Design and Architecture of Feature Extraction based Learning Image Search Engine (FELIS)

Ms. Sukhwant Kaur
Assistant Professor,
sukhwantsagar@gmail.com

Ms. Sandhya Pati
Lecturer,
sandhyachandranp@gmail.com

Mr. Mahesh Gupta
Student,
maheshmc2@gmail.com

Ms. Sneha Kamat
Student,
skamat49@gmail.com

Mr. Abdul Mohsin
Student,
mohsink129@gmail.com

Fr. C. Rodrigues Institute of Technology, Mumbai University
Vashi, Navi Mumbai, India.

ABSTRACT

This paper presents architecture and various design related issues for FELIS which is content based image retrieval system. FELIS works by recognizing various objects present in an image and storing the user defined tags on them. Each object is recognized and stored in the database based on its color, shape and texture feature. This paper also discusses about how image segmentation can be achieved while still maintaining composition relationship among objects.

KEYWORDS

CBIR, Feature Representation, Object Composition, Learning Algorithm.

GENERAL TERMS

Computer vision, Shape feature, Color histogram, Texture feature.

1. INTRODUCTION

Image searching is a task of finding desired image from large collection of images. It is said that “A picture is worth a thousand words”. Hence, reading the true information present in an image is not an easy task. Thus, traditional approaches for image searching were based image metadata. This metadata includes image title, user defined tags and descriptions, the source from which image was taken (in case of web based image search) etc. These metadata have to be generated manually for each image; which is very time consuming and erroneous approach and can also be fooled easily.

Content based image search engines are designed to overcome these problems. Unlike traditional search engines, CBIR is based on the concept of retrieving images based on the content. The content of image is retrieved by using various morphological image processing techniques.

Most of CBIR search engines designed till now were based on the concept of applying morphological operations on an entire image [1] or dividing it into similar looking regions [2] and then applying morphological operations on each region one at a time. FELIS, on the other hand works by recognizing different objects present in an image and then extracting features from individual object. Thus, each image can be represented as a list of objects and each object can be represented as a feature vector.

This paper is organized as follows. In section 2, various existing CBIR systems and their limitations have been discussed. In section 3, scope of the FELIS has been presented. In section 4, various design issues have been discussed. In section 5, architecture of FELIS has been discussed.

2. EXISTING SYSTEMS

In this section PatSeek, Blobworld and Photobook systems have been discussed.

2.1. PatSeek [1]

This system was designed to work for US patent system and hence can process on gray scale images only. It considers only the shape feature of images. This puts limitation on the system to be used as a general purpose image search engine.

The major advantage of this system is that it uses Edge Oriented Auto-correlogram (EOAC) to represent shape feature, which is rotation invariant.

2.2. Blobworld [2]

This search engine was designed to be a web based general purpose image search engine. It divides the image into similar regions and extracts color, texture and shape feature of each region separately.

Color is represented by color histogram. Texture feature is represented by mean contrast and anisotropy. Shape feature is represented by area, eccentricity and other orientation dependent features.

The major disadvantage of this search engine was that shape feature is orientation dependent and hence same image when rotated will be treated as different image. Also, being web based search engine, it can work only on predefined set of images.

2.3. Photobook [3]

This system was designed for image archival system where images are previously stored into database and it provides a better and convenient way of browsing. The system was designed with a complex interface and hence it is not very user friendly.

An image is treated as sum of 3 orthonormal components i.e. periodicity, directionality and randomness. Shape is represented by building a finite elemental model.

3. Scope of FELIS

A. Image Pre-processing

Noise is random in nature which can affect the result of image processing. It may occur during image capturing or while transmitting an image over the network. With this idea, image pre-processing module has been added in the project. It will apply general noise removal algorithms on an image and will remove most of noise content from the image.

B. Search based on Image Content

From the given image database, images will be searched based on their contents rather than metadata. Different objects present in an image will be recognized. These objects can be considered as content of an image. Hence retrieved image will be more relevant to user's requirement.

C. Image Auto-tagging

In the image database, once an object is tagged, the tag and its description in terms of object color, texture and shape will be stored in the database. Later, if the object appears again in any other image, it will automatically be tagged. Hence same object need not be tagged again and again.

D. Different modes of Querying

Querying module helps the user to search the database and retrieve the most relevant results. While querying the system, user will be allowed to query the system by.

i. *Drawing a sketch:* In which user will draw a sketch and will give it as input. All the images containing objects which are similar to query sketch are retrieved.

ii. *Image as Query*: In which user will choose an image and will select a particular object from it. The system will search all images and retrieve those images containing similar objects.

iii. *Tag Based Searching*: User can search all the images based on a particular tag. All the images containing the tag related object will be retrieved.

E. Dynamic Organization of Image

When using the system for general usage purpose, it is possible that user can change the path of some images or changes organization completely. In such cases, instead of reloading images into database, it's better to only change corresponding path in the database. FELIS provides the facility to maintain the database consistent without user's extra effort when the user changes image organization.

