



INTEGRATED CONTENT-BASED IMAGE RETRIEVAL USING TEXTURE, SHAPE AND SPATIAL INFORMATION

By

Michael Eziashi Osadebey

February 2006

Master Thesis Report in Media Signal Processing

**Department of Applied Physics and Electronics,
Umeå University, Umeå
Sweden**

Supervisors:

Prof. Haibo Li & Apostolos, Georgakis, PhD

CONTENTS

	ABSTRACT	3
	ACKNOWLEDGEMENT	5
CHAPTER 1	INTRODUCTION	6
CHAPTER 2	REVIEW OF RELATED WORKS	11
CHAPTER 3	VISUAL FEATURES INTRODUCED	16
CHAPTER 4	TEXTURE	21
CHAPTER 5	SHAPE	44
CHAPTER 6	SPATIAL INFORMATION	51
CHAPTER 7	SYSTEM DESIGN	60
CHAPTER 8	SYSTEM OPERATION	72
CHAPTER 9	CONCLUDING REMARKS	79
APPENDIX A	REFERENCES	112
APPENDIX B	TEST RESULTS - UW	114
APPENDIX C	TEST RESULTS - PSU	134
APPENDIX D	TEST RESULT - SU	154

ABSTRACT

Content-based image retrieval (CBIR) systems demonstrate excellent performance at computing low-level features from pixel representations but its output does not reflect the overall desire of the user. The systems perform poorly in extracting high-level (semantic) features that include objects and their meanings, actions and feelings. This phenomenon, referred to as the semantic gap, has necessitated current research in CBIR systems towards retrieving images by the type of object or scene depicted.

Analyzing and interpretation of image data in large and diverse image database, as in a CBIR system is obviously difficult because there is no prior information on the size or scale of individual structures within the images to be analysed. The image processing and computer vision community had developed scale-space theory to deal with this problem. Scale-space theory incorporates multi-scale representation of a signal, which itself is an ordered set of derived signals intended to represent the original signal at different levels of scale. Scale-space theory is based on the fact that real-world objects exist as meaningful entities over certain ranges of scale, and human perceive them at coarse or fine scales depending on the scale of observation. According to the principle of scale-space theory, to be able to extract information from image data, a probe, sensor or operator is required to interact with the actual image structure. The information extracted is dependent on the relationship between size of the image structures and the size of the operators (probes). The probe is equivalent to the human view of the image while the size of the operator determines the resolution at which the image is observed.

Scale-space theory is in line with Gabor's formulation. A particular size of probe is simultaneously represented in two domains, time and frequency by dividing it into finite number of elementary information cells. Each of these information cells is the different scale at which the image data can be viewed, analysed and interpreted.

A lot of research effort had been devoted towards image retrieval using multiresolution analysis. In particular, Manjunath and Ma chose a single probe of arbitrary size, decomposed it simultaneously into two domains using Gabor formulation and derived for the probes a formula such that all the elementary information cells derived from it covers as much as possible the Fourier frequency space.

To strictly apply the principle of scale-space theory, all the variables that characterize the probe should be taken into consideration. They are the time-space, frequency and a window size. The third variable window size became necessary because since the quality of information we can get from probing any image is determined by the relationship between size of structures in the image and the size of the probe, it is logical to reason that a better analysis and interpretation of the image data can be obtained by using probes of various sizes. It is for this reason that I am of the view that the work of Manjunath and Ma, novel as it is, did not follow strictly the principle of scale-space theory. This is because, like other researchers, a single probe of arbitrary size was used in the CBIR design.

In this thesis it is shown that since objects concepts are usually related to visual characteristics high-level scene properties may be inferred from weighted combination of low-level image features – texture, shape and spatial information.

By using probes of various sizes and decomposing each sized-probe simultaneously in the frequency and time domains, artificial intelligence is impacted to the CBIR system to be able to locally 'scroll' through the images in the database and give better analysis and interpretation towards semantics, thereby bridging the semantic gap.

Of all the visual features, texture feature possess the properties for capturing semantic features in images because it is the aggregate contribution of various grey levels within the image.

The space/spatial-frequency tuning property of the neurons in the visual cortex as observed in physiological study is replayed by Gabor filters in image database.

Texture features derived from six grid sizes of independent and different Gabor filter banks were incorporated into the CBIR system by taking advantage of the fact that each grid size of filter is suited to capture particular set of localized frequency-images in diverse image database.

It is shown that Gabor filters can replay their efficient texture feature extraction in pure texture images in complex and real-world images because these images, though constituted by constant grey levels, the various constant grey levels within the global image constitute texture that can be captured by the tuneable characteristics of Gabor filters.

An integrated but simple, robust, flexible and effective image retrieval system using weighted combination of integrated Gabor texture features, shape features of texture regions and spatial information features of texture regions is hereby proposed.

The shape and spatial features are quite simple to derive and effective, and can be extracted in real time.

The system is integrated because it incorporates Gabor filters of six grid sizes namely 5 x 5, 15 x 15, 25 x 25, 35 x 35, 45 x 45, and 55 x 55.

The system is simple because of the ease with which the system can be operated and display results.

The system is flexible because the feature weights can be adjusted to achieve retrieval refinement according to user's need.

It is robust because the system's algorithm is applicable to retrieval in virtually all kinds of image database.

The system can successfully retrieve not only visual-content based queries but also queries based on semantics such as objects or scenes, hence it is a contribution towards the current research in semantic image retrieval.

In current CBIR systems the common method of improving retrieval performance is by weighting the feature vectors. In this thesis a new and reliable method of improving retrieval performance, and which complement feature weighting is proposed. Since the system use Gabor filter for texture feature extraction, the proposal is weighting the output (features) of the system as derived from various sizes of Gabor filter.

The system has the potential of developing into a semantic based CBIR system by proper mathematical modelling of the texture features obtained from the six grid sizes of Gabor filter.

Based on results obtained from this thesis, i hereby state that the key to a breakthrough in current research in semantic image retrieval lies in the use of Gabor texture feature. Its benefits of Fourier (global) as well as local analysis of images enables analysis of gradual changes of texture and texture variations which are essential properties of real-world scenes. Incorporating and integrating various sizes of Gabor filters in a CBIR system to derive texture feature and flexible weighting with shape and spatial features can be used to model and extract high level features in images.

ACKNOWLEDGEMENT

The primary motivation for this thesis was the knowledge acquired in media signal processing, image processing, digital vision and biomedia. For this reason I hereby express my gratitude to my lecturers in these areas of knowledge namely, Apostolos Georgakis, Sara Sjörgren, Ulf Holmgren, Erik fällman, Adi Anani, Liu Li and Professor Haibo Li.

I am particularly grateful to my supervisors for this thesis, Apostolos Georgakis and Professor Haibo Li whose strict supervision gave me the enabling environment and a real taste of thesis work.

My gratitude will not be complete without mention of Alexander Jayasundara, I prefer to call him Alexander the great. His friendly encouragement and down-to earth tutorial on graphical users interface was of immense help to me.

Finally, I give thanks to Almighty God, the Lord Jesus Christ and the Holy spirit who had been my guardian, and source of strength and hope on this earthly journey.

1

INTRODUCTION

1.1 The use of images

Historical records show the use of images date back to paintings on walls of cave by early man. In the pre-Roman times images were seen mostly in the form of building plans and maps [1]. The need and use of images grew with the ages, particularly with the advent of photography in the sixteenth century. In the twentieth century, introduction of computer and advances in science and technology gave birth to low cost and efficient digital storage devices and the worldwide web, which in turn became the catalyst for increasing acquisition of digital information in the form of images.

In this computer age virtually all spheres of human life including commerce, government, academics, hospitals, crime prevention, surveillance, engineering, architecture, journalism, fashion and graphic design and historical research are in need of, and use of images for efficient services. A large collection of images is referred to as ***image database***. Image database is a system where image data are integratedly stored [2]. Image data include the raw images and information extracted from images by automated or computer assisted image analysis.

The police maintain image database of criminals, crime scenes and stolen items. In the medical profession X-rays, and scanned image database are kept for diagnosis, monitoring and research purposes. In architectural and engineering design image database exist for design projects, finished projects and machine parts. In publishing and advertising journalists create image database for various events and activities such as sports, buildings, personalities, national and international events, and product advertisements. In historical research image database are created for archives in areas that include arts, sociology and medicine.

1.2 Image retrieval problem

In small collection of images simple browsing can identify an image. This is not the case for large and varied collection of images, where the user encounters image retrieval problem. Image retrieval problem is the problem of searching and retrieving images that are relevant to a user's request from a database. A typical retrieval problem example is a design engineer who needs to search his organisation database for design projects similar to that required by his clients or the police seeking to confirm the face of a suspected criminal among faces in the database of renowned criminals. In the commerce department before trademark is finally approved for use there is need to find out if such or similar ones ever existed. In the hospital some ailments require the medical practitioner to search and review similar X-rays or scanned images of a patient before proffering solution.

1.3 Visual content levels

Images are naturally endowed with attributes or information content that can help in resolving the image retrieval problem. The information content that can be derived from an image is classified into three levels. See Fig.1

- **Low level** – They include visual features such as colour, texture, shape, spatial information and motion.
- **Middle level** – Examples include presence or arrangement of specific types of objects, roles and scenes.
- **High level** – Include impressions, emotions and meaning associated with the combination of perceptual features. Examples include objects or scenes with emotional or religious significance.

The image content level is also a measure of level of feature extraction. At the low level, also regarded as primary level the features extracted (color, shape, texture, spatial information and motion) are called primitive features because they can only be extracted by information obtained at the pixel level, that is pixel

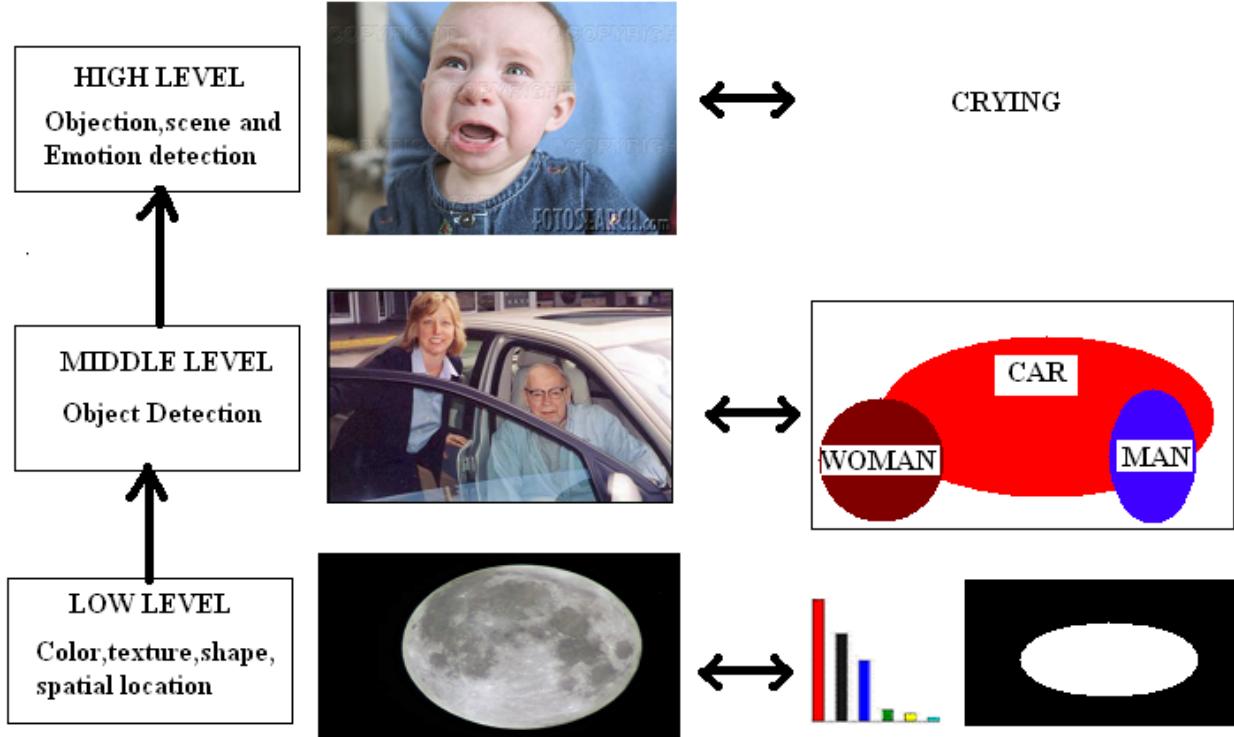


Fig. 1 Examples of image content levels

representation of the images. The middle level features are features that can be extracted by collection of pixels that make up the image, while high level features goes beyond the collection of pixels. It identifies the impressions, meanings and emotions associated with the collection of pixels that make up the object.

1.4 Text-based retrieval and Content-based retrieval

An image retrieval system is a computer system for browsing, searching and retrieving images in an image database. Text-based and content-based are the two techniques adopted for search and retrieval in image database.

In text-based retrieval, images are indexed using keywords, subject headings or classification codes, which in turn are used as retrieval keys during search and retrieval. Text-based retrieval is non-standardized because different users use different keywords for annotation. Text descriptions are sometimes subjective and incomplete because it cannot depict complicated image features very well. Examples are texture images that cannot be described by text. In text retrieval, humans are required to personally describe every image in the database, so for a large image database the technique is cumbersome, expensive and labour-intensive.

Content-based image retrieval (CBIR) technique use image content to search and retrieve digital images . Content-based image retrieval system was introduced to address the problems associated with text-based image retrieval, [3]. Advantages of content-based image retrieval over text-based retrieval will be mentioned in the next sections.

However, text-based and content-based image retrieval techniques complement each other. Text-based techniques can capture high-level feature representation and concepts. It is easy to issue text queries but text-based techniques cannot accept pictorial queries. On the other hand, content-based techniques can capture low-level image features and accept pictorial queries. But they cannot capture high-level concepts effectively. Retrieval systems exist which combine both techniques for more efficient retrieval [1], [4], [5].

1.5 Principle of CBIR

A typical CBIR system as shown in Fig.2 automatically extract visual attributes (colour, shape, texture and spatial information) of each image in the database based on its pixel values and stores in a different database within the system called **feature database**. The feature data for each of the visual attributes of each image is very much smaller in size compared to the image data. Thus the feature database contains an abstraction (compact form) of the images in the image database; each image is represented by a compact representation of its contents (colour, texture, shape and spatial information) in the form of a fixed length real-valued multi-component **feature vectors or signature**.

The users usually formulate query image and present to the system. The system automatically extract the visual attributes of the query image in the same mode as it does for each database image, and then identifies images in the database whose feature vectors match those of the query image, and sorts the best similar objects according to their similarity value.

During operation the system processes less compact feature vectors rather than the large size image data thus giving CBIR its cheap, fast and efficient advantage over text-based retrieval.

CBIR system can be used in one of two ways. First, exact image matching, that is matching two images, one an example image and the other, image in image database. Second is approximate image matching, which is finding most closely match images to a query image.

1.6 Outline of this thesis

This report is divided into nine chapters. The next chapter reviews related work. Visual features are introduced in chapter three. The next three chapters discuss the three visual features used in the proposed CBIR system. The design of the CBIR system is in chapter seven followed by its operation. The report ends with concluding remarks in chapter 9.

