2 0 0 9 年 2月

碩士學位論文

LOCAL REGIONAL OBJECT BASED RETRIEVAL TECHNIQUES FOR CBIR APPLICATIONS

朝 鮮 大 學 校 　大 學 院

情報通信工學科

Rasheed Waqas

2009年 2月

碩士學位論文

LOCAL REGIONAL OBJECT BASED RETRIEVAL TECHNIQUES FOR CBIR APPLICATIONS

Rasheed Waqas

LOCAL REGIONAL OBJECT BASED RETRIEVAL TECHNIQUES FOR CBIR APPLICATIONS

2 0 0 9 年 2月 日

朝 鮮 大 學 校 大 學 院

情報通信工學科

Rasheed Waqas

LOCAL REGIONAL OBJECT BASED RETRIEVAL TECHNIQUES FOR CBIR APPLICATIONS

指導敎授 朴 鍾 安

이 論文을 工學碩士學位申請 論文으로 提出함.

2 0 0 9 年 2月 日

朝 鮮 大 學 校 大 學 院

情報通信工學科

Rasheed Waqas

Rasheed Waqas의 碩士學位 論文을 認准함

委員長 光州科技院 敎授 韓 承 朝 印

委 員 朝鮮大學校 敎授 邊 宰 瑩 印

委 員 朝鮮大學校 敎授 朴 鍾 安 印

2009年 2月 日

朝 鮮 大 學 校 大 學 院

Table of Contents (목차)

List of Figures

List of Tables

ABSTRACT

Local Regional Object-based Retrieval Techniques for CBIR Applications

Rasheed, Waqas
Advisor : Prof. Park, Jong-An, Ph. D
College of Electronics and Information Engineering
Graduate School of Chosun University

The study mainly covers the subject of content based image retrieval (CBIR). Some specialized applications for image based query and retrieval are researched and some useful results are produced. The research includes image structural and color feature analysis in order to ease retrieval of similar images from huge databases. HSV, RGB and gray-scale nature of images was analysed and most prominent and noteworthy parameters were short listed and ranked against various sorts of images.

Many references have been studied in order to rank various notable features of images, which prove key indices or identification marks of various images. Under the light of most reputable and most remarkable research, the features were rigorously tested and confirmed for their solidity. The most useful of them were used to devise several algorithms that can achieve promising retrieval results. Simulation was performed using MATLAB, and tested against existing algorithms.

Several valuable image characteristics were studied separately and collectively in the form of sets. The most useful combinations are devised after collective and iterative simulation of the most fruitful set of image identification features.

I. *Introduction*

A. *History of CBIR*

CBIR, or "Content-based image retrieval", also known as "query by image content" (QBIC) and "content-based visual information retrieval" (CBVIR) is the application of computer vision to the image retrieval problem, that is, the problem of searching for digital images in large databases. "Content-based" means that the search will analyze the actual contents of the image. The term 'content' in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself. Without the ability to examine image content, searches must rely on metadata such as captions or keywords, which may be laborious or expensive to produce.

The term CBIR seems to have originated in 1992, when it was used by T. Kato [53] to describe experiments into automatic retrieval of images from a database, based on the colors and shapes present. Since then, the term has been used to describe the process of retrieving desired images from a large collection on the basis of syntactical image features. The techniques, tools and algorithms that are used originate from fields such as statistics, pattern recognition, signal processing, and computer vision.

B. *Research Background*

There is a growing interest in CBIR because of the limitations inherent in metadata-based systems, as well as the large range of possible uses for efficient image retrieval. Textual information about images can be easily searched using existing technology, but requires humans to personally describe every image in the database. This is impractical for very large databases, or for images that are generated automatically, e.g. from surveillance cameras. It is also possible to miss images that use different synonyms in their descriptions. Systems based on categorizing images in semantic classes like "cat" as a subclass of "animal" avoid this problem but still face the same scaling issues.

The main issues in improving CBIR systems are:

1. Which features should be derived to describe the images better within database?
2. Which data structure should be used to store the feature vectors?
3. Which learning algorithms should be used in order to make the CBIR wiser?
4. How to participate the user's feedback in order to improve the searching result?

Potential uses for CBIR include:

* Art collections
* Photograph archives
* Retail catalogs
* Medical diagnosis
* Crime prevention
* The military
* Intellectual property
* Architectural and engineering design
* Geographical information and remote sensing systems

Different implementations of CBIR make use of different types of user queries.

1. Query by Example
2. Semantic retrieval
3. Browsing for example images
4. Navigating customized/hierarchical categories
5. Querying by image region (rather than the entire image)
6. Querying by multiple example images
7. Querying by visual sketch
8. Querying by direct specification of image features
9. Multimodal queries (e.g. combining touch, voice, etc.)

CBIR systems can also make use of relevance feedback, where the user progressively refines the search results by marking images in the results as "relevant", "not relevant", or "neutral" to the search query, then repeating the search with the new information.

Some common methods for extracting content from images are:

1. Color
2. Texture
3. Shape

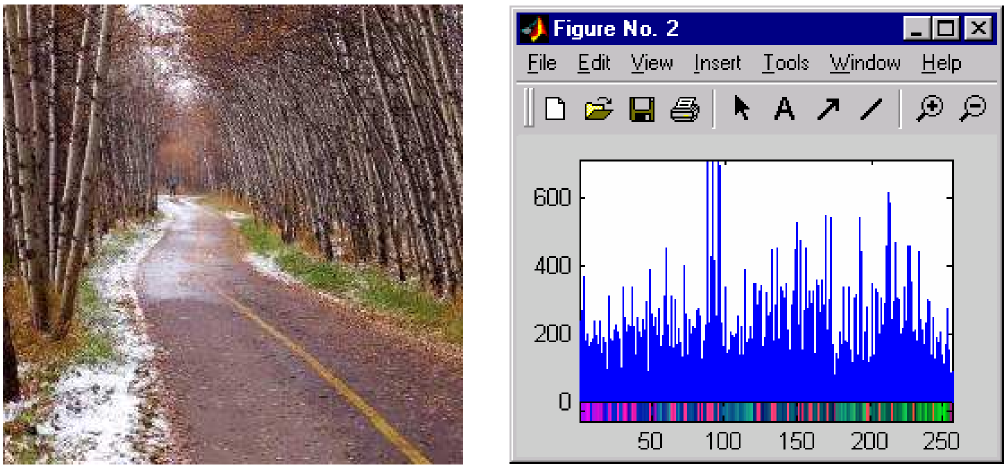
1. *Color*

One of the most important features that make possible the recognition of images by humans is color. color is a property that depends on the reflection of light to the eye and the processing of that information in the brain. We use color everyday to tell the difference between objects, places, and the time of day [54]. Usually colors are defined in three dimensional color spaces. These could either be RGB (Red, Green, and Blue), HSV (Hue, Saturation, and Value) or HSB (Hue, Saturation, and Brightness). The last two are dependent on the human perception of hue, saturation, and brightness.

Most image formats such as JPEG, BMP, GIF, use the RGB color space to store information [54]. The RGB color space is defined as a unit cube with red, green, and blue axes. Thus, a vector with three co-ordinates represents the color in this space. When all three coordinates are set to zero the color perceived is black. When all three coordinates are set to 1 the color perceived is white [54]. The other color spaces operate in a similar fashion but with a different perception.

The main method of representing color information of images in CBIR systems is through color histograms. A color histogram is a type of bar graph, where each bar represents a particular color of the color space being used. In MatLab for example you can get a color histogram of an image in the RGB or HSV color space. The bars in a color histogram are referred to as bins and they represent the x-axis. The number of bins depends on the number of colors there are in an image. The y-axis denotes the number of pixels there are in each bin. In other words how many pixels in an image are of a particular color.

As one can see from the color map each row represents the color of a bin. The row is composed of the three coordinates of the color space. The first coordinate represents hue, the second saturation, and the third, value, thereby giving HSV. The percentages of each of these coordinates are what make up the color of a bin. Also one can see the corresponding pixel numbers for each bin, which are denoted by the blue lines in the histogram.



(a) (b)

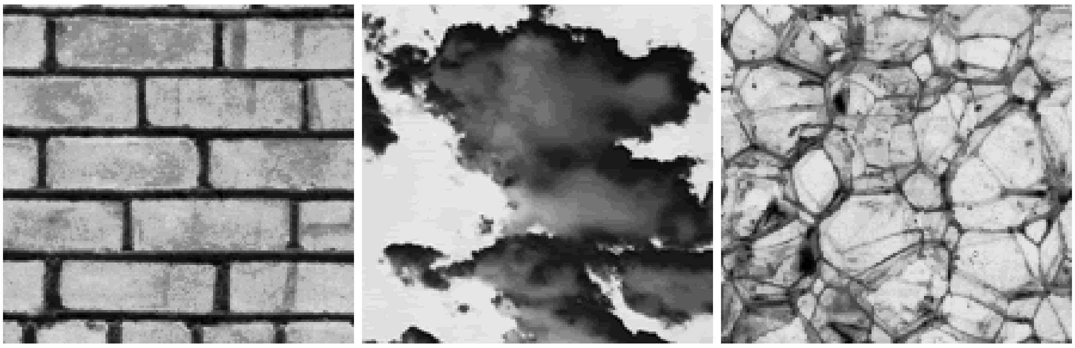
Fig 1.1 (a) Image in HSV color space. (b) Color histogram

Quantization in terms of color histograms refers to the process of reducing the number of bins by taking colors that are very similar to each other and putting them in the same bin. By default the maximum number of bins one can obtain using the histogram function in MATLAB is 256. For the purpose of saving time when trying to compare color histograms, one can quantize the number of bins. Obviously quantization reduces the information regarding the content of images but as was mentioned this is the tradeoff when one wants to reduce processing time.

There are two types of color histograms, Global color histograms (GCHs) and Local color histograms (LCHs). A GCH represents one whole image with a single color histogram. An LCH divides an image into fixed blocks and takes the color histogram of each of those blocks [54]. LCHs contain more information about an image but are computationally expensive when comparing images. "The GCH is the traditional method for color based image retrieval. However, it does not include information concerning the color distribution of the regions [54]" of an image. Thus when comparing GCHs one might not always get a proper result in terms of similarity of images.

2. *Texture*

Texture is that innate property of all surfaces that describes visual patterns, each having properties of homogeneity. It contains important information about the structural arrangement of the surface, such as; clouds, leaves, bricks, fabric, etc. It also describes the relationship of the surface to the surrounding environment. In short, it is a feature that describes the distinctive physical composition of a surface.



(a) (b) (c)

Fig 1.2 Various forms of texture

Texture properties include:

- Coarseness
- Contrast
- Directionality

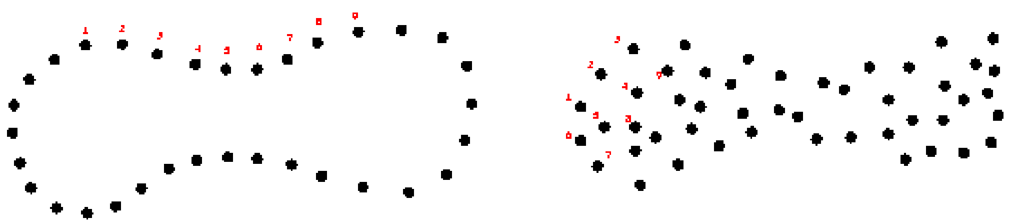
- Line-likeness
- Regularity
- Roughness

Texture is one of the most important defining features of an image. It is characterized by the spatial distribution of gray levels in a neighborhood [55]. In order to capture the spatial dependence of gray-level values, which contribute to the perception of texture, a two-dimensional dependence texture analysis matrix is taken into consideration. This two-dimensional matrix is obtained by decoding the image file; JPEG, BMP, etc.

3. *Shape*

Shape may be defined as the characteristic surface configuration of an object; an outline or contour. It permits an object to be distinguished from its surroundings by its outline. Shape representations can be generally divided into two categories:

- Boundary-based, and
- Region-based.



(a) (b)

Fig 1.3 (a) Boundary based, and (b) Region based.

Boundary-based shape representation only uses the outer boundary of the shape. This is done by describing the considered region using its external characteristics; i.e., the pixels along the object boundary. Region-based shape representation uses the entire shape region by describing the considered region using its internal characteristics; i.e., the pixels contained in that region.

C. *Image Retrieval in the Real World*

Invention of the digital camera has given the common man the privilege to capture his world in pictures, and conveniently share them with others. One can today generate volumes of images with content as diverse as family get-togethers and national park visits. Low-cost storage and easy Web hosting has fueled the metamorphosis of common man from a passive consumer of photography in the past to a current-day active producer. Today, searchable image data exists with extremely diverse visual and semantic content, spanning geographically disparate locations, and is rapidly growing in size. All these factors have created innumerable possibilities and hence considerations for real-world image search system designers.

As far as technological advances are concerned, growth in content-based image retrieval has been unquestionably rapid. In recent years, there has been significant effort put into understanding the real world implications, applications, and constraints of the technology. Yet, real-world application of the technology is currently limited. We devote this section to understanding image retrieval in the real world and discuss user expectations, system constraints and requirements, and the research effort to make image retrieval a reality in the not-too-distant future. Designing an omnipotent real-world image search engine capable of serving all categories of users requires understanding and characterizing user-system interaction and image search, from both user and system points-of-view. From a user perspective, embarking on an image search, journey involves considering and making decisions on the following fronts.

1. Clarity of the user about what is needed
2. Where the user intends to search
3. The form in which the user sets up query.

In an alternative view from an image retrieval system perspective, a search translates to making arrangements as per the following factors.

1. How does the user wish the results to be presented.
2. Where does the user desire to search
3. What is the nature of user input/interaction.

A typical browser values ease of use and manipulation. A browser usually has plenty of time at hand and expects surprises and random search hints to elongate her session (e.g., picture of the day, week, etc.). On the other hand, a surfer would value a search environment which facilitates clarity of her goal. A surfer planning a holiday would value a hint such as "pictures of most popular destinations". At the other extreme, the searcher views an image retrieval system from a core utilitarian perspective. Completeness of results and clarity of representation would usually be the most important factors. The impact of real-world usage from the user viewpoint has not been extensively studied.

systems.

An image retrieval system designed to serve a personal collection should focus on features such as customization, flexibility of browsing, and display methodology. Domain-specific collections may impose specific standards for presentation of results. Searching an archive for content discovery could involve long user search sessions. Good visualization and a rich query support system should be the design goals.A system designed for the Web should be able to support massive user traffic. One way to supplement software approaches for this purpose is to provide hardware support to the system architecture. Unfortunately, very little has been explored in this direction, partly due to the lack of agreed-upon indexing and retrieval methods.

Presentation of search results is perhaps one of the most important factors in the acceptance and popularity of an image retrieval system. We characterize common visualization schemes for image search as follows.

