A Survey of Image Retrieval Techniques for Search Engine

[1]Mohammad Al Oudat [2]Miad Faezipour

[1]Faculty of Sciences, University of Bridgeport, 170 Lafayette St, Bridgeport, CT, 06604, USA
{ maloudat@my.bridgeport.edu, mfaezipo@bridgeport.edu }

Abstract. With the emergence of the latest progression of the World Wide Web, a tremendous need has arisen for efficient retrieval and extraction of information. The development of image retrieval and media extraction mechanisms is of paramount significance for storing, transmitting, generating, analyzing and accessing multimedia. Color, texture, shapes and spatial layout are the core characteristics used to signify and catalog the images. These odd features of images are extracted to assess the similarity of the images in several applications, particularly airport security checks and immigration processing. To extract useful information from this notably large amount of data, several information and image retrieval approaches have been introduced in recent decades. In this paper, we present a comprehensive survey of the latest technical accomplishments in image retrieval and extraction, especially for search engine applications. This survey covers well-known approaches from 2007 to 2014.
This survey comprehensively identifies the strengths and limitations of each technique. Additionally, certain other related issues are discussed in detail, i.e., text-based image search, semantic web-based search, and efficient web image search. Finally, based on current technology and the demand from practical real-world applications, future challenges and research directions are discussed.

Keywords: Image retrieval, Image extraction, Search engine applications, multimedia, and edge detection.

1. INTRODUCTION

Retrieval and extraction of images have emerged as an appealing research area in recent decades. Although improvements in imaging technology have made it possible to capture and display digital images, their use in search engines has been limited. To enhance image-based content retrieval accuracy, representative feature extraction algorithms are needed to reduce the gap between pictorial features and the range of human semantics. It has been observed that the size of the information on the web has increased exponentially, and similarly, information has been retrieved at a high pace due to the emergence of the latest technology. However, the techniques are not sufficient to handle all of the available information efficiently [1].

Similarly, images create difficulties during storage in computers, mobile phones, and other electronic devices, especially in the network. Nevertheless, the images stored in the web search engines are not satisfactory or accurate [2]. Most of the image search engines rely on text or tags related to the images to index and search the images [3]. Certain image search engines, e.g., Flickr, allow users to write a textual query and compare it with the stored images in the database. Such images are organized using tags marked by the users. In contrast, Google, Bing and Yahoo count on the text surrounding the images to index the images. However, those search engines do not carry out the search using image content because texts and tags are sufficient to describe the image content [4].

Several techniques are available to extract image content. One of the common techniques for images is edge detection, which is a process that locates the edges of an image. This process works on grayscale images to locate the absolute gradient magnitude that is considered an important factor in image processing [5]. Another technique is the color coherence vector (CCV) [6], which provides a decent mechanism for distinguishing between images. The CCV technique divides the coherent pixels from the incoherent pixels by storing a coherent versus incoherent pixel for each color [7]. Certain search engines use image compression techniques applied to the original image to produce a new image. The new image is either of the same sizes as the original image (lossless compression) or has a size that is smaller than that of the original image (lossy compression). The key objective in using image compression is to minimize the size of the images [8]. The most popular partitioning mechanism for images is fractal image compression (FIC), which partitions the image in a tree structure known as the quad-tree partitioning method [9]. The cross model search engine is a C++ library that grants such visual feature extractions as color, sharpness, texture and face. Furthermore, this method provides indexing and search functionalities [10].

A relatively new technique known as the Scale Invariant Feature Transform (SIFT) also has been introduced to convert image data into scale invariants that are coordinated relative to the local features. The importance of this technique lies in its ability to obtain a large number of features when applied to an image. This process also densely covers the entire image over the full range of scales and locations [11]. Therefore, SIFT is one of the emerging technologies that searches images using a color sketch. This technology allows end users to create a color map that informs the search engine how to distribute the colors in the desired image and results in improvement in text-based image searching and enhancement matching using color maps.

In this paper, we present a comprehensive survey of image extraction and retrieval techniques from 2007 to 2014. We also identify the strengths and weaknesses of each technique. Additionally, we discuss other related issues, i.e., text-based image search, semantic web-based search, and an effective web image search. Finally, we discuss the correctness and expected challenges of these techniques with respect to real-world applications.

2. CHARACTERIZATION OF IMAGE RETRIEVAL

Image retrieval covers such features as shape, texture and color. The traditional methods of image indexing have become obsolete due to their existing limitations, including speed, complexity, space, robustness, etc. Thus, new techniques are introduced with rich features and multiple utilities. However, these techniques still lack completeness. The image retrieval process consists of two important steps: the image features are extracted in the first step, and matching of extracted features is performed in the second step based on visual similarities. Feature extraction involves many characteristics, and the details of the image retrieval process, including taxonomy of image extraction feature techniques, are depicted in Figure 1.

2.1. Color Feature Extraction

Color is a commonly used feature in image retrieval. Several colors and varieties of color spaces are defined, and color is a significant feature that makes image recognition possible for humans. Color characteristics have a high impact on human eyes for processing valuable information in the brain. By focusing on the color characteristics and studying the properties of impending morphological operators for content depiction, color is characterized into different categories based on a state-of-art ordering methodology [1]. Several systems have been developed to represent color and texture properties, i.e., the Red-Green-Blue (RGB) color histogram used to detect texture properties. In addition, wavelet transform and statistical texture measurements are performed. Furthermore, the coefficients are collected based on the histogram, and subsequently, the Euclidean distance is applied for comparison of the wavelet coefficients and texture properties of two images.

2.2. Texture Feature Extraction

Texture extraction is one of the significant features in content-based information retrieval (CBIR). In real life, texture provides indispensable information for images based on image classification [12], and texture plays an important role in the high-level semantic image retrieval process. The Gabor filtering and wavelet features are used for many texture feature retrieval and extraction processes. Because these filters are particularly designed for rectangular-shaped images, the CBIR system involves arbitrary-shaped regions. Texture image retrieval is in high demand for many applications, i.e., the medical field and immigration processing. The 2-D Gabor filter is one multi-resolution filtering method that proves a better choice for texture retrieval and extraction [13].

2.3. Low-Level Feature Extraction

Low-level feature extraction is highly significant in content-based information retrieval systems. Image feature extraction is performed within certain specific regions of the image or on the entire image. Particularly, image extraction is highly focused on specific regions for determining exact state-of-the-art outcomes. Thus, several region-specific CBIR methodologies have been developed. In these methodologies, image segmentation is performed first, and based on the segmentation, the color, shape, texture and spatial location are subsequently extracted [14]. A dual-feature extraction mechanism of global and local extraction is applied for better accuracy [15].