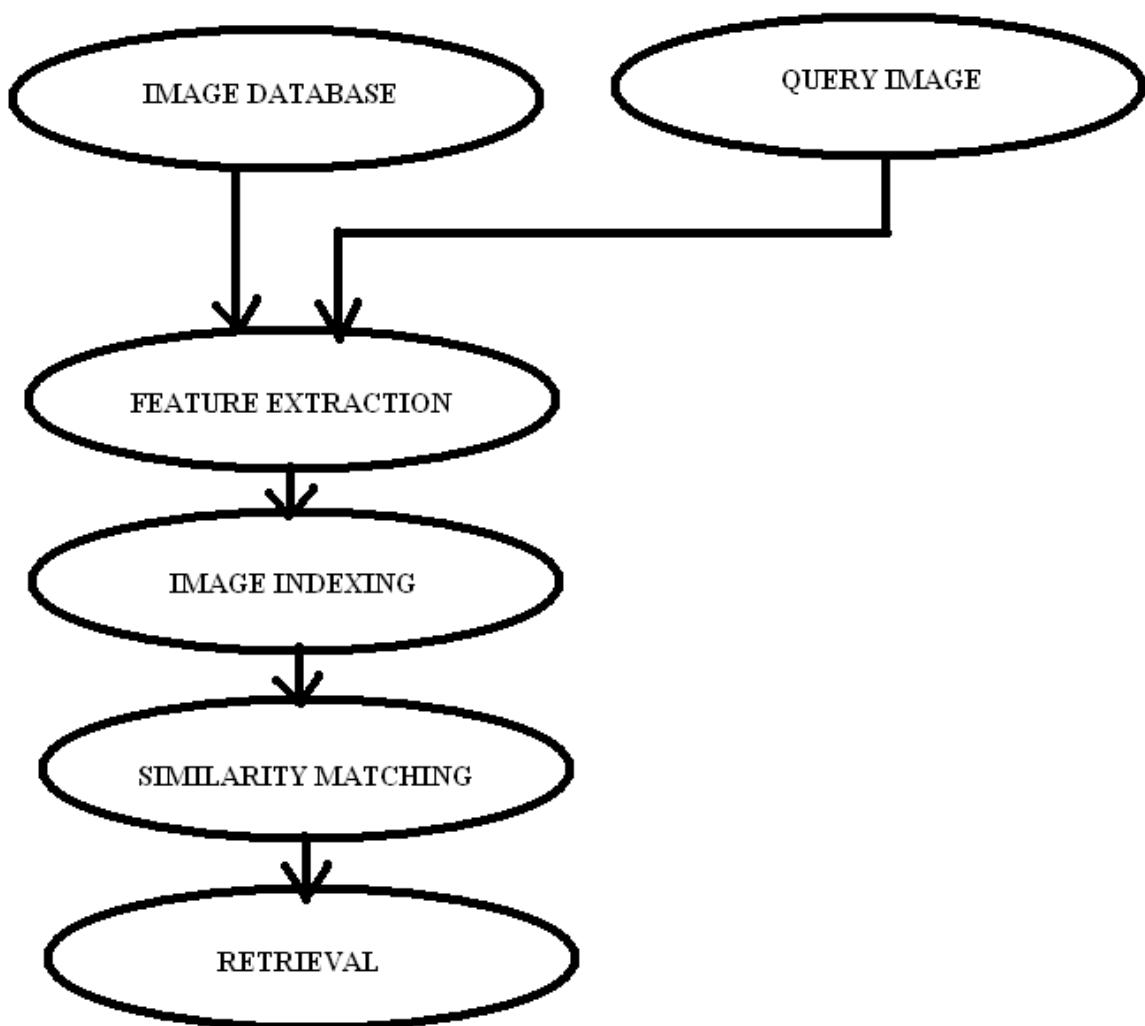


Fig. 2 Flow chart showing principle of CBIR system

2

REVIEW OF RELATED WORKS

2.1 Introduction to the chapter

A considerable number of image retrieval systems have been developed for commercial use and demonstrations versions are in existence in the academic world. The commercial systems include **IBM's QBIC**, **VIR** image engine search from Virage Inc and Excalibur from Excalibur technologies. Several experimental systems exist in the academia, notable among them are **Photobook** system from Massachusetts institute of technology (MIT), **VisualSEEK** system of Columbia University and **MARS** of University of Illinois and NETRA of University of California, Santa Barbara, [1]. A total of fifty-eight CBIR systems were reviewed in [5]. I will discuss only five CBIR systems whose systems of operation are most closely related to this project work. They are **NETRA**, **RETIN** (Recherche et Traque Interactive) developed by ENSA/University of cergy-Pointoise, France, **KIWI** (Key-points Indexing Web Interface) of INSA Lyon, France, **iPURE** (Perceptual and User-friendly Retrieval of Images) developed by IBM India Research lab, New Delhi, India, and **ImageMiner** developed by Technologie-Zentrum Informatik, Uinversity of Bremen, Germany.

2.2 NETRA

NETRA CBIR system use colour, texture, shape and spatial location. The descriptor for the colour is colour histogram obtained using training set of images. Texture feature descriptor is the normalized mean and standard deviation of the Gabor wavelet transform of the images. Shape feature descriptors are the curvature function of the contour, the centroid distance function of the contour and the complex coordinate function of the contour. Spatial location descriptor is bounding box

The system allows query by example.

Similarity matching is carried out in the Euclidean space.

2.3 RETIN

RETIN CBIR system use colour and texture. The system unlike the **NETRA** system does not process image as a single entity. Rather for each image random pixels are selected of which colour and texture feature are computed. The descriptor for colour is colour histogram while the texture descriptor is the output of Gabor transformation of the images.

The system allow query by example. However the system is flexible enough to allow the user to locate region of interest within a query image and use it for search.

Similarity matching is carried out using weighted minkowski distance and cross matching between bins.

2.4 KIWI

KIWI system like the **RETIN** system detects key points in an image rather than the entire image. It uses colour and shape. The colour descriptor is colour histogram while the shape descriptors are computed with the aid of Gabor filters.

The system allows query by example.

Similarity matching is carried out in the Euclidean space.

2.5 iPURE

iPURE system operates using colour, texture, shape and spatial location. The system segments the image into regions before deriving the features. Colour descriptor is the average colour in CIE's (Comission Internationale de l'Eclairage) Luv colour space. Texture descriptors are by means of wold decomposition. Shape descriptors are size, orientation axes and Fourier descriptor. The spatial location descriptors are centroid, and bounding box.

The system accept query by example.

Similarity matching s carried out in the Euclidean space.

2.6 ImageMiner

ImageMiner system use colour, texture and shape. Colour descriptors is colour histogram, texture description is grey level co-occurrence matrix. Shape description is carried out using image contour size, centroids and boundary coordinates. The system feature description has the capability to classify scenes and generate description of the scene with keywords and values.

The system allows query by example.

Special module within the system carries out similarity matching.

2.7 Review

CBIR system effectiveness is measured by its **precision** and **recall**. Precision is the ratio of relevant images to the total number of images retrieved. Recall is percentage of relevant images among all possible relevant images. Though manufacturers of CBIR systems give good figures about its precision and recall, it is difficult to evaluate how successful content-based image retrieval systems are in terms of effectiveness, efficiency and flexibility. For example if a database is narrow with only airplanes the system will only return airplanes as similar images even if the query image is something other than airplane. Also if the database is diverse but contains only a single chicken, the best that can be achieved is return of only one chicken, in response to a chicken query. Thus the more diverse and larger the collection of images in the database the more chance that it contains images similar to the query image [5]. This review points to the fundamental issues in the design of CBIR including its image database, which are

- Efficient selection of images which satisfy a user's query
- Data modelling
- Feature extraction
- Selecting appropriate features for content representation
- Query languages, and
- Indexing techniques.

Now a closer look at the CBIR systems as reviewed in [5].

Virtually all the systems incorporate colour in their feature extraction. Most of the colour descriptors are based on colour histograms. Merely more than fifty percent of the systems use texture feature. Shape features are utilized in half of the systems reviewed. Most of the shape features are the simple ones, for example, eccentricity, area and orientation. Less than half of the systems use spatial information. And of course the spatial layout feature does not go beyond the simple type, which is bounding box and centroid.

Colour, texture, shape and spatial information feature descriptors that are highly efficient, robust to noise and invariant to rotation, scale and translation have been developed. However these seems unattractive to the CBIR system developers. According to [5] the frequency of usage of the feature descriptors is inversely proportional to the computational complexity involved in extracting the features. Colour is commonly used because colour features are easily designed and effectively implemented. It was also observed that the systems perform poorly in retrieval using only shape features. This is not surprising because the elementary shape descriptors adopted are not much effective because they are prone to noise.

All the systems operate with low-level features. They do not operate at higher semantic level, hence cannot recognise object and scenes except for the attempt by developers of **ImageMiners** to classify objects and scenes.

It is said that existing CBIR systems and their users are at cross-purpose. The reason is that the output of the system does not represent the entire meaning of what the user have in mind. While the systems perform excellently at automatically computing low-level features from pixel representations its output does not reflect the overall desire of the user. In other words the system perform poorly in extracting high-level (semantic) features that include objects and their meanings, actions and feelings. Experiences have shown that no matter the sophistication and effectiveness of a low-level feature representation they do not adequately model abstract semantic concepts within images. This is the limitation of existing CBIR systems. This gap between the outputs of the system as computed from low-level features and the high-level semantic descriptions in the user's mind is often referred to as the *semantic gap*. If computers can be seen to exhibit intelligence in numerical calculation, semantic gap is a clear indication that visual intelligence cannot be produced by mere calculation.

Current research in CBIR systems is geared towards bringing some measure of automation to the processes of indexing and retrieving images by the type of object or scene depicted. Since objects concepts are usually related to visual characteristics it is not out of place to expect that high-level scene properties may be inferred from low-level image features such as colour and texture information. While some researchers [6] are of the view that bridging the semantic gap between the low-level features and the high-level semantics is within the interface between the user and the system, other research direction is towards improving aspects of CBIR systems by finding the latent correlation between low-level visual features and high-level semantics and integrating them into a unified vector space model [7]

2.8 MOTIVATIONS

Remarkable observations in the review of related works are as follows

- Despite the fact that Gabor filters are widely acclaimed natural and excellent tool in texture feature classification, segmentation and extraction, only two CBIR systems utilize Gabor filters for texture feature extraction.
- In the literature texture feature extraction methods using Gabor filters are widely applied to only pure texture images. Do the preferences of CBIR system designers imply that the efficiency of Gabor feature extractors does not extend to real-world scenes?

- Current CBIR systems shy away from the combination of texture, shape and spatial information for retrieval. Rather than use solely these three features the system designer add a fourth feature usually colour, most probably to help improve its efficiency.

Gabor filters are known to model the receptive fields of the neurons in the human visual cortex. Since objects and scenes detection is possible through human intelligence, it implies that proper treatment and manipulation of Gabor function can lead to high level feature extraction

Real-world scenes constitute combination of colours, and since texture is derived from independent contribution of various brightness levels of colour pixels, texture feature can be said to be established in real-world images. Thus efficiency of Gabor filter in pure texture images can be replayed in real-world scenes.

In most current CBIR systems each of the feature extraction techniques used for retrieval are treated with equal emphasis. In other words, no one particular feature dominates in similarity matching and retrieval. However it is a statement of fact that every real-world image is particularly suited to one particular feature extraction method, and that no particular feature can best express all types of images. If the feature most suitable for retrieval of a particular image is used in flexible weighted combination with other features it is expected that a higher level of precision and recall can be achieved. Flexible weighted combination of the image features can provide the basis for elevating current CBIR system to recognise objects and scenes.

Since objects concepts are usually related to visual characteristics high-level scene properties may be inferred from weighted combination of low-level image features – texture, shape and spatial information. These are the motivations for the Masters thesis.

3

VISUAL FEATUURES INTRODUCED

3.1 Feature defined

Feature is anything that is localized, meaningful and detectable. In an image noticeable features include corners, lines, objects, colour, shape, spatial location, motion and texture.

3.2 Feature explained

Features extracted from images define and describe the image content. In simple images such as that shown in Fig. 3 (1) and Fig.3 (2), the feature is easily identified as an object that is circular in shape and a textured zebra respectively. However this is not the case for complex images such as that shown in Fig. 3 (3) and Fig.3 (6). Complex images are complex because of the complex task involved in identifying appropriate features with which to describe the image. Complex images or real-world scene images consist of multiple objects. As there are many objects in the image so many features are competing as candidates for its description. An appropriate way out of this problem is identifying appropriate image feature(s) within the image that best represent the image and represent or combine them such that during retrieval process meaningful result can be obtained. Identifying suitable image feature for describing a particular type or class of image reduces storage size of indexing features used in programming, leading to efficient and fast CBIR system.

Features are used to represent an image instead of using the original pixel values because of the significant simplification of image representation and the improved correlation with image semantics. The beauty of CBIR is seen from the fact that though computers cannot understand images, comparison of visual features as enabled by the feature vectors enables comparison of the real-world visual scenes. ***Hence it is said that indexing bridge the gap between image semantics and the pixel representation.***

No particular visual feature is most suitable for retrieval of all types of images. Colour visual feature is most suitable for describing and representing colour images. Texture is most suitable for describing and representing visual patterns, surface properties and scene depth. CBIR system using texture is particularly useful in satellite images, medical images and natural scenes like clouds, while shape is suitable for representing and describing boundaries of real world objects and edges. In reality no one particular feature can completely describe an image. Consider the images of a Zebra shown in Fig. 3 (2), natural scene shown in Fig. 3 (4), a typical apartment shown in Fig. 3 (6) and that of a lion in grassland shown in Fig. 3 (5). All these examples depict the challenges a typical CBIR system designer will face in seeking features with which to describe images in database. Natural scenes are best described by colour and texture while shape is the least feature for its description. Shape can only be included in its description only when all images in the database consist of natural scenes. In this case shape and spatial location can be used to discriminate between images in the database. For simple images as the circular moon and zebra shown in Fig. 3 (1) and Fig. 3 (2), shape is the best descriptor followed by texture and colour. Spatial location is its least descriptor.

3.3 Visual features explained

In this project work level 1 or low level image content relating to the human visual perception are considered for solving the retrieval problem. Visual features of an image is the output of the human visual system as represented by human visual perception of images, and these include color, texture,shape, spatial information and motion. CBIR system based on human visual perception is also known as content-based visual information retrieval (CBVIR). At this point it is necessary to explain how the human visual system works. This is because it will be referred to in later section of this report.