1. *Relevance-Ordered*

The most popular way to present search results is relevance ordered, as adopted by Google (http://www.google.com) and Yahoo! (http://www.yahoo.com) for their image search engines. Results are ordered by some numeric measure of relevance to the query.

2. *Time-Ordered*

In time-ordered image search, pictures are shown in a chronological ordering rather than by relevance.

3. *Clustered*

Clustering of images by their meta-data or visual content has been an active research topic for several years. Clustering of search results, besides being an intuitive and desirable form of presentation, has also been used to improve retrieval performance.

4. *Hierarchical*

If meta-data associated with images can be arranged in tree order, it can be a very useful aid in visualization. Hierarchical visualization of search results is desirable for archives, especially for educational purposes.

5. *Composite*

Combining consists of mixing two or more of the preceding forms of visualization scheme, and is used especially for personalized systems. Hierarchical clustering and visualization of concept graphs are examples of composite visualizations.

The prevalent research topics which have potential for improving multimedia retrieval by bridging the semantic gap are human-centered computing, new features, new media, browsing and summarization, and evaluation/benchmarking. In human-centered computing, the main idea is to satisfy the user and allow the user to make queries in their own terminology. User studies give us insight directly into the interactions between human and computer. Experiential computing also focuses on methods for allowing the user to explore and gain insights in media collections. On a fundamental level, the notion of user satisfaction is inherently emotional. Affective computing is fascinating because it focuses on understanding the user's emotional state and intelligently reacting to it. It can also be beneficial toward measuring user satisfaction in the retrieval process.

Learning algorithms are interesting because they potentially allow the computer to understand the media collection on a semantic level. Furthermore, learning algorithms may be able to adapt and compensate for the noise and clutter in real world contexts. New features are pertinent in that they can potentially improve the detection and recognition process or be correlated with human perception. New media types address the changing nature of the media in the collections or databases. Some of the recent new media include 3D models (i.e. for virtual reality or games) and biological imaging data (i.e. towards understanding the machinery of life). As scientists, we need to objectively evaluate and benchmark the performance of the systems and take into account factors such as user satisfaction with results. Currently, there are no large international test sets for the wide problems such as searching personal media collections, so significant effort has been addressed toward developing paradigms which are effective for evaluation. Furthermore, as collections grow from gigabyte to terabyte to peta-byte sizes, high performance algorithms will be necessary toward responding to a query in an acceptable time period.

The research included in this thesis includes color and shape specific methods used in CBIR applications. Chapter 4, especially, deals with a proposal about video stream supporting CBIR application over a wide range of network topologies. The results generated through simulations produced through MATLAB software are also added where needed.

II. *Image Retrieval Algorithm based on Incremental CBIR using Color Histogram*

A. *Introduction*

In this paper, a modified scheme based on histogram refinement [1] method is presented. Our method has two distinct stages. Detailed stages are described in section (C) onwards. For retrieval, we use incremental content based retrieval (CBIR) by separately using the features extracted in each of two stages.

B. *Related Work*

Work by Arnold et. al. [2] is an excellent review of content based image retrieval till 2000. Hsu [3] exploits the degree of overlap between regions of the same color. They used a database of 260 images. Paisarn [4] proposed an unsupervised learning network to incorporate a self-learning capability into image retrieval systems. Smith & Chang's method also partitions the image. Histogram back-projection method [5] is used for back projecting set of colors onto the image. They used database of 3100 images for testing purposes. Zhang [6] discussed a generic Fourier descriptor (GFD) to overcome the drawbacks of existing shape representation techniques.

Rickman and Stonham [7] provide a method based on small equilateral triangles with fixed sides. They used a database of 100 images. Djeraba [8] tried to add the generalization capability for indexing and retrieval. Stricker and Dimai [9] finds the first three moments of the color distributions in an image. They used a database of about 11,000 images. Huang et al. [10] method is called Color Correlogram and it captures the spatial correlation between colors. They used a database of 18,000 images.

Pass and Zabih [1] method called Histogram Refinement partitions histogram bins by the spatial coherence of pixels. Their database consists of 14,554 images. Jong-An, Bilal et al. [11,12] provided shape description based on histogram based chain codes. Vasileios [13] presented an image retrieval methodology suited for search in large collections of heterogeneous images.

C. *Feature Extraction Algorithm*

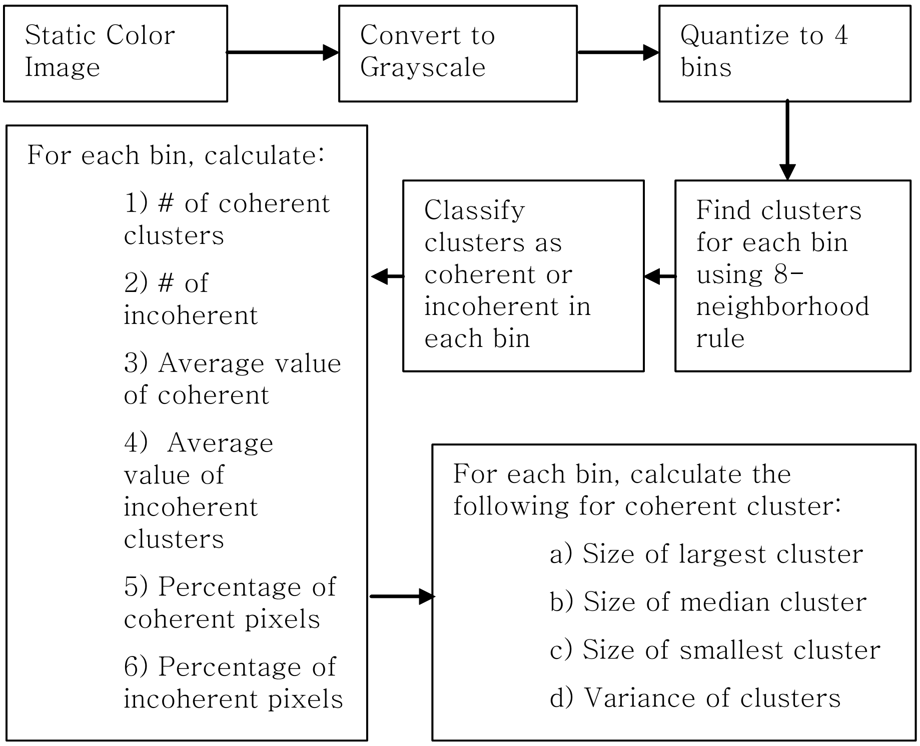


Fig 2.1 Block diagram of the feature extraction algorithm

Color histogram buckets are partitioned based on spatial coherence just like computed by Pass and Zabih [1]; A pixel is coherent if it is a part of some sizable similar colored region, otherwise it is incoherent. So the pixels are classified as coherent or incoherent within each color bucket. If a pixel is part of a large group of pixels of the same color which form at least one percent of the image then that pixel is a coherent pixel and that group is called the coherent group or cluster. Otherwise it is incoherent pixel and the group is incoherent group or cluster.

For each discretized color j, let the number of coherent pixels as α_j, the number of coherent connected components as $C_{\alpha j}$ and the average of coherent connected compo-nent as $\mu_{\alpha j}$. Similarly, let the number of incoherent pixels as β_j, the number of incoherent connected components as $C_{\beta j}$ and the average of incoherent connected component as $\mu_{\beta j}$. For each discretized color j, the total number of pixels are $\alpha_j+\beta_j$ and the color histogram summarizes the image as $<\alpha_1+\beta_1, ..., \alpha_n+\beta_n>$.

Since there are 4 bins, so we get total of 6 features for each bin. Therefore, a total of 24 features are considered for image retrieval.

At this stage, additional features are selected for content based image retrieval on the coherent clusters only. Four features are selected among the coherent clusters. Three of them are based on the size of the clusters while one is statistical in nature. They are:

(a) Size of largest cluster in each bin ($L_{\alpha j}$)

(b) Size of median cluster in each bin ($M_{\alpha j}$)

(c) Size of smallest cluster in each bin ($S_{\alpha j}$)

(d) Variance of clusters in each bin ($V_{\alpha j}$)

Therefore, a total of 40 additional features are considered for image retrieval.

D. *Incremental Image Retrieval Approach*

1. *First Level Retrieval*

We use the L_1 distance to compare two images I and I'. Using the L_1 distance, the j_{th} bucket's contribution to the distance between I and I' is:

$$\Delta_1 = |(\alpha_j - \alpha'_j) + (\beta_j - \beta'_j)| \quad ...(1)$$

$$\Delta_2 = |(C_{\alpha j} - C'_{\alpha j}) + (C_{\beta j} - C'_{\beta j})| \quad ...(2)$$

$$\Delta_3 = |(\mu_{\alpha j} - \mu'_{\alpha j}) + (\mu_{\beta j} - \mu'_{\beta j})| \quad ...(3)$$

So we get the initial retrieval result with this method. In original scheme [1], only equation (1) is used and for comparison, the following equation is used:

$$\Delta_4 = |(\alpha_j+\beta_j)-(\alpha'_j+\beta'_j)| \quad ...(4)$$

2. *Second Level Retrieval*

Here we use the L_1 distance to compare two images I and I′. Using the L_1 distance, the j_{th} bucket's contribution to the distance between I and I′ is:

$$\Delta_5 = |(L_{\alpha j}-L'_{\alpha j})| \quad ...(5)$$

$$\Delta_6 = |(M_{\alpha j}-M'_{\alpha j})| \quad ...(6)$$

$$\Delta_7 = |(S_{\alpha j}-S'_{\alpha j})| \quad ...(7)$$

$$\Delta_8 = |(V_{\alpha j}-V'_{\alpha j})| \quad ...(8)$$

(a) Query Image

(b) First Level Retrieval

(c) Second Level Retrieval

Fig 2.2 Image Retrieval from the database

E. *Results and Discussion*

The proposed algorithm is tested with the database provided by James S. Wang et. al [14]. Figure 2.2 shows results obtained through various phases of retrieval.

F. *Conclusions*

In this paper, we have proposed an algorithm that is based on color histogram. We have shown that the features obtained using this algorithm are quite useful for relevant image retrieval queries. The feature selection is based on the number, color and shape of objects present in the image. The grayscale values, mean, variance and various sizes of the objects are considered as appropriate features for retrieval and are independent of image orientation. Color refinement method takes care of the color as well as the spatial relation feature.

We have also presented a two stage approach for image retrieval. Hence, this ap-proach is computationally efficient and provides refined result. Results show that the algorithm presented in this paper provides very relevant retrieval results.

III. *Image Retrieval using Maximum Frequency of Local Histogram based Color Correlogram*

A. *Introduction*

A huge amount of text and image data is available on the World Wide Web (WWW) for access to internet users. The amount of data especially still images and videos is increasing exponentially. The users not only want to retrieve the text data, but they would be very happy if images and videos can also be retrieved as fast as text data retrieves on the WWW. The amount of image and video data has remarkably increased and hence the demand of image retrieval systems that are able to effectively index a large amount of images and to effectively retrieve them based on their visual contents. Content Based Image Retrieval (CBIR) [2,15-17] is an interesting but difficult research topic in multimedia information technology. CBIR uses visual contents such as color, shape, texture etc to retrieve images from a huge image databases according to the user's visual queries.

In content based image retrieval (CBIR), a user has an image and he/she is interested to search the similar images from image database. For this, a feature vector, characterizing some image properties, is computed and stored in the image feature database. The user gives the query image, and the CBIR system computes the feature vector for the query and then compare with the features of database images. The comparison is done using some distance measure, and the minimum distances are the metrics for the matched or similar images. The features extractions and their matching should be efficient enough to retrieve the similar images from the image database.

Color feature is one of the most reliable and easier visual features used in image retrieval. It is robust to background complication and is independent of image size and orientation. The most common technique for extracting the color features is based on color histograms of images [5,10,18-20]. A color histogram tells the global distribution of colors in the images. It is very easy to compute and insensitive to small variations in the images. There are two main problems in color histogram based CBIR. First, color histogram does not take into account the spatial information. The second is that the histogram is not unique and also not robust. Two different images with similar color distribution give rise to very similar histograms. Similarly, the images of the same view with different conditions of lighting create very different histograms.

To deal with the first problem, many researchers suggested the use of color Correlogram for taking into account the spatial information [10]. It is suggested in [20, 21] the use of multiresolution histogram for image retrieval. In [21], Gaussian filtering is used for multiresolution decomposition of an image.

In this paper, we try to solve the second problem, again by using the concept of a color Correlogram. A single image histogram suffers from the inability to encode spatial image variations.

Fig 3.1 An image selected at random as a sample from a collection of images

B. *Method*

Digital images undergo the following process in order to produce an effective Correlogram describing an eminent feature set targeted to avoid the lack of robustness of a common histogram.

1. *Pre-processing*

RGB and indexed images carry high values that require more computation time. Hence, the images are converted to grayscale in order to reduce the vast spectrum of indexed images or the 3D components of RGB to a 2D component carrying values between 0 and 255 (containing the end points). This process promises reduction in the computation time and power required for extracting features from an image. The resulting image undergoes histogram equalization in order to enhance contrast of values of an image by generating its flat histogram. Figures 3.2 show the pre-processing applied to the image shown in figure 3.1.

2. *Splitting Histogram Values by Fixed Frequency Range*

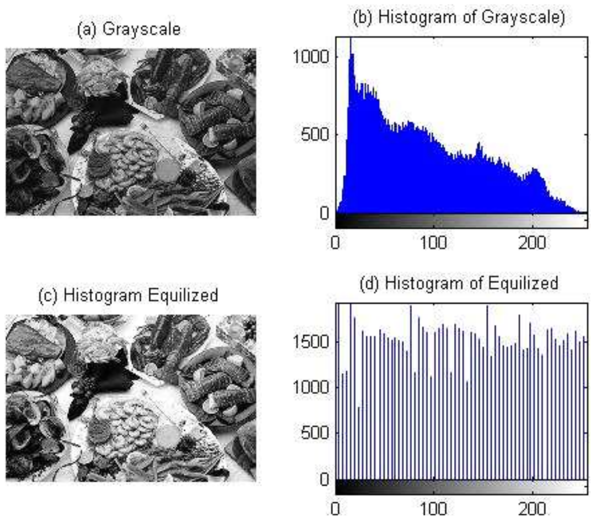


Fig 3.2 (a) Grayscale (intensity) representation of image in figure 1 (b) Histogram of the grayscale image (c) Grayscale image after histogram equalization (d) Histogram of the flat (equalized) image

The histogram equalized image is split into four fixed bins in order to extract more distinct information from it. The frequencies of 256 values of gray scale are split into four bins carrying 64 values each (0~63, 64~127, 128~191, and 192~255).

This is done by turning off the gray values of image which do not lie between the four bins.

This gives four images carrying objects which lie in the specific frequency ranges, and all different from each other. This provides a better illustration of image segments and simplifies the computation of features for the distinct portion of image. An example of the mechanism is shown in figure 3.3, which clearly shows the distribution of frequencies in various bins.