F. Learning Module

Different users have different views for similarity measure of images. Hence the system provides users with the facility of giving feedback to the system. Using these feedbacks, the system will automatically modify the similarity measure criteria. This feedback will be used to train the system for user's preference and will be used to obtain best possible results accordingly. With time, if the user preference changes then the system will also learn accordingly and adapt it.

4. Design issues

Following are different issues regarding design of FELIS.

A. Domain

One of serious issues was regarding choice of domain of project i.e. designing as a desktop search engine or web-based search engine. Desktop search engines are useful to organize user's personal image collection. But with growth of internet and growth in usage of photo hosting sites future of computing is cloud or network-based system. Hence it is a serious matter of choice whether the design should be web-based or desktop based.

B. Image Composition

When Image segmentation is applied to an image, all the image objects are collected separately. For e.g. if human face is segmented we get face, eyes, nose and mouth as separate objects. But we need to consider an algorithm or some way to combine these objects together to represent a face. To do this, we need to consider composition relationship. But while doing composition some of unexpected results can be obtained.

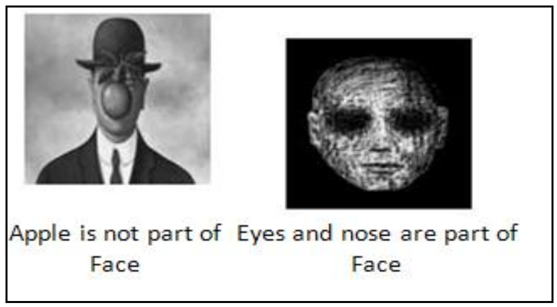

Fig 1. Challenges in Composition

For e.g. consider a picture where an apple is kept on a table and a man is standing behind it. In this case if apple and man are composed together then we will get some of unexpected result. So, whether image composition should or should not be considered is a serious issue.

C. Image Format

Since the project has been designed to work as a general purpose search engine, it needs to be able to process almost every format. Thus, which all image formats to be allowed and which all to be left is one of issue.

5. Architecture of FELIS

Based on our study work and various issues and tradeoffs, following design of FELIS have been prepared. Figure 1 shows architecture of FELIS.

In this system, initially, Loader module loads all the images into database. For this, it uses Image to Feature Vector Converter (IFV) module which has been explained in Figure 3.

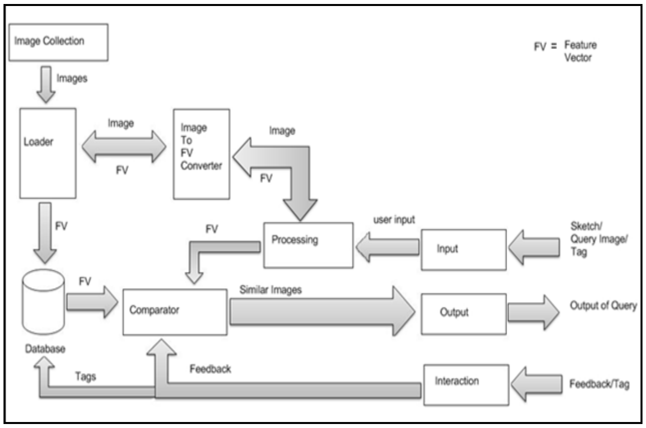

Fig 2. Architecture of FELIS

Three tables will be present in the database, first table will index the images, second table will index objects and store the pointer to other sub-objects contained in it and third table will store the feature of each identified object. Each image table will contain pointer to multiple objects present in the object table, which can be interpreted as presence of multiple objects in the image. Object table will contain 3 columns. First column will store the corresponding pointer in the feature table. Using the feature descriptor (in feature table), a unique key will be generated which will act as a primary key in the object table. The last column will store the user defined tag. In the image table also, a unique hash key specific to image will be stored. This key will be unique to each image. Thus, if image organization changes then this key will be useful to identify the image and hence corresponding path will be modified in the database. Thus, this architecture will allow dynamic organization of the images.

The Comparator module compares the query image and database value. This module takes weighted similarity measure. Here, weights mainly depend on user feedback.
User can manually adjust the weighting factor as per his requirement. Thus, user feedback directly affects the comparator module of the system.

The Image to Feature Vector Converter Module has been depicted in figure 3.

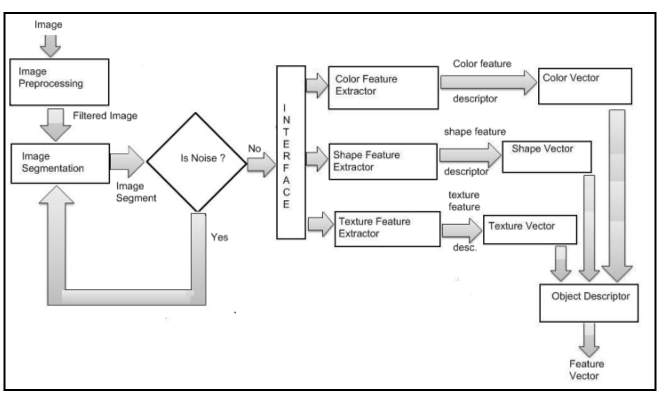

Fig 3. Image to Feature Vector converter

This module will convert the input image into its feature vector representation.