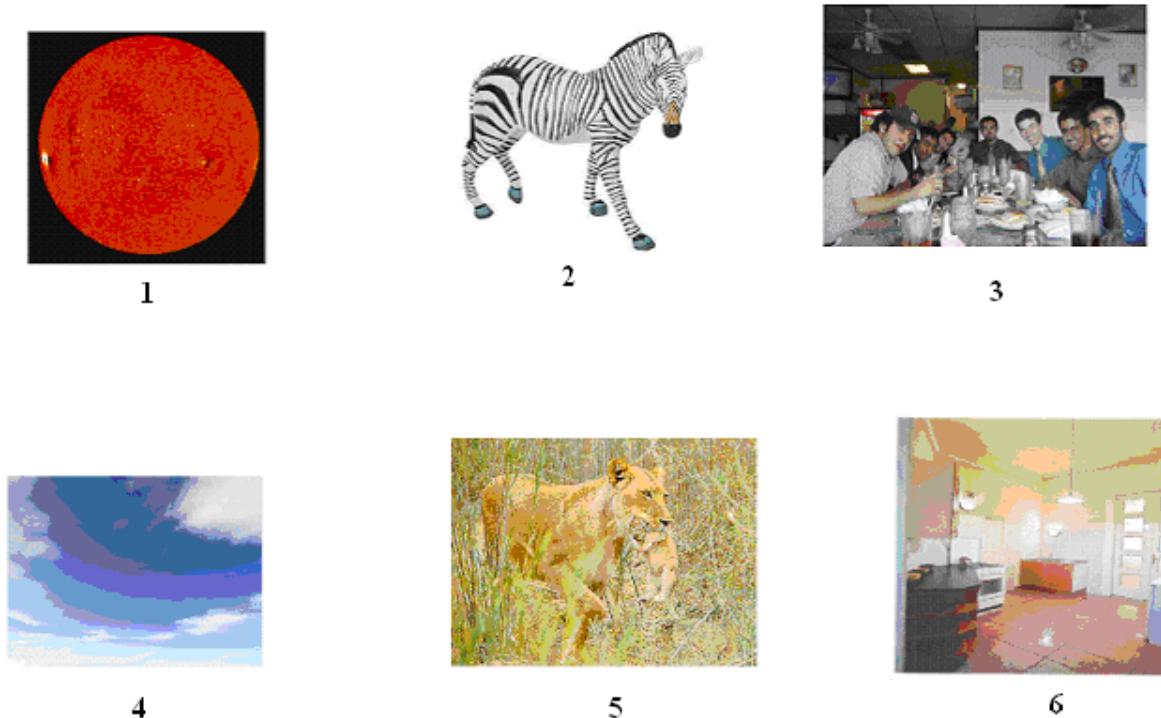


Fig.3. Examples of images with simple and complex features

3.4 The human Visual system

The human eye is composed of layers and structures as shown in Fig.4 . Each distinct layer and structure performs distinct functions for vision to take place.

Figure 5 shows light rays reflected by a real world object and enter the eye first through the **cornea**, before passing through the **aqueous humour** and **pupil**. The light then strikes the **lens**. The lens projects an inverted image of the object onto the retina at the back of the eye. Visual processing of the light signal begins in the retina. At the retina are millions of photoreceptors called **rod cells** and **cone cells** which are stimulated by the light rays that strike them, and thus produce electrical signals. The **optic nerve** consists of retina fibers that transport the electrical signal to the **visual cortex**, located at the back of the brain. The visual cortex is the most massive system in the human brain and is responsible for higher-level processing of the visual image.

At the visual cortex, the **primary visual cortex** processes the signal and gives it local interpretation. The signal fans out to other areas of the visual cortex where global interpretation of the object is obtained after processing. When all parts of the visual system are working, the eyes can move together, can adapt to light and dark environment, perceive colour, shape, texture, and motion and accurately evaluate an object's location in space, [8].

An important term in visual information processing is the term 'receptive fields'. It is the region of the visual field of the neurons in which spatial patterns of light influence the neuron's behaviour. In neurophysiological study of visual receptive fields, 1981 Nobel laureates David Hubel and Thorsten Wiesel discovered that some of the cortical receptive fields respond best when the stimulus was of a certain shape, had a given orientation and or moved in a given direction. For example, one receptive field might respond best when a real world object

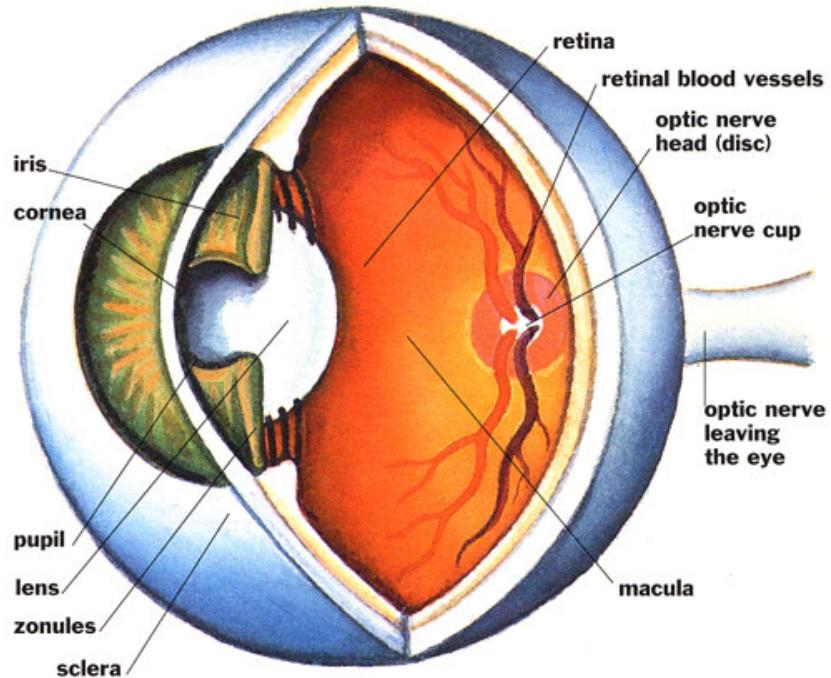


Fig. 4 Structure of the eye

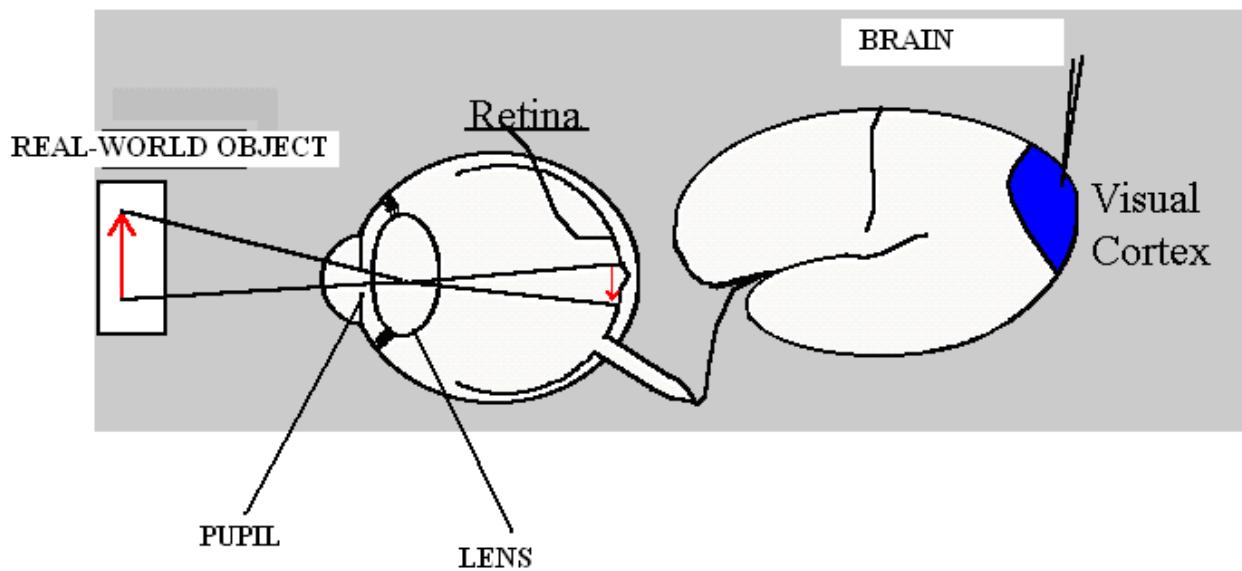


Fig. 5. How the brain relates to the eye in the human visual system

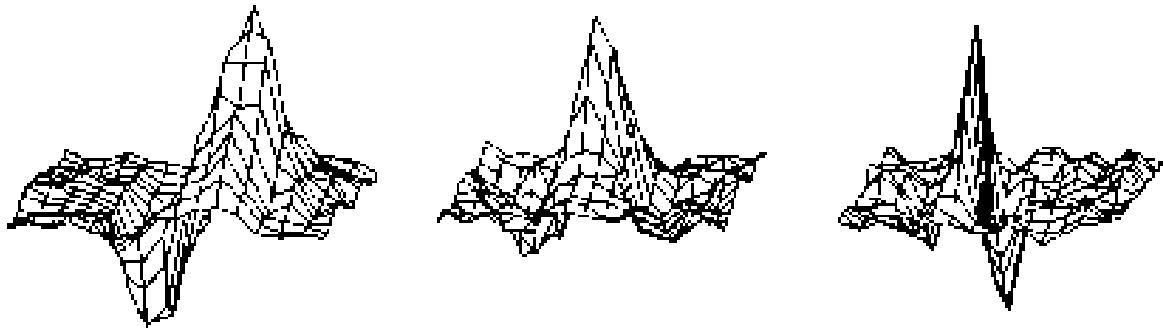


Fig. 6 Typical receptive field of the neurons in the visual cortex

moves to the right, but not when it moves in other directions. Three representative receptive fields typical of the neurons in the visual cortex are shown in Fig. 6.

The works of Drs. Hubel and Wiesel established the basic architecture for how the brain analyzes visual patterns. Their work strongly influenced the direction of other work in sensory physiology especially in the studies of somatosensory mechanisms, audition, and the coordination of movement. Computer algorithms incorporating similar analysis and logic are now a mainstay of automatic systems for image and speech recognition.

Scientists are of the view that the human visual system appears to have multiple mechanisms (channels) tuned to different spatial frequencies; neurons or cells which constitute the visual cortex show tuning properties including selectivity for orientation, motion, direction, and are sensitive to color, contrast and spatial frequency. Current consensus among scientists seems to be that the visual cortex consists of tiled sets of selective spatiotemporal filters. In the spatial domain, characteristics of the visual cortex can be thought of as similar to many spatially local, complex Fourier transform. Theoretically, these filters together can carry out neuronal processing of spatial frequency, orientation, motion, direction, speed, and many other spatiotemporal features. Experiments of the visual cortex neurons substantiate this theory.

While the receptive fields of the cortical simple cells of the visual cortex all differ from each other, they have some common features. The receptive field profiles consist of spatially **local, oriented, decaying bands** of excitation and inhibition. The term locality implies that two cells may have similar looking receptive field profiles that differ only in the region of the field of view where the active response occurs. Oriented, decaying bands implies that the receptive fields decay from their centers. Experiments with oriented bars and sinusoidal gratings acting as real-world objects suggest that these cortical cells act as band pass filters with a bandwidth of approximately 1.5 octaves and an orientation bandwidth of approximately 45 degrees, [9], [10].

4

TEXTURE

4.1 Texture defined

According to the American Heritage dictionary texture is

- A structure of interwoven fibres or other elements such as repetitive patterns.
- The distinctive physical composition or structure of something, especially with respect to the size, shape, and arrangement of its parts.
- The appearance and feel of a surface.
- Distinctive or identifying quality or character.

In the field of computer vision and image processing there is no clear-cut definition of texture. This is because available texture definitions are based on texture analysis methods and the features extracted from the image. However texture can be thought of as repeated patterns of pixels over a spatial domain, of which the addition of noise to the patterns and their repetition frequencies result in textures that can appear to be random and unstructured. Texture properties are the visual patterns in an image that have properties of homogeneity that do not result from the presence of only a single colour or intensity.

The different texture properties as perceived by the human eye are regularity, directionality, smoothness and coarseness. See Fig. 7 (A).

In real world scenes texture perception can be far more complicated. The various brightness intensities give rise to a blend of the different human perception of texture as shown in Fig. 7 (B).

Image textures have useful applications in image processing and computer vision. They include

- Recognition of image regions using texture properties, otherwise known as texture classification.
- Recognition of texture boundaries using texture properties, otherwise known as texture segmentation.
- Texture synthesis, generation of texture images from known texture models.
- Extraction of image shape using texture properties.

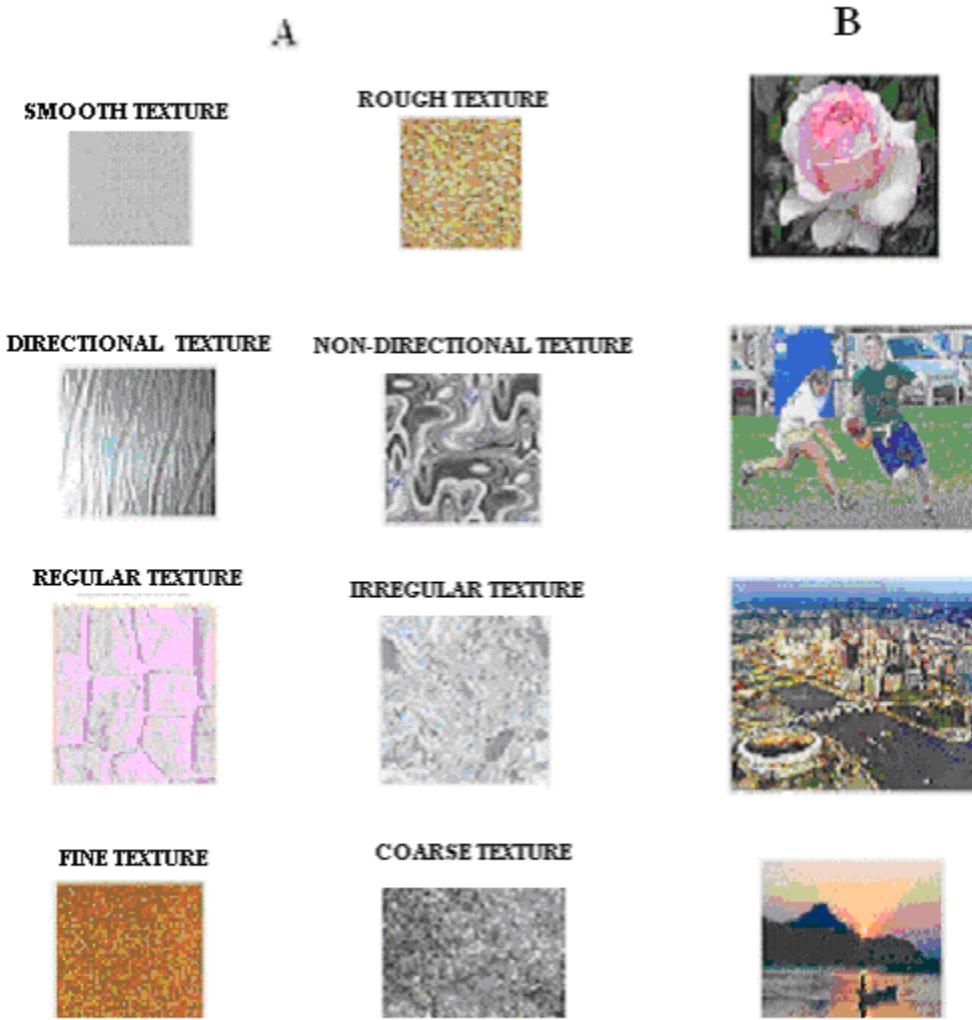


Fig. 7 Examples of simple and complex texture images

4.2 Texture feature extraction

Since there is no accepted mathematical definition for texture, many different methods for computing texture features have been proposed over the years. Unfortunately, there is still no single method that works best with all types of textures. The commonly used methods for texture feature description are statistical and transform-based methods [11], [12].