3. *Sub-divisions of Histogram Values from each Bin*

The bins produced in the previous section (C-2) are further subdivided into four fixed ranges carrying 16 values each. For example, for the first bin carrying value range of 0 to 63, four sub-divisions of the frequency values will be 0~15, 16~31, 32~ 47, and 48~63. Similar procedure is repeated for rest of the bins. Hence, the four bins used in the beginning give rise to (4x4=) 16 equal sub-divisions.

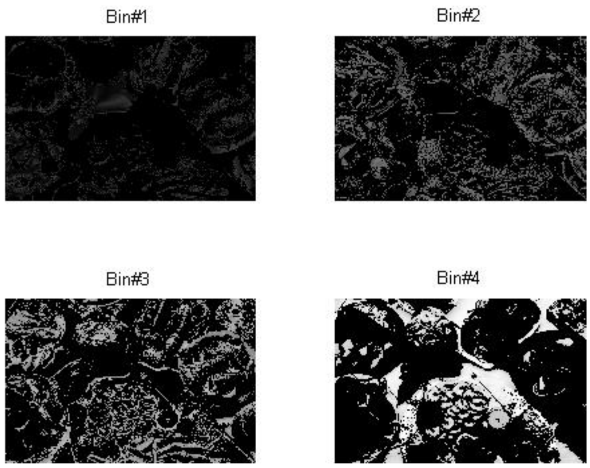


Fig 3.3 Bins generated from the image in figure 3.2(c)

4. *Calculating Maximum Frequency of the Most Recurring Value from the Bin Sub-divisions*

The bins produced in the previous section (C-2) are further subdivided into four fixed ranges carrying 16 values each. For example, for the first bin carrying value range of 0 to 63, four sub-divisions of the frequency values will be 0~15, 16~31, 32~ 47, and 48~63. Similar procedure is repeated for rest of the bins. Hence, the four bins used in the beginning give rise to (4x4=) 16 equal sub-divisions.

5. *Correlogram Formation*

The information from bins and sub-divisions are stored in the form of a Correlogram as shown in table 3.1 as shown below.

Table 3.1 Correlogram of bins vs. subdivisions

Sub-divisions	**1**	**2**	**3**	**4**
Bin 1	1181	1019	834	820
Bin 2	782	610	582	585
Bin 3	554	510	432	730
Bin 4	452	672	584	549

6. *Similarity Measurement*

The distance between the Correlogram of query image and the Correlograms of image stored in the database can be calculated by using L_1 or Euclidean distance. The distance measurement process comprises of three steps. The Correlogram matrices are subtracted, at first, under simple subtraction rules. Secondly, the sum of the matrix components is calculated. Finally, the third step comprises of sorting the absolute values of the sums obtained in the second step.

For any two vector images Q and D, let their corresponding correlograms be 4 row and 4 column matrices. Let, Correlogram of image Q be Corr(Q). Similarly, let the Correlogram of image D be Corr(D). According to Euclidean distance algorithm described above, let:

$$Distance(Q,D) = Corr(Q) - Corr(D) \cdots (1)$$

Where, *Distance(Q,D)* is a vector containing the result of difference calculation as described in equation 1. The components of the resulting matrix are summed together and the absolute value of this sum is considered.

These absolute values are sorted in order to display the top matches.

C. *Results*

The process described in section 2 is rigorously tested against various conditions and types of images. The algorithm of the paper has been tested against the database of James Z. Wang et al. [14, 25]. Some minor changes to some of the images were made in order to test the robustness (figure 3.8; image Hue, saturation, color values, and object positions are altered). Tests prove that the worth of this algorithm because it can be useful in three ways:-

1. Less computation is required, which makes the feature extraction and retrieval very fast.
2. This sort of Correlogram representation is independent of differences in displacements and hue of images.
3. Complex images, which carry many objects of a very wide range of color values, can also be retrieved to a considerable extent

Figures 3.4-3.7 clearly show the robustness of the results generated from the algorithm devised in this paper. Figure 3.4 and 3.5 carry the highest degree of likeness of results generated by this algorithm. Less likeliness to the query image can be noticed in figures 3.6 and 3.7, because of their complicated nature due to too many colored regions.

Moreover, figure 3.8 confirms the second inference stated earlier in this section. There is a major area displacement in the second and third result in figure 3.8 (2: horizontal shift and 3: vertical shift). Results 5 up to 9 have a noticeable difference in the hue, saturation, and color value.

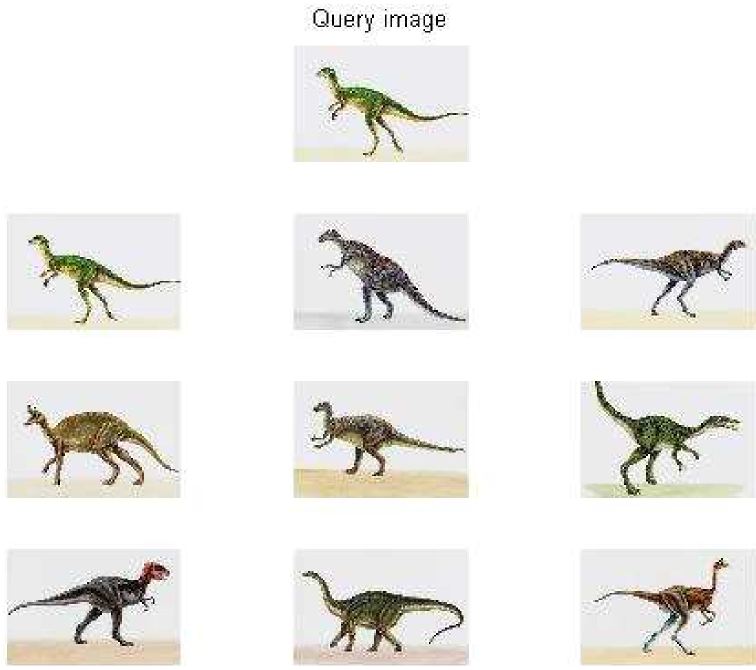


Fig 3.4 Top 9 matches of a dinosaur query image

Fig 3.5 Top 9 images of a flower query image

Fig 3.6 Top 6 images of a painted face query image

Fig 3.7 Top 6 images of a horse query image

Fig 3.8 Top 9 images of a food query image

These images are still marked similar to the query image which proves the second inference.

D. *Conclusion and Discussion*

Experiments prove that region specific histogram properties can be very useful, because, they add robustness to the histograms. Hence, it can be stated that two dissimilar images can be distinguished by using the color Correlogram approach based on some local property (or properties).

The promising results displayed in the previous section can prove the worth of the feature extraction process, which also works well for complex images. Therefore, it can be stated that this algorithm solves the problem rooted at inability of uniqueness and robustness of histogram matching. Different distance measures can also be chosen.

IV. *Key Objects based Profile for a Content-based Video Information Retrieval and Streaming System using Viewpoint Invariant Regions*

A. *Introduction*

With advances in visual technologies, scientists have greatly concentrated on feature retrieval from images and motion picture. The modern day techniques include all-embracing phenomenon for extracting feature sets from structural and color characteristics of images. 3D imagery and stereoscopy have also been the interests for image retrieval. The introduction of HDTV and high and low bandwidth streaming of video content over the internet have proved to be a promising power in steering the direction of research in the fields of computer vision, image processing, neural networks, data mining, and robotics.

One of the best review of CBIR till 2000 is provided by Arnold et al. [2], reviewing some 200 references in content based image retrieval. Content based video compression is very decently described by HongJiang Zhang et al. [26] where they provide an idea about using key-objects individually from key-frames of a video. The idea of this paper is rooted in their research. The research in this paper may sound similar to Anil K. Jain et al. [27], which is also based on the research of Zhang et al. [26], but, it only uses their research for acquiring key-objects from the key frames, and rest of the process and the target achieved is entirely different. Similarly, Ling-Yu Duan et al. [28] have proposed a fast search based on index structure of objects in the key frames.

R. Lienhart [30] has given a very thorough account on reliable transition detection in videos. S. Agarwal et al. [31] researched for methods involving learning algorithm for a sparse representation for object detection. Mosaic based clustering of movie scenes algorithm, as devised by A. Aner et al. [32], also proved a major improvement in video retrieval by proposing a shot and scene clustering system. Content-Based Indexing algorithm, proposed by S. Eickeler [33], offer refined results for face detection and recognition systems by indexing faces existing in images and video. S. Lazebnik et al. [35] suggested an affine transform based method locating Affine-invariant local descriptors and neighborhood statistics for texture recognition. Similarly, B. Tseng et al. [36] devised a method for personalization and summarization of videos. T. Tuytelaars et al. [37] have also worked on affine-invariant regions and stereo matching. P. Hong et al. [34] had a good dealing with mining of inexact spatial patterns.

B. *Information Acquisition Process*

Amount of information in a movie depends upon the duration, frame size, and quality of the movie. The information retrieval process for the ease of the user can be segmented into two phases: Object recognition and Object Logging.

1. *Object Recognition*

Keeping in mind the complexity of isolating and searching for a specific object, only those calculation methods can be applied which utilize least computation power time. Color based CBIR methods can be thought of for this reason, but, they do not carry information about changes in the orientation of any particular object. A unified solution for content based video retrieval, such as the one described by HongJiang Zhang, et al. [26], can be used in order to locate “key objects”. A key object is an idea extended from the concept of “key frame”. A key object consists of region within a key-frame that move with similar motion.

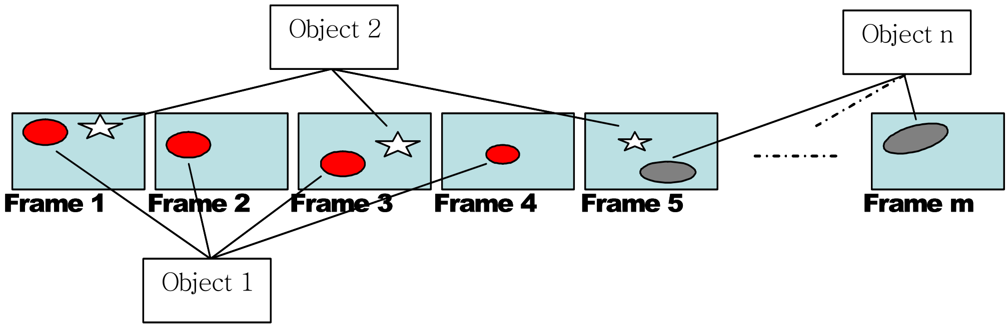


Fig 4.1 Identification of similar key object occurrences in various key frames

.

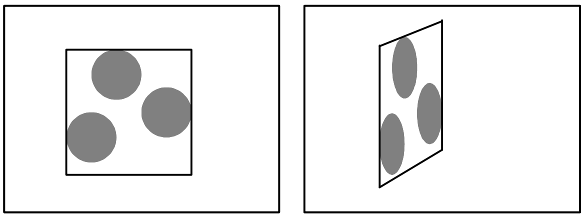


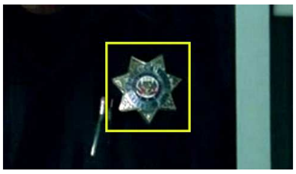
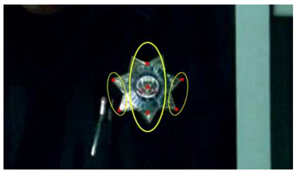
Fig 4.2 Spatial configuration: a square are centered at affine covariant region.

The key objects are used as the reference objects to be located throughout the movie key frames and are logged. Figure 4.1 shows the key objects being searched and marked from entire movie, which are logged according to their occurrences in the profile of table 4.1.

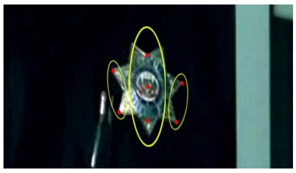
Our experiments show that lookup process for the next occurrences of the key object is remarkably performed by using the concept of viewpoint invariant regions, as described by Josef Sivic and Andrew Zisserman [29].

According to their method, the spatial configuration and extent have to be noted followed by any viewpoint invariant match across the frames of a motion picture. Hence, start from a detected elliptical region p encapsulating the key object in one frame and define it's neighborhood as all detected regions within an area A centered on p. The size of A determines the scale of the configuration, and the neighbors of p. The detected elliptical regions matching p are determined in rest of the frames, and a match between p and p' in a second frame also determines the 2D affine transformation between the regions, which in turn can be used to map A surrounding p to its corresponding skewed area in the second frame. The neighbors of p', in the second frame, as those elliptical regions lying inside the skewed area are determined. When all the elliptical neighbors of p can be mapped onto corresponding elliptical neighbors of p' through affine transformation between the two neighborhood, it implies a match. This phenomenon is illustrated in figure 4.2.

(a)

(b) (c)

(d) (e)

Fig 4.3 Example of scene from the TV series “Prison Break” (a) A key frame carrying a key object (b) Close-up of a key object from a key frame (c) elliptical region around encapsulating sub-key-objects (d) affine transformation of key object (e) elliptical region around encapsulating sub-key-objects after affine transformation

The neighborhood of an elliptical region p is the convex hull of its N spatial nearest neighbors in the frame and the neighborhood of the matching region p' is the convex hull of its N spatial nearest neighbors as shown in figure 4.3. Hence, the two configurations are deemed matched if M of the neighbors also matches, where usually M is a small fraction of N.

Josef Sivic et al. [29] also devised a three stage algorithm to efficiently compute the frequency of occurrence of the neighborhoods defined as above.

1. Neighborhoods occurring in more than a minimum number of key frames are considered for clustering.
2. Significant neighborhoods are matched by a progressive clustering algorithm.
3. Resulting clusters are merged based both on spatial and temporal overlap.

2. *Object logging*

This information is meant to be stored at the movie hosting or streaming servers as a separate database or in the form of profile bound AVI (movie) files (probably in the form of some new format). Any user who is searching for any particular object over the internet may need this information readily available. Similarly, key objects and key frames from various movies can be used to gather some really important information about occurrences involving similar object from the greatest movie database; the internet.

The information gathered in the previous section (2.1) will eventually prove as the useful information required by the multimedia application users looking for a specific object from one or many movies at a time.

Table 4.1 shows such a profile derived from information in figure 4.1. There are many frames in figure 4.1, where similar object is repeated in various sizes and orientations seen at various locations. All of this data and any other data can be stored in the profile.

Redundant objects may be removed from the objects list and placed as the containing frame information (frame number, location, size, and orientation). It will further reduce the size of the profile, and it will surely make it more web databases friendly due to the fact that it will further reduce the amount of data any database engine would require to browse and search during retrieval operation.

