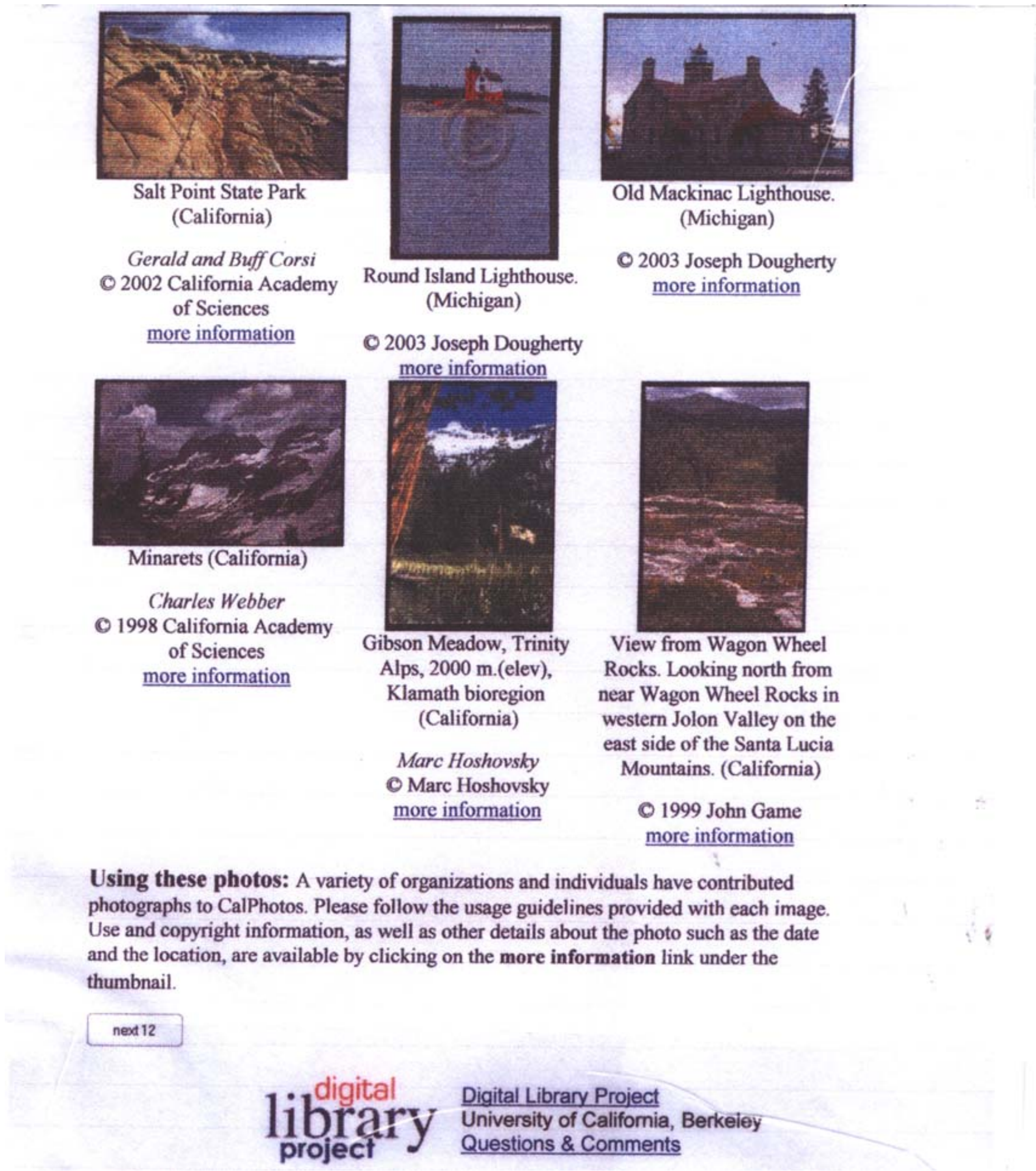


Figure 7.8 Outputs of Digital Library Project

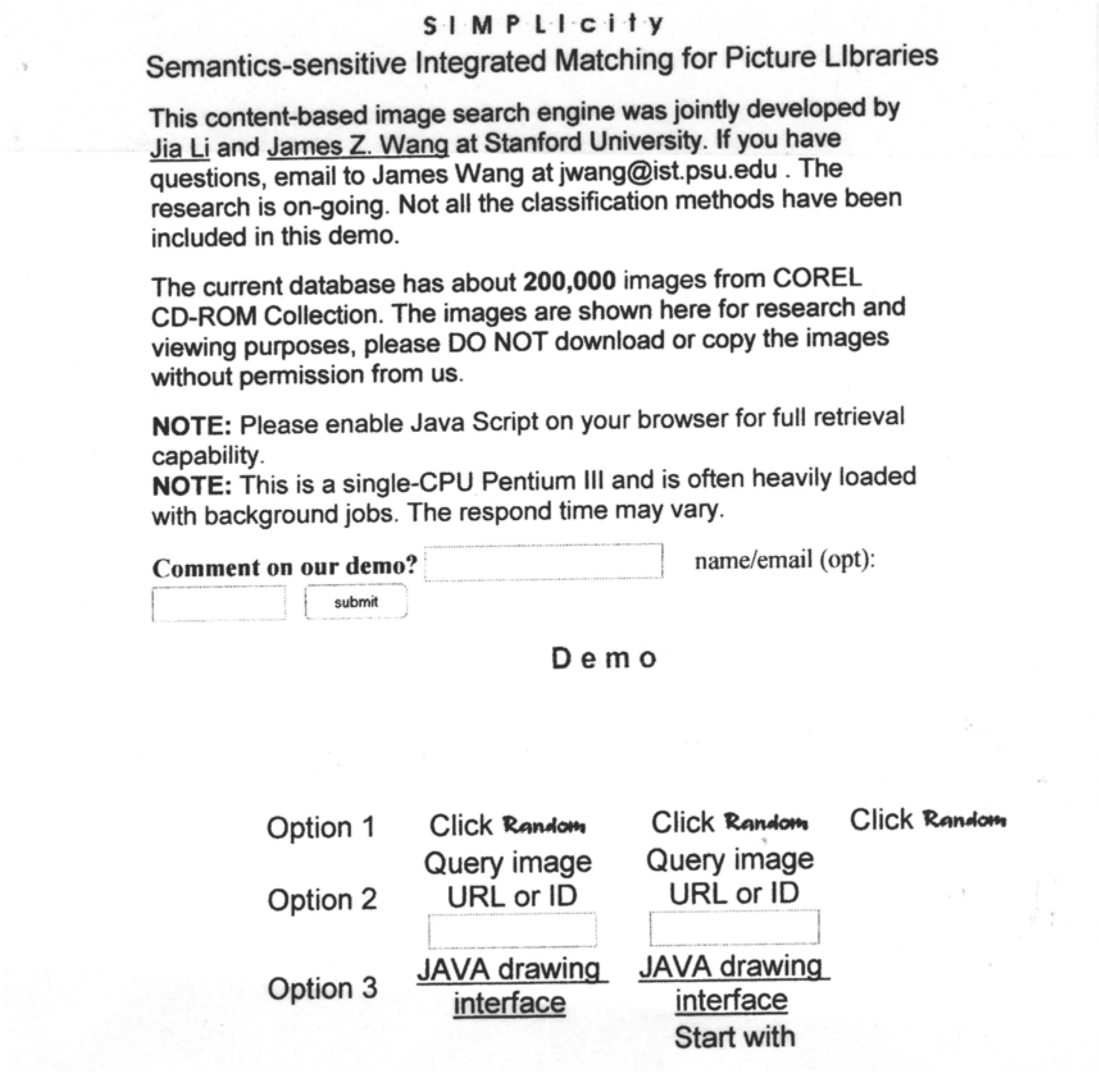


Figure 7.9 Interface of SIMPLIcity

Figure 7.10 Input of SIMPLIcity

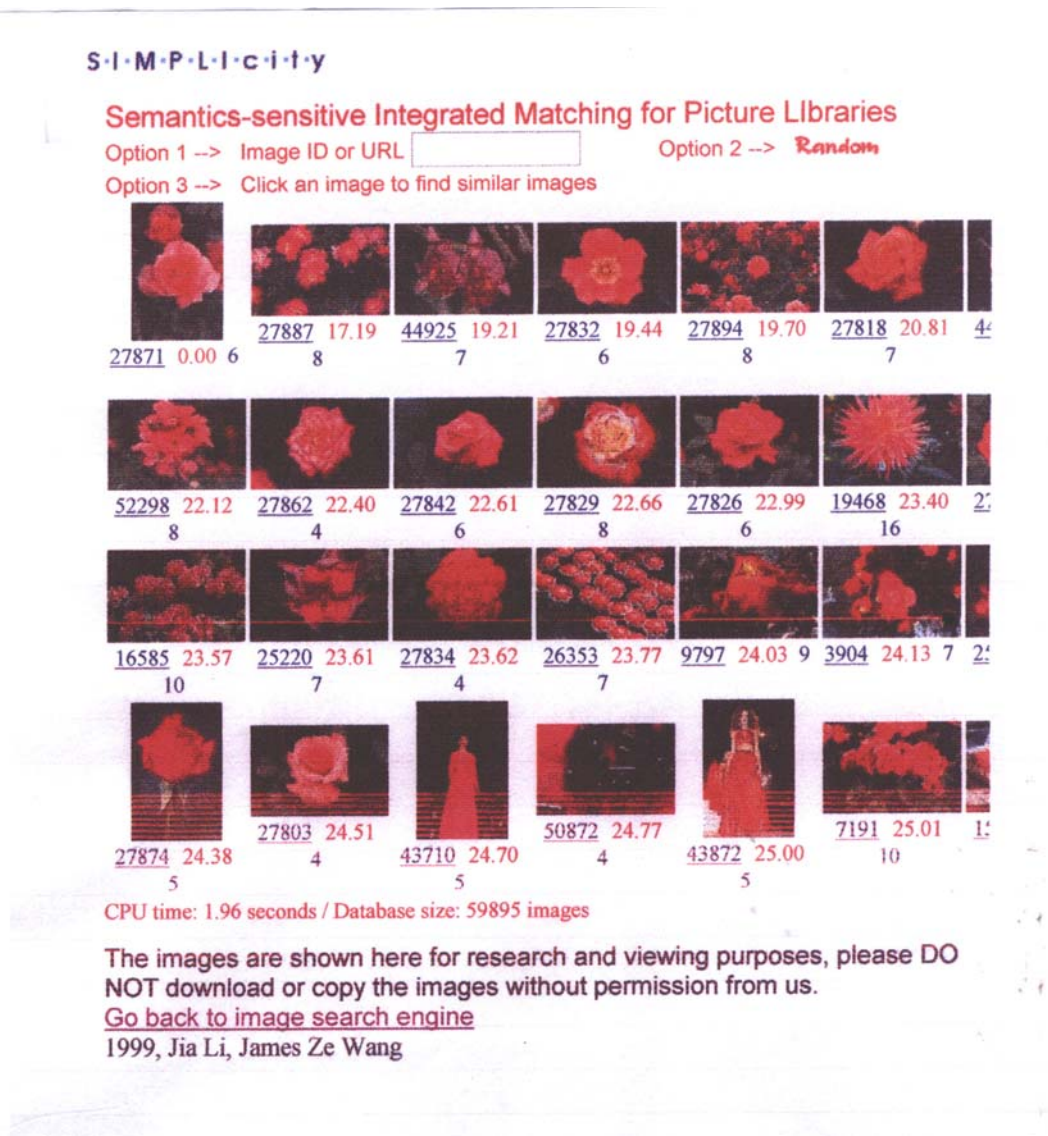


Figure 7.11 Outputs of SIMPLIcity

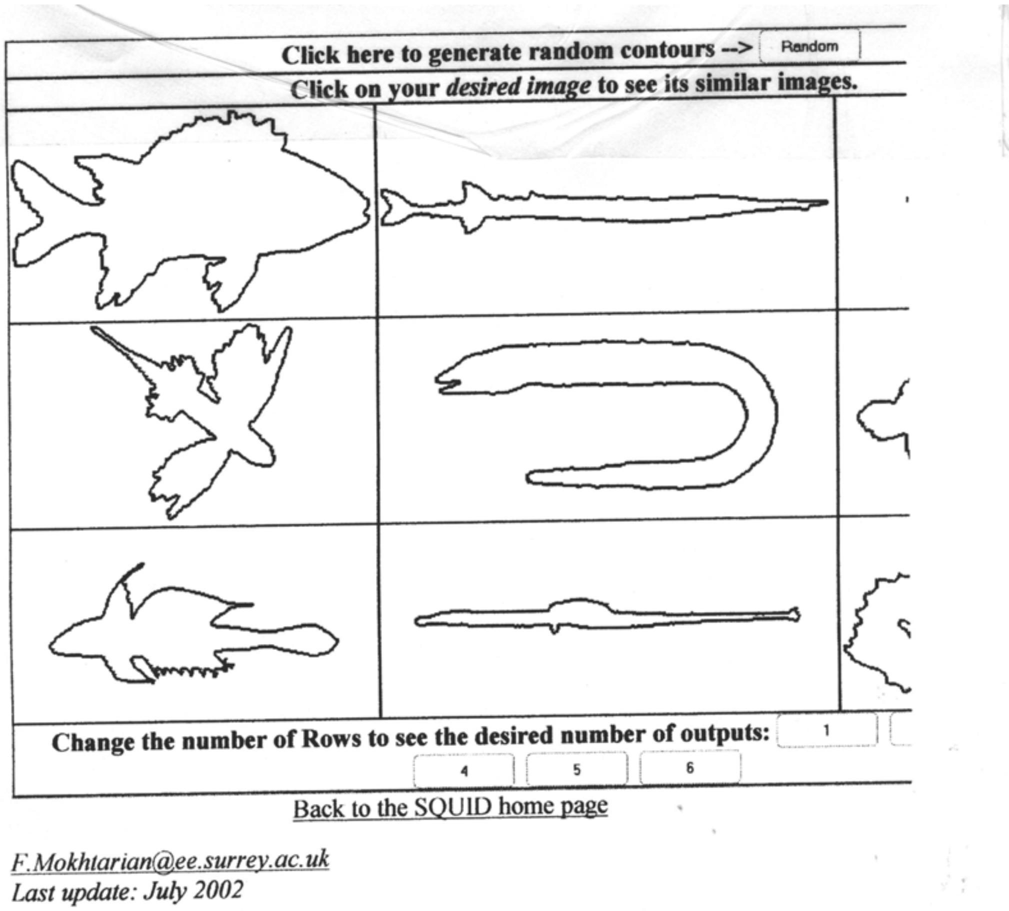


Figure 7.12 Input of SQUID

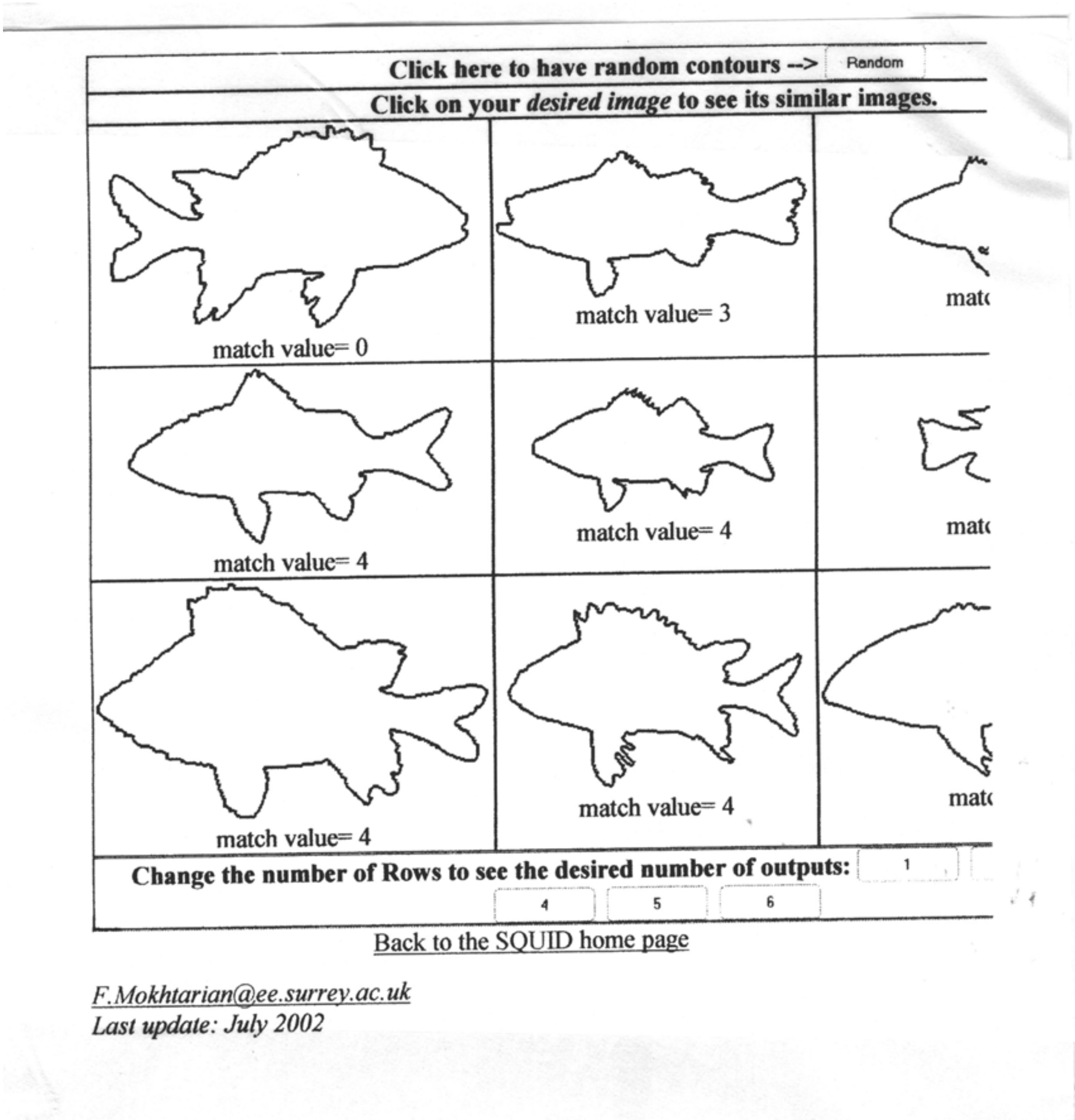


Figure 7.13 Outputs of SQUID

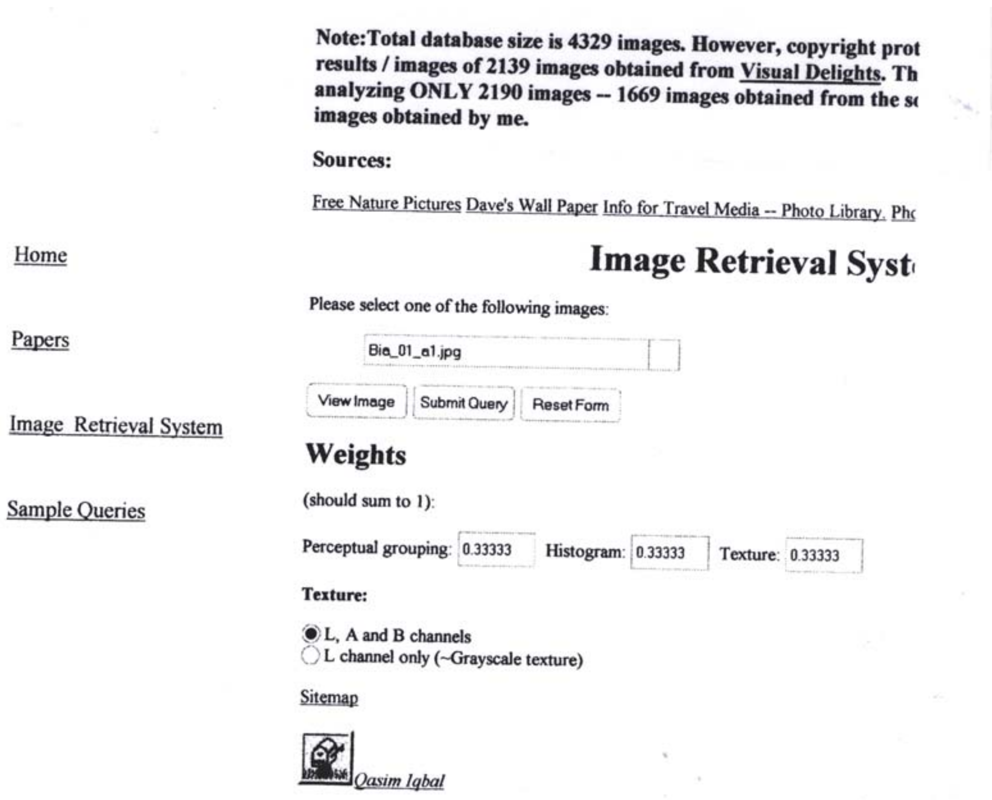


Figure 7.14 Interface of CIRES

Figure 7.15 Outputs of CIRES

7.7 Conclusions

7.7.1 Summary

In the thesis, we started with providing the general backgound of content-based image retrieval. We then defined the problems of the research done so far in this field. After this, we described the aims, scope and limitations of the research we are undertaking and mentioned the plan of the thesis. We have discussed concepts, definition, structure and construction of emergence index with examples. Approach to work on the problem of content-based image retrieval with emergence index has been described. We have presented the theory of semantic representation of images. We then covered calculation of emergence index and accessing multimedia databases with emergence index. Then we discussed how to find symmetry for a three dimensional image. How to apply the concepts of emergence index in geographic location has also been discussed. Algorithm for accessing database with emergence index is presented. We covered implementation of the concepts considering the global view of the image with experimental results. We have also shown how emergence index could relate effectively in the direction of content-based image retrieval in the context of multimedia databases. We have shown how emergence can give rise to altogether different meaning of an image and hence a different search outcome in image retrieval than when emergence is not considered. This could help us explain and interpret images with more precision as we have shown in the example of geographic location in 6.2. In that example we have shown how a particular geographic location, which could be the shape of a bowl in a map, could be located much more easily when emergence is considered than when it is not considered from a database containing huge volumes of images.