4.3 Statistical method

Statistical methods analyse the spatial distribution of grey values by computing local features at each point in the image, and deriving a set of statistics from the distribution of the local features. They include co-occurrence matrix representation, statistical moments, grey level differences, autocorrelation function and grey level run lengths.

4.4 Gray-level Co-occurrence matrix

Gray level co-occurrence method use grey-level co-occurrence matrix to sample statistically the way certain grey-levels occur in relation to other grey-levels. Grey-level matrix is a matrix whose elements measure the relative frequencies of occurrence of grey level combinations among pairs of pixels with a specified spatial relationship. The theory of grey level co-occurrence matrix is explained below.

Given an image $Q(i, j)$, let $p(i, j)$ be position operator, and A be a $N \times N$ matrix whose element $A(i, j)$ is the number of times that points with grey level (intensity) $g(i)$ occur, in the position specified by the relationship operator, p , relative to points with grey level $g(j)$.

Let P be the $N \times N$ matrix that is produced by dividing A with the total number of point pairs that satisfy p . $P(i, j)$ is a measure of the joint probability that a pair of points satisfying p will have values $g(i)$, $g(j)$. P is called a co-occurrence matrix defined by p . The relationship operator is defined by an angle θ and distance d .

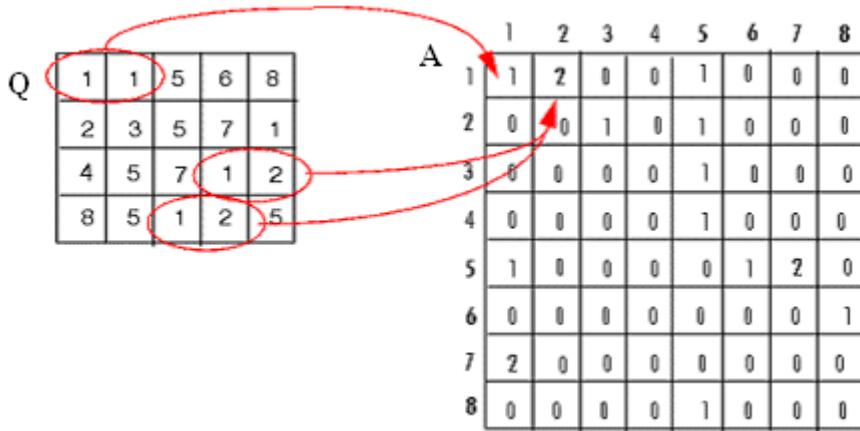


Fig. 8 Demonstration of co-occurrence matrix representation

From P the following texture descriptors can be computed

Energy,

$$\sum_{i,j} P_{i,j}^2 \quad (1)$$

Inverse difference moment,

$$\sum_{\substack{i,j \\ i \neq j}} \frac{P_{i,j}^2}{|i-j|} \quad (2)$$

Entropy,

$$\sum_{i,j} P_{i,j} \log P_{i,j} \quad (3)$$

Maximum probability,

$$\max(P_{i,j}) \quad (4)$$

Contrast,

$$\sum_{i,j} |i - j| P_{i,j}^2 \quad (5)$$

Correlation,

$$\sum_{i,j} \frac{(i - \mu)(j - \mu)P_{i,j}}{\sigma^2} \quad (6)$$

The diagram in Fig. 8 explains the generation of co-occurrence matrix for a 4×5 matrix image, Q , having grey levels ranging from 1 to 8. The position operator p is defined as a unit pixel distance, $d = 1$ and $\theta = 90$ degrees towards the east. The matrix $A(i, j)$ is an 8×8 matrix showing number of times that points with grey level (intensity) $g(i)$ occur, in the position specified by the relationship operator, p , relative to points with grey level $g(j)$. In this example $A(1,1)$ contains the value 1 because there is only one instance in the image where two, horizontally adjacent pixels have the values 1 and 1. $A(1,2)$ contain the value 2 because there are two instances where two horizontally adjacent pixels have the values 1 and 2.

Another matrix $P(i, j)$ which is a measure of the grey level co-occurrence matrix, also a 8×8 can be produced by dividing A with the total number of point pairs that satisfy p .

Gray-level co-occurrence matrix method of representing texture features have found useful application in recognizing fabric defects. The basis of the application is derived from the fact that a fabric defect image is characterized by its primitive properties as well as the spatial relationships between them. A gray level co-occurrence is specified in a matrix of the relative frequencies with which two neighboring pixels separated by a distance occur on the image. By applying the co-occurrence matrix and gray relational analysis of the gray theory, characteristic values of a fabric defect image is extracted and the defects classified to recognize common problems such as broken warps, broken wefts, holes, and oil stains [13]. Other useful applications of gray level co-occurrence matrix methods are in rock texture classification and retrieval [14], [15]

4.5 Transformed-based method

Transform methods analyse the frequency content of the image to determine texture features. Examples include the use of Fourier transform to describe the global frequency content of the image and multi-resolution analysis (wavelet transform and Gabor wavelets) that uses a window function whose width changes as the frequency changes. Multiresolution analysis, the representation or analysis of signals at different scales subjects the image to a linear transform followed by energy measurement. In the following eight sections, Gabor wavelets the basic tool of this project work will be discussed in detail.

4.6 Gabor wavelet building block: The Complex sinusoid

A sinusoid is the curve of the sine function. It is sometimes called circular function as shown in Fig. 9(A) because the essential feature of the sine function can be thought of as a point moving around a circle in a uniform way, and the value of sine being the height of the point. Fig. 9(B) details the profile of the sinusoid.

The complex sinusoid is a two-dimensional sinusoid. The first dimension is the real axis. It contains a cosine wave. The second dimension is perpendicular to the first dimension. It is the imaginary axis and contains a sine wave. Fig. 10 shows a complex sinusoid in the time domain. The profile of the complex sinusoid, which is the vector combination of the real and imaginary components, as shown in the Fig. 10 is a spiral circular wave.

Mathematically it is expressed as

$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + P)) \quad (7)$$

u_0, v_0 are the spatial frequencies in the cartesian coordinates, that is along the x - and y -axes

P is the phase

The mathematical expressions for the real and imaginary parts of the complex sinusoid are

$$\text{Re}(s(x, y)) = \cos(2\pi(u_0x + v_0y) + P) \quad (8)$$

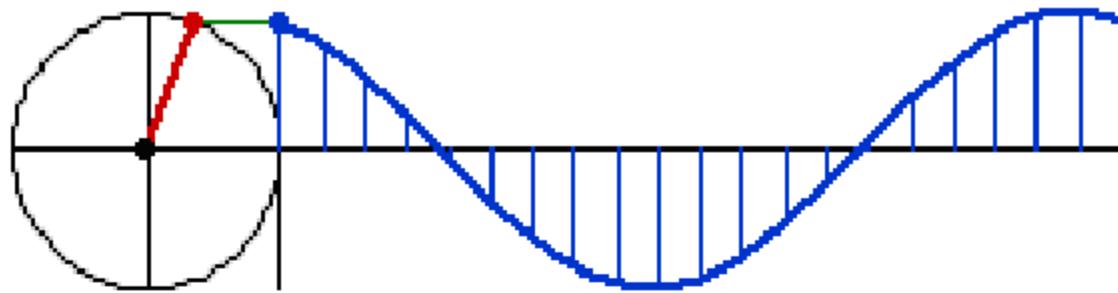
$$\text{Im}(s(x, y)) = \sin(2\pi(u_0x + v_0y) + P) \quad (9)$$

In polar coordinates the spatial frequency is expressed as having a magnitude

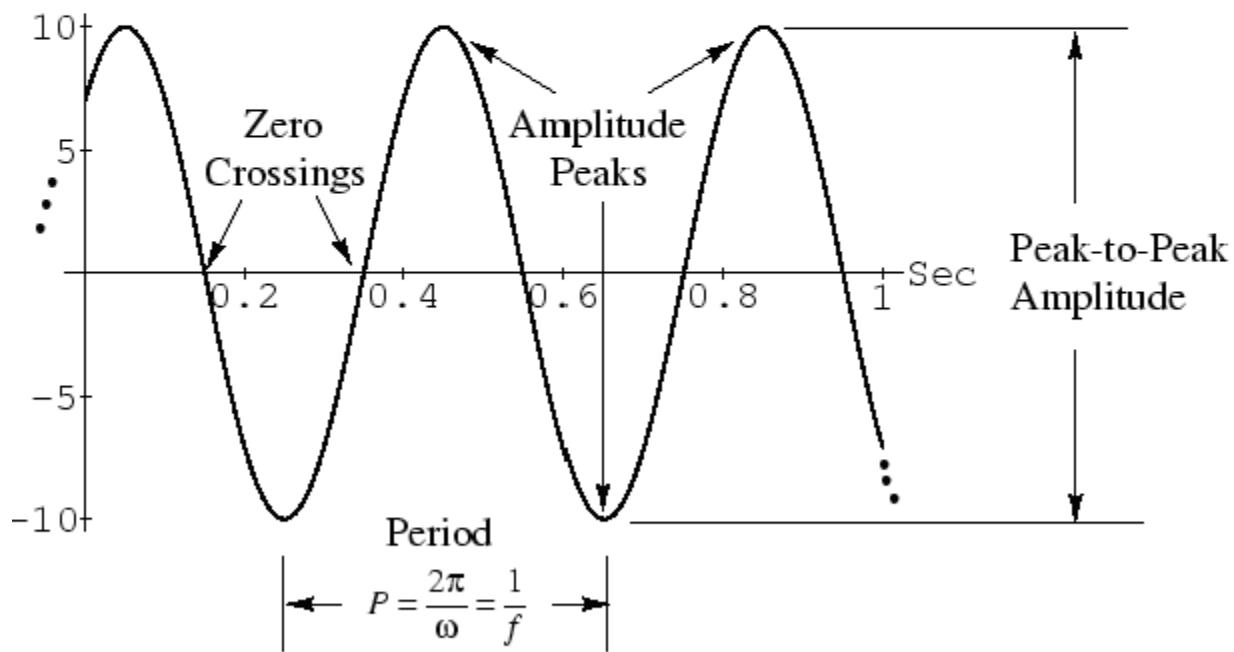
$$F_0 = \sqrt{u_0^2 + v_0^2} \quad (10)$$

and direction

$$\omega_0 = \tan^{-1}\left(\frac{v_0}{u_0}\right) \quad (11)$$



A. Demonstrating complex sinusoid as a circular function



B. Profile of a sinusoid

Fig. 9 The sinusoid

Thus in polar coordinates the complex sinusoidal is expressed as

$$s(x, y) = \exp(j(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P)) \quad (12)$$

4.7 Gabor building block: Gaussian function

The Gaussian function is the mathematical expression for data which are randomly distributed about a mean value of M and with a dispersion σ . Fig. 11 show a one and two dimensional Gaussian function having mean at the origin and standard deviation of unity.

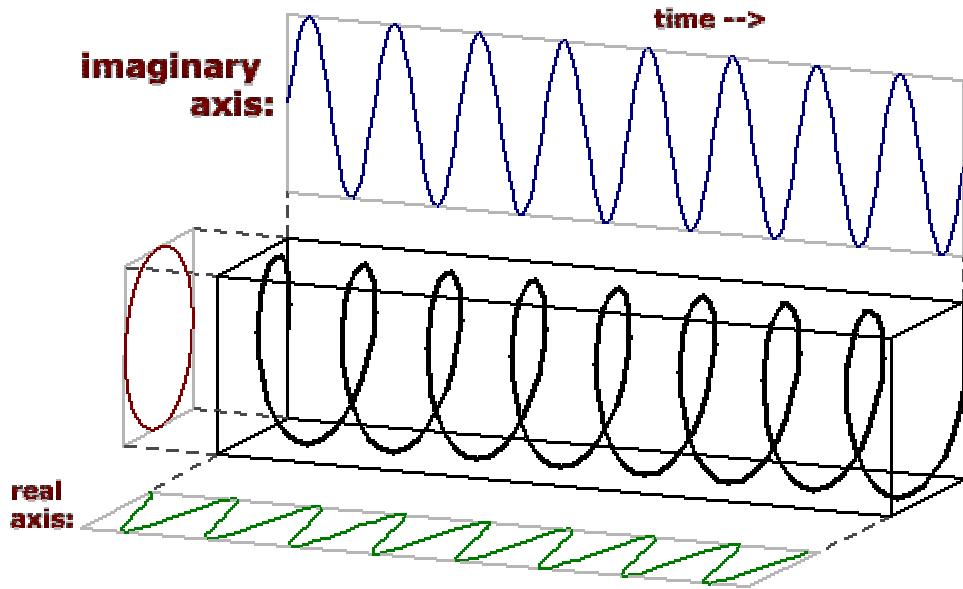


Fig. 10. The complex sinusoid shown as a spiral circular wave

The one dimensional Gaussian function is expressed as

$$\exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right) \quad (13)$$

Since the area under the curve is

$$\int_{-\infty}^{\infty} \exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right) dx \quad (14)$$

The normalized one -dimensional Gaussian function is

$$f(x) = \frac{1}{\sqrt{2\pi a^2}} \exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right) \quad (15)$$

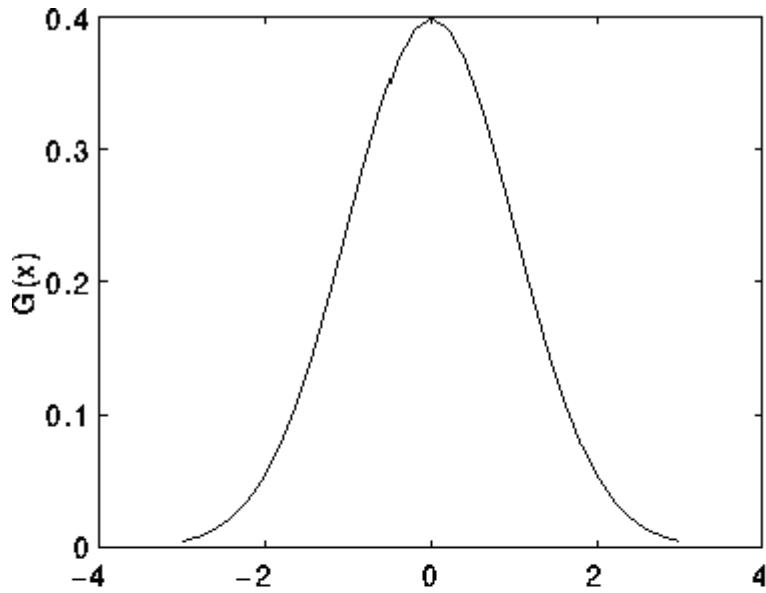
The peak value of the function is

$$f(x_0) = \left(\frac{1}{\sqrt{2\pi a^2}}\right) \quad (16)$$

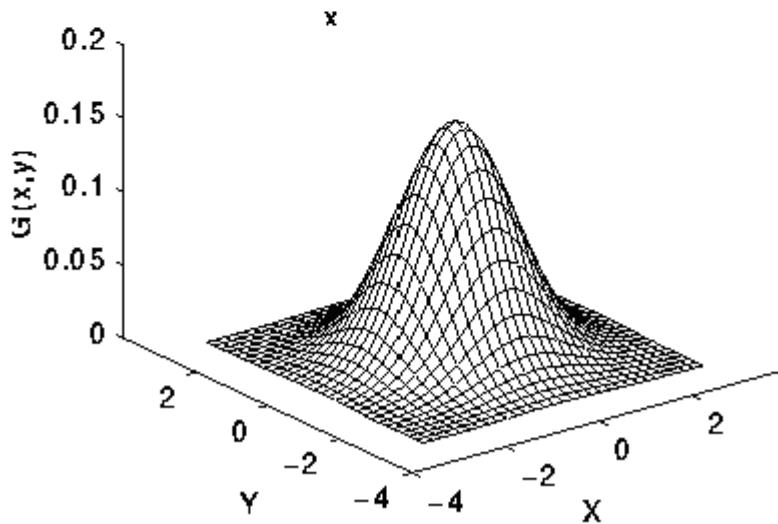
Using the same argument as in the one-dimensional case, the two-dimensional Gaussian function in cartesian coordinate is

$$f(x, y) = K \exp\left(-\pi(R_x(x-x_0)^2 + R_y(y-y_0)^2)\right) \quad (17)$$

x , the horizontal location in space
 y , the vertical location in space
 K , the peak amplitude of the Gaussian
 x_0 , the horizontal midpoint in space
 y_0 , the vertical midpoint in space
 R_x , the horizontal scaling parameter
 R_y , the vertical scaling parameter



A. ONE-DIMENSIONAL GAUSSIAN FUNCTION



B. TWO DIMENSIONAL GAUSSIAN FUNCTION

Fig. 11 One and two dimensional Gaussian function

4.8 The complex Gabor function: How it evolved

Before 1946, the Fourier system was the state-of-the art in signal analysis. The basis of the Fourier system is representation of arbitrary signal with trigonometric functions called Fourier series. The Fourier analysis is a powerful tool with applications in diverse areas including mathematics, engineering and physics. However it has limitation. It is ideally suited to the study of stationary signals and processes that are statistically invariant over time.