2.4. High-Level Feature Extraction

High-level feature retrieval involves semantic extraction that uses text, keywords and descriptions. In this type of retrieval, the color, texture and shape extraction is performed automatically over the image. Certain methodologies were introduced to support this type of extraction, i.e., the audiovisual queries discussed in [16]. A clue methodology

ISSN 2229-5518

was introduced for cluster image retrieval to remove the semantic gaps. Thus, CLUE shows the top matching targeted images in CBIR to the users. Based on the query, the targeted images are chosen as similar or near to the query image. Subsequently, clustering semantic classification is carried out. Next, the system shows the image clusters, and finally, a similarity measurement model is built with respect to the response of the users.

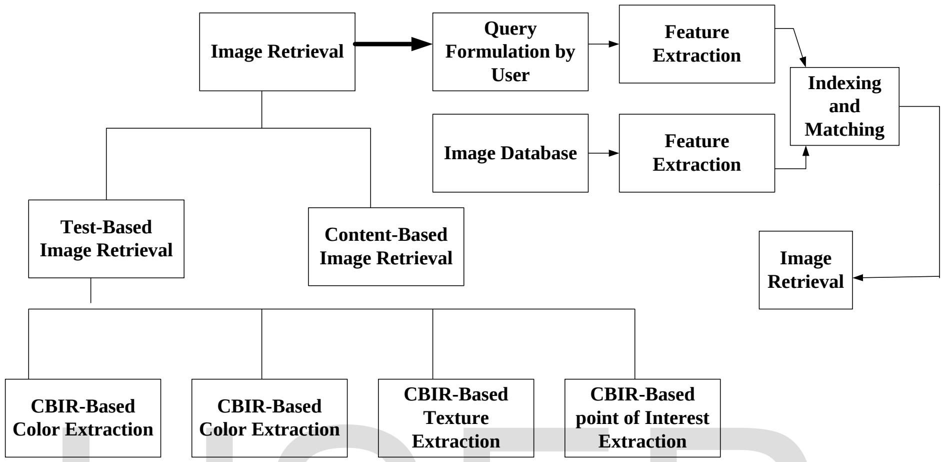

Figure 1: Content-based image retrieval process, including image-extraction feature techniques

3. REVIEW OF THE TECHNIQUES

In this review, we discuss a subset of the well-known techniques with a particular focus on search engine applications.

3.1. Sobel Edge Detection and Color Coherence Vector

In [17], the authors proposed an image-query-based search engine system that allows users to upload images from a local database to the system. The system extracts features from the uploaded image, i.e., the color coherence vector, and matches them with the images on the internet. The images on the internet, that have the most similar matches with the uploaded image, are identified. This approach is based on a search methodology that uses images as input from the user to obtain hypertext information from the web. The image contents are used as a basis for the search. Furthermore, this approach uses textual information with respect to the image from the source code of the website to find hypertextual information for the query by the user.

This approach is based on two major techniques, i.e., Sobel edge detection and Color Coherent Vector Matrix (CCV). Sobel edge detection is used as a pre-processing technique that identifies the edges of an image at the points where the intensity changes drastically, and CCV matches the extracted features from the image. This technique is efficient overall; however, it inherits the limitations of metadata-based systems. The textual information of the images is easily determined with the existing technology, but human support is required to describe each image in the database manually. Moreover, this approach is not suitable for large databases and automatically generated images.

3.1.1. Edge Detection

The concept of the Sobel operator originated in 1965 from the Robert edge operator, which consists of two 2×2 convolution masks. The benefit of the Sobel operator is the use of a larger mask for reducing errors due to noise by calculating the average of the surrounding pixels in the mask. Sobel image edge detection was introduced in [2] to convert a 2D pixel array into a statistical uncorrelated dataset. This technique removes redundant data; as a result, the reduced data volume is used for the digital image. The Sobel edge detector uses a 3×3 convolution mask that creates one estimation gradient at the *x*-axis, and the other gradient is created on the *y*-direction.

The beauty of Sobel edge detection is the ability to highlight the edges. Sobel edge detection is normally used to identify the horizontal and vertical edges. The edge detection process works well on noiseless images, but edges detected by the Sobel operator are thick and not suitable for certain applications [18]. Additionally, it is also difficult to detect edges on noisy images because the parameters for evaluation and detection are rigid [19]. Moreover, the Sobel operator indicates a slightly inaccurate approximation with an image gradient. The approximations of these derived functions can be demarcated at flexible degrees of accuracy that slow down its performance.

3.1.2. Color Coherence Vector

In [7], the authors proposed image comparison techniques using color coherence vectors (CCV). This technique is based on color coherence in terms of the level where pixels of a specific color are a component of selected large adjacent regions. The images, that possess a similar color, are known as the coherent regions. The value of the region is counted to illustrate the image. The CCV technique works in a manner similar to that of a color histogram in image comparison. The color histogram is commonly used to compare images and is easy to compute. The CCV frequently uses different objects with special color histograms and distinguishes between images by separating the coherent from incoherent pixels but does not allow any matching between coherent pixels in one image and incoherent pixels in another. As previously mentioned, coherent pixels represent adjacent pixels with the same color where incoherent pixels are not available. With the use of this feature in CCV, it is easier to distinguish between images than with the use of a color histogram. Computation of the CCV is similar to the computation of a color histogram.

The CCV can distinguish between coherence and incoherence using connected components C, where ‘C’ is the number of pixels between two specific points P and P’, and P, P’$\in$ C. The pixels between P and P’ are known as the path. Furthermore, adjacent pixels are considered to be coherent. A decision must be made as to whether two pixels are adjacent and whether one pixel is among the eight neighbors of the other. Each pixel will fit into only one connected component. Finally, classification of pixels as either coherent or incoherent is determined by counting the size of the pixels in its connected component. A pixel is classified as coherent if the size of its connected component exceeds a threshold value; otherwise, the pixel is classified as incoherent. These characteristics make CCV a better choice for distinguishing between images. However, this method has several weaknesses, e.g., slight changes in camera angle can affect the color histograms. The CCV also experiences problems in distinguishing between coherent and incoherent pixels because it does not have the ability to illustrate the variation between the two images [7],[20].