7.7.2 Possible Future Research

- More research works need to be done to apply this concept in practical problems of fingerprint analysis, understanding cloud behavior in weather forecasting or in medical fields and other scientific and engineering fields as well. That way we should be able to find more meanings and hidden patterns of those images which not only would enable us to define them with more precision but also should establish more appropriate symmetry with other images when needed.

- In implementation, we considered the retrieval of image from global point of view. Here the whole image is considered at a time instead of its various objects and sections that could be present in it. But there is scope to carry out further research for efficient and meaningful image segmentation into various objects and sections. Then we could establish symmetry of the input image with individual object and section of the image of the database. This will be able to provide more efficiency and also better result than when the whole image is considered globally.
- We can carry out research to find the semantic or high-level meanings of objects from low-level features like color, shape, texture or spatial locations. At the moment it is proving very difficult to achieve with existing technology for the simple reason that there exists a vast gap between human and computer perceptions. For the computer to sense what could be the object or what could be the meaning of a picture from low-level features like color, shape, texture is still very difficult. It requires the application of sophisticated neural technology to make computer perceive the object or the scenerio. Lot more research efforts are needed in this field to bring the technology to a matured level.
- After proper image segmentation is done and links between low-level features and high-level features are established, we can further implement our theories of emergence index based on those segmented objects and their semantic meanings using the Model of Emergence presented in section 3.4.1 and the Model of Emergence Index presented in section 1.6.2. As we pointed out, application of the emergence concepts for individual objects and sections within an image would lend more precise result than when whole image is considered at a time.

References

References

1. Aslandogan, Y.A., Yu, C.T. (1999). Techniques and systems for image and video retrieval, *IEEE Transactions on Knowledge and Data Engineering*, 11(1), 1999, pp. 56-63

2. Attrasoft (2001). *Attrasoft*, P.O. Box 13051, Savannah, GA. 31406, USA, May 2001

3. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B. (1990). The R* -tree: An efficient and robust access method for points and rectangles, *ACM International Conference on data Management (SIGMOD)*, 1990, pp. 322-331

4. Berchtold, S., Keim, D.A., Kriegal, H.P. (1996). The X-tree: An index structure for high-dimensional data, *International Conference on Very Large Databases (VLDB)*, Bombay, India, 1996, pp. 28-39

5. Bjarnestam, A. (1998). Description of an image retrieval system, *The Challenge of Image Retrieval Research Workshop*, Newcastle open Tyne, February 5, 1998

6. Boujemma, N. and others. (2000). *IMEDA Project*, INRIA, Feb. 2000

7. Burkhard, W.A., Keller, R.M. (1973). Some approaches to best-match file searching, *Communications of the ACM*, 16(4), 1973, pp. 230-236

8. Cariani, P. (1992). Emergence and Artificial life, in Langton, C., Taylor, C., Farmer, J.D. and Rasmussen, S. (eds), *Artificial Life II*, Addision-Wesley, Reading, pp. 775-797

9. Chakrabarti, K. and Mehrotra, S. (1999). The hybrid tree: An index structure for high dimensional feature spaces, *International Conference on Data Engineering (ICDE)*, Sydney, Australia, 1999, pp. 440-447

10. Chang, C., Chan, C., Hsiao, J. (2002). A Color Image Retrieval Method Based on Local Histogram, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 2001, pp. 831-836

11. Chang, S. K, Yan, C. W., Dimitroff, D. C., Arndt, T. (1988). An intelligent image database system, *IEEE Transactions on Software Engineering*, 14(5), 1988

12. Chen, Z., Liu, W., Zhang, F., Li, M., Zhang, H. J. (2001). Web mining for web image retrieval, *Journal of the American Society for Information Science and Technology*, 52(10), August 2001, pp. 831-839

13. Claccia, P., Patella, M., Zezula, P. (1997). M-tree: An efficient access method for similarity search in metric spaces, *International Conference on Very Large Databases (VLDB),* Athens, Greece, ed. M. Jarke, 1997, pp. 426-435

14. Comer, D. (1979). The ubiquitous B tree, *ACM Computing Surveys*, 11(20), 1979, pp. 121-137

15. Computer Vision Homepage (2001). *Computer Vision Online Demos*

16. Computer Vision Meets Digital Libraries (2003). *Digital Library Project*, University of California, Berkeley, USA

17. Constantopoulos, P., Drakopoulos, J., Yeorgaroudakis, Y. (1991). Retrieval of Multimedia Documents by Pictorial Content: A Prototype System, *International Conference on Multimedia Information Systems '91,* The Institute of Systems Science, National University of Singapore, pp. 35-48

18. Eakins, J. P. and Graham, M. E. (1999). Content-based image Retrieval: A report to the JISC Technology Application Program, Institute for Image Data Research, University of Northumbria at Newcastle, U.K., January 1999

19. Esperanca, C., Samet, H. (1997). A differential code for shape representation in image database application, *International Conference on Image Processing*, 1997