Table 4.1 Profile of object occurrences in figure 4.1

Objects	**Occurrences**				
O1	Frame	1	2	3	4
	Location	(x11,y11)	(x12,y12)	(x13,y13)	(x14,y14)
	Size	100%	100%	100%	65%
	Orientation	0°	0°	0°	0°
O2	Frame	1	3	5	
	Location	(x21,y21)	(x23,y23)	(x25,y25)	
	Size	100%	100%	70%	
	Orientation	0°	0°	0°	
...	…				
On	Frame	5	6	…	m
	Location	(xn5,yn5)	(xn6,yn6)	…	(xnm,ynm)
	Size	100%	…	…	130%
	Orientation	0°	…	…	18°

Our experiments show that logging such objects in the profile produces better results; however, other methods may be use for object recognition.

C. *Streaming and Support*

The profile described in section B-2, not only can ease object search from video(s), but also it serves helpful when it comes to dealing with streaming media over various network quality and speed issues.

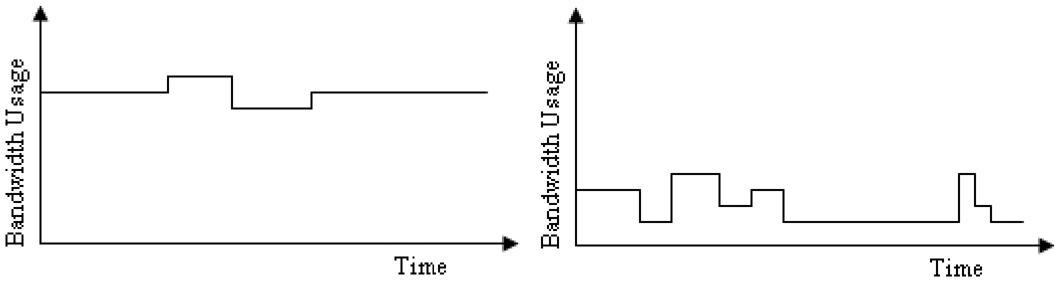
The objects are sending to the client on priority basis. The priority is defined by taking into account, both, the size of object and its order of appearance over various frame sequences within a video. These objects are then steered remotely from the server only by sending their occurrence information described in table 4.1.

Another issue, dealing with the time of triggering the object download and steering signals, arises by considering the mentioned procedure. This issue is dealt with by understanding the network quality and bandwidth, which can be specified by the user at the time of view. Otherwise, some information on the network bandwidth can help any server application, used for streaming the devised method of profile, to adjust automatically by reducing or retaining the streamed object quality to meet the viewing requirements. This, hence, controls and reduces the network traffic generated due to streaming.

D. *Results*

Fig 4.4 A bunch of frames retrieved using the key object from example in figure 4.3

The results were generated both separately and collectively for the whole process. The key frame lookup that results in gathering key objects present a more efficient information retrieval system. At the same time, the precision/recall measurements as described by Josef Sivic et al. [29] were confirmed and drew affective results. The key-object retrieval phase (described in section 2.1) produced very promising results, hence, populating the profile representation with very distinct results. The redundancy check for repeating key-objects (described at the end of section B-2) will also reduce the data size and refine the existence information for a particular object. Figure 4.4 shows the retrieval results for a particular key object from figure 4.3(a) when it is used as query by example (QBE) type searching.



(a) (b)

Fig 4.5 Bandwidth usage results while streaming video content over a LAN using (a) existing systems (b) proposed algorithm

According to readings generated through our network simulation software for our algorithm, we found out that the method may be equally useful for high definition (HD) and low bit rates. Figure 4.5 shows the remarkable difference between the bandwidth consumption through classic video streaming systems against the algorithm proposed in this research.

E. *Conclusion and Discussion*

This paper provides another effective application of QBE systems. The process devised in this paper in order to retrieve images from motion picture is based on very authentic research from some of the most adroit scientists. It can be clearly noticed that all of the potential key objects were mined. The search process is biased towards lightly textured regions. It may seem like the system carries a drawback of big monochrome objects cannot be logged due to lack of texture properties. However, it is noticed that occurrences may be saved from the shape perspective, which does provide reasonable amount of information about monochrome regions.

Also, the streaming mechanism described in this paper can serve as a food for though for systems, where bandwidth bottlenecks are needed to be avoided or reduced. Thus, we can say that this paper introduces a reduced profiling mechanism which simultaneously deals with streaming video quality and computation costs.

Hence, the proposed system can prove fruitful for low bandwidth requiring and consuming multimedia searching and streaming systems, as well as HD systems with less computation latency due to high bit rates and frame sizes.

V. *Defining a New Feature Set for Content Based Image Analysis using Histogram Refinement*

A. *Introduction*

Research in content based image retrieval is an active discipline, which is expanding in length and breadth. The deeper problems in computer vision, databases and information retrieval are being emphasized with the maturation of content based image retrieval technology. The web has huge collection of digital media which contains all sorts of digital content including still images, video, audio, graphics, animation etc. We concentrate on the visual content especially on still images. Therefore, to access and retrieve the visual information, we need specific human centered tools. As the amount of information available in the form of digital media continuously increases as well as the continuously increasing number of people with access to such image collections, we need image-retrieval schemes for visual information that are user-friendly and flexible.

One of the most effective ways of accessing visual data is content-based image retrieval (CBIR). The visual content such as color, shape and image structure is considered for the retrieval of images instead of an annotated text method. However, one major problem with CBIR is the issue of predicting the relevancy of retrieved images. This retrieval is based on various image features. The objective is the selection of such features which can provide accurate and precise query results.

Color histograms are widely used for retrieval of results based on queries. There are queries that require the comparing of the images on their overall appearance. For such queries, color histograms can be employed because they are very efficient regarding computations as well as they offer insensitivity to small changes regarding camera position. But the main problem with color histograms is their coarse characterization of an image. That may itself result in same histograms for images with different appearances. Color histograms are employed in systems such as QBIC [38], Chabot [39] etc. They all utilize the advantages of color histogram.

Various applications of content based image retrieval have been discussed by Arnold [2]. They make three broad categories based on user aims. First is the type of users who have no specific aim other than finding the interesting things. This search is refined by iterative methodology. In this case, the CBIR Systems need to be highly interactive because the specification may be an example or a sketch [40]. Relevance feedback can help in improving the result [41, 42].

Second is the type of search where user is looking for a precise copy of the image in mind or search for another image of the same object of which the user has an image. Examples can be search for stamps, art, catalogues etc. Third is the type of search where the user has an example and the search is for other elements of the same class. Hence, the user may have available a group of images and the search is for additional images of the same class [43]. Other factors affecting the retrieval include illumination problem, window size etc [44].

In this paper, a modified scheme based on histogram refinement [1] method is presented. The histogram refinement method provides that the pixels within a given bucket be split into classes based upon some local property and these split histograms are then compared on bucket by bucket basis just like normal histogram matching but the pixels within a bucket with same local property are compared. So the results are better than the normal histogram matching. So not only the color features of the image are used but also the spatial information is incorporated to refine the histogram.

The proposed algorithm starts with the estimation of each of the bins that is based on spatial coherence of pixels just like computed by Pass and Zabih [1]. A pixel is coherent if it is a part of some sizable similar colored region, otherwise it is incoherent. Then two more properties are calculated for each of the coherent and incoherent pixels in each bin. First the number of clusters is found for each case, i.e., coherent and incoherent case. Secondly, the average of each cluster is computed. Then the numbers of coherent and incoherent pixels are calculated. After these calculations, next only the coherent clusters are considered for additional features. This information includes size of the largest cluster, size of the smallest cluster, size of the median cluster and variance of the clusters. Additionally, we find the major axis length, minor axis length and angle of the ellipse for each of the largest, median and smallest cluster. For retrieval, we use the L_1 (Euclidean) distance for all the set of features. However, we divide the set of features for intelligent incremental content based retrieval.

The various sections in this paper are arranged in the following manner. The related work on content based image retrieval is discussed in section two. Sections three explains the pre-processing required for feature extraction. The selection and extraction of the set of features is provided in section four. Section five details the image retrieval approach. Results are provided in section six. The results obtained by testing the algorithm on a database of images. Conclusions are given is section seven followed by references.

B. *Related Work*

Content based image retrieval is an active field of research and various research groups are around the globe are actively involved trying to make breakthroughs in this field. We find many methods and algorithms related with CBIR research.

The best review of CBIR till 2000 is provided by Arnold et al. [2]. They reviewed 200 references in content based image retrieval. They discussed the working conditions of content-based retrieval: patterns of use, types of pictures, the role of semantics, and the sensory gap. They reviewed algorithms for retrieval sorted by color, texture, and local geometry. Similarity of pictures and objects in pictures is reviewed for each of the feature types, in close connection to the types and means of feedback the user of the systems is capable of giving by interaction. They also presented their view on: the driving force of the field, the heritage from computer vision, the influence on computer vision, the role of similarity and of interaction, the need for databases, the problem of evaluation, and the role of the semantic gap.

Histogram refinement method was first proposed by Pass and Zabih [1]. They partition histogram bins by the spatial coherence of pixels. They further refine it by using additional feature. The additional feature used is the center of the image. The center of the image is defined as the 75% centermost pixels. Their database consists of 14,554 images.

An unsupervised learning network to incorporate a self-learning capability into image retrieval systems was proposed by Paisarn [4]. The adoption of a self-organizing tree map (SOTM) is introduced, to minimize the user participation in an effort to automate interactive retrieval. In addition, a semiautomatic version is proposed to support retrieval with different user subjectivities. Image similarity is evaluated by a nonlinear model, which performs discrimination based on local analysis.

Zhang [6] discussed a generic Fourier descriptor (GFD) to overcome the drawbacks of existing shape representation techniques. Their proposed shape descriptor is derived by applying two-dimensional Fourier transform on a polar-raster sampled shape image. The acquired shape descriptor is application independent.

Special emphasis was made on content-based indexing and retrieval by Djeraba [8]. They try to add the generalization capability for indexing and retrieval. They propose exploiting the common associations among basic features (e.g., textures and colors) that the user cannot specify explicitly. They present an approach that discovers hidden associations among features during image indexing. The best associations are selected on the basis of measures of confidence. To reduce the combinatory explosion of associations, because images of the database contain very large numbers of colors and textures, they consider a visual dictionary that group together similar colors and textures. Thus, the visual dictionary summarizes the image features. An algorithm based on a clustering strategy creates the visual dictionary.

Jong-An, Bilal et al. [11, 12] provided shape description based on histogram based chain codes. Shape descriptions based on the traditional chain codes are very susceptible to small perturbations in the contours of the objects. Therefore, direct matching of traditional chain codes could not be used for image retrieval based on the shape boundaries from the large databases. Therefore they proposed histogram based chain codes which could be used for image retrieval. The modified chain codes matching are invariant to translation, rotation and scaling transformations, and have high immunity to noise and small perturbations.

One of the problems is the search in large collections of heterogeneous images. Vasileios [13] presented an image retrieval methodology for this problem. Their proposed approach employs a fully unsupervised segmentation algorithm to divide images into regions and endow the indexing and retrieval system with content-based functionalities. Low-level descriptors for the color, position, size, and shape of each region are subsequently extracted. These arithmetic descriptors are automatically associated with appropriate qualitative intermediate-level descriptors, which form a simple vocabulary termed object ontology.

The relevance feedback in CBIR was discussed by Paisarn [45]. They proposed a method that allows the users to directly modify the system characteristics by specifying their desired image attributes. This is done through the training samples. They used radial basis function (RBF) method for implementing an adaptive metric which progressively models the notion of image similarity through continual feedback from the users. They tested their algorithm using images compressed by wavelet transform and vector quantization coders.

A comparative study on CBIR using various shape descriptors was made by Zhang [46].

They considered several properties such as affine invariance, robustness, compactness, low computation complexity and perceptual similarity measurement. Against these properties, they studied several shape descriptors such as Fourier descriptors (FD), curvature scale space (CSS) descriptors (CSSD), Zernike moment descriptors (ZMD) etc.

Moreover, from the most recent advancements in CBIR, Tao et al. [47 – 52] have produced some high quality results through their vast research and analysis of CBIR applications and their advantages to modern technology.

C. *Proposed Algorithm & Feature Selection*

According to our research through various experiments, we have concluded that the features derived from histogram refinement method of Pass and Zabih [1] can serve as a promising essence for the image retrieval systems. However, there are also proposed some more features, which are proven to be useful based on experimental evidence from the results of our rigorous research.

1. *Pre-Processing*

We use the L_1 distance to compare two images I and I′. Using the L_1 distance, the jth bucket's contribution to the distance between I and I′ is:

After the image acquisition, the image needs to be pre-processed before feature extraction process. We consider the grayscale images for feature extraction. Therefore, first the image is converted to grayscale image using threshold. The RGB image is changed to grayscale image, also known as the intensity image, which is a single 2-D matrix containing values from 0 to 255. The RGB image is a 3-D matrix with values ranging from 0~255. This is done to reduce the number of computations.

For grayscale image, we do not consider all the 256 levels. Hence after the conversion from RGB to grayscale image, we perform quantization to reduce the number of levels in the image. Again the number of levels is reduced to increase the computational speed. The number of total levels is 256 in the grayscale image. We reduce the 256 levels to 16 levels in the quantized image. These levels are also known as bins. Hence, we get 16 bins after the application of quantization. By increasing the number of bins, the precision also increases. However, by experimentation we found that 16 bins give reasonable retrieval results.

For reducing the number of levels from 256 to 16, we use uniform quantization. The uniform quantization provides us with 16 separate bins with equal range. Figure 5.1 shows the block diagram of the algorithm. The steps in the pre-processing stage can be observed from the first three blocks in figure 5.1.

2. *Features from Quantized Bins*

We use color refinement method for feature extraction from the quantized bins. Color refinement is based on histogram refinement [1] method. The histogram refinement method provides that the pixels within a given bucket be split into classes based upon some local property and these split histograms are then compared on bucket by bucket basis and the pixels within a bucket are compared.

Color histogram buckets are partitioned based on spatial coherence just like computed by Pass and Zabih [1]. A pixel is coherent if it is a part of some sizable similar colored region, otherwise it is incoherent. So the pixels are classified as coherent or incoherent within each color bucket. If a pixel is part of a large group of pixels of the same color which form at least one percent of the image then that pixel is a coherent pixel and that group is called the coherent group or cluster. Otherwise it is incoherent pixel and the group is incoherent group or cluster.

Then two more properties are calculated for each bin. First the numbers of clusters are found for each case, i.e., coherent and incoherent case in each of the bin. Secondly, the average of each cluster is computed. So for each bin, there are six values: one each for percentage of coherent pixels and incoherent pixels, number of coherent clusters and incoherent clusters, average of coherent cluster and incoherent cluster. This is shown in the form of block diagram in figure 1.