3.2. Quad Tree Partitioned Iterated Function and Fractal Image Compression

The work in [21] proposed a Quad-tree-based image search engine (QBISE) that allows the user to retrieve similar images from the database. The QBISE is split into four major components: normalization of image, retrieval of eigenvalues from QPIFS, image storage, and analysis. Images on the internet have different sizes, colors and intensities, and therefore, a normalization phase is required. The size of the image cannot be too large, and the average size is approximately 100×100 pixels. In the color scale normalization, the RGB transformation formula is used to transform colored charts to grayscale:

$$Pixel\ intensity = 0.299 \times R + 0.587 \times G + 0.114 \times B \qquad (1)$$

In the intensity normalization, we adopt a statistical color degree method. This method calls for 256 pixel values, and the color ranges from 0 to 255. We accumulate the sums of the pixel values starting from the 0th group and the 255^{th} group while moving toward the 128^{th} group. When the sum of two sides of the pixel quantity reaches 5%, we stop the accumulation and distribute the pixels into the 0^{th} and 255^{th} groups proportionally assigned to 1-254 groups based on the degree of distance. The QBISE extracts the eigenvalues from an image and stores them into the database rather than storing the entire image because the images take up a notably large amount of storage space. The eigenvalues are used for comparison between images, and thus, rather than retrieving the entire image, the system will retrieve the eigenvalues from the image by applying the quad tree iterated function system. Moreover, the proposed system uses the mean value as the first step to eliminating images with large color differences. The mean value is the sum of all pixel image color values divided by the number of pixels.

3.2.1. Quad tree Partitioned Iterated Function (QPIF)

To discuss QPIF, we must illustrate fractal image processing [22]. This technique splits an image into range blocks and domain blocks. Range blocks do not overlap, and thus, they are different. Domain blocks are double the size of range blocks (e.g., if the range block size is 4×4, then domain block size is 8×8), and domain blocks are allowed to overlap. The position of a domain block is determined by shifting the new domain block one pixel parallel from the previous domain block. All domain blocks are referred to as the domain pool. Applying this shape for range blocks and domain blocks guarantees that nothing is missed when comparing range blocks with domain blocks. However, this shape will increase the complexity of calculations. The iterated function system (IFS) [23] is considered an important connection between range blocks and domain blocks because the range blocks are compared with all domain blocks. If a similarity exists between a range block and domain block, the range block will save all information from that domain block, i.e., index value location and affine transformation component, but not the entire content of the domain block. The degree of similarity is obtained by applying a peak signal-to-noise ratio (PSNR) and performing a comparison between the range block and domain block, where both must have the same size. Therefore, affine transformation is applied on the domain block to make it the same size as the range block. Affine transformation is contractive. For example, suppose we have an image f, and after applying fractal image compression, we have several range blocks R_1, R_2… R_n and domain blocks D_1, D_2… D_k.

To make a domain block (e.g., Dj) have the same size as the range block Ri, we apply contractive transformation W, such that $Ri \simeq W(Dj)$, and this process is repeated for all domain blocks. Thus, we obtain W_1, W_2…W_k. The union of all transformations is referred to as the Partitioned IFS. To encode a range block, each domain block in the domain pool is scaled to the size of the range block, and a comparison is made between the range block and domain block with respect to intensity offset, contrast parameters, and isometric transformations, i.e., identity, reflections about the mid-horizontal and the mid-vertical axes, reflections about each diagonal, and rotations through 90°, 180°, and 270°.

The results of this process are domain blocks that have the best matches with range blocks, and this new pool of domain blocks is known as the codebook pool. After comparing the range blocks with the domain blocks, if the PSNR result is infinity, then both are one hundred percent similar. If a difference exists between the range block and domain block, then a threshold value is considered. If the PSNR exceeds the threshold value, both are considered similar; otherwise, the range block will be compared with another domain block. The equation below shows the PSNR.

$$PSNR = log10\left(\frac{255}{MSE}\right).10dB \qquad (2)$$

where dB is a logarithmic unit representing the ratio between two values of a physical quantity.

$$\text{MSE}= \sum_{i=1}^{x}\sum_{j=1}^{y}\frac{(\left|a_{ij-}b_{ij}\right|)^2}{x.y} \qquad (3)$$

MSE is the mean square error a_{ij}, which denotes the pixel value in *a* domain block; b_{ij} donates the pixel value in the range block; and $x.y$ denotes the pixel quantity in the range block.

To return the image to its original state, an empty image (i.e., canvas) is initiated to store the recovered domain blocks. The domain block is rebuilt from a range block by applying IFS and recovered to the canvas, which is similar to the original image. It is too difficult to visualize how fractal image processing operates, and therefore, for ease of visualization, we consider Figure 2.

ISSN 2229-5518

Figure 2: Fractal image processing

The image indicates that fractal image compression has already divided the image into blocks. As noted in the image, the upper left corner block (range block) is similar to the upper right corner block (domain block). Affine transformation is applied to the upper right corner block (domain block) to ensure that its size the same as that of the upper left corner block (range block). Next, all information related to the domain block is stored into the range block. Finally, this process is repeated for all range blocks. In the Quad tree partitioned IFS technique, the image is divided into four equal blocks. For each range block, a matching domain block is searched, and if it is found, then the range block is coded by storing the information for the location of the matching domain block and the transformation applied. If a matching domain block is not found within the exact error tolerance, then the corresponding range block will be subdivided into four equal sub-blocks. This process continues repeatedly and recursively, starting from the entire image and continuing until the range blocks are sufficiently small to be able to find a matching domain block within the specified error tolerance. Figure 3 illustrates how the first, second and third partitions work.

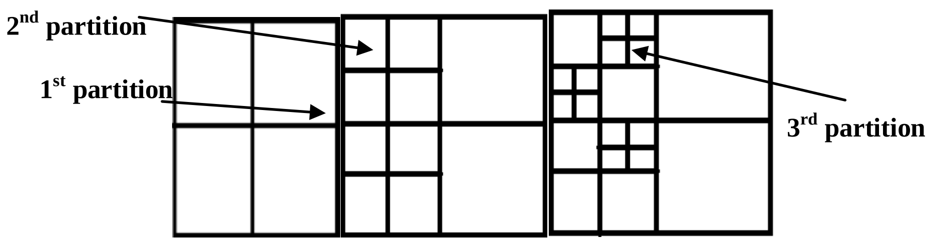

Figure 3: First, second and third partitions generated

The weakness of this technique is the fixed sized partition. The image is segregated without deliberating on the image contents. The block regions of the image are difficult to cover using fixed-size range blocks. Similarly, certain regions require larger size range blocks and must reduce the total number of maps. Moreover, an imperfect matching block causes image quality loss.

3.2.2. Fractal Image Compression (FIC)

In [24], the authors proposed fractal image compression. The concept of this technique is to reduce the number of bits that form an image. This technique generates identical images with the same size as the original image, known as lossless compression. Techniques that create images that are less than the original size are referred to as lossy compression. The use of fractal image compression with an iterated function system (IFS) involves the process of fractal encoding that partitions the images into range blocks and domain blocks. Each range block is mapped onto the domain blocks using contractive transforms known as affine transforms. The shortcoming of FIC is its computational complexity. Due to its complexity, the FIC can achieve real-time speeds only with the use of highly parallel array structures [25]. Furthermore, FIC involves a large amount of matching and geometric operations that are time-consuming. Additionally, the coding process is highly asymmetrical, and thus, image encoding takes much longer than decoding [26].