20. Forrest, S. (ed.) (1991). *Emergent Computation*, Elsevier, New York

21. Forsyth, D. A and others. (1996). Finding pictures of objects in large collections of images, *Digital Image Access and Retrieval: 1996 Clinic on Library Applications of Data Processing* (Heidon, P. B. and Sandore, B., eds.), pp. 118-139, Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign

22. Forsyth, D. and others. (1996). Finding pictures of objects in large collections of images, *Report of the NSF/ARPA Workshop on 3D Object Representation for Computer Vision*, 1996, pp. 335

23. Fournier, F. and Cord, M. (2002). A flexible search-by-similarity algorithm for content-based image retrieval, *Proceedings of 6th Joint Conference on Information Sciences*, March 8-13, 2002, Research Triangle Park, North Carolina, USA, pp. 672-675

24. Gaede, V., Gnther, O. (1998). Multidimensional access methods, *ACM Computing Surveys*, 30(2), 1998, pp. 170-231

25. Gagandakis, G., Rosin, P. I. (2001). Shape measure for image retrieval, *International Conference on Image Processing*, Thessaloniki, Greece, 2001, pp.757-760

26. Garcia-Molina, H., Ulhman, J. D., Widow, J. (2000). *Database System Implementation*, Prentice-Hall, 2000

27. Gatzer, U., Baike, W. T., Kiessing, W. (2000). Optimizing multi-feature queries for image databases, *International Conference on Very Large Databases (VLDB)*, Cairo, Egypt, 2000, pp. 419-428

28. Gero, J. S. (1992b). Shape emergence and symbolic reasoning using maximal lines.*Unpublished Notes, Design computing Unit,* Department of Architectural and Design Science, University of Sydney, Sydney

29. Gero, J. S. (Year Unknown). Visual emergence in design collaboration, Key Center of Design Computing, University of Sydney

30. Gero, J. S. and Maher, M. L. (1994). Computational Support for Emergence in Design, *Information Technology in Design Conference,* Moscow, September 1994

31. Gero, J. S. and Yan, M. (1994). Shape Emergence by symbolic reasoning.", *Environment and Planning B:Planning and Design,* 21:191-212

32. Gimel'farb, G. I., Jain, A. K. (1996). On retrieving textured images from an image database, *Pattern Recognition*, 29(90), 1996, pp. 1461-1483

33. Gudivada, V. N. and Raghavan, V. V. (1995). Content-Based Image Retrieval Systems. *IEEE*, September 1995

34. Guttman, A. (1984). R-tree : A dynamic index structure for spatial searching, *ACM International Conference on Data Management (SIGMOD)*, Boston, MA, 1984, pp. 47-57

35. Healey, G. and Jain, A. (1996). Retrieving Multispectral Satellite Images Using Physics-Based Invariant Representations, *IEEE Transactions on pattern Analysis and Machine Intelligence,* Vol. 18, No. 8, August 1996

36. Hsieh, J., Huang, L. W., Huang, Y. (2002). Multiple-Person Tracking System for Content Analysis, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China,* October 24-26, 2001, pp. 897-902

37. Idrissi, K., Ricard, J., Anwander, A., Baskurt, A. (2001). An image Retrieval System Based on Local and Global color Descriptors, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 2001, pp. 55-62

38. Jagadish, H. V. (1990). Spatial Search with Polyhedra, *IEEE,* pp. 311-319

39. Jain, R. (2003). Semantics in Multimedia Systems, *The 9th International Conference on Multi-media Modeling*, January 7-10, 2003, Tamkang University, Taiwan

40. Jing, E., Li, M., Zhang, H. J., Zhang, B. (2002). An effective region-based image retrieval framework, *Proceedings of ACM Multimedia 2002*, Juan-les-Pins, France, December 1-6, 2002

41. Jlayi, C., Tan, T., Mullhem, P., Kankanhall, M. (2003). An Improved Method for Image Retrieval Using Speech Annotation, *The 9th International Conference on Multi-media Modeling*, January 7-10, 2003, Tamkang University, Taiwan

42. Joubert, G. R. and Kao, O. (2002). Efficient Dynamic Image Retrieval Using the A Trous Wavelet Transformation, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 2001, pp. 343-350

43. Jun, H. J. (1994). Emergence of Shape Semantics in CAD System, *Ph.D. thesis proposal,* Design Computing Unit, Department of Architectural and Design Science, University of Sydney, Sydney

44. Kamel, I., Paloutsos, C. (1994). Hilbert R-tree: An improved R-tree using fractals, *20th International Conference on Very Large Data Bases (VLDB),* Santiago del Chic, eds. J.B.Bocca, M. Jarke and C. Zaniolo, 1994, pp. 500-509

45. Ko, B., Lee, H. S., Byun, H. (2000). Image retrieval using flexible image subblocks, *ACM Symposium on Applied Computing,*Vol. 2, 2000, pp. 574-578

46. Langton, G .L. (1989). *Artificial Life,* Addision-Wesley, Reading

47. Lin, K. L. D., Jagadish, H. V., Paloutsos, C. (1994). The TV-tree: An index structure for high-dimensional data, *VLDB Journal*, 3(4), 1994, pp. 517-542

48. Lu, Y. and others. (2000). A unified framework for semantics and feature based relevance feedback in image retrieval systems, *Proceedings of ACM MM2000,* 2000

49. Ma, W. Y., Manjunath, B. S., Luo, Y., Deng, Y., Sun, X. (Year Unknown). NETRA: A Content-Based Image Retrieval System, Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA

50. Mahmoudi, S. and Daoudi, M. (2002). 3D Models Retrieval and Indexing, *Proceedings of 6th Joint Conference on Information Sciences,* March 8-13, 2002, Research Triangle Park, North Carolina, USA, pp. 972-975