But there are many physical processes and signals that are non-stationary. Examples are speech and music. When sampled over a finite duration of time using the Fourier system the Fourier representation gives the prevailing notes in terms of the corresponding frequencies, but not information about the duration and emission of the notes. These information are masked in the phase of each component of the Fourier series. Thus we say that the Fourier transform does not provide frequency information local to a point in the signal. It only provides a global measure of all the frequencies present. It does not tell which bits of the signal give rise to bits of the frequency spectrum as illustrated in Fig.12.

As far back as 1946, Dennis Gabor, the 1971 Nobel prize winner in Holography was, like every other scientist, interested in the problem of obtaining simultaneous localization in both time/space and frequency domains. He was motivated by developments in quantum mechanics including Heisenberg's uncertainty principle, and the fundamental results of Nyquist and Hartley on the limits for the transmission of information over a channel.

He examined the two extreme cases of localization, the sine wave which is not localized in time space domain but extremely localized in frequency domain, and the delta function which is perfectly localized in time/space domain but with no localization in the frequency domain as shown in Fig.13.

He proposed to represent signals in time and frequency domain by the use of ‘elementary’ functions constructed from a single block or ‘mother’ signal by translation and modulation. By his proposal any signal $f(t)$ of finite duration T and bandwidth F can be divided into a finite number of elementary “information cells” called **logons**. Each logon is of duration Δt and bandwidth ΔF , and is localized at a different time and frequency in each of the domains.

If we let

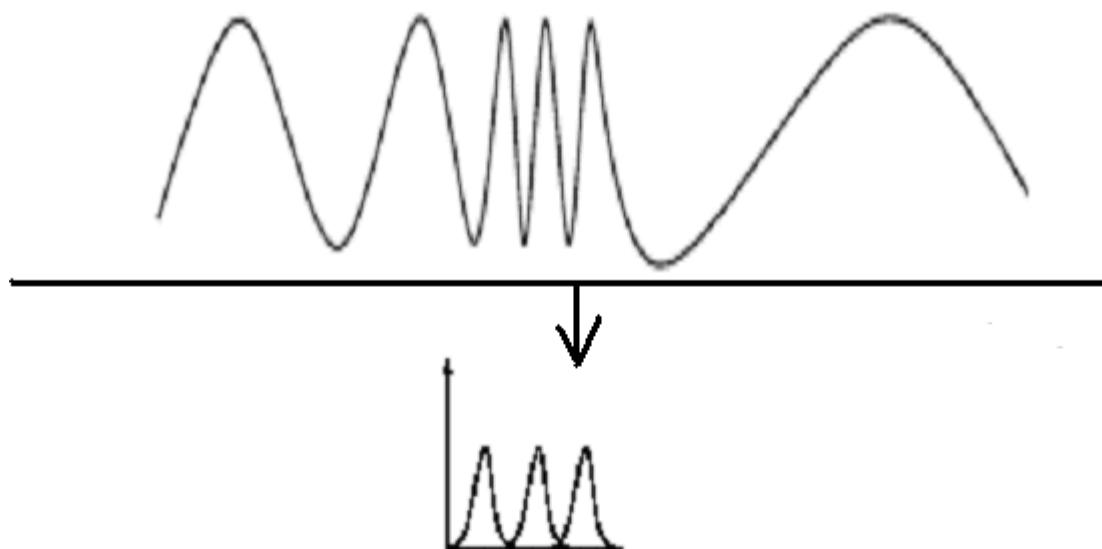
$$M = T / \Delta t \quad (18)$$

and

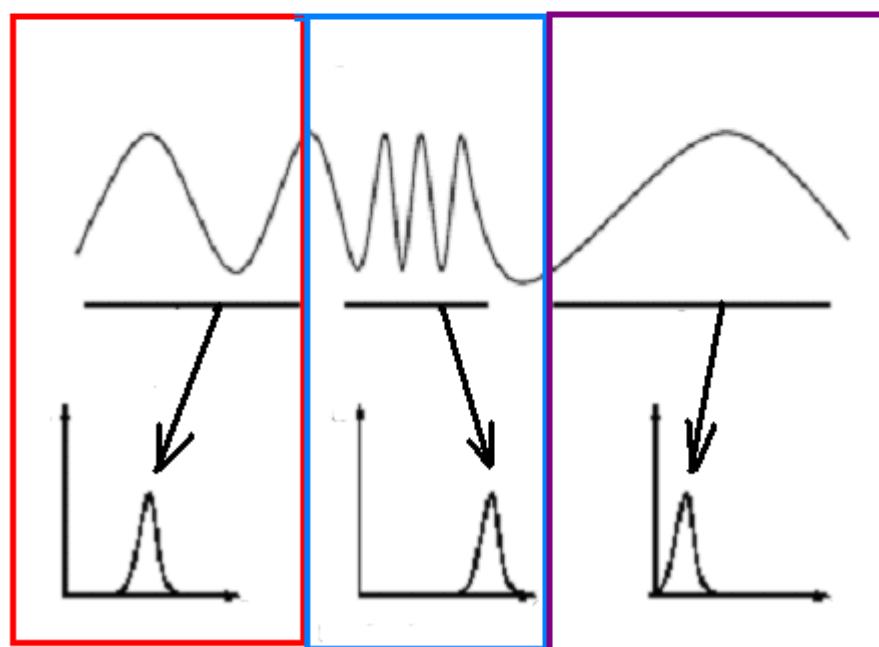
$$N = F / \Delta f \quad (19)$$

then there are MN elementary information cells; the signal represents MN logons of information, namely, the MN coefficients associated with the cells. This is the most information that can be represented by the signal, and these MN complex coefficients are sufficient to regenerate the signal [16].

The elementary signals he proposed is the Gaussian function and its translations using time shift parameter a , and modulations using frequency shift parameter b . Gabor proved that this elementary signal that is now referred to as the Gabor function occupies the minimal area in the time frequency plane so that there is the minimal amount of simultaneous information in time and frequency. Thus Gabor elementary function represents the minimal quantum information.



A. GLOBAL FREQUENCY SPECTRUM PROVIDED BY FOURIER SYSTEM



B. DESIRED LOCAL FREQUENCY SPECTRUM

Fig. 12 Global frequency spectrum obtained from Fourier spectrum and desired local frequency spectrum

Mathematically,

$$f(t) = \sum_{n,m \in Z} c_{m,n} g_{m,n}(t) \quad (20)$$

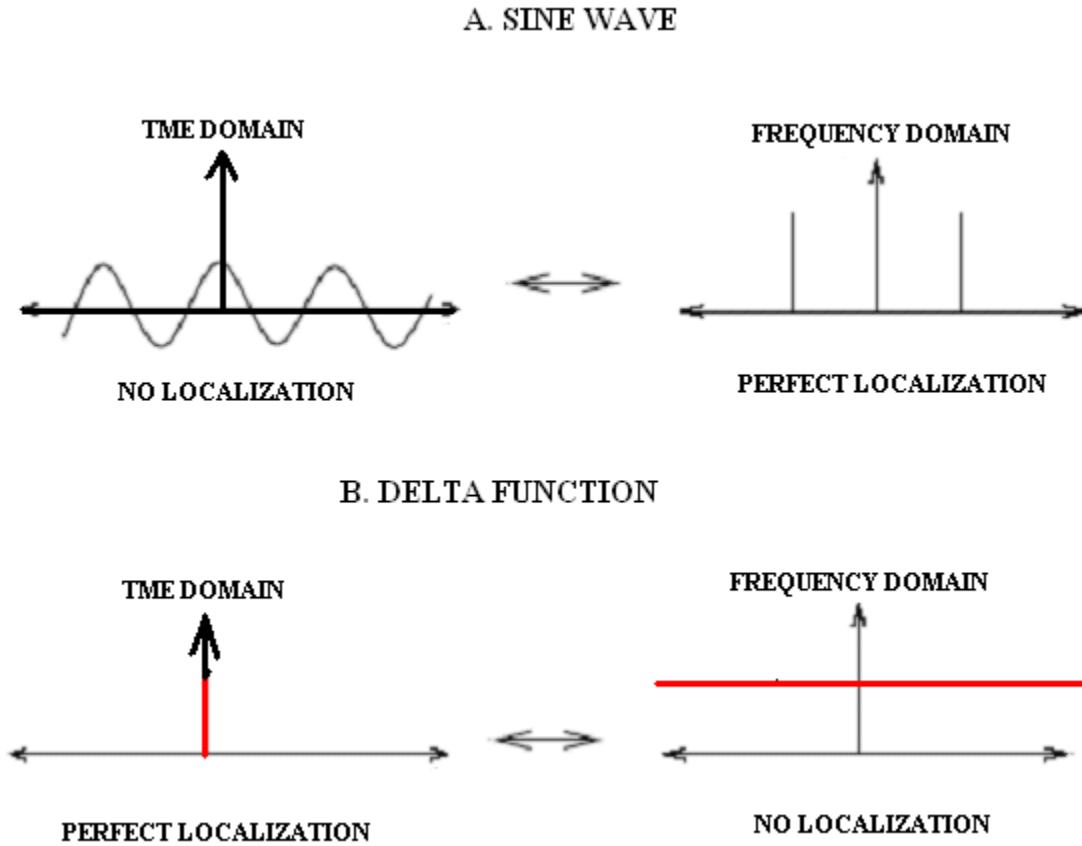


Fig. 13 The sine and delta functions as two extremes in frequency and space localization

the elementary functions, $g_{m,n}$ are given by

$$g_{m,n}(t) = g(t - na)e^{2\pi imb t} \quad (21)$$

a, b are the respective time and frequency shift parameters.

Generation of Gabor function from a Gaussian by translation and modulation is illustrated in Fig. 14.

4.9 The complex Gabor function

A Gabor function is a complex sinusoidal modulated by a 2-D Gaussian envelope. Gabor function can be visually appreciated if one look at Fig. 15 showing a one-dimensional Gabor function obtained by modulating a sinusoid with a Gaussian envelope. It is in the same manner that a 2-D Gabor complex function is obtained. However a 2-D Gabor function is anchored on a rectangular grid as shown in the two views of the complex Gabor function shown in Fig. 16. The rectangular grid is the equivalent of mask, kernel or filter size we are familiar with in image filters. It is important to always visualize the rectangular grid size on which the complex Gabor is sited for a better visualization and understanding of Gabor function operations.

Using the usual notations for the parameters, a 2-D Gabor function is expressed in the spatial domain as

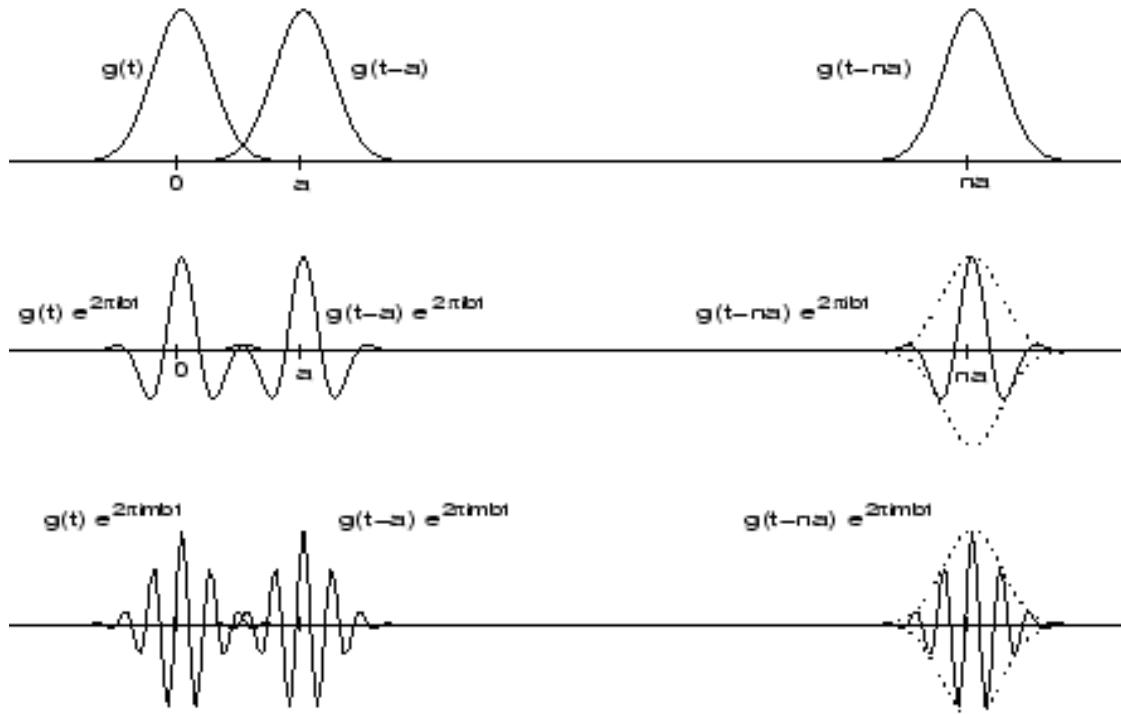


Fig.14 Generation of Gabor elementary function by translation and modulation

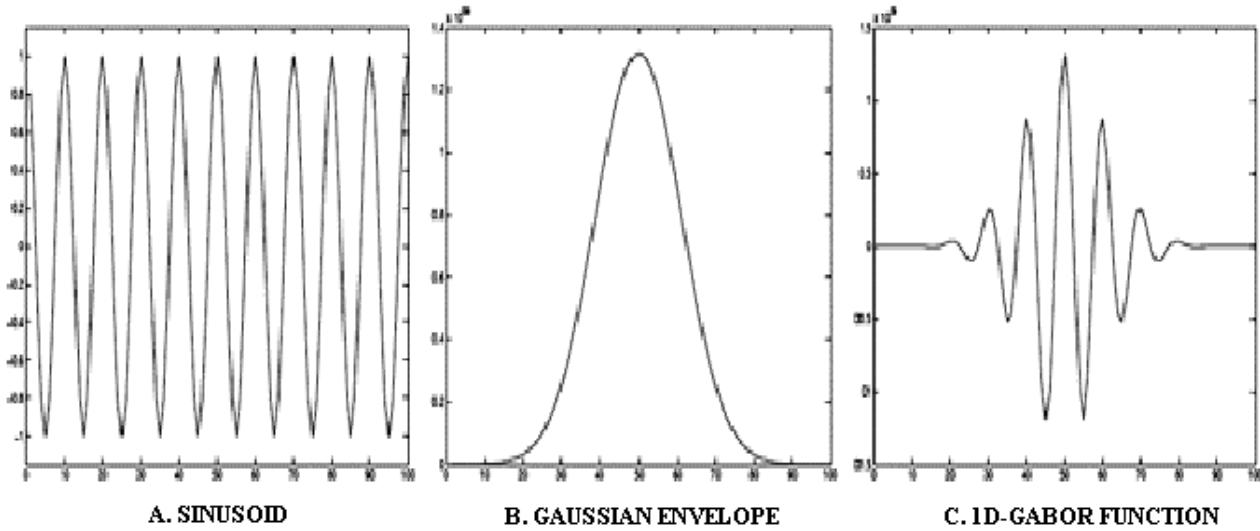


Fig. 15 1D-Gabor function and its constituent elements

$$g(x, y) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) \exp(j(2\pi(u_0x + v_0y) + P)) \quad (22)$$