4. PARALLEL CROSS-MODAL SEARCH ENGINE

In [10], the authors introduced a Parallel Cross Model Search Engine (PCMSE). This technique uses a cross model search engine that involves a C++ library. The PCMSE provides feature extraction, indexing, and search functionality and proposes visual feature extraction, i.e., color, shape, texture and face. The scheme uses more than million images from the Image-Net database. The queries are generated using the Image-Net database among 20 group images. The group is composed of one positive example, and the remainder consists of negative examples, also known as the Rank Boost learning strategy. The scheme involves the response time of active nodes and uses 1 to 40 cores for 20 nodes.

Thus, each node uses its two cores and supports its access to the local disk and memory. The increase in the number of cores is first carried out with one core per machine and subsequently with two cores per node. For this simple usage scenario, the response time from the perspective of the user (from submission of the query to obtaining the response) is 22 seconds (sequential version) to approximately 2 s (40 core version). Additionally, this process uses map reduction to simplify the data processing within clusters, specifically for indexing. This scheme directly copes with the issue of mass storage access and memory using a cluster-based distributed strategy and requires highly scalable software; a lack of this capability causes failure of parallel extraction and indexing. The PCMSE encounters a significant problem with failure of components due to high overload and also requires high security to protect the database from malicious attacks, which are the major issues in this technique.

5. WEB-SCALE IMAGE SIMILARITY SEARCH SYSTEM

In [27], the authors proposed the web scale image similarity search system (WISSS). This system was developed as a large-scale architecture for indexing and searching of

ISSN 2229-5518

image collections based on the visual characteristics of its contents. The system fulfills two major goals. The first goal involves highly packed images of over 50 million with the Flicker photo-sharing system and extraction of five MPEG-7 features for each and every image. The second goal describes the route used to index images (from several thousand up to a million) carried out via experiments to ensure unique comparison of individual steps. The proposed system for support of scalability challenges is shown in Figure 4.

Content-based image retrieval (CBIR) is considered a stepping stone for a proposed system that searches images depending on their identical visual content. This approach also extracts features from the image, and those features are considered as a signature. The signature is used for indexing and searching. Collection of images from the web is not an easy task. Therefore, this approach uses the Flicker photo-sharing site, which provides the richest additional metadata and also offers an efficient API to access its content at various levels. Moreover, one of the properties of Flicker is that it has a unique ID for each photo.

The approach uses five MPEG-7 descriptors to support a content-based search to obtain scalable color, color structure, color layout, edge histogram, and homogeneous texture. The extracted MPEG-7 visual descriptor is computationally quite expensive. Therefore, the MPEG experimental model (MPEG-XM) is chosen to guarantee the correctness of the extracted features. The application used in this approach involves three components: image-id server, crawling agent, and repository manager. Figure 3 shows the working process of the used application.

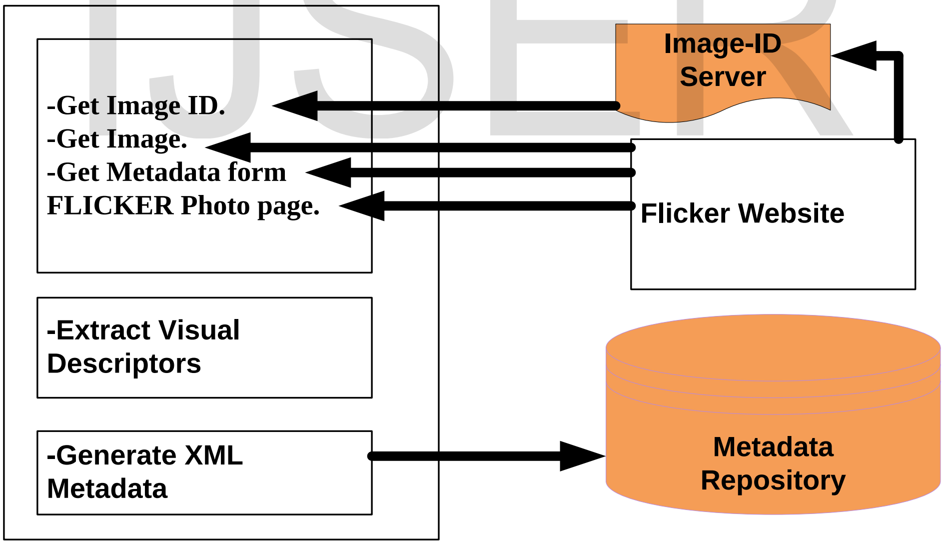

Figure 4: Application to processing images in parallel

The system used for this approach is based on GRID and supported by DILIGENT IST project. The approach divides the crawling agent into two modules. The first module is responsible for establishing a connection with the image-id server and crawls the Flicker website using the Flicker APIs. The result is sent to the repository manager. The

ISSN 2229-5518

second module consists of MPEG-XM software with feature extraction that is encapsulated into the crawling agent. The experiment depends on the results from the crawling agent and image processing using a content image retrieval method and its scalability characteristics. The result is composed of the Content Based Photo Image Retrieval (CoPIR) Test collection. This approach involves highly rich features. However, the system used for the experiments in the application is difficult to configure due to its heterogeneous characteristics, which involve complicated software and hardware. As a result, construction of the crawling agent is complicated. The MPEG-XM software module with extraction features is encapsulated in the crawling agent. However, the crawling agent is not compatible with new compilers and libraries due to the heterogeneity of the Graphic Retrieval and Information Display (GRID) system.

6. WEB-SCALE IMAGE SEARCH BY COLOR SKETCH (WISCS)

In [3], the authors proposed an image search system using color sketches that improves text-based image search and enables users to develop a color map. This method also shows how colors are distributed in the anticipated image. In this approach, the system searches images by color rather than by shapes. The system deploys two color-sketch modes composed of stroke scribbling and image dragging. Stroke scribbling allows the end user to scrawl a few color strokes in a blank canvas, and image dragging allows the end users to drag and drop images onto the blank canvas and mark the color region of interest to display the desired spatial distribution of colors. Figure 5 shows the image representation and similarity evaluation, which can be divided into three major processes. First, the system objective is to target the color map to extract the main colors and excavate the intention map. Second, the main colors (11 colors: black, brown, blue, green, gray, orange, purple, pink, red, white, and yellow) are used to sift the image search results related to the text query. Third, similarity scores are calculated among the top 1000 images and the queried intention map that remains after leading color filtering, and finally, the images are reordered.