Image preprocessing is used to process the images initally to remove noise present in the system to certain extent so that processing at later stages becomes easy. Hence preprocessing mdoule is present.

Once image is pre-processed its quality improves and can be processed. Later image is segmented into different segments and each segment is processed separately. During processing, whether to treat the segment as noise or as object is decided. If segment is quite small then it is taken as a noise and discarded, otherwise it is processed later. In the later stage and composition is performed to group similar segments into one category.

During processing of segments, 3 separate modules are used which are responsible for converting given object segment into its shape, color and texture representation respectively. Later, output of each of these modules is merged to get the overall feature vector of an object.

6. ALGORITHMS

This section discusses about various algorithms which were selected for implementation of project.

A. Pre-processing

Pre-processing stage is divided into 3 passes. In the first pass, noise from the image is removed using Alpha trimmed filter. This filter removes some of very prominent noise. In the second pass, low pass filter is applied on an image so that edges are blurred and hence any discontinuity is removed and each area becomes closed. In the third pass, contrast of image is enhanced so that very low and very high illumination effects are removed.

B. Segmentation and Clustering

Image segmentation is a serious issue as most of algorithms designed for image segmentation are quite processor intensive to produce very accurate segments. One of the desired things for segmentation is that once image has been segmented each segment should be a closed region. To achieve this, morphological open operation can be applied on it.

Once all the segments are obtained, next thing comes is clustering of segments. While clustering assumption made is that if one object is completely contained inside another object, then it is part of that object. With this assumption two object are said to belong to same cluster if they are inside same segment. For doing this clustering we perform inside outside test of one object within another object. For this purpose, we extend the general mathematical algorithm of point inside another object [6]. In this approach, we take 8 sample points i.e. minimum x, maximum x, minimum y and maximum y. Other 4 points are in between them. If all these points are within another object then it is quite probable that the object to be considered is completely inside other object.

Although there are some exception to this, but it is sufficiently good for the same purpose.

C. Shape Representation

Shape representation method should be such that it should have following properties.

1) It should be invariant to translation, rotation and scaling.
2) It should be able to represent spatial relation and composition relationship between edges in an object.
3) It should not be processor intensive.

With these requirements, Distance Auto-correlogram (DAC) algorithm [4] can be chosen as it satisfies all of these criteria.

D. Color Representation

Color representation algorithm should have following properties.

1) It should be simple.
2) It should be able to represent (or scalable enough to cover) entire possibility of infinite colors in real world.
3) It should not take large amount of space.
4) It should take less retrieval time.

With these considerations, color histogram can be chosen as it

E. Texture Representation

Texture of an object is one of most important and difficult to calculate property of an object. Each object has different texture based on its physical material. Existing texture calculation algorithms are time-consuming, which will directly affect the processing time of the entire application. Hence, for texture representation, Gabor filter [5] can be used which calculates texture of individual object efficiently and accurately.

7. System Requirements

Following are requirements for the FELIS

A. Hardware Requirements

a) Any Intel or AMD x86 processor supporting SSE2 instruction set
b) Minimum 1GB Hard Disk Space
c) Minimum 1GB RAM

B. Software Requirements

a) Operating System: Window XP or Higher version
b) Java Runtime Environment (JRE) version 6
c) Image Processing Toolbox (from www.mathworks.com) version 6 or Higher Oracle Database Server (9i or Higher)

8. Conclusion

In this paper, design issues and architecture of FELIS have been presented. The key modules are accurate segmentation, composition accuracy and implementation of feature extraction algorithms, which have been discussed. This paper also addresses the tradeoff between accurate feature extraction and processing time. Taking these issues into consideration, we are looking forward for the successful implementation of the project.

9. References

[1]. 'PATSEEK: Content Based Image Retrieval System for Patent Database', Avinash Tiwari, Veena Bansal. Industrial and Management Engineering Department, IIT Kanpur.

[2]. 'Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying.' Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 24, NO. 8.

[3]. 'Photobook: Content based Manipulation of Image Database', A. Pentland, R. W. Picard, S. Sclaroff. Perceptual Computing Section, The Media Laboratory, Massachusetts Institute of Technology, SPIE Storage and Retrieval of Image and Video Database II, No. 2185.

[4]. 'A Novel Shape Representation and Retrieval Algorithm: Distance Autocorrelogram', Jie-xianZeng, Yong-gang Zhao, Xiang Fu, Nanchang Hang Kong University, China, Journal of Software, volume 5, number 9.

[5]. 'Content-based Image Retrieval Using Gabor Texture Features', Dengsheng Zhang, Aylwin Wong, Maria Indrawan, Guojun Lu, Monash University, Australia.

[6]. http://en.wikipedia.org/wiki/Inside-outside_test