The first parts of the equation represent the Gaussian envelope while the second part is the complex sinusoid. It can be rewritten as

$$g(x, y) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) \exp(j(2\pi(u_0x + v_0y) + P)) \quad (23)$$

By separating variables,

$$g(x, y) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) (\cos(2\pi(u_0x + v_0y) + P) + j \sin(2\pi(u_0x + v_0y) + P)) \quad (24)$$

Hence,

The real part of Gabor is

$$\operatorname{Re}(g(x, y)) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) \cos(2\pi(u_0x + v_0y) + P) \quad (25)$$

The imaginary part is

$$\operatorname{Im}(g(x, y)) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) \sin(2\pi(u_0x + v_0y) + P) \quad (26)$$

In the polar coordinates the equation is

$$g(x, y) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) \exp(j(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P)) \quad (27)$$

This can also be rewritten as

$$g(x, y) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) + j(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P) \quad (28)$$

Also by separating variables the respective real and imaginary parts of Gabor in polar coordinate are

$$\operatorname{Re}(g(x, y)) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) \cos(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P) \quad (29)$$

and

$$\operatorname{Im}(g(x, y)) = K \exp\left(-\pi(a^2(x - x_0)^2 + b^2(y - y_0)^2)\right) \sin(2\pi F_0(x \cos \omega_0 + y \sin \omega_0) + P) \quad (30)$$

The complex sinusoid gave the complex Gabor function its complex characteristics. Just as the complex sinusoid has its real and imaginary parts the complex Gabor function can also be decomposed into its **real** and **imaginary** parts as shown in Eqn 25 and Eqn 26. They are perpendicular to each other, hence they are said to be a **quadrature pair**. The real part of Gabor complex function is also called the **even** Gabor function because it is evenly symmetric, while the imaginary part is called the **odd** Gabor function because it is odd symmetric.

4.10 Fourier transform of Gabor function

The Fourier transform of the complex Gabor function was derived in [17]. It is expressed in the spatial domain as

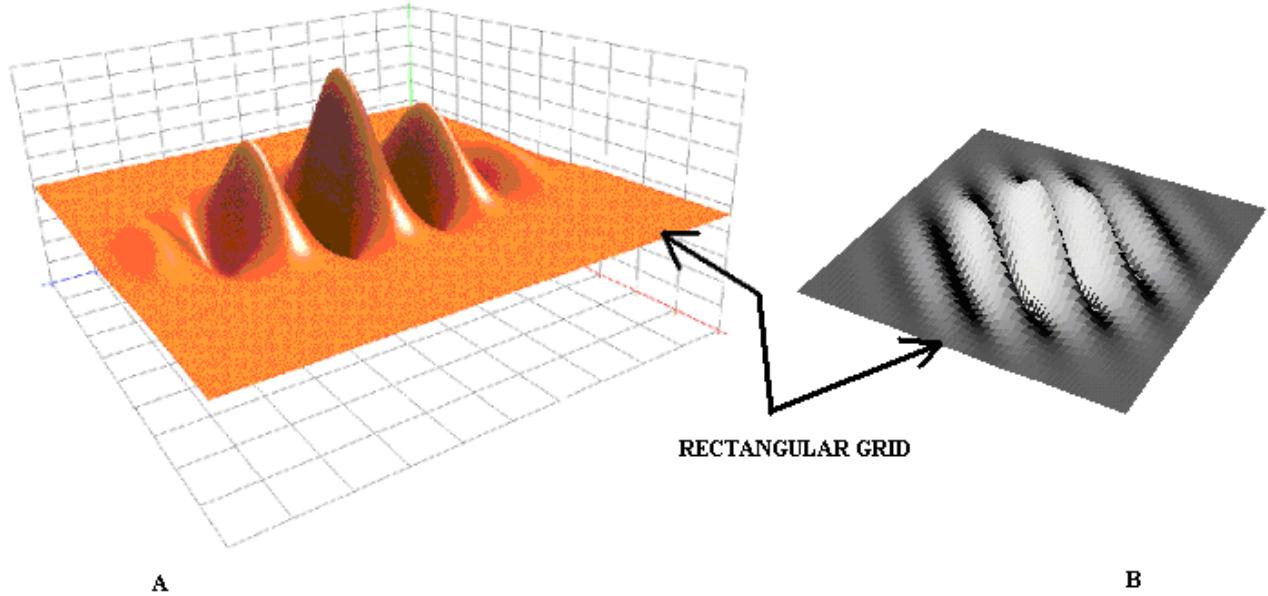


Fig. 16. Two views of the complex Gabor function

$$g(u, v) = \frac{K}{ab} \exp(j(-2\pi(x_0(u - u_0) + y_0(v - v_0)) + P)) \exp\left(-\pi\left(\frac{(u - u_0)^2}{a^2} + \frac{(v - v_0)^2}{b^2}\right)\right) \quad (31)$$

The frequency response of cartesian one and two dimensional complex Gabor function is shown in Fig. 17(A) and Fig. 17(B). Fig. 18 shows the polar frequency response

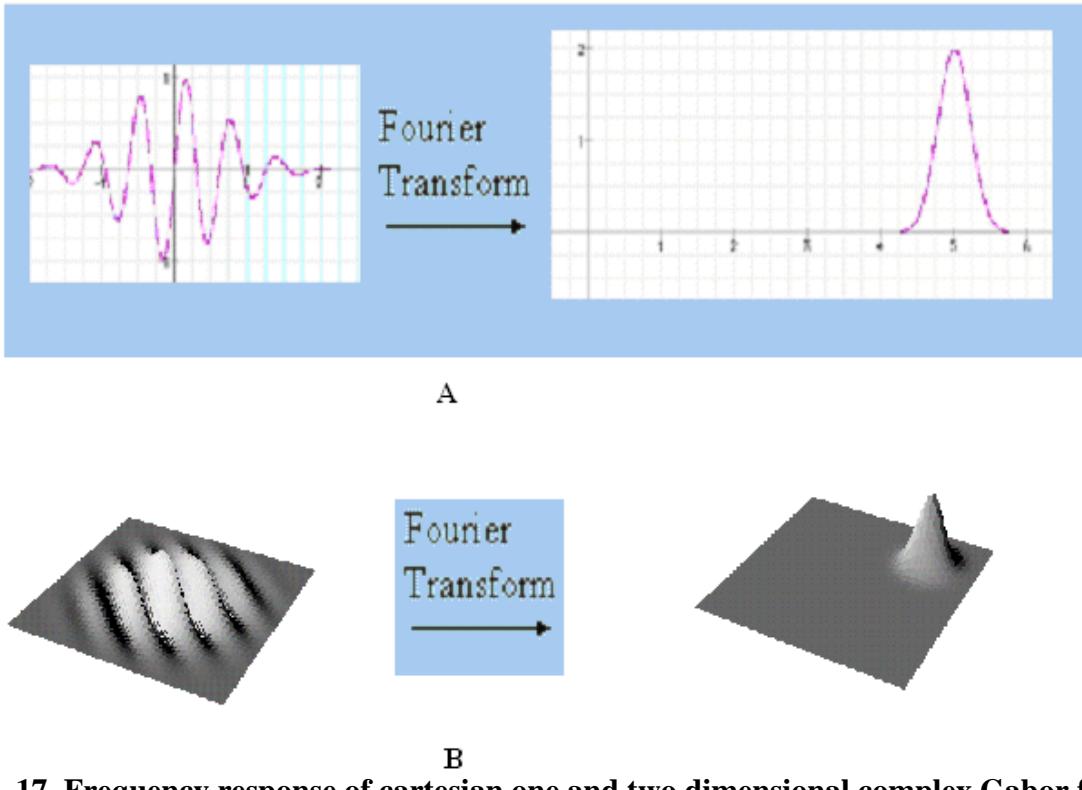
In polar coordinate, its magnitude is

$$F = \frac{K}{ab} \exp\left(-\pi\left(\frac{(u - u_0)^2}{a^2} + \frac{(v - v_0)^2}{b^2}\right)\right) \quad (32)$$

The phase or direction is

$$\theta = -2\pi(x_0(u - u_0) + y_0(v - v_0)) + P \quad (33)$$

The characteristic of the complex Gabor function in the frequency domain is measured in terms of a so called **half-magnitude response** or **half-peak response**. The half-peak response is the region of points in the frequency domain with magnitude equal to one-half the peak magnitude. This mode of measuring Gabor function characteristics should not sound strange. The bell shape of its frequency response as shown in Fig. 17 necessitated it.

**Fig. 17. Frequency response of cartesian one and two dimensional complex Gabor function**

From Eqtn. 32 the peak value of the magnitude of the frequency response is

$$F = \frac{K}{ab} \quad (34)$$

It is obtained by setting $(u, v) = (u_0, v_0)$. The set of points with half the peak magnitude is obtained by setting F to $K/2ab$ and solving Eqtn. 32, resulting in

$$\frac{1}{2} \frac{K}{ab} = \frac{K}{ab} \exp\left(-\pi\left(\frac{(u-u_0)^2}{a^2} + \frac{(v-v_0)^2}{b^2}\right)\right) \quad (35)$$

or

$$-\log 2 = -\pi\left(\frac{(u-u_0)^2}{a^2} + \frac{(v-v_0)^2}{b^2}\right) \quad (36)$$

Solving gives

$$\left(\frac{(u-u_0)}{aC}\right)^2 + \left(\frac{(v-v_0)}{bC}\right)^2 = 1 \quad (37)$$

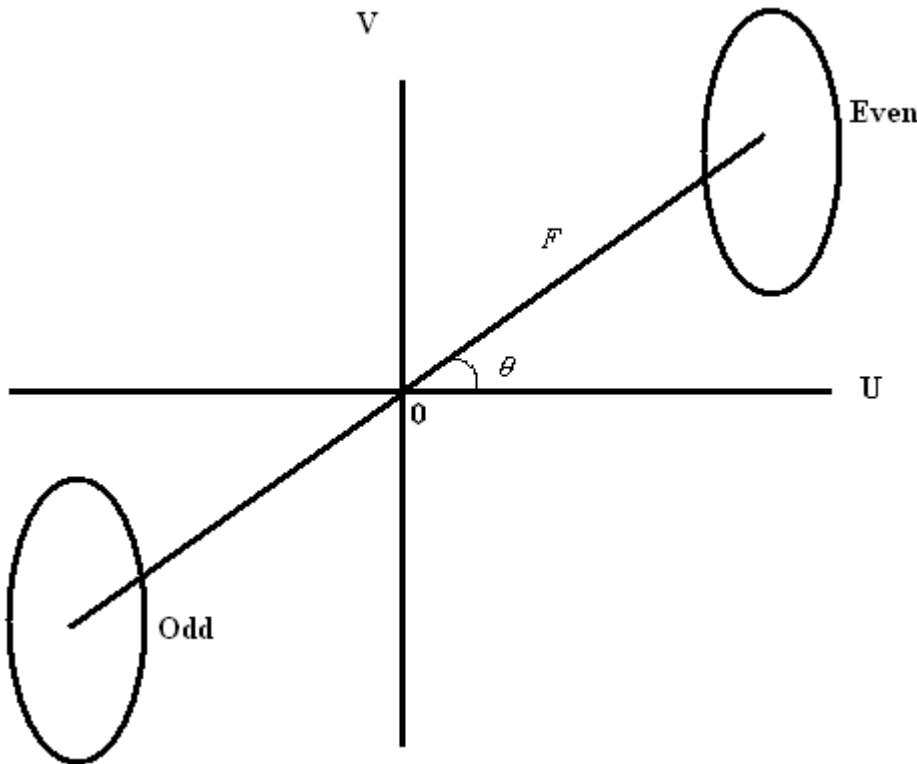


Fig. 18. Frequency response of polar two dimensional complex Gabor function

$$C = \sqrt{\frac{\log 2}{\pi}} \approx 0.5 \quad (38)$$

Eqtn. 37 shows that the half-peak frequency response of the complex Gabor function is an ellipse centred at the frequency of the complex sinusoid, with major axis of length $2aC$ and minor axis of length $2bC$.