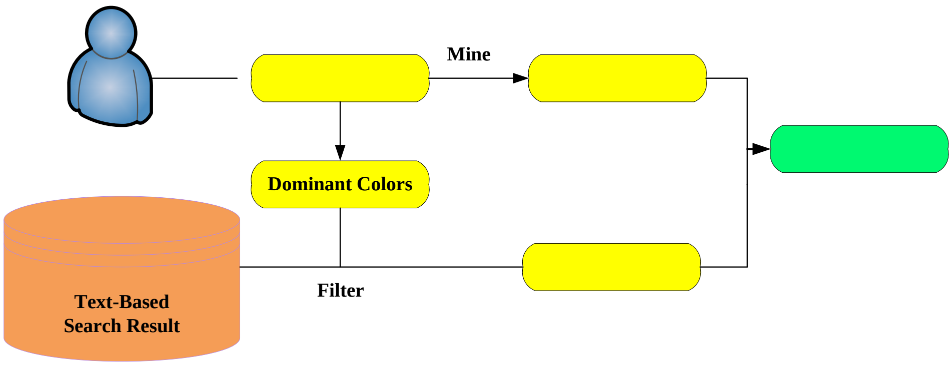

Figure 5: Flow chart of image representation and similarity evaluation

Initially, the system that searches images by color is based on a text-based search, and the obtained results are sent by the user to the target color map. Next, a comparison is conducted between the sent color map and the images using the back-end algorithm. The ordered images are returned to the users, and the users can apply an additional operation on the obtained result to edit the target color map and obtain a newly ordered result. The system involves two phases in the color map. The first phase is used to select the representative colors for each grid. In the first phase, the system also splits the image into $g \times g$ grids. As a result, the dominant colors $C_{xy}=\{c^i_{xy}\}$ are identified for each grid. Thus, the system analyzes the frequencies for the quantized colors presented in the grid. The system selects the corresponding color according to the 't' largest frequencies such that f_i-1<2f_i, $\forall i \leq t$ and $i \neq 1$.

After mining the colors in each grid, the system determines the representative colors for the entire image, $C=U_{xy}\ C_{xy}$. The second phase consists of color mapping that concatenates the mined colors into three-dimensional maps. For each color $c \in C$, the system calculates its three-dimensional map as a binary vector' m' consisting of $g \times g$ dimension, which can be obtained using equation (4).

$$m_p = \begin{cases} 1 \quad if\ C^i_{xy} = C, \quad \exists C^i_{xy} \in C_{xy} \\ 0\ else, \end{cases} \tag{4}$$

where 'p' is the one-dimensional index corresponding to the two-dimensional indices (x, y). Intuitively, if color c appears in the grid (x,y), the corresponding entry 'm_p' is set to 1 or to 0 otherwise. The scheme claims to show a well-designed trade-off between feasibility and scalability. However, it is not clear in this approach how the system obtains the feasibility and scalability features. The interface is spontaneous and requires users to scrawl a few color strokes. Without the correct scrawling, images cannot be dragged efficiently into the region of interest. As a result, the intended search cannot be obtained.

7. SEMANTIC WEB-BASED SEARCH ENGINE

In [2], the authors proposed a semantic web-based search engine used to retrieve a human facial image by counting facial features. The technique categorizes the human face into two ontologies. The first is an internal ontology that focuses on absolute data, and the second is an external ontology that focuses on relative data. Figure 6 shows the SemFace architecture.

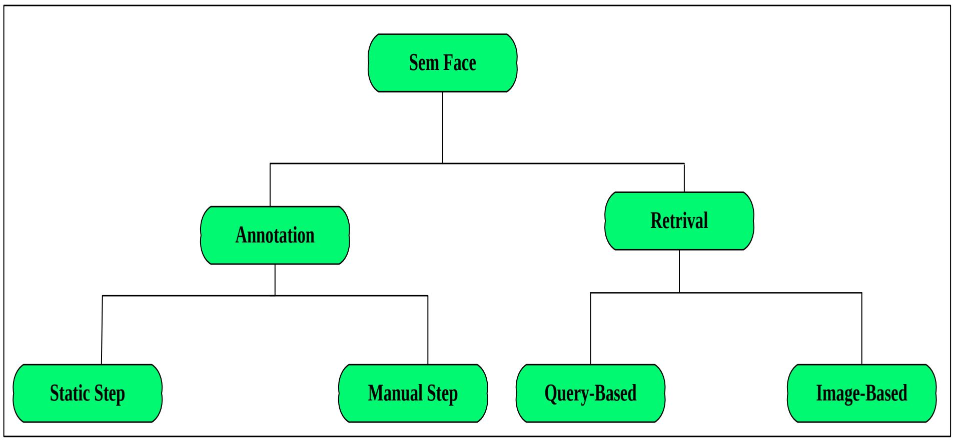

Figure 6: SemFace Architecture

SemFace is divided into two phases, i.e., the annotation phase and retrieval phase. Both of the phases work independently while interacting with each other through the annotation files. The annotation phase encompasses Query Based Retrieval (QBR) and Image Based Retrieval (IBR). The technique represents the abstract idea of searching the image and fails to produce the desired result. Further, the proposed architecture is not well suited for searching large numbers of images. The SemFace architecture requires improvement at the annotation phase because the static setup limits the number of images. This phase only aids in determining the known stored images.

8. SCALE INVARIANT FEATURE TRANSFORM(SIFT) ALGORITHM

The SIFT algorithm was proposed by [27] and used for the Effective Web Image Searching Engine (EWISE) based on SIFT feature-matching and an effective content-based web image searching engine (ECBWISE) based on the Scale Invariant Feature Transform. The working processes of both schemes are similar and are based on SIFT. In both schemes, the SIFT feature is the local feature of an image that is constant during transformation, image scaling and rotation. This technique is also partially constant to radiance changes, affine with noise, and angle of view changes. The distinguishing image features are suitably used for rapid and correct feature matching in a mass feature database. The SIFT method is extendible and has the ability to combine with other features. In both approaches, for decreasing inaccessible feature matching, the dynamic probability function swaps the original static value to identify the identical distance and database using training images. Next, key points are stored in XML format using source image pretreatment. The SIF also improves the search efficiency. Finally, the results are presented to the user via HTML.