These values are calculated by computing the connected components. A connected component C is a maximal set of pixels such that for any two pixels p, $p' \in C$, there is a path in C between p and p'. Eight connected neighbors method is used for computing connected component. A pixel is classified as coherent if it is part of a connected component whose size is equal to or greater than τ (τ = 5%

of the image size). Otherwise it is classified as incoherent. And the connected component is classified as coherent connected component if it equals or exceeds 'τ'. Otherwise it is classified as incoherent connected component. Finally the average for coherent and incoherent connected component is calculated.

For each discretized color j, let the number of coherent pixels as α_j, the number of coherent connected components as $C_{\alpha j}$ and the average of coherent connected compo-nent as $\mu_{\alpha j}$. Similarly, let the number of incoherent pixels as β_j, the number of incoherent connected components as $C_{\beta j}$ and the average of incoherent connected component as $\mu_{\beta j}$. For each discretized color j, the total number of pixels are $\alpha_j+\beta_j$ and the color histogram summarizes the image as $<\alpha_1+\beta_1, \ldots, \alpha_n+\beta_n>$.

Since there are 4 bins, so we get total of 6 features for each bin. Therefore, a total of 24 features are considered for image retrieval.

3. *Features from Coherent Clusters*

At this stage, additional features are selected for content based image retrieval. These additional features are based on the coherent clusters only. At this stage, incoherent clusters are ignored. The reason for selecting coherent clusters only is based on the assumption that objects of significant size are considered only, i.e., cluster size is equal to or greater than 5% of the image.

Four features are selected among the coherent clusters. Three of them are based on the size of the clusters while one is statistical in nature. They are:

(a) Size of largest cluster in each bin
(b) Size of median cluster in each bin
(c) Size of smallest cluster in each bin
(d) Variance of clusters in each bin

Let us denote the largest cluster in each bin as $L_{\alpha j}$, the median cluster in each bin as $M_{\alpha j}$, the smallest cluster in each bin as $S_{\alpha j}$ and variance of clusters in each bin as $V_{\alpha j}$. Since there are 4 bins, so we get additional 4 features for each bin.

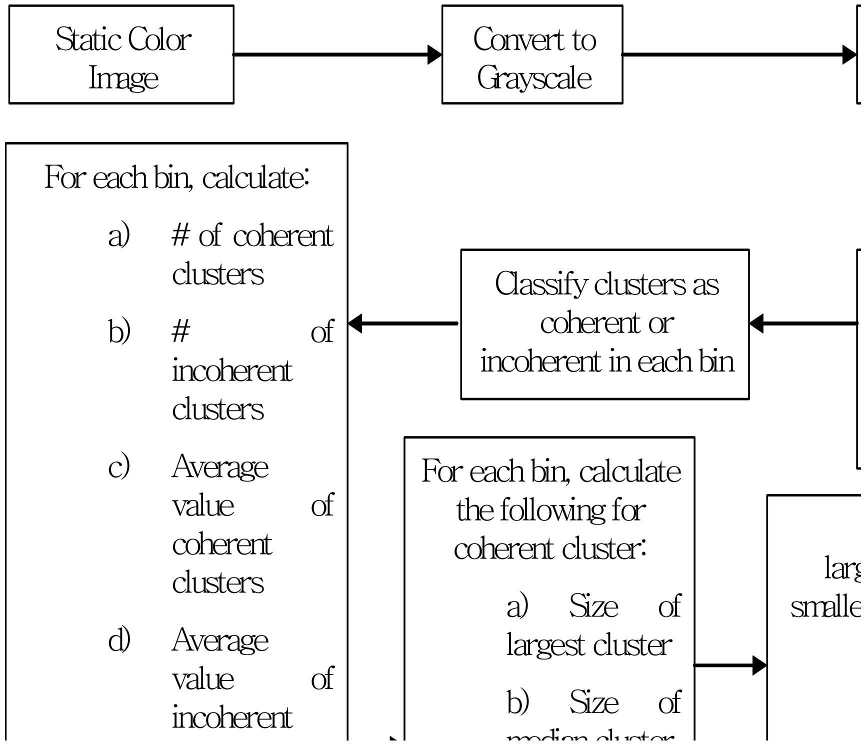


Fig 5.1: Block diagram of the feature extraction algorithm

Therefore, a total of 40 additional features are considered for image retrieval. Considering section C-2, initially 6 features per bin are selected for image retrieval and later 4 additional features per bin are considered in this section for refining the result of image retrieval. These features are shown in figure 5.1.

4. *Additional Features based on Size of Cluster*

At this stage, another set of features are selected for content based image retrieval. Again these additional features are based on the coherent clusters only. The following features are selected for retrieval for each of the largest cluster, median cluster and smallest cluster in each of the bin:

(a) Major axis length

(b) Minor axis length

(c) Angle between x-axis and major axis of ellipse

Let us denote the major axis length of the largest cluster in each bin as $MAL_{\alpha Lj}$, the minor axis length of the largest cluster in each bin as $MIL_{\alpha Lj}$, and angle as $Ang_{\alpha Lj}$. Similarly, let us denote the major axis length of the median cluster in each bin as $MAL_{\alpha Mj}$, the minor axis length of the median cluster in each bin as $MIL_{\alpha Mj}$, the angle of median cluster as $Ang_{\alpha Mj}$, the major axis length of the smallest cluster in each bin as $MAL_{\alpha Sj}$, the minor axis length of the smallest cluster in each bin as $MIL_{\alpha Sj}$ and the angle of smallest cluster as $Ang_{\alpha Sj}$. This is shown in figure 5.1.

Since there are 16 bins, so we get additional 9 features for each bin. Therefore, a total of 144 additional features are considered for image retrieval. Considering section D-1, initially 6 features per bin are selected for image retrieval and later 4 additional features per bin are considered in section D-2 and now further 9 features per bin are selected for refining the result of image retrieval.

D. *Image retrieval*

Image retrieval is done in three stages hence we can call it incremental image retrieval approach. In the first stage, the features defined in section D-1 are considered for retrieval while in stage 2; the features defined in section D-2 are taken into account for image retrieval. Finally, the features defined in section D-3 are used for the final retrieval result. The first stage gives us a coarse result while stage 2 refines the result obtained in stage 1. Finally, the stage 3 provides us with the most relevant retrieval results. Therefore, the result is more relevant and accurate image retrieval from the image databases.

1. *Stage 1*

The features obtained in section D-1 are used for retrieval at first level. We use the L_1 distance to compare two images I and I′. Using the L_1 distance, the j_{th} bucket's contribution to the distance between I and I′ is:

$$\Delta_1 = |(\alpha_j - \alpha'_j) + (\beta_j - \beta'_j)| \qquad ...(1)$$

$$\Delta_2 = |(C_{\alpha j}-C'_{\alpha j})+(C_{\beta j}-C'_{\beta j})| \quad ...(2)$$
$$\Delta_3 = |(\mu_{\alpha j}-\mu'_{\alpha j})+(\mu_{\beta j}-\mu'_{\beta j})| \quad ...(3)$$

So we get the initial retrieval result with this method. In original scheme [43], only equation (1) is used. Also equations (1) to (3) provide for incorporating the scalability and remove problems identified by Huang et al. [46] which cannot be removed by only using CCV (Color Coherent Vector) defined in [43].

2. *Stage 2*

This level of retrieval is used for further refining the result obtained in section D-1. The additional features obtained in section D-2 are used at this level of retrieval. Again we use the L_1 distance to compare two images I and I′. Using the L_1 distance, the jth bucket's contribution to the distance between I and I′ is:

$$\Delta_4 = |(L_{\alpha j}-L'_{\alpha j})| \quad ...(4)$$
$$\Delta_5 = |(M_{\alpha j}-M'_{\alpha j})| \quad ...(5)$$
$$\Delta_6 = |(S_{\alpha j}-S'_{\alpha j})| \quad ...(6)$$
$$\Delta_7 = |(V_{\alpha j}-V'_{\alpha j})| \quad ...(7)$$

3. *Stage 3*

This level of retrieval is used for final retrieval of images the result obtained in section D-2. The additional features obtained in section D-3 are used at this level of retrieval. Again we use the L_1 distance to compare two images I and I′. Using the L_1 distance, the j_{th} bucket's contribution to the distance between I and I′ is:

$$\Delta_8 = |(MAL_{\alpha Lj}-MAL'_{\alpha Lj})| \quad ...(8)$$
$$\Delta_9 = |(MIL_{\alpha Lj}-MIL'_{\alpha Lj})| \quad ...(9)$$
$$\Delta_{10} = |(Ang_{\alpha Lj}-Ang'_{\alpha Lj})| \quad ...(10)$$
$$\Delta_{11} = |(MAL_{\alpha Mj}-MAL'_{\alpha Mj})| \quad ...(11)$$
$$\Delta_{12} = |(MIL_{\alpha Mj}-MIL'_{\alpha Mj})| \quad ...(12)$$

$$\Delta_{13} = |(Ang_{aMj} - Ang'_{aMj})| \quad ...(13)$$
$$\Delta_{14} = |(MAL_{aSj} - MAL'_{aSj})| \quad ...(14)$$
$$\Delta_{15} = |(MIL_{aSj} - MIL'_{aSj})| \quad ...(15)$$
$$\Delta_{16} = |(Ang_{aSj} - Ang'_{aSj})| \quad ...(16)$$

E. *Results*

For testing the proposed algorithm, we used databases initially carrying 1,000, and then with 10,000 images provided by James S. Wang et al. [14, 25]. The databases carry large sets of similar class images; for example, 30 dinosaurs, 60 human, 50 flowers, 40 satellite images, etc. First the images were preprocessed and converted to grayscale images. Then the images were quantized and the features described in section D-1 were calculated. Also, the features described in section D-2 and D-3 were calculated.

These features were stored in a vector form. Figure 5.2 shows one of the image from the database, its corresponding grayscale image and then the corresponding quantized images. This algorithm is based on coherent and incoherent clusters. Figure 5.3 and 5.4 shows the coherent and incoherent clusters for one of the images from the database.

One important point is the benchmarking of CBIR solutions. This is still an open problem and the research community is actively working on the evaluation criteria for this purpose. There are recommendations issued by the technical committee from the International Association for Pattern Recognition (IAPR) regarding benchmarking of the visual information retrieval. Based on those recommendations, we devised the following guidelines for the implementation of our algorithm:

a) We use the image database that is freely available to researchers and is free from any conditions or restrictions.

b) To minimize the dependence on hardware, the entire image collection is stored in main memory during evaluation.

c) The initial number of images in the database is 1000.

d) Images are in JPEG format.

e) 10 different types of objects are available in the database.

f) A set of about 30 evaluation queries are designed covering all the objects in the database.

g) For each query, there are known answers that do not exceed 10 images or 1% of the database.

h) Images returned for possible browsing are 8 images for each query.

i) The measures used for evaluating the system include False Acceptance ratio (FAR) and precision using visual inspection.

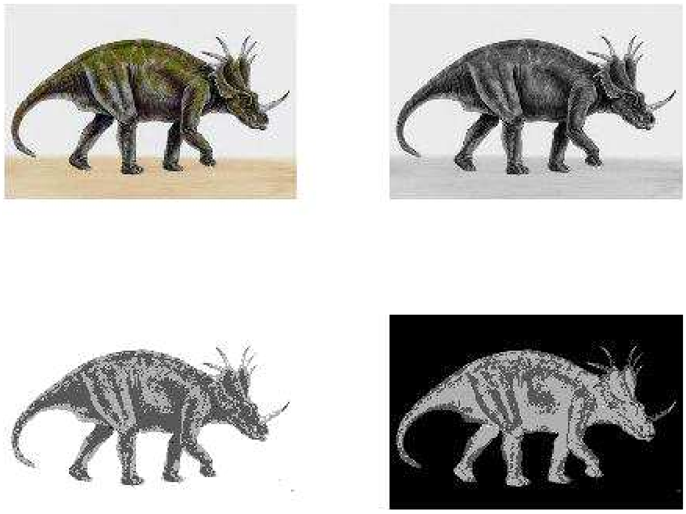


Fig 5.2 One of the image from the database, converted to grayscale & quantized

Based on the above guidelines, there are ten different object categories including people, sea, building, bus, dinosaur, elephant, flower, horse, mountain and food. A set of queries are designed as query 1, 2, 3, ..., 10 corresponding to these ten image categories. Table 5.1 shows the results of FAR with all the ten queries for comparing the proposed method with the traditional Pass & Zabih method (TR). The results are shown for various options of the proposed method, namely, 4 bins (M4), 8 bins (M8) and 16 bins (M16). If the FAR is less than 50% then the result is declared as unsuccessful. In table 5.1, 'F' indicates that the method is unsuccessful for that specific query. It can be seen from table 5.1 that the proposed method with 16 bins show exceptionally good results followed by 8 bins. The traditional method is unsuccessful for all queries. In addition, proposed method with 4 bins is also unsuccessful for majority of queries hence it is not recommended to use 4 bins.

Table 5.1 Comparison using FAR in percentage

Query	*TR*	*M4*	*M8*	*M16*
1 - People	F	F	~35%	~20%
2 - Sea	F	F	~20%	~10%
3 - Building	F	F	~50%	~20%
4 - Bus	F	F	~20%	~10%
5 - Dinosaur	F	~33%	~10%	~10%
6 - Elephant	F	F	~20%	~20%
7 - Flower	F	~25%	~20%	~25%
8 - Horse	F	F	~10%	~5%
9 - Mountain	F	F	~20%	~20%
10 - Food	F	~37%	~10%	~25%

Fig 5.3 Coherent clusters for one of the images in the database

The results were compared with the L_1 distance as described in section 4. First, we used equation (1) to equation (3) for identifying the similarity between images. Then we used equation (4) to equation (7) to further refine the results. Finally, we used equations (8) to (16) to retrieve the closest 8 matches. Figures 5.5 to 5.8 show some of the query images and the retrieved results for the proposed method. Each figure contains the query image on the top followed by 8 closest matches.

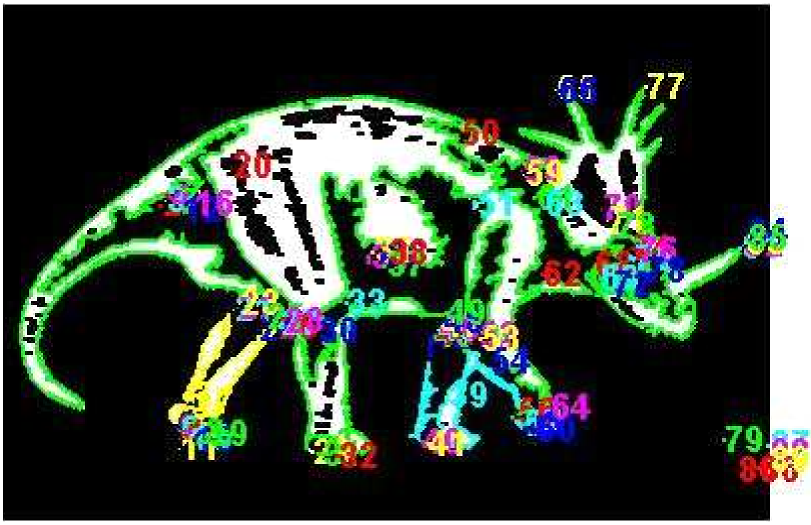


Fig 5.4 Incoherent clusters for one of the images in the database

The number on the top of matching images is the distance measurement from query image. Also, some noise was added to the query images in order to test the strength of the proposed algorithm, which can also be noticed by a non-zero distance of the closest match.