51. Manjunath, B. S. and Ma, W. Y. (1996). Texture Features for Browsing and Retrieval of Image Data, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 18, No. 8, August 1996

52. Marques, O. and Furht, B. (2002). Content-based Visual Information Retrieval, *Distributed Multimedia Databases: Techniques and applications*, Idea Group Publishing, Hershey, PA 17033-1117, USA, pp. 37-57

53. Mehrotra and Gary (1995). Similar-Shape Retrieval In Shape Data Management, *IEEE,* September 1995

54. Mitchell, M. and Hofstadter, D. (1994). The Copycat Project: A Model of Mental Fluidity and Analogy-making, *Fluid Concepts & Analogies:Computer Models of the Fundamental Mechanisms of Thought*, BasicBooks, New York

55. Pass, G., Zabih, R., Miller, J. (1996). Comparing images using color coherence vector, *ACM Multimedia*, Boston, MA, 1996, pp. 65-73

56. Pentland, A., Picard, R., Sclaroff, S. (1996). Content-based manipulation of image databases, *International Journal of Computer Vision*, 18(3), 1996, pp. 233-254

57. Qiu, G. and Sudirman, S. (2002). Retrieving Landscape Images Using Scene Structural Matrix, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia*, *Beijing*, *China*, October 24-26, 2001, pp. 921-926

58. Ratha, N K., Karu, K., Chen, S., Jain, A K. (1996). A Real-Time Matching System for Large Fingerprint Databases, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, August 1996

59. Robinson, J. T. (1981). The K-D-B-tree: A search structure for large multidimensional dynamic indexes, *ACM International Conference on Management of Data (SIGMOD)*, Ed. Y.E.Lien, 1981, pp. 10-18

60. Safar, M., Shahabi, C., Sun, X. (2000). Image retrieval by shape: A comparative study, *Proceedings of IEEE International Conference on Multimedia and Exposition (ICME)*, USA, 2000

61. Samet, H. (1995). Spatial data structures in modern database systems: The object model, interoperability and beyond, Addison-Wesley/ACM Press, 1995, pp. 361-385

62. Samet, H. (1996). MARCO: MAp Retrieval by COntent, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, August 1996

63. Sandsmark, F. (1997). Multimedia Databases bring Order to Images, Nov. 1997

64. Santos, R. F., Traina, A. J. M., Traina Jr, C., Faloutsos, C. (2001). Similarity search without tears: The OMNI family of all purpose access methods, *International Conference on Data Engineering (ICDE)*, Heidelberg, Germany, 2001, pp. 623-630

65. Seaborn, M. A. (1997). Image Retrieval Systems, November 1997

66. Search for similar shapes in the SQUID system: Shape Queries Using Image Databases (2002). *SQUID Demo*

67. Sellis, T. K., Roussopoulos, N., Faloutsos, C. (1987). The R+tree: A dynamic index for multi-dimensional objects, *International Conference on Very Large Databases (VLDB)*, Brighton, England, 1987, pp. 507-518

68. Shahabi, C., Safar, M. (1999). Efficient retrieval and spatial querying of 2D objects, *IEEE International Conference on Multimedia Computing and Systems (ICMCS99)*, Florence, Italy, 1999, pp. 611-617

69. Shapiro, L.G. and others (2000). Efficient Content-Based Image Retrieval, Department of Computer Science and Engineering, University of Washington, Feb. 2000

70. Sheikholeslami, G., Zhang, A. (1997). An approach to clustering large visual databases using wavelet transform, *Proceedings of the IEEE International Conference on Image Processing*, 1994, pp. 407-411

71. Shneier, M. and Abde-Mottaleb, M. (1996). Exploiting the JPEG Compression Scheme for Image Retrieval, *IEEE Transactions on pattern Analysis and machine Intelligence*, Vol. 18, No. 8, August 1996

72. Simplicity (2001). *SIMPLIcity Online Demos*

73. Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., Jain, R. (2000). Content-based image retrieval at the end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 2000

74. Smith, J. R., Chang, S. (1996). Transform features for texture classification and discrimination in large image databases, *Proceedings of the IEEE International Conference in Image processing*, 1994, pp. 407-411

75. Smith, J. R., Chang, S. F. (1996). VisualSeek: a fully automated content-based image query system, *Proceedings of ACM Multimedia 96*, Boston, MA, 1996, pp. 87-98

76. Song, Y. (2002). Monotonic tree and its application to multimedia information retrieval, *Ph.D. dissertation,* Department of Computer Science and Engineering, State University of New York at Buffalo, 2002

77. Song, Y., Zhang, A. (2002). Monotonic tree, *The 10th International Conference on Discrete Geometry for Computer Imagery*, Bordeaux, France, April 3-5, 2002

78. Song, Y., Wang, W., Zhang, A. (2003). Automatic Annotation and Retrieval of Images, *World Wide Web Internet and Web Information Systems*, 6, 2003, pp. 209-231

79. Srisuk, S. and Kurutach, W. (2002). An Efficient Algorithm for Face Detection in Color Images, *Proceedings of 6th Joint Conference on Information Sciences,* March 8-13, 2002, Research Triangle Park, North Carolina, USA, pp. 688-691

80. Su, Z., Zhang, H. J., Ma, S. (2001). Relevant feedback using a Bayesian classifier in content-based image retrieval, *SPIE Electronic Imaging 2001,* San Jose, CA. January 2001

81. Su, Z., Li, S., Zhang, H.J. (2001). Extraction of feature subspaces for content-based retrieval using relevance feedback, *ACM Multimedia 2001*, Ottawa, Canada, 2001