The experimental outcomes demonstrate that the image search engine accuracy is increased. The search engine used in both schemes has the significant characteristic that it produces many features that compactly handle the image using complete scaling ranges and locations. The fixed value is used for the feature-matching threshold. Thus, each feature vector in XML is the same in making the decision whether it is matching

Both schemes focus on image retrieval that involves two vital steps: feature matching and feature extraction. The current image pixel point provides the feature vectors for feature extraction. Thus, the feature vector is known as a SIFT descriptor represented by 128-dimensional descriptors. The approach uses an improved SIFT algorithm that involves major states of calculation to produce the set of Image Feature Vectors (IFVs). These IFVs consist of detection for scale-space extrema, accurate key-point localization, and orientation assignment. The detection for scale-space extrema is used to create multi-scale-space images using the convolution of a flexible scale Gaussian formula.

Accurate key-point localization is used to increase feature-matching accuracy and noise invulnerability. The orientation assignment uses an orientation histogram to determine the orientation of the key points. Both schemes also involve local feature matching supported by the Euclidean distance and dynamic ratio of the distance. The Euclidean distance aids in identifying the nearest neighbor in the database that is declared as the key point. The best candidate is matched with each key point for the static descriptor vector. The dynamic ratio of the distance identifies whether the matching is successful.

9. A NEW WEB IMAGE SEARCHING ENGINE USING SIFT ALOGRITHM

In [28], the authors introduced a new descriptor that combines the SIFT algorithm with co-occurrence histograms (CCH). The CCH idea is comprehensively applied to object recognition applications and color texture retrieval. The approach analyzes the new descriptor in the form of image matching. The new descriptor is validated to a greater extent than the existing SIFT descriptor based on the experimental results. This approach detects a significant number of extra accurately matched feature points, and the dissimilarity ratio remains constant. In this approach, CCH transfers the information of a three-dimensional correlation for the neighboring pixels. Additionally, CCH counts the number of events of the color pixel pair in the image. Figure 7(a) shows the orientation histogram 4×4×8 for SIFT, and spatial dependencies of CHH are illustrated in Figure 7(b).

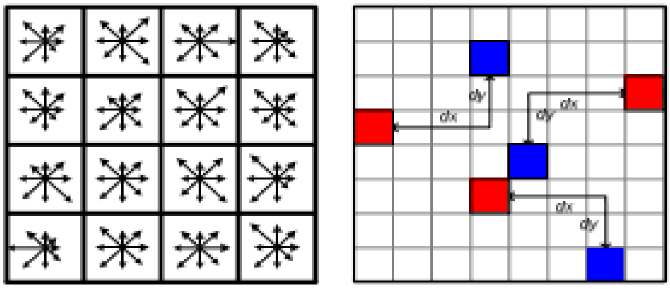

Figure 7: SIFT 4×4×8 histogram (a) and spatial dependencies for CCH (b)

In [29], the authors compared the features of the scale invariant feature transform (SIFT) and Iterative Scale invariant Feature Transform (ISIFT). In this comparison, ISIFT finds exact matches of the images. Additionally, the relative views and lighting between the images are projected iteratively. The matching efficiency is not affected by

ISSN 2229-5518

view and lighting modifications in the threshold ranges. Furthermore, the threshold values illustrate the maximum permissible changes in view and lighting between the images. The comparison is carried out between the SIFT and ISIFT matching algorithms. Based on the results, the ISIFT algorithm provides an exact matching rate and a scale static to lighting the change via a five-step process.

Step 1: The process initially assumes that the estimated transformation is given as in equation (5).

$$H_0 = \{E_0, F_0\} = \{\, D, 1\}(1) \tag{5}$$

Where, $H = \ \ H_0, \sigma E, \sigma E;$

Step 2: The iteration process begins:

$$K = K + 1 \tag{6}$$

Step 3: The transformation is estimated iteratively.

$$H_k: H_k \, , F_k \tag{7}$$

In step-4:

$$H = \ H_k {}^\circ H \tag{8}$$

Thus,

$$H = \ E_k * F_k \tag{9}$$

Finally Step-5: Transformation: $\mathrm{Z_{K\text{-}1}}$ to $\mathrm{Z_K}$ using equation (4) until

$$(E_k - P) < \ \sigma E \, , (F_{k-1} \ < \sigma L) \tag{10}$$

where '*P*' is the unit matrix, and E_k and F_k are the view and lightening factors, respectively. One of the benefits of ISIFT is that it estimates the lighting change that improves the matching. These SIFT-based techniques work perfectly, but they experience issues in searching the object. The schemes only extract a few objects from a large number of SIFT and ISIFT features. Thus, certain feature vectors for the targeted image are lost and affected. The features also remain invariant to image scale and rotation, which affects the quality.

10. DISCUSSION AND COMPARISON OF TECHNIQUES

Several techniques for image retrieval have been introduced, but no single algorithm solves the inherent issues completely. Automatic image retrieval, processing and storage are still focal subjects of research. The techniques vary in their strengths and weaknesses, and the selection of which algorithm to use depends on the requirements modeled by the application. Certain techniques neither introduce objects nor distort the original organization of the image. However, the method might fail to recollect all significant objects when the image encompasses dispersed important regions or significant objects larger than the anticipated size. Pixels are always superfluous, and the image composition is spoiled or improved in certain cases. In this work, we discussed a subset of leading image extraction and matching algorithms. The SAD and CVV techniques were introduced for extracting features and matching the extracted features from the image. This approach is better overall, but it inherits the limitations of metadata-based systems; as a result, it cannot work for large databases and automatically generated images. Sobel edge detection was introduced to determine the horizontal and vertical edges and works well on noiseless images but is not suitable for certain applications; it also gives slightly inaccurate approximations with an image gradient. The CVV approach is a better choice for differentiating between images. However, slight changes in angle can affect the color histograms.

The QPIFS is considered the most common partitioning technique in fractal image compression (FIC) in which the original image is reconstructed by contractive transforms and selected mathematical functions, but QPIFS experiences problems due to a fixed size partition that affects the image quality. The Fractal Quadtree Partitioned Iterated Function System was introduced to retrieve similar images from the database using eigenvalues and links for each stored image; however, it suffers due to use of compression that causes quality image degradation. To control the disadvantage of compression for a Fractal Quadtree Partitioned Iterated Function System, the FIC technique is introduced to generate identical images with the same size as the original image (lossless compression), but FIC has a high computational complexity. To reduce the complexity, the Parallel Cross Model Search Engine (PCMSE) was introduced to provide feature extraction, indexing, and search functionality and proposed visual feature extraction, i.e., color, shape, texture and face.

This approach covers a broad area of image extraction and retrieval but requires highly scalable software, and lack of this capability causes failure of parallel extraction and indexing. To reduce the cost of highly scalable software, a web-scale image similarity search system was introduced for indexing and searching the image collections based on visual characteristics of its contents. However, it is difficult to configure the system due to its heterogeneous characteristics. To overcome the problem of heterogeneity, an image search system using color sketches was introduced to improve the text-based image search and enable users to create a color map. The scheme also offered a well-designed trade-off between feasibility and scalability. However, it is not clear in this approach whether the system obtains feasibility and scalability features. The interface is quite spontaneous and requires users to scrawl a few color strokes.