The precision, recall, and fallout values have been calculated and some of them are mentioned in table 2 against a few classes of similar type images from James S. Wang et al. [18, 19].

Table 5.2 Precision, recall and fallout

	Precision	*Recall*	*Fallout*
Dinosaurs	0.98	1	0.02
Humans	0.92	0.98	0.09
Buses	0.94	0.95	0.05
Horses	0.89	0.96	0.10

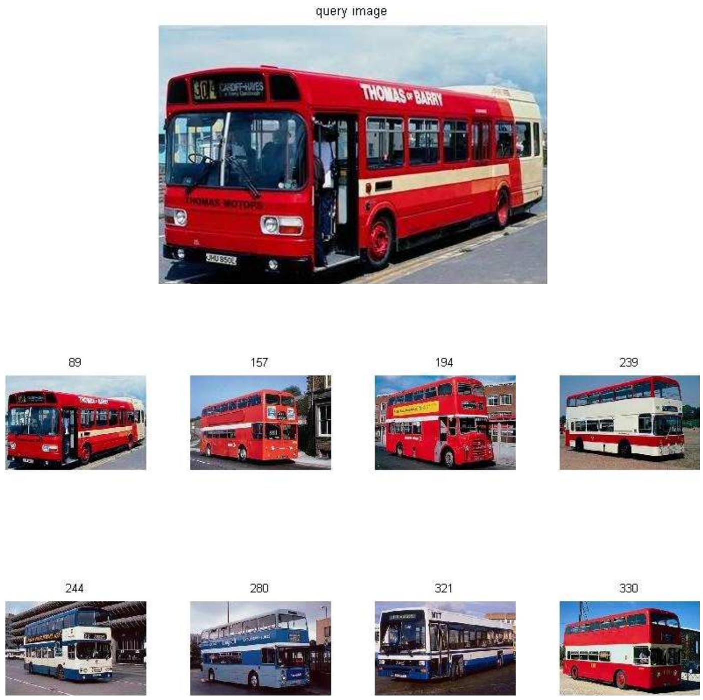


Fig 5.5 An example of "bus" image retrieval from the database

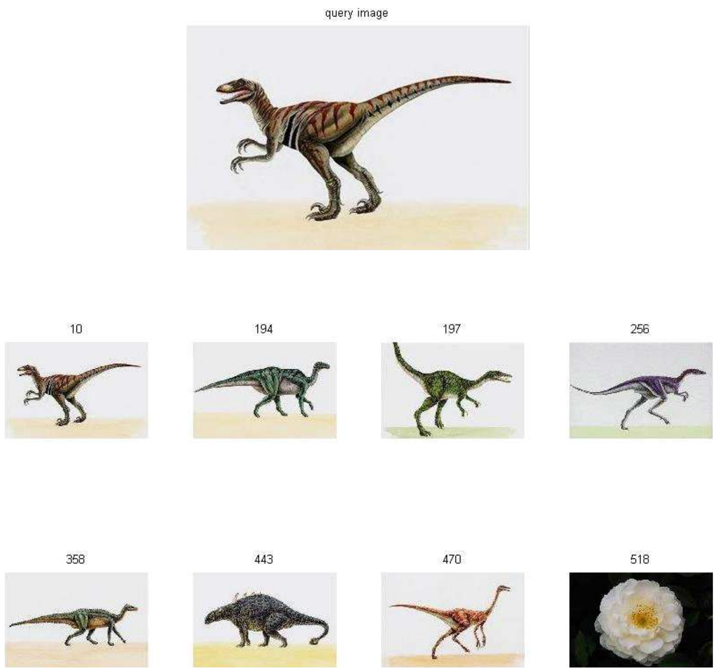


Fig 5.6 An example of "dinosaur" image retrieval from the database

Fig 5.7 An example of "food" image retrieval from the database

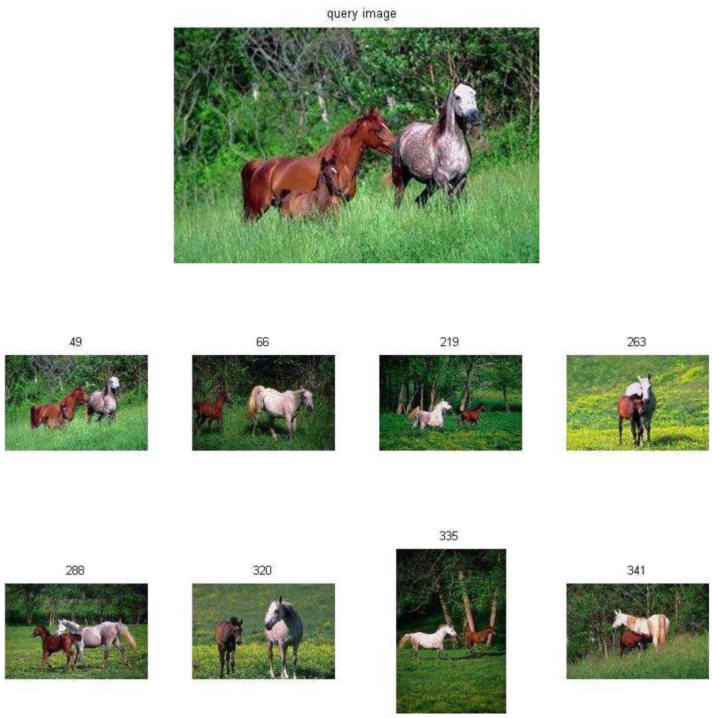


Fig 5.8 An example of "hosres" image retrieval from the database

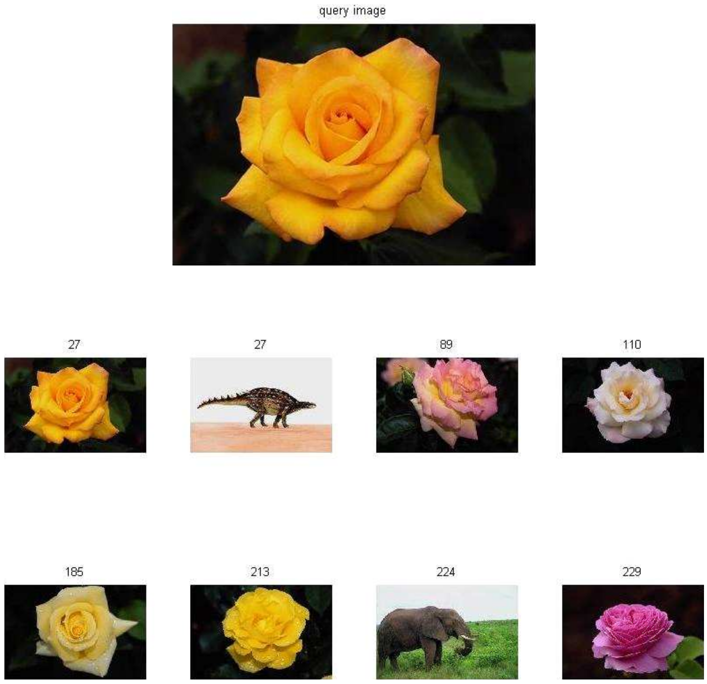


Fig 5.9 An example of "rose" image retrieval from the database

F. *Discussion & conclusion*

In this paper, we have proposed an algorithm that is based on color histogram. We have shown that the features obtained using this algorithm is quite useful for relevant image retrieval queries. The feature selection is based on the number, color and shape of objects present in the image. The grayscale values, mean, variance, and various sizes of the objects are considered as appropriate features for retrieval. These features are independent of image orientation. The features are defined in section D-1, D-2 and section D-3. Color refinement method takes care of the color as well as the spatial relation feature. The shape features are extracted in section D-2 and D-3 from the color based features defined in section D-1.

We have also presented a three stage approach for image retrieval. At first stage, the initial set of features described in section D-1 is used for image retrieval. At next stages, the additional features described in section D-2 and D-3 are considered for retrieval. Hence, this approach is computationally efficient and provides refined result. Results show that the algorithm presented in this paper provides very relevant retrieval results. We recommend using the proposed algorithm with 16 bins.

VI. *Petite Feature Set Defining Solid Structural and Color Assets in CBIR Processes*

A. *Introduction*

Query by example (QBE) or content based image retrieval (CBIR) has been a key focus throughout the world of image analysis, processing, and matching. The visual content such as color, shape and image structure is considered for the retrieval of images instead of an annotated text method. One major problem with CBIR is the issue of predicting the relevancy of retrieved images. Usually, the process is based on various basic image features. The objective is the selection of such features which can provide accurate and precise query results.

The most common method for predicting the characteristics of the image is Color histograms, which are widely used for retrieval of results based on queries that require the comparison of images on their overall appearance. Color histograms can be employed because they are very efficient regarding computations, as well as they offer insensitivity to small changes regarding camera position, but the main problem with color histograms is their coarse characterizations. That may imply same histograms for images with different appearances. Color histograms are employed in systems such as Chabot [39], QBIC [38] etc. They all utilize the advantages of color histogram.

Various applications of content based image retrieval have been discussed by Smeulders [2]. They make three broad categories based on user aims. First is the type of users who have no specific aim other than finding the interesting things. This search is refined by iterative methodology. In this case, the CBIR Systems need to be highly interactive because the specification may be an example or a sketch [40]. Relevance feedback can help in improving the result [41, 42].

Second is the type of search where user is looking for a precise copy of the image in mind or search for another image of the same object of which the user has an image. Examples can be search for stamps, art, catalogues etc. Third is the type of search where the user has an example and the search is for other elements of the same class. Hence, the user may have available a group of images and the search is for additional images of the same class [43].

This paper provides a brief analysis of histogram independent structural features defining identity assets of an RGB type image. The R, G, and B channel local properties of the image are individually and collectively analyzed, and results are fetched after quantitative analysis of the extracted features.

The proposed algorithm states that the total number of colored objects, horizontal edges, vertical edges, and slanting edges for each R, G, and B component can short list the more likely matches, which can be further analyzed on their overall properties in order to further improve precision and recall. The image database provided by James S. Wang et al. [14, 25] was used in order to test the proposed method.

B. *Related Work*

The best review of CBIR till 2000 is provided by Arnold et al. [2]. They reviewed 200 references in content based image retrieval. They discussed the working conditions of content-based retrieval: patterns of use, types of pictures, the role of semantics, and the sensory gap. They reviewed algorithms for retrieval sorted by color, texture, and local geometry. Similarity of pictures and objects in pictures is reviewed for each of the feature types, in close connection to the types and means of feedback the user of the systems is capable of giving by interaction. They also presented their view on: the driving force of the field, the heritage from computer vision, the influence on computer vision, the role of similarity and of interaction, the need for databases, the problem of evaluation, and the role of the semantic gap.

An unsupervised learning network to incorporate a self-learning capability into image retrieval systems was proposed by Paisarn [4]. The adoption of a self-organizing tree map (SOTM) is introduced, to minimize the user participation in an effort to automate interactive retrieval. In addition, a semiautomatic version is proposed to support retrieval with different user subjectivities. Image similarity is evaluated by a nonlinear model, which performs discrimination based on local analysis.

Zhang [46] discussed a generic Fourier descriptor (GFD) to overcome the drawbacks of existing shape representation techniques. Their proposed shape descriptor is derived by applying two-dimensional Fourier transform on a polar-raster sampled shape image. The acquired shape descriptor is application independent. Special emphasis was made on content-based indexing and retrieval by Djeraba [12]. They try to add the generalization capability for indexing and retrieval. They propose exploiting the common associations among basic features (e.g., textures and colors) that the user cannot specify explicitly. They present an approach that discovers hidden associations among features during image indexing. The best associations are selected on the basis of measures of confidence. To reduce the combinatory explosion of associations, because images of the database contain very large numbers of colors and textures, they consider a visual dictionary that group together similar colors and textures. Thus, the visual dictionary summarizes the image features. An algorithm based on a clustering strategy creates the visual dictionary.

Jongan Park, et al. [11, 53] provided shape description based on histogram based chain codes. Shape descriptions based on the traditional chain codes are very susceptible to small perturbations in the contours of the objects. Therefore, direct matching of traditional chain codes could not be used for image retrieval based on the shape boundaries from the large databases. Therefore they proposed histogram based chain codes which could be used for image retrieval. The modified chain codes matching are invariant to translation, rotation and scaling transformations, and have high immunity to noise and small perturbations.

One of the problems is the search in large collections of heterogeneous images. The relevance feedback in CBIR was discussed by Paisarn [4]. They proposed a method that allows the users to directly modify the system characteristics by specifying their desired image attributes. This is done through the training samples. They used radial basis function (RBF) method for implementing an adaptive metric which progressively models the notion of image similarity through continual feedback from the users. They tested their algorithm using images compressed by wavelet transform and vector quantization coders.

A comparative study on CBIR using various shape descriptors was made by Zhang [11]. They considered several properties such as affine invariance, robustness, compactness, low computation complexity and perceptual similarity measurement. Against these properties, they studied several shape descriptors such as Fourier descriptors (FD), curvature scale space (CSS) descriptors (CSSD), Zernike moment descriptors (ZMD) etc.

C. *Method*

Fig 6.1 Example of an RGB image

The process takes place in two levels of retrieval, described below in sections C-1 and C-2. Euclidian, or L_1 distance is used for similarity measurement for each level.

1. *Processing of individual RGB features of an image*

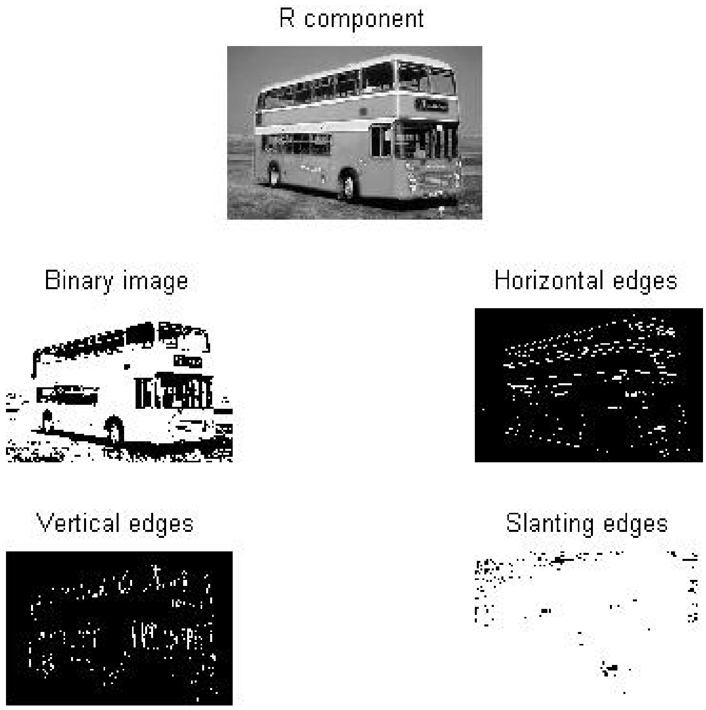


Fig 6.2 Example of R (red) image component processing from figure 6.1

The image is acquired and split into its R (red), G (green), and B (blue) components. Each component is processed separately through same process, which is described as:

1. Image component is converted to binary or logical object under a fixed or dynamic threshold according to its appearance.
2. Total number of connected components found in the image is noted.
3. The percentage of ON area with respect to the whole image area is also calculated.
4. Horizontal, vertical, and slanting edges are found for the image component under a fixed or dynamic threshold according to its appearance.
5. Total number of horizontal, vertical, and slanting edges is noted.