Another term in the frequency domain is the **half-magnitude frequency bandwidth** measured in octaves. It is defined as the base two logarithm of the ratio of the upper and lower frequency region defining the half-magnitude profile of the complex Gabor function,

$$\Delta F_{1/2} = \log_2 \left(\frac{F_{MAX}}{F_{MIN}} \right) \quad (39)$$

From the elliptical geometry of the Gabor Fourier transform shown in Fig. 19 the half-magnitude bandwidth is approximately equal to the length of the major axis. In the same vein the orientation bandwidth, $\omega_{1/2}$ is approximately equal to

$$\Delta \omega_{1/2} = 2 \tan^{-1} \left(\frac{bC}{F_0} \right) \quad (40)$$

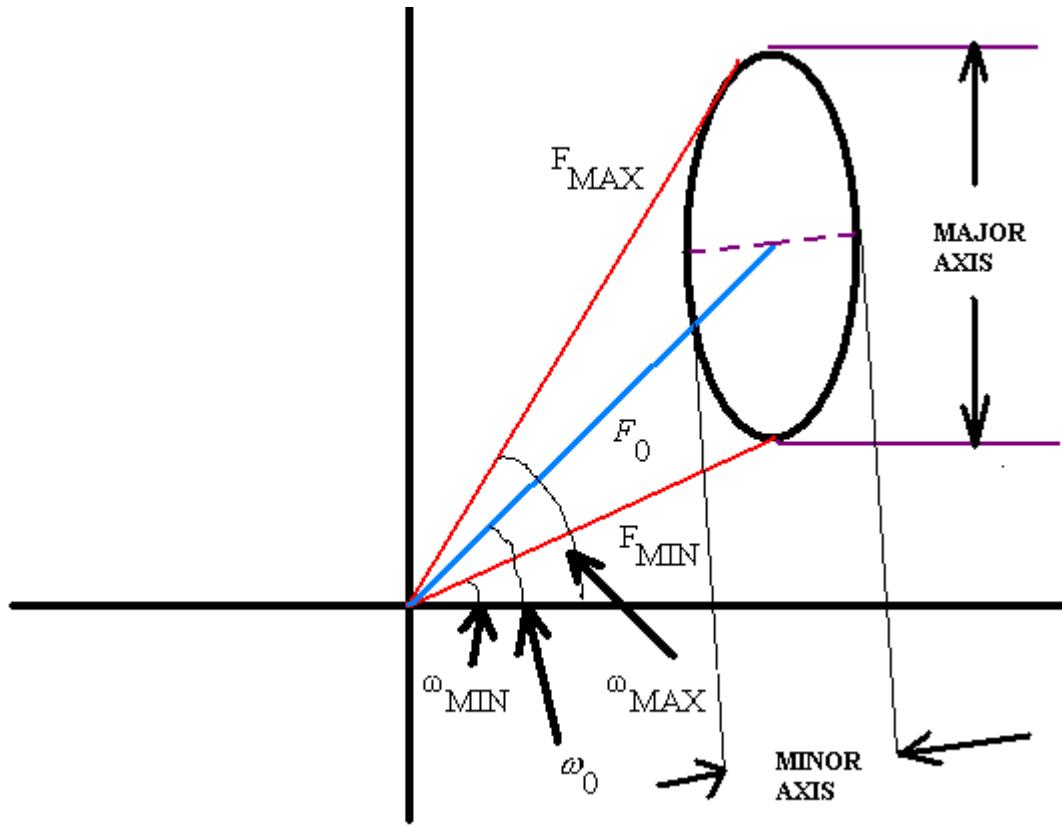


Fig. 19 Parameters for computing half-magnitude bandwidth and orientation bandwidth of the Gabor function

4.11 Generation of Gabor wavelets

Wavelets are families of basis functions generated by dilations (scaling) and translations of a basic wavelet called the mother wavelet. The basis functions are themselves basic functional building block of any wavelet family.

Before extending this notion to the Gabor family it is necessary to note that the complex Gabor function so far discussed is a function that has a **fixed frequency magnitude** F_0 as determined by the complex sinusoid, with a **fixed orientation** ω_0 . It is also characterized by half-peak magnitude frequency bandwidth $\Delta F_{1/2}$, and half-magnitude orientation bandwidth $\Delta\omega_{1/2}$

For convenience regard the complex Gabor function discussed so far as the ‘mother’ Gabor or basic wavelet. Scaling in the frequency domain and rotation in the space-time domain of this basic wavelet generate Gabor wavelets.

Rotation in the space-time domain is achieved by applying coordinate transformation to the two dimensional spatial-time domain using the rotated coordinate system,

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (41)$$

The orientation range or orientation bandwidth is

$$0 \leq \theta \leq \pi \quad (42)$$

For practical purposes the orientation bandwidth is discretized into N intervals, that is,

$$\theta_k = \frac{k\pi}{N} \quad (43)$$

$$k = 1, 2, \dots, N$$

In accordance with Gabor proposal, for every rotation, or orientation there is associated frequency domain set of values. This is obtained by scaling the wavelet. But scaling the wavelet changes the frequency that it is tuned to, that is its center frequency, which in turn changes the bandwidth that it responds to. For example where the scaling factor α is 1/2, each successive wavelet is tuned to twice the frequency of the previous one, and has twice the bandwidth. It is for this reason that for practical consideration the scaling parameter in the frequency domain is exponential (logarithmic). The reason is that on a logarithmic frequency scale the transfer function of each wavelet is identical. Thus the logarithmic scale factor (a power of 2) introduced in the design of Gabor wavelet ensures that energy of each wavelet is independent of scaling the mother function.

Scaling is addressed through the standard deviation of the modulating Gaussian and the frequency of the complex sinusoid.

Having defined the dilation (scaling) and orientation parameters, the Gabor wavelet family is

$$\left(g\left(\alpha^j(x - x_0, y - y_0), \theta_k\right) \right), \alpha \in \mathfrak{R}, j = \{0, -1, -2, \dots\} \quad (44)$$

For a ‘mother’ Gabor g scaled at m scales of frequencies and n orientations a so-called self-similar filter dictionary can be obtained through the generating function

$$g_{mn}(x, y) = \alpha^{-m} g(x', y') \quad (45)$$

$$\alpha > 1, \quad m, n \text{ are integers}$$

In the domain of functional analysis, the generated Gabor wavelets do not form an orthogonal basis set. This implies that for a given family of Gabor wavelets it is not possible to calculate a weight by simple projection of the Gabor wavelet onto the image. Since the weights or coefficients needed to expand the signal are difficult to determine it implies that there is redundant information in the filtered images. This is a disadvantage of using Gabor wavelet in signal analysis. Nevertheless Gabor wavelets can form a complete representation of the image (signal).

4.12 Bandwidth Characteristics of the Gabor wavelets

The center frequency of each generated wavelet is directly related to the frequency of the modulating complex sinusoid. The bandwidth itself is related to the standard deviation of the Gaussian envelope and is independent

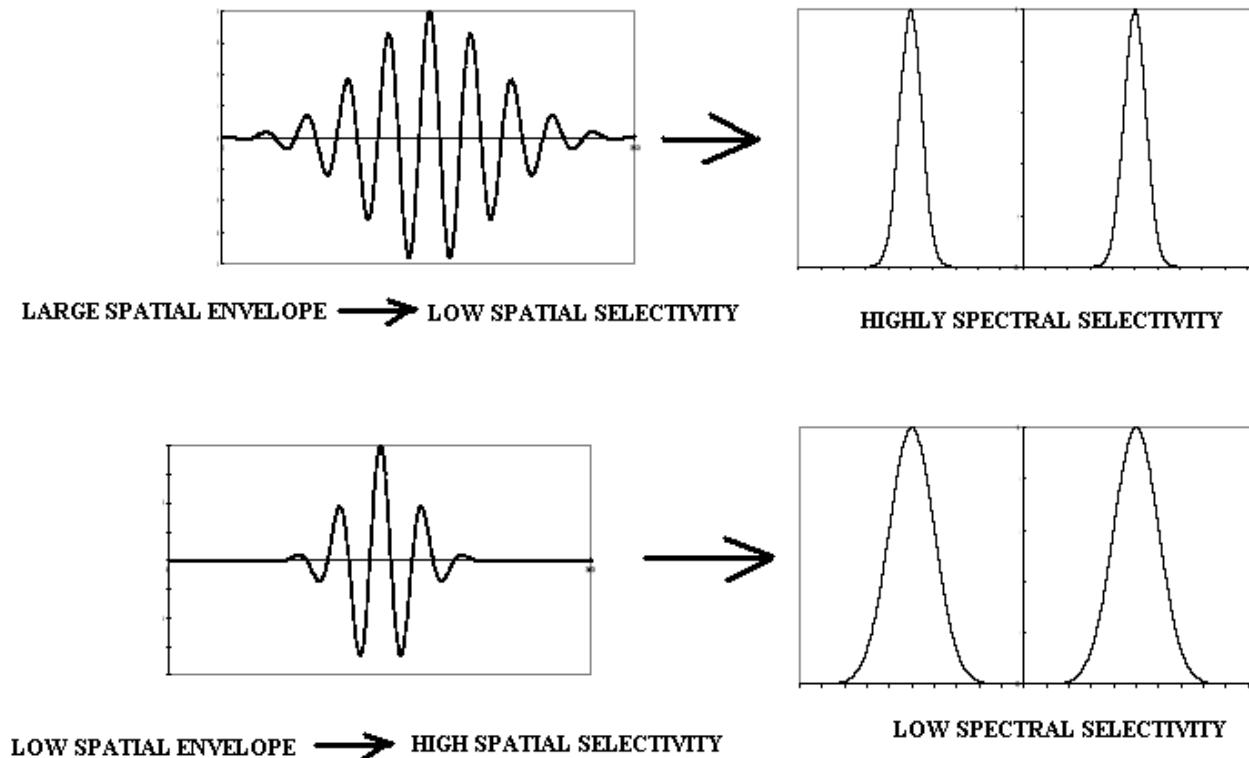


Fig. 20 Illustration of trade-off between spatial selectivity and frequency selectivity

of the frequency of the modulating function. This implies that a filter with a large spatial variance will have high spectral selectivity or highly localized, while filter with lower spatial variance will have low spectral selectivity or less localization. Thus a larger Gaussian envelope will increase the spectral resolution at the cost of decreased spatial resolution. This trade off between spatial localization and spectral selectivity is in accordance with Heisenberg uncertainty principle and is encountered in the use of Gabor filter for texture analysis. It is illustrated in Fig. 20.

4.13 Biological importance of Gabor wavelets

In 1980 Marcelja [18] studied the receptive field profiles of the visual cortex and observed spatial frequency tuning characteristics of simple cortical cells of the visual system; representation of an image in the visual cortex involve both spatial and spatial frequency variables. Receptive fields and spatial frequency tuning curves of the visual cortex conformed closely to the functional form of Gabor elementary signals.

Marcelja's claim was supported by Daugman [19] and Jones [20]. Daugman proposed two-dimensional Gabor functions to model the spatial summation properties of the receptive fields of simple cells in the visual cortex. Motivated by this biological findings Daugman, Moshe Porat and Yehoshua Y. Zeevi [21] proposed the use of Gabor functions for image processing applications such as image analysis and image compression. Since then Gabor functions have shown to be useful for different applications such as texture analysis and medical signal processing.

4.14 Application of Gabor wavelets to image processing: scale-space defined

According to the web dictionary, scale is the size or apparent size of an object seen in relation to other objects, people, or its environment or format. The same dictionary says that scale is either a weighing scale used for measurement of weight (mass or force), or a series of ratios against which different measurements can be compared. The latter need not always be a linear ratio, and is often logarithmic.

The dictionary gave another version of space definition as a set of dimensions in which objects are separated and located, have size and shape, and through which they can move.

To apply this definition to image processing we can view scale in two dimensions. The first dimension is the spatial scale which has to do with relative lengths, areas, distances and sizes, and uses the metre as basis of measurement. The other dimension is the frequency scale that uses logarithmic scale as basis of measurement.

4.15 Application of Gabor wavelets to image processing: multiscale representations introduced

A look around a living room, the equivalent of view in the spatial domain shows different images such as chairs, books, wall clock, wall photographs, lightings, television, computer and communication equipments. All these objects that constitute the living room have different sizes or scales. In the universal space, the living room exist in the range of say 25 square metres, the books exist in the range, say 100 square centimetres, the television and computer equipments exist within 2 – 5 square metres. The handy mobile phone exist within 10 to 30 square centimetres. If it were possible to view the living room from afar say at a distance of 25 metres, only the living room itself will be clearly visible to the naked eyes. Other objects will be blurred. If we move closer to the living room the photograph first appears blurred but as we enter the living room the object(s) of the photograph becomes clearly visible. A very close look at the photograph will not only reveal the objects in clear detail but also presence of texture if it exist.

The aforementioned example is a demonstration that real-world objects exist as meaningful entities over certain ranges of scale, and human perceive them at coarse or fine scales depending on the scale of observation.

It is based on these real-world examples that the features in an image are said to be generally present at various scales or range of scales. This implies that the features can best be observed over this range, and this makes multiscale decomposition of the image an efficient way to analyse and interpret images.

To be able to extract information from image data, a probe, sensor or operator is required to interact with the actual image structure. The information extracted is dependent on the size of the image structures and the size of the operators (probes). The probe is equivalent to the human view of the image while the size of the operator determines the resolution at which the image is observed.

In practice, as in a large image database there is no prior information on the size or scale of individual structures within the images to be analysed. Obviously this makes analysis and interpretation of the image difficult. In image processing and computer vision community this problem is dealt with by representing the image data at multiple scales.

Since quality of information we can get from probing any image is determined by the relationship between size of structures in the image and the size of the probe, it is therefore logical to reason that a better analysis and interpretation of the image data can be obtained by using probes of various sizes.

4.16 Application of Gabor wavelets to image processing: multiscale representations defined

Multiscale techniques or multiscale resolution such as Gabor wavelet representations was developed by scientists in the image processing and computer vision community to provide a way to isolate, analyze and interpret structures (objects or features) of different scales within an image [22].

Multi-scale representation of a signal is an ordered set of derived signals intended to represent the original signal at different levels of scale. Example is Gabor decomposition of an image shown in Fig. 21 The different scales are obtained in the frequency and spatial domains and represented respectively as \mathbf{M} and \mathbf{N} .

The main idea of multiscale representation is to generate a one-parameter family of derived images $I(x, y, s)$ by convolving the original image $I(x, y)$ with a Gabor kernel $G(x, y, s)$ in a filter bank,

i.e

$$I(x, y, s) = I(x, y) * G(x, y, s) \quad (46)$$

$s = [M, N]$ is the length scale factor of the Gabor filter in the filter bank.

The effect of convolving a signal by a Gabor kernel is to suppress most of the structures in the signal with a characteristic length less than the scale factor, s . This is quite obvious in Fig. 21 particularly for the derived images with length scale factor $[M= 1, N=1]$ and $[M=2, N=1]$.

In multiscale representation process, a crucial requirement is that structures at coarse scales in the multi-scale representation should constitute simplifications of corresponding structures at finer scales as shown in Fig.22.

4.17 Application of Gabor wavelets to image processing: summary

The current state-of-the-art in texture feature extraction and segmentation use Gabor wavelets. This was fuelled by physiological research evidence that Gabor filters model the neurons in the visual cortex of the human visual system. The neurons are characterized by a number of independent narrow band filters tuned to different space-time and spatial frequencies.

The most interesting feature of Gabor functions is that they permit a joint sampling of the space and frequency domains, with a maximum joint localization in both domains; i.e. they permit a simultaneous local analysis of both domains.

In image processing and in particular multi-resolution analysis, Gabor function is used as basis function to decompose a signal by forming inner products between the signal and the basis functions. The process of taking the inner product with the Gabor function is repeated with the function successively indexed across the signal (image). This corresponds to convolving the signal with the function. At each position in the signal the local amplitude and phase corresponding to the frequency that the filter is tuned to is extracted.