One of the feasible and scalable semantic web-based search engines was introduced to retrieve human face image by counting facial features. This approach introduced external and internal ontologies that focus on absolute data and relative data. However, this approach suffers due to an annotation phase because the static setup of the annotation phase limits the search of the images. To improve this problem, the SIFT algorithm was introduced to provide local features of an image that are static to transformation, image scaling and rotation and is also partially static to radiance changes, affine with noise, and angle of view changes. The distinguishing image features are suitably used for rapid and correct feature matching in a mass feature database. The SIFT is extendible and has the ability to combine with other features. However, SIFT and its variants remain invariant rather than dynamic with image scale and rotation, which affect the quality. In addition, SIFT extracts a few objects from a large number of features. All of these existing techniques have flaws, but the development of robust algorithms for new methodologies and flexible designs could retrieve image contents efficiently. The strength and limitations of the existing approaches are listed in Table 1.

Table 1: Image Retrieval Techniques for Search Engine Applications

Parameters	**SAD**	**CVV**	**SOBEL**	**QIFS**	**FIC**	**PCMSE**	**WISSS**	**WISCS**	**Semantic Web**	**SIFT**
Feature Extraction Capability	Moderate	Low	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	High	Moderate
Feature Retrieval Capability	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Low	Moderate	Moderate
Visual Feature Extraction Capability	No	Yes	Yes	No	Yes	Yes	Yes	May be	May be	May be
Complexity	High	High	Moderate	High	High	Low	Moderate	High	High	Moderate
Parallel Extraction and Indexing	No	No	May be	No	No	No	Low	No	No	No
System Compatibility	Moderate	Moderate	Moderate	Low	Low	Moderate	Low	Low	Low	Moderate
Mapping	No	Yes	No	No	Yes	No	No	Yes	No	May be
Feasibility	No	No	No	No	Low	Low	Low	Low	Moderate	Moderate
Reliability	Moderate	Moderate	Low	Not reliable for fixed size	Low	Low	Low	Moderate	Moderate	Moderate
Scalability	Low	Low	Moderate	Moderate	Low	Low	Low	Low	Moderate	Low
Metadata-based Limitations	High	High	Moderate	High	Moderate	Moderate	Moderate	Moderate	Moderate	High
Automation of images	No	No	No	No	No	No	No	No	Yes	May be
Robustness	Moderate	Moderate	Moderate	Low	Low	Moderate	Low	Moderate	Low	Moderate

11. RESEARCH DIRECTIONS AND FUTURE ISSUES

Most of the recent image retrieval and extraction techniques focus on the image retrieval accuracy. However, several other issues must be addressed. The query-based approaches play a significant role in reducing the semantic gap, which could resolve many problems associated with image retrieval and extraction. A few techniques addressed the problem of sketch query and example query [30]. However, future work must thoroughly focus on these issues, particularly the semantic contents of a query for retrieval and extraction in the web service for finding metadata and objects in databases. As the size of image databases increases, the retrieval speed of the images is a significant factor that must be addressed. In addition, off-line multimedia image indexing is also necessary for scalable retrieval. Consistency, isolation and data independency are the most important factors to be considered because extraction of color and organization of data require isolation to differentiate one image from another image.

The need also exists for new content retrieval methods to find images similar uploaded images on the Internet. Hybrid approaches might be the best choice for extraction and retrieval of images. However, it is highly important to choose the best features of each algorithm without computational complexity. Based on the emerging technologies and the needs of practical applications, a few open issues are highlighted from the system point of view, i.e., continuing integration of image retrieval, query language design, high-dimensional image feature indexing, etc. Proposal of a complete image retrieval system with advanced semantics and the ability to handle big data requires the combination of prominent low-level feature extraction, a friendly user interface, successful learning of high-level semantics, and a well-organized indexing tool. Most of the systems comprehensibly bind their contributions to one or two of these mechanisms.

A content-based image retrieval system that offers a highly composed view of all essential components is in great demand. In addition, high-dimensional image feature indexing for practical image database retrieval is new research direction.
Efficient multidimensional approaches could improve the image retrieval and extraction system. Thus, these approaches should be highly scalable and faster for the image retrieval and extraction process. Improvement in image search performance is highly desirable for real-time applications [31]. Multimodal fusion with a combination of pattern-mining forms could be used for speedy processing.

Text-based image retrieval is an important research area that requires additional focus. In this area, structural methods based on morphological operators and adjacency graphs must be optimized to identify placement rules and structural primitives effectively. The need also exists to improve the existing statistical methods for image retrieval and extraction, i.e., wavelet transform, shift-invariant component analysis, concurrence matrices, fractal models, Markov random fields, and counting of co-occurrence pixels. The dimensionality of image features also must be addressed because the image size is usually quite high. Thus, traditional indexing algorithms (i.e., k-d-b tree [32], quad-tree [33] and R-tree [34]) failed to have an impact. To replace these traditional algorithms, the X-tree [35], i-Distance [36] and VA-file [37] have been introduced for indexing but have failed to determine which features to index. For these reasons, the approaches are premeditated, without consideration of the image feature properties. Multidimensional indexing approaches must be explored for efficient and scalable content-based retrieval and extraction.

Future research directions are based on differentiating the computer vision pattern recognition approaches and image retrieval approaches. This research direction has already been revealed in the advancement of content-based image retrieval and extraction processes. Image storage consists of huge collections, and image retrieval and extraction algorithms have a limited capacity for retrieving the image. However, retrieval speed is highly bottlenecked. Certain research is in progress in this area. However, efficient high-dimensional indexing approaches are still a critical need that must be addressed. Human beings are the ultimate end users of image retrieval and extraction; therefore, research on human insight into image content from a psychophysical point of view is also of paramount significance. In all, a combination of interdisciplinary fields and sources of human and computer interaction will lead to powerful image retrieval and extraction systems.

12. CONCLUSION

The wide range of experiments on image retrieval and extraction demonstrate that low-level image features cannot illustrate the high-level semantically rich ideas from the user point of view. Content image retrieval and extraction should offer substantial support for linking the semantic gap between pictorial features and the wide range of human semantics. The paper provides a detailed survey of the latest work leading to semantics and ontology. The paper focuses on content retrieval, image search, fractal image compression, image similarity, semantic web, Scale Invariant Feature Transform (SIFT) and its variants.