After the image passes through the above process as shown in figure 6.2, five features are calculated for each R, G, and B image component, and hence, fifteen (=5x3) features are stored. These features prove useful for the first level of image retrieval.

For any two images I and Q, let Icolor(oc) and Qcolor (oc) be the number of connected components, let Icolor (hc) and Qcolor (hc) be the number of horizontal edge components, let Icolor (vc) and Qcolor (vc) be the number of vertical edge components, and let Icolor (sc) and Qcolor (sc) be the number of slanting edges found in the color components in the images, where color={R, G, B}. Hence, the similarity is measured by the following twelve equations.

$\triangle R(oc) = | IR(oc) - QR(oc) |$ … (1)

$\triangle G(oc) = | IG(oc) - QG(oc) |$ … (2)

$\triangle B(oc) = | IB(oc) - QB(oc) |$ … (3)

$\triangle R(hc) = | IR(hc) - QR(hc) |$ … (4)

$\triangle G(hc) = | IG(hc) - QG(hc) |$ … (5)

$\triangle B(hc) = | IB(hc) - QB(hc) |$ … (6)

$\triangle R(vc) = | IR(vc) - QR(vc) |$ … (7)

$\triangle G(vc) = | IG(vc) - QG(vc) |$ … (8)

$\triangle B(vc) = | IB(vc) - QB(vc) |$ … (9)

$\triangle R(sc) = | IR(sc) - QR(sc) |$ … (10)

$\triangle G(sc) = | IG(sc) - QG(sc) |$ ... (11)

$\triangle B(sc) = | IB(sc) - QB(sc) |$ … (12)

Similarly, for any two images I and Q, let Icolor(pc) and Qcolor (pc) be the percentage of connected components to the size of image color component, where color={R, G, B}, and the similarity can be measured by the following three equations.

$\triangle R(pc) = | IR(pc) - QR(pc) |$ … (13)

$\triangle G(pc) = | IG(pc) - QG(pc) |$ … (14)

$\triangle B(pc) = | IB(pc) - QB(pc) |$ … (15)

2. *Processing of overall appearance features of an image*

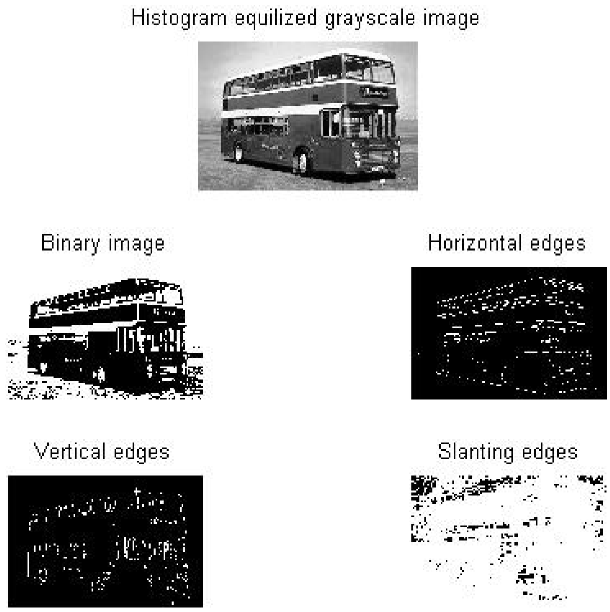


Fig 6.3 Example of histogram equalized grayscale image processing from figure 6.1

The results obtained after the image goes through the first step as described in C-1, are further refined by an iteration, which compares images on their overall appearances and basic spatial characteristics. This process carries following steps.

1. The RGB Image is converted to grayscale.
2. Histogram equalization is applied to the grayscale image.
3. This histogram equalized grayscale image is converted to binary or logical object under a fixed or dynamic threshold according to its appearance.
4. Total number of connected components found in the image is noted.
5. The percentage of ON area with respect to the whole image area is also calculated.
6. Horizontal, vertical, and slanting edges are found for the equalized image under a fixed or dynamic threshold according to its appearance.
7. Total number of horizontal, vertical, and slanting edges is noted.

Therefore, five more features are calculated for the histogram equalized grayscale image as shown in figure 6.3, which show promising results in the second level of image retrieval process.

For any two images I and Q, let I(oc) and Q(oc) be the number of connected components, let I (hc) and Q(hc) be the number of horizontal edge components, let I(vc) and Q(vc) be the number of vertical edge components, and let I(sc) and Q(sc) be the number of slanting edges found in the color components in the images. Hence, the similarity is measured by the following four equations.

$\triangle OC(oc) = | I(oc) - Q(oc) |$ … (16)

$\triangle HC(oc) = | I(hc) - Q(hc) |$ … (17)

$\triangle VC(oc) = | I(vc) - Q(vc) |$ … (18)

$\triangle SC(oc) = | I(sc) - Q(sc) |$ … (19)

Similarly, for any two images I and Q, let I(pc) and Q (pc) be the percentage of connected components to the size of image color component, and the similarity can be measured by the following three equations.

$\triangle PC(pc) = | I(pc) - Q(pc) |$ … (13)

D. *Results*

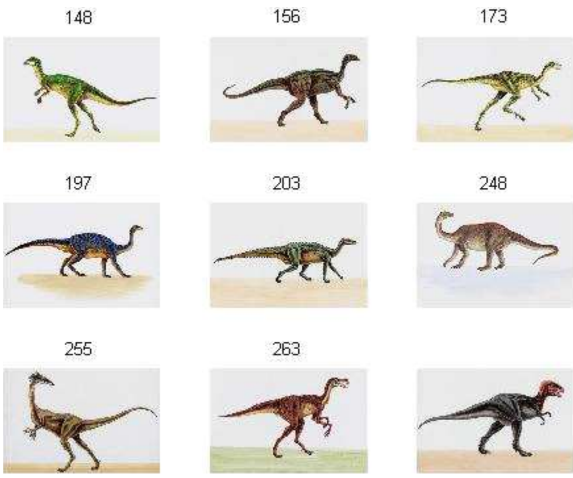


Fig 6.4 Results for query for dinosaur type images

Several experiments performed using the proposed method have generate quality results with very precise outputs. The results generated in the first level of retrieval, described in section C-1, are provided as input to the second level process, described in section C-2. Experiments have proved that the fallouts found in the first level are improved in the second level, in general.

Fig 6.5 Results for query for yellow flower type images

It was also noticed that using both structural and color features confer a very precise color and shape matching instrument. Figures 6.4, 6.5 and 6.6 show a remarkable matching result for dinosaur and flower sort of input images. Especially, figure 6.5 and 6.6 show a very precise color and structure matching.

Fig 6.6 Results for query for red flower type images

However, the algorithm has also proven its worth while searching for an image carrying a large number of colored objects and structures as shown in figures 6.7 and 6.8.

Also, it can be noticed from figure 6.8 that the algorithm has visibly searched for similar structures and colors, and the bus colors gradually shift from R through G towards B domain.

It was also noticed that the algorithm also works well for searching a very large database of images, and the extraction and retrieval process too is very quick. It was also noticed that the results from this research provide incredible values for precision and recall followed by very less fallouts.

Fig 6.7 Results for query for horses type images

Fig 6.8 Results for query for bus type images

E. *Conclusions*

A two level incremental approach is proposed in this paper which caters retrieval of images based on image queries. A set of very basic features are proposed for content based image retrieval in this paper. These features, if added as a part of image header, can prove vital identity elements for any RGB image, which could also be used during the retrieval process at basic levels. Moreover, the numerous values of histogram matching algorithms me be useful, but, they utilize very high computation and processing power, on the other hand, the fifteen values of the first and five values of the second level retrieval as described in this paper utilize less CPU power. The feature selection in the proposed algorithm is based on the color and structure of objects present in the image, and a very petite and basic set of retrieval assistants is generated.

VII. *Sum of Values of Local Histograms for Image Retrieval*

A. *Introduction*

The humungous and ever increasing variety of multimedia data usually requires computationally effective and quick indexing and retrieval systems. Images and movies are the most commonly used retrieval multimedia elements. User friendliness demands a multimedia data retrieval system that can retrieve images and videos as fast as text data retrievals on the WWW. Content Based Image Retrieval (CBIR) [2, 15-17] is a highly focused, yet difficult, research topic in image analysis, and retrieval technology. CBIR research usually involves visual contents (color, shape, texture etc.) in order to retrieve images from huge image databases according to the visual queries from the user.

The CBIR process consists of calculating a feature vector that characterizes some image properties, and stored in the image feature database. The user provides a query image, and the CBIR system computes the feature vector for it, and then compares it with the particular image feature database images. The relevance comparison is done by using some distance measurement technique, and the minimum or permissible distances are the metrics for the matched or similar images. The features vector should be able enough to fully characterize image structural and spatial properties, which retrieve the similar images from the image database.

Color is one of the most reliable visual features that are also easier to implement in image retrieval systems. Color is independent of image size and orientation, because, it is robust to background complication. Color histogram is the most common technique for extracting the color features of colored images [5, 10, 18-20]. Color histogram tells the global distribution of colors in the images. It involves low computation cost and it is insensitive to small variations in the image structure. However, color histogram hold two major shortcomings. They are unable to fully accommodate the spatial information, and they are not unique and robust. Two dissimilar images with similar color distribution produce very similar histograms. Moreover, similar images of same point of view carrying different lighting conditions create dissimilar histograms.

Many researchers suggested the use of color correlogram for avoiding inconsistencies involving the spatial information [10]. Multiresolution histograms [20, 21] are also suggested to ameliorate image retrieval process. Gaussian filtering may also be used for multiresolution decomposition of an image [21]. This paper tends to solve the second problem.

B. *Method*

The proposed method strives for a light weight computation with effective feature extraction. Digital images undergo the following process in order to produce an effective feature vector describing an eminent feature set targeted to avoid the lack of robustness of a common histogram.

1. *Pre-processing*

Fig 7.1 An image selected at random as a sample from a collection of images.

RGB and indexed images carry high values that require more computation time. Hence, the images are converted to grayscale in order to reduce the vast spectrum of indexed images or the 3D components of RGB to a 2D component carrying values between 0 and 255 (containing the end points). This process promises reduction in the computation time and power required for extracting features from an image. The resulting image undergoes histogram equalization in order to enhance contrast of values of an image by generating its flat histogram. Figure 7.2 show the pre-processing applied to the image shown in figure 7.1.

2. *Splitting histogram values by fixed frequency range*

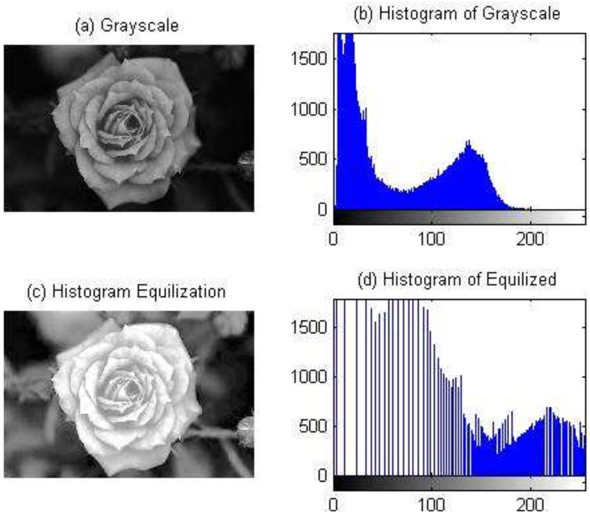


Fig 7.2 (a) Grayscale (intensity) representation of image in figure 7.1 (b) Histogram of the grayscale image (c) Grayscale image after histogram equalization (d) Histogram of the flat (equalized) image

The histogram equalized image is split into four fixed bins in order to extract more distinct information from it. The frequencies of 256 values of gray scale are split into sixteen (16) bins carrying 16 values each (0~15, 16~31, 32~ 47, 48~63, and so forth). This is done by turning off the gray values of image which do not lie between the four bins. This gives four images carrying objects which lie in the specific frequency ranges, and all different from each other. This provides a better illustration of image segments and simplifies the computation of features for the distinct portion of image. An example of the mechanism is shown in figure 7.3, which clearly shows the distribution of frequencies in various bins.

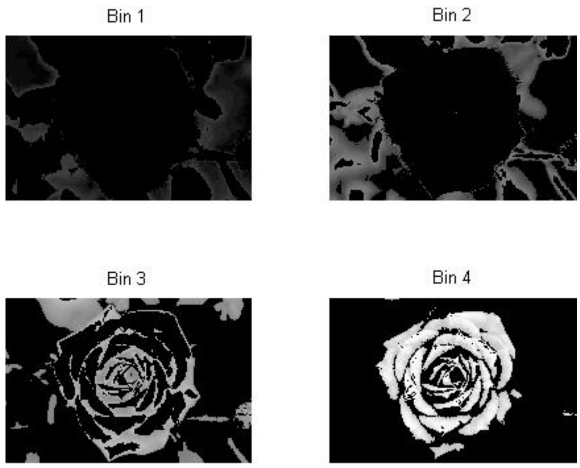


Fig 7.3 Bins generated from the image in figure 7.2(c)

3. *Calculating sum of values from the bin sub-divisions*

The values from each bin are summed together and noted down against each bin. This provides a more distinctive set of values for an image. Therefore, these sums from the local regions give a, somewhat, robust information from the histograms.

4. *Storing information*

Table 7.1 Sum of values listed against bins

Bin#	1	2	3	4	5	6	7	…	16
Sum	122	334	22	1	334	5	67	…	231

The information from bins is stored in the form of a feature vector as shown in table 1.

5. *Similarity measurement*

The distance between the feature vector of query image and the feature vector of image stored in the database can be calculated by using L_1 or Euclidean distance. The distance measurement process comprises of three steps. The feature vectors are subtracted, at first, under simple subtraction rules. Secondly, their sum is calculated. Finally, the third step comprises of sorting the absolute values of the sums obtained in the second step.