82. Sural, S., Qian, G., Pramanik, S. (2002). A Histogram with Perceptually Smooth Color Transition for Image Retrieval, *Proceedings of 6th Joint Conference on Information Sciences*, March 8-13, 2002, Research Triangle Park, North Carolina, USA, pp.664-667

83. Swain, M., Ballard, D. (1991). Color indexing, *International Journal of Computer Vision*, 7(1), 1991, pp. 11-32

84. Swiderski, Z. (1998). Multimedia Information Systems: Multimedia Databases, February 13, 1998

85. Tao, Y., Grosky, W. (1999). Delaunay triangulation for image object indexing: A novel method for shape representation, *Proceedings of the Seventh SPIE Symposium on Storage and Retrieval for Image and Video Databases,* San Jose, CA, 1999, pp. 631-942

86. Tomita, F., Saburo, T. (1990). *Computer Analysis of Visual Textures*, Kluwer, 1990

87. Torratha, A., Oliva, A. (1999). Semantic organization of scenes using discriminant structural templates, *International Conference on Computer Vision (ICCV99)*, 1999, pp. 1253-1258

88. Traina, A. J. M., Traina Jr, C., Bueno, J. M., Chino, F. J. T. (2003). Efficient Content-based Image Retrieval through Metric Histograms, *World Wide Web Internet and Web Information Systems*, 6, 2003, pp. 157-185

89. Traina Jr, C., Traina, A. J. M., Seeger, B., Faloutsos, C. (2000). Slim-trees: High performance metric trees minimizing overlap between nodes, *International Conference on Extending Database Technology*, eds. C. Zaniolo, P. C. Lockemann, M. H. Scholl and T. Grust, Lecture Notes in Computer Science, Vol 1777, Konstanz, Germany, 2000, pp. 51-65

90. Tuytelaars, T., Gool, L. J. V. (1999). Content-based image retrieval based on local affinely invariant regions, *Third International Conference on Visual Information and Information Systems – VISUAL'99*, eds. D. P. Huijsmans and A. W. M. Smeulders, Lecture Notes in Computer Science, VOL. 1614, Amsterdam, The Netherlands, 1999, pp. 493-500

91. Vailaya, A., Figuciredo, A. T., Jain, A. K., Zhang, H. J. (2001). Image classification for content-based indexing, *IEEE Transactions on Image Processing*, 10(1), 2001, pp. 117-130

92. Xu, Y. and Zhang, Y. (2002). Association Feedback: A Novel Tool for Feature Elements Based Image Retrieval, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 2001, pp. 506-513

93. Wang, Y. (2002). Image Indexing and Similarity Retrieval Based on Spatial Relation Model, *Proceedings of 6th Joint Conference on Information Sciences*, March 8-13, 2002, Research Triangle Park, North Carolina, USA, pp. 980-983

94. Wu, P., Manjunath, B. S., Chandrasekaran, S. (2002). An Adaptive Index Structure for High-Dimensional Similarity Search, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 2001, pp.71-77

95. Yang, J., Li, Y., Zhuang, Y. (2003). Modeling Data and User Characteristics by Peer Indexing in Content-based Image Retrieval, *The 9th International Conference on Multi-media Modeling*, January 7-10, 2003, Tamkang University, Taiwan

96. Yoshitaka, A., Kishida, S., Hirakawa, M. (1994). Knowledge-Assisted Content-Based Retrieval for Multimedia Databases, *IEEE Multimedi,*Winter 1994, pp.12-21

97. Yu, H. and Grimson, E. L. (2002). Combining Configurational and Statistical Approaches in Image Retrieval, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 2001, pp. 293-300

98. Zhao, R., Grosky, W. I. (2002). Bridging the Semantic Gap in Image Retrieval, *Distributed Multimedia Databases: Techniques and applications*, Idea Group Publishing, Hershey, PA 17033-1117, USA, pp. 14-36

99. Zhao, T., Tang, H. L., Ip, H. H. S., Qi, F. (2002). Content-Based Trademark Recognition and Retrieval Based on Discrete Synergetic Neural Network, *Distributed Multimedia Databases: Techniques and applications*, Idea Group Publishing, Hershey, PA 17033-1117, USA, pp. 58-72

100. Zhang, D. and Lu, G. (2002). A Comparison of Shape Retrieval Using Fourier Descriptors and Short-Time Fourier Descriptors, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 2001, pp. 855-860

101. Zhang, H., Chen, Z., Li, M., Suu, Z. (2003). Relevance Feedback and Learning in Content-Based Image Search, *World Wide Web Internet and Web Information Systems*, 2003, pp. 131-155

102. Zhou, P., Feng, J. F., Shi, Q. Y. (2001). Texture feature based on local Fourier transform, *International Conference on Image Processing*, Vol. 2, Thessaloniki, Greece, 2001, pp. 610-613

103. Zhou, X. S., Huang, T. S. (2002). Relevance Feedback in Content-Based Image Retrieval: Some Recent Advances, *Proceedings of 6th Joint Conference on Information Sciences*, March 8-13, 2002, Research Triangle Park, North Carolina, USA, pp. 15-18

104. Zhu, L., Tang, C., Zhang, A. (2002). Using Keyblock Statistics to Model Image Retrieval, *Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Beijing, China*, October 24-26, 2001, pp. 522-529

105. Zhuang, L., AI, H., XU, G. (2002). Video Based Face Recognition By Support Vector Machines, *Proceedings of 6th Joint Conference on Information Sciences*, March 8-13, 2002, Research Triangle Park, North Carolina, USA, pp. 700-703