The paper also provides valuable insights on how to obtain low-level significant features using scalable and heterogeneous components without computational complexity. In addition, the advantages and disadvantages of the related techniques are discussed, which should provide a platform for the introduction robust techniques for image retrieval, extraction and rapid and accurate searching. Experiments and components of related techniques are discussed to demonstrate the performance efficiency of each technique. Finally, based on the available current technologies and the needs of real-world applications, new research directions are summarized in introduction of a promising algorithm for image retrieval and extraction.

BIBLOGRAPHY

[1] E. Aptoula and S. Lefèvre, "Morphological description of color images for content-based image retrieval," *Image Processing, IEEE Transactions on*, vol. 18, pp. 2505-2517, 2009.

[2] Z. Wang, Q. Zheng, and J. Sun, "An effective content-based web image searching engine algorithm," in *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, 2010, pp. 1008-1012.

[3] J. Wang and X.-S. Hua, "Web-scale image search by color sketch," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 751-752.

[4] C. Gode and M. A. Ganar, "Image Retrieval by Using Colour, Texture and Shape Features," *Image*, vol. 3, 2014.

[5] O. Vincent and O. Folorunso, "A descriptive algorithm for sobel image edge detection," in *Proceedings of Informing Science & IT Education Conference (InSITE)*, 2009, pp. 97-107.

[6] A. Al-Hamami and H. Al-Rashdan, "Improving the Effectiveness of the Color Coherence Vector," *Int. Arab J. Inf. Technol.*, vol. 7, pp. 324-332, 2010.

[7] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the fourth ACM international conference on Multimedia*, 1997, pp. 65-73.

[8] P. E. Oppenheimer, "Real time design and animation of fractal plants and trees," in *ACM SiGGRAPH Computer Graphics*, 1986, pp. 55-64.

[9] S. Padmashree and R. Nagapadma, "Comparative Analysis of Quad-tree Partitioned Iterated Function Systems for Medical Images," *International Journal of Computer Applications*, vol. 59, pp. 25-27, 2012.

[10] M. Von Wyl, H. Mohamed, E. Bruno, and S. Marchand-Maillet, "A parallel cross-modal search engine over large-scale multimedia collections with interactive relevance feedback," in *Proceedings of the 1st ACM international conference on multimedia retrieval*, 2011, p. 73.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91-110, 2004.

[12] J. Huang, S. R. Kumar, and R. Zabih, "An automatic hierarchical image classification scheme," in *Proceedings of the sixth ACM international conference on Multimedia*, 1998, pp. 219-228.

[13] S. E. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on Gabor filters," *Image Processing, IEEE Transactions on*, vol. 11, pp. 1160-1167, 2002.

[14] X. S. Zhou, Y. Rui, and T. S. Huang, "Water-Filling: a novel way for image structural feature extraction," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, 1999, pp. 570-574.

[15] S. Nandagopalan, D. B. Adiga, and N. Deepak, "A universal model for content-based image retrieval," *jiP*, vol. 1, p. 5, 2008.

[16] M. R. Naphade and T. S. Huang, "Extracting semantics from audio-visual content: the final frontier in multimedia retrieval," *Neural Networks, IEEE Transactions on*, vol. 13, pp. 793-810, 2002.

[17] D. V. Ragatha and D. Yadav, "Image Query Based Search Engine Using Image Content Retrieval," in *Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on*, 2012, pp. 283-286.

[18] R. Maini and H. Aggarwal, "Study and comparison of various image edge detection techniques," *International Journal of Image Processing (IJIP)*, vol. 3, pp. 1-11, 2009.

[19] N. Senthilkumaran and R. Rajesh, "Edge detection techniques for image segmentation–a survey of soft computing approaches," *International Journal of Recent Trends in Engineering*, vol. 1, 2009.

[20] K. Kaur and S. Malhotra, "A Survey on Edge Detection Using Different Techniques."

[21] S. V. Pandit and M. Shreevastava, "A Technique for Image Search Using Fractal Quadtree Partitioned Iterated Function System-A Survey," *Image*, vol. 5, p. 6, 2013.

[22] G. Lu and T. Yew, "Image compression using quadtree partitioned iterated function systems," *Electronics Letters*, vol. 30, pp. 23-24, 1994.

[23] A. H. Fan and K.-S. Lau, "Iterated function system and Ruelle operator," *Journal of mathematical analysis and applications*, vol. 231, pp. 319-344, 1999.

[24] N. K. M. S. AH, "A survey on fractal image compression key issues," *Information Technology Journal*, vol. 7, pp. 1085-1095, 2008.

[25] K. Satish, T. Jayakar, C. Tobin, K. Madhavi, and K. Murali, "Chaos based spread spectrum image steganography," *Consumer Electronics, IEEE Transactions on*, vol. 50, pp. 587-590, 2004.

[26] E. K. Kaur, E. R. Chawla, and E. I. Kaur, "IMAGE COMPRESSION USING WAVELET TRANSFORM," *International Journal of Information Technology*, vol. 5, pp. 421-423, 2012.

[27] Z. Wang, Y. Mei, and K. Jia, "An Effective Web Image Searching Engine Based on SIFT Feature Matching," in *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on*, 2009, pp. 1-5.

[28] C. Ancuti and P. Bekaert, "SIFT-CCH: Increasing the SIFT distinctness by Color Co-occurrence Histograms," in *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*, 2007, pp. 130-135.

[29] Y. Yu, K. Huang, W. Chen, and T. Tan, "A novel algorithm for view and illumination invariant image matching," *Image Processing, IEEE Transactions on*, vol. 21, pp. 229-240, 2012.

[30] C. Town and D. Sinclair, "Language-based querying of image collections on the basis of an extensible ontology," *Image and Vision Computing*, vol. 22, pp. 251-267, 2004.

[31] T. Yao, C.-W. Ngo, and T. Mei, "Circular reranking for visual search," *Image Processing, IEEE Transactions on*, vol. 22, pp. 1644-1655, 2013.

[32] J. T. Robinson, "The KDB-tree: a search structure for large multidimensional dynamic indexes," in *Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, 1981, pp. 10-18.

[33] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta informatica*, vol. 4, pp. 1-9, 1974.

[34] A. Guttman, *R-trees: a dynamic index structure for spatial searching* vol. 14: ACM, 1984.

[35] R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *VLDB*, 1998, pp. 194-205.

[36] C. Yu, B. C. Ooi, K.-L. Tan, and H. Jagadish, "Indexing the distance: An efficient method to knn processing," in *VLDB*, 2001, pp. 421-430.

[37] S. Berchtold, D. A. Keim, and H.-P. Kriegel, "The X-tree: An index structure for high-dimensional data," *Readings in multimedia computing and networking*, vol. 451, 2001.