For any two vector images Q and D, let their corresponding feature vectors be 1 row and 16 column matrices. Let, feature vector of image Q be FV(Q). Similarly, let the feature vector of image D be FV(D). According to Euclidean distance algorithm described above, let:

$$\text{Distance}(Q,D) = FV(Q) - FV(D) \qquad \cdots (1)$$

Where, Distance(Q,D) is a vector containing the result of difference calculation as described in equation 1. The components of the resulting matrix are summed together and the absolute value of this sum is considered.

These absolute values are sorted in order to display the top matches.

C. *Results*

The process described in section 2 is rigorously tested against various conditions and types of images. The algorithm of the paper has been tested against the database of James Z. Wang et al. [14, 25]. Some minor changes to some of the images were made in order to test the robustness (figure 7.7; image Hue, saturation, color values, and object positions are altered). Tests prove that the worth of this algorithm because it can be useful in three ways.

1. Less computation is required, which makes the feature extraction and retrieval very fast.
2. This sort of feature vector representation is independent of differences in displacements and hue of images.

3. Complex images, which carry many objects of a very wide range of color values, can also be retrieved to a considerable extent

The process described in section 2 is rigorously tested against various conditions and types of images. The algorithm of the paper has been tested against the database of James Z. Wang et al. [14, 25]. Some minor changes to some of the images were made in order to test the robustness (figure 7.8; image Hue, saturation, color values, and object positions are altered). Tests prove that the worth of this algorithm because it can be useful in three ways.

Figure 7.4-7.6 clearly show the robustness of the results generated from the algorithm devised in this paper. Figure 7.4 carries the highest degree of likeness of results generated by this algorithm. Less likeliness to the query image can be noticed in figures 7.5 and 7.6, because of their complicated nature due to too many colored regions.

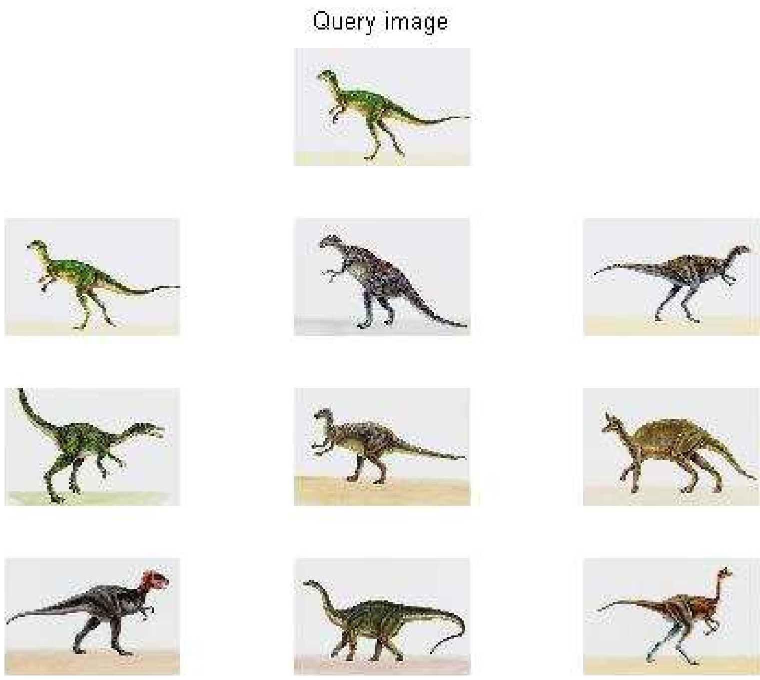


Fig 7.4 Top 9 matches of a dinosaur query image

Fig 7.5 Top 9 images of a human query image

Fig 7.6 Top 9 images of a horse query image

Fig 7.7 Top 9 images of a food query image

Moreover, figure 7.7 confirms the second inference stated earlier in this section. There is a major area displacement in the second and third result in figure 7.7 (2: horizontal shift and 3: vertical shift). Results 5 up to 9 of figure 7.7 have a noticeable difference in the hue, saturation, and color value. These images are still marked similar to the query image which proves the second inference.

D. *Conclusion and Discussion*

Experiments prove that region specific histogram properties can be very useful, because, they add robustness to the histograms that, in turn, add uniqueness of characterization among a set of similar images. Hence, it can be stated that two dissimilar images can be distinguished by considering the local feature set, and similar images can be apparently retrieved holding a low computational cost and improved characterization of image features.

Results displayed in the previous section prove the worth of the feature extraction process proposed in this research, which also works well with complex images. Therefore, it can be stated that the proposed algorithm solves the problem rooted at inability of uniqueness and robustness of histogram matching. Moreover, a wide variety of other local properties of colored and grayscale images can be tried and many distance measures can be used in order to further improve the results.

References

[1] G. Pass and R. Zabih, "Histogram refinement for content-based image retrieval," 1996.

1. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1349-1380, 2000.
2. W. Hsu, S. T. Chua, and H. H. Pung, "An integrated color-spatial approach to content-based image retrieval," 1995.
3. P. Muneesawang and L. Guan, "Automatic machine interactions for content-based image retrieval using a self-organizing tree map architecture," Neural Networks, IEEE Transactions on, vol. 13, pp. 821-834, 2002.
4. M. J. Swain and D. H. Ballard, "Color indexing," International Journal of Computer Vision, vol. 7, pp. 11-32, 1991.
5. D. Zhang and G. Lu, "Shape-based image retrieval using generic Fourier descriptor," Signal Processing: Image Communication, vol. 17, pp. 825-848, 2002.
6. R. M. Rickman and T. J. Stonham, "Content-based image retrieval using color tuple histograms," 1996.
7. C. Djeraba, "Association and Content-Based Retrieval," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, pp. 118-135, 2003.
8. M. Stricker and A. Dimai, "Color indexing with weak spatial constraints," Storage and Retrieval for Image and Video Databases IV, vol. 2670, 1996.
9. J. Huang, S. Kumar, M. Mitra, W. J. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms," 1997.
10. J. A. Park, M. H. Chang, T. S. Choi, and M. B. Ahmad, "Histogram Based Chain Codes for Shape Description," IEICE TRANSACTIONS on Communications, vol. 86, pp. 3662-3665, 2003.
11. J.-A. P. a. T.-S. C. Muhammad Bilal Ahmad, "Modified chain code and statistical invariant based shape matching for MPEG-7," in IEEE International

Symposium on Consumer Electronics. Erfurt, Germany, 2002, pp. 23-26.
12. V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Region-based image retrieval using an object ontology and relevance feedback," EURASIP Journal on Applied Signal Processing, vol. 2004, pp. 886-901, 2004.
13. J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 947-963, 2001.
14. C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "QBIC project: querying images by content, using color, texture, and shape," 1993.
15. Y. Rui, T. S. Huang, and S. F. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," Journal of Visual Communication and Image Representation, vol. 10, pp. 39-62, 1999.
16. T. Gevers and A. W. M. Smeulders, "PicToSeek: combining color and shape invariant features for imageretrieval," Image Processing, IEEE Transactions on, vol. 9, pp. 102-119, 2000.
17. M. J. Carlotto, "Histogram analysis using a scale-space approach," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, pp. 121-129, 1987.
18. J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient Color Histogram Indexing for Quadratic Form Distance Functions," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 729-736, 1995.
19. J. Engel, "The multiresolution histogram," Metrika, vol. 46, pp. 41-57, 1997.
20. E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar, "Multiresolution Histograms and Their Use for Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 831-847, 2004.
21. J. Malik and P. Perona, "Preattentive texture dissimilarity with early vision mechanisms," J. Optical Soc. Am. A, vol. 7, pp. 923-932, 1990.
22. A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," Pattern Analysis and Machine Intelligence, IEEE

Transactions on, vol. 12, pp. 55-73, 1990.
23. T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelettransform," Image Processing, IEEE Transactions on, vol. 2, pp. 429-441, 1993.
24. J. Li and J. Z. Wang, "Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1075-1088, 2003.
25. H. J. Zhang, J. Y. A. Wang, and Y. Altunbasak, "Content-based video retrieval and compression: A unified solution," 1997.
26. A. K. Jain, A. Vailaya, and X. Wei, "Query by video clip," Multimedia Systems, vol. 7, pp. 369-384, 1999.
27. J. Yuan, L. Y. Duan, Q. Tian, and C. Xu, "Fast and robust short video clip search using an index structure," 2004.
28. J. Sivic and A. Zisserman, "Video data mining using configurations of viewpoint invariant regions."
29. R. Lienhart, "Reliable Transition Detection in Videos: A Survey and Practitioner's Guide," International Journal of Image and Graphics, vol. 1, pp. 469-486, 2001.
30. S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," LECTURE NOTES IN COMPUTER SCIENCE, pp. 113-130, 2002.
31. A. Aner and J. R. Kender, "Video Summaries through Mosaic-Based Shot and Scene Clustering," LECTURE NOTES IN COMPUTER SCIENCE, pp. 388-402, 2002.
32. S. Eickeler, F. Wallhoff, U. Iurgel, and G. Rigoll, "Content-Based Indexing of Images and Video Using Face Detection and Recognition Methods," presented at IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001.
33. P. Hong and T. Huang, "Mining Inexact Spatial Patterns," 2002.
34. S. Lazebnik, C. Schmid, and J. Ponce, "Affine-invariant local descriptors and neighborhood statistics for texture recognition," 2003.

35. B. L. Tseng, C. Y. Lin, J. R. Smith, I. Center, and N. Y. Hawthorne, "Video personalization and summarization system," 2002.
36. T. Tuytelaars and L. Van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," 2000.
37. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, and D. Petkovic, "Query by Image and Video Content: The QBIC System," Computer, pp. 23-32, 1995.
38. V. E. Ogle and M. Stonebraker, "Chabot: Retrieval from a Relational Database of Images," Computer, pp. 40-48, 1995.
39. J. M. Corridoni, A. Del Bimbo, and P. Pala, "Image retrieval by color semantics," Multimedia Systems, vol. 7, pp. 175-183, 1999.
40. G. Frederix, G. Caenen, and E. J. Pauwels, "PARISS: Panoramic, Adaptive and Reconfigurable Interface for Similarity Search," 2000.
41. A. Hiroike, Y. Musha, A. Sugimoto, and Y. Mori, "Visualization of Information Spaces to Retrieve and Browse Image Data," LECTURE NOTES IN COMPUTER SCIENCE, pp. 155-162, 1999.
42. S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," Journal of the ACM (JACM), vol. 45, pp. 891-923, 1998.
43. A. S. Malik and T. S. Choi, "Consideration of illumination effects and optimization of window size for accurate calculation of depth map for 3D shape recovery," Pattern Recognition, vol. 40, pp. 154-170, 2007.
44. P. Muneesawang and L. Guan, "Interactive CBIR using RBF-based relevance feedback for WT/VQ codedimages," 2001.
45. D. Zhang and G. Lu, "Content-based shape retrieval using different shape descriptors: a comparative study," 2001.
46. D. Tao, X. Tang, and X. Li, "Which Components are Important for Interactive Image Searching?," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, vol. 18, pp. 3, 2008.
47. D. Tao, X. Li, and S. J. Maybank, "Negative Samples Analysis in Relevance

Feedback," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, pp. 568-580, 2007.

48. J. Li, N. Allinson, D. Tao, and X. Li, "Multitraining Support Vector Machine for Image Retrieval," IEEE TRANSACTIONS ON IMAGE PROCESSING, vol. 15, pp. 3597, 2006.
49. X. T. Dacheng Tao, Xuelong Li, and Xindong Wu, "Asymmetric Bagging and Random Subspace for Support Vector Machine-Based Relevance Feedback in Image Retrieval," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, pp. 1088-1099, 2006.
50. D. Tao, X. Tang, X. Li, and Y. Rui, "Direct Kernel Biased Discriminant Analysis: A New Content-Based Image Retrieval Relevance Feedback Algorithm," IEEE TRANSACTIONS ON MULTIMEDIA, vol. 8, pp. 716, 2006.
51. L. I. Xuelong, T. A. O. Dacheng, Y. Yuan, Y. U. Nenghai, L. I. U. Zhengkai, and T. Xiaoou, "Illumination-invariant Color Histogram Spectrum Feature for Content-based Image Retrieval," 2002.
52. J. Park, S. Kang, I. Jeong, W. Rasheed, S. Park, and Y. An, "Web Based Image Retrieval System Using HSI Color Indexes," 2007.
53. K. Hirata and T. Kato, "Query by visual example: Content based image retrieval", Advances in database technology - EDBT92, Springer-Verlag, Berlin 1992.
54. Shengjiu Wang, "A Robust CBIR Approach Using Local Color Histograms," Department of Computer Science, University of Alberta, Edmonton, Alberta, Canada, Tech. Rep. TR 01-13, October 2001.
55. R. Jain, R. Kasturi, and B. G. Schunck, Machine Vision, McGraw Hill International Editions, 1995.

(별 지)

저작물 이용 허락서

학 과	정보통신공학과	학 번	20077714	과 정	석사
성 명	영문 : Rasheed Waqas				
주 소	광주광역시 동구 서석동 375 조선대학교 전자정보공과대학				
연락처	E-MAIL : iwaqas@gmail.com				
논문제목	영문 : LOCAL REGIONAL OBJECT BASED RETRIEVAL TECHNIQUES FOR CBIR APPLICATIONS				

본인이 저작한 위의 저작물에 대하여 다음과 같은 조건아래 조선대학교가 저작물을 이용할 수 있도록 허락하고 동의합니다.

- 다 음 -

1. 저작물의 DB구축 및 인터넷을 포함한 정보통신망에의 공개를 위한 저작물의 복제, 기억장치에의 저장, 전송 등을 허락함
2. 위의 목적을 위하여 필요한 범위 내에서의 편집형식상의 변경을 허락함. 다만, 저작물의 내용변경은 금지함.
3. 배포전송된 저작물의 영리적 목적을 위한 복제, 저장, 전송 등은 금지함.
4. 저작물에 대한 이용기간은 5년으로 하고, 기간종료 3개월 이내에 별도의 의사표시가 없을 경우에는 저작물의 이용기간을 계속 연장함.
5. 해당 저작물의 저작권을 타인에게 양도하거나 또는 출판을 허락을 하였을 경우에는 1개월 이내에 대학에 이를 통보함.
6. 조선대학교는 저작물의 이용허락 이후 해당 저작물로 인하여 발생하는 타인에 의한 권리 침해에 대하여 일체의 법적 책임을 지지 않음
7. 소속대학의 협정기관에 저작물의 제공 및 인터넷 등 정보통신망을 이용한 저작물의 전송출력을 허락함.

2009 년 2 월 일

저작자: Rasheed Waqas (서명 또는 인)

조선대학교 총장 귀하