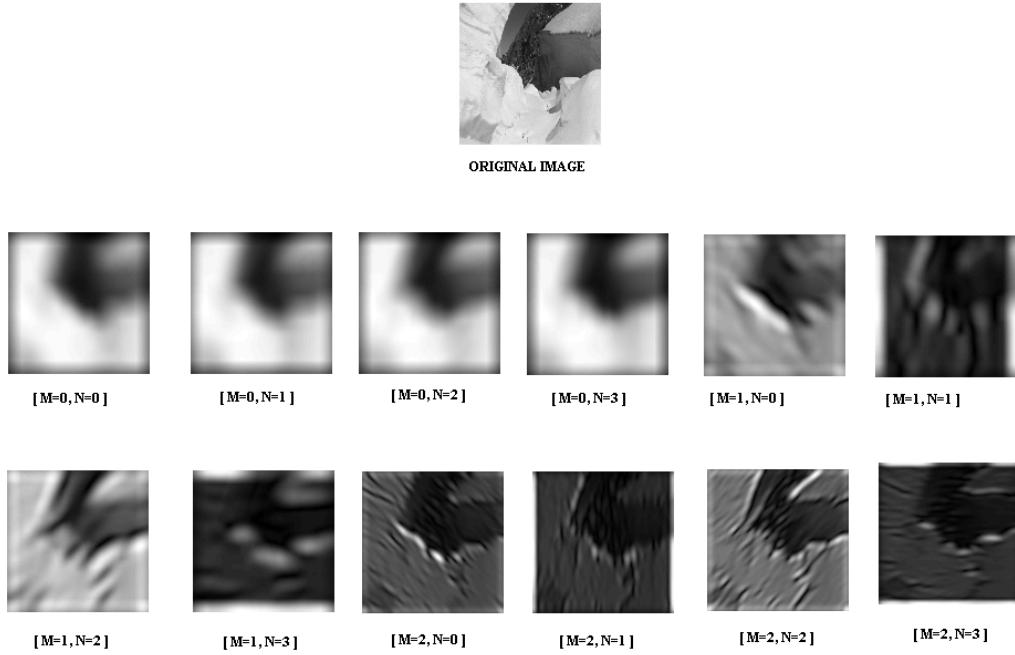


Fig. 21 Multiscale representation of an image using Gabor grid size of 15 x 15

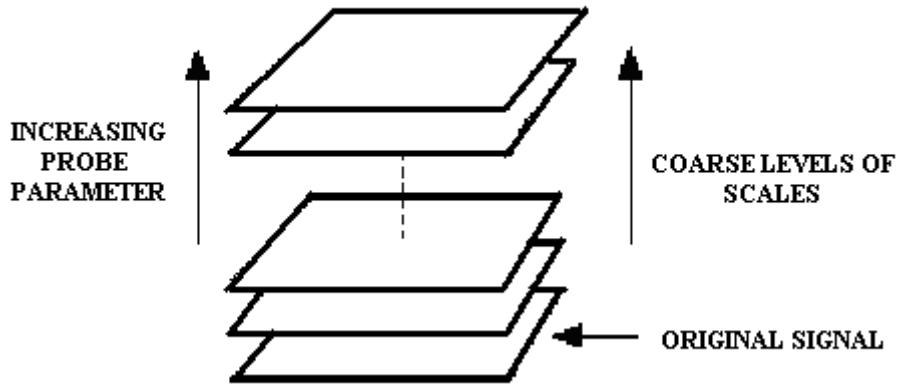


Fig. 22 Demonstration of multiscale representation

Since features of interest in an image is present at various scales and localized, Gabor wavelets possess the properties required for segmentation of images based on local spatial variations of intensity or colour. This is because a particular Gabor filter reacts strongly to specific textures and weakly to all others. Each generated wavelets are called channel in a filter bank. Thus Gabor filters are band-pass filters with tunable center frequency, orientation and bandwidth, and this provides Gabor filters the distinct advantage to optimally achieve joint resolution in space-time and spatial frequency. By arranging a set of filters in a filter bank, thereby forming Gabor wavelets, the output of each of the filters provide information about the spatial location of textures within the image. When the texture of a test image exhibits the frequency and orientation characteristics to which a Gabor filter is tuned the magnitude of the filter output is large. Otherwise if the test image local texture is not dominated by similar characteristics as the Gabor filters the magnitude of the filter output is low.

5

SHAPE

5.1 Shape defined

Shape of an object is the characteristic surface configuration as represented by the outline or contour. Shape recognition is one of the modes through which human perception of the environment is executed. It is important in CBIR because it corresponds to region of interests in images.

In image processing, shape is the binary image consisting of contour or outline of objects, obtained after segmentation.

In CBIR system designed for specific domain such as trademarks and silhouettes of tools, shape segmentation can be automatic and effective. However this is not the case for CBIR system having heterogeneous database. In this case shape segmentation may be difficult or sometimes impossible.

Shape feature representations are categorized according to the techniques used. They are boundary-based and region-based. Boundary-based technique describes the shape region by using its external characteristics, for example pixel along the object boundary, while the region-based technique describes the shape region by using its internal characteristics, for example the pixel contained in the region.

Simple boundary-based shape descriptors include area, perimeter, compactness, eccentricity, elongation, and orientation. Complex boundary-based descriptors include Fourier descriptors, grid descriptors, chain codes and statistical moments [23].

5.2 Area

Area is the number of pixels in the region described by the shape. The real area of each pixel may be taken into consideration to get the real size of a region. In Fig. 23 each pixel has area of one square unit. The total area represented by the shape is 28 square units because the total pixel inside the shape region is 28.

5.3 Perimeter

Perimeter is the number of pixels in the boundary of the shape. In Fig. 24 the total number of pixels on the boundary of the shape is 32

5.4 Compactness

Compactness is a measure of how closely packed is the shape. The most compact shape is a circle while all other shapes have compactness larger than that of a circle (4π). Example of compact and non-compact shape is shown in Fig.25.

$$\text{Compactness} = \frac{(\text{region_border_length})^2}{\text{area}} \quad (47)$$

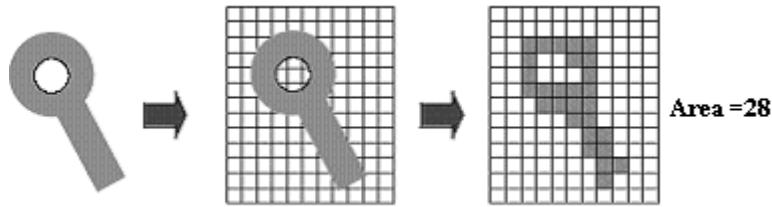


Fig. 23 Demonstration of Area as shape descriptor

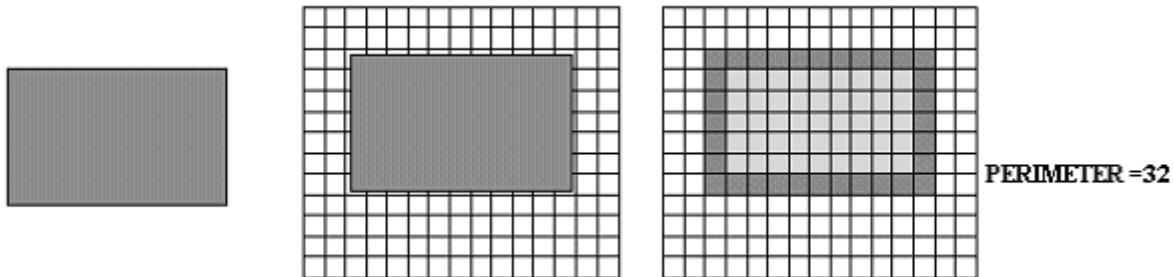


Fig 24. Demonstration of perimeter as shape descriptor

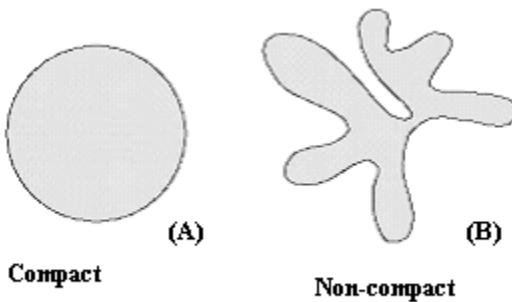


Fig. 25 Example of compact and non-compact shape

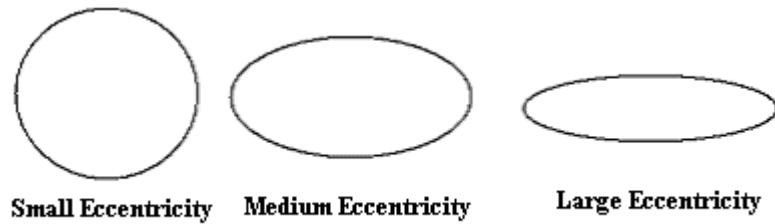
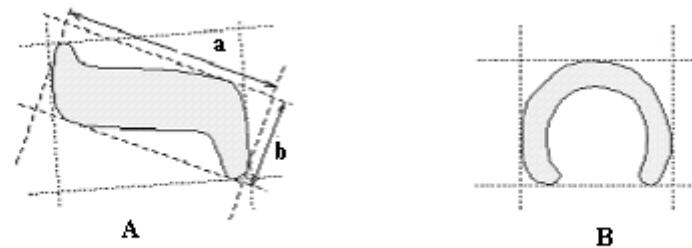


Fig. 26 Objects with different eccentricities



Elongatedness is defined for this shape **Elongatedness is not defined for this shape**

Fig. 27 Demonstration of Elongatedness

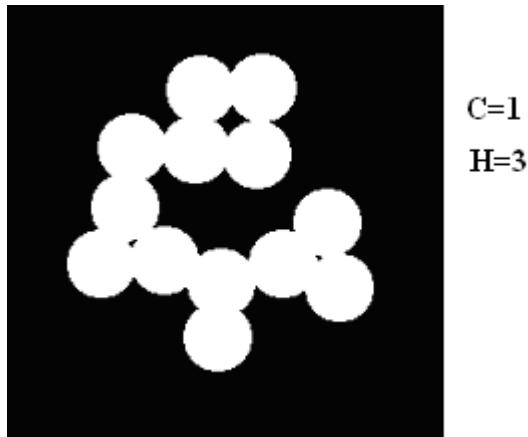


Fig. 28 A binary image having Euler number of -2

5.3 Eccentricity

Eccentricity is the ratio of the longest chord of a shaped object to longest chord perpendicular to it. Eccentricity is a measure of how circular a shape is. For a perfectly circular shape the eccentricity is zero. Elliptical orbits have eccentricities between zero and one. Objects with different eccentricities are shown in Fig. 26.

5.4 Elongation

Elongation is the ratio of the height and width of a rotated minimal bounding box. Bounding box is the smallest rectangle containing the shape.

In Fig. 27(A) elongatedness is,

$$\text{Elongatedness} = \frac{b}{a} \quad (48)$$

The definition of elongatedness does not hold for curved regions as shown in Fig.27 (B). In that case the definition of elongatedness must be based on maximum region thickness; ratio of the region area and the square of its thickness,

$$\text{Elongatedness} = \frac{\text{Area}}{(2d)^2} \quad (49)$$

d = thickness

5.5 Euler number

Euler number or connectivity factor E is the difference between the number of connected components C and the number of holes H in the image. It is a topological property of the shape that describes the structure of a shape regardless of its geometric structure.

$$E = C - H \quad (50)$$

The binary image in Fig.28 has only one connected component and three holes. Therefore its Euler number is -2 .

5.6 Moment of inertia

Moment of inertia or **second-order moments** are shape descriptor that measures the distribution of mass relative to axes through the center of gravity. It is based on the physics concept of moment of inertia. In theory distinct shapes have distinct moments.

The ij th discrete central moment m_{ij} of a region is defined by

$$m_{ij} = \sum \left(x - \bar{x} \right)^i \left(y - \bar{y} \right)^j \quad (51)$$

where the sums are taken over all points (x, y) contained within the region. The centre of gravity of the region are

$$\bar{x} = \frac{1}{n} \sum_x x \quad (52)$$

and

$$\bar{y} = \frac{1}{n} \sum_y y \quad (53)$$

n , the total number of points contained in the region, is a measure of its area

From the moment a lot of useful information about the object can be obtained. For example a binary image

$$I(x, y) = \begin{cases} 1 & \text{for foreground} \\ 0 & \text{for background} \end{cases} \quad (54)$$

- The area, A is given by the zeroth moment,

$$A = \sum_x \sum_y I(x, y) \quad (55)$$

- The center of mass is the first moment,

$$\bar{x} = \frac{\sum_x \sum_y x I(x, y)}{\sum_x \sum_y I(x, y)}, \quad \bar{y} = \frac{\sum_x \sum_y y I(x, y)}{\sum_x \sum_y I(x, y)} \quad (56)$$

- The orientation of the object is the axis of minimum inertia.

Seven moments are derived from the normalized second and third central moments. They are labelled $\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7$, and are invariant to changes of position, scale and orientation [24]. The values of each of seven invariant moments provide information about the shape of the object [25]. ϕ_1, ϕ_2 are always positive. Higher ϕ_2 values means that a shape is wider than it is tall, whereas ϕ_2 will be higher if the shape is taller. ϕ_4, ϕ_5 are measures of covariance. What this means is that shapes that are strongly diagonal, or skewed, will give higher values. ϕ_6, ϕ_7 are measures of asymmetry. If ϕ_6 is positive, it means that the shape is bulkier to the left and more outstretched to the right of the centroid. If ϕ_7 is negative the shape is more outstretched upwards and downwards if it is positive.

5.7 Fourier descriptors

Fourier descriptors are used to represent the shape of an image object that is in the form of a closed curve as shown in Fig.29.

To obtain the Fourier descriptors, a reference point on the boundary, \mathbf{O} is chosen. The distance of points from the reference point, \mathbf{O} on the closed curve is computed and normalized to 2π . Then angular variation between the tangent at the reference point \mathbf{O} and any point on the curve \mathbf{S} is given by

$$\begin{aligned} \phi(t) &= \theta(t) - t \\ t &= \frac{2\pi s}{L} \end{aligned} \quad (57)$$

Since $\phi(t)$ is real, continuous and periodic with a period 2π , it can be described by a Fourier series given by

$$\phi(t) = \sum_{k=0}^{\infty} a_k \exp(-jkt) \quad (58)$$



Fig 29. Parameters for computing Fourier descriptors of a shape boundary

BASIC 4-CONNECTIVITY CHAIN CODE 4 CONNECTIVITY CODED SUBSAMPLED IMAGE



Fig. 30 Example of chain code implementation

The set of coefficients a_k is called Fourier descriptors. Its distinctive property is they stay constant under translation, rotation, scale and origin transformation.

With these characteristics one can compare two shapes by comparing subsets of the Fourier descriptors, beginning with the lower order coefficients, and then using higher order coefficients.

Fourier descriptors have the advantage that it can be able to capture some human perceptual characteristics of shape because coarse or general shape features are captured by lower order coefficients of the Fourier transform and finer features are captured by higher order coefficients.

5.8 Chain codes

Chain codes are obtained by first resampling shape boundary on a grid, and using 4- or 8- connected sequence of straight-line segments of specified length and direction to trace the boundary. The chain codes are obtained by using a numbering scheme. An example is shown in Fig. 30

Chain codes describe objects boundaries by a sequence of unit-size line segments with a given orientation. The first element of the sequence is indexed with information about its position to permit the region to be reconstructed. In similarity matching chain codes must be normalized so that it is independent of the choice of the first border pixel in the sequence.

6

SPATIAL INFORMATION

6.1 Spatial information defined

By definition, spatial information is any information that can be geographically referenced by describing a location, or any information that can be linked to a location. An example is reference to a point in a two or three dimensional space.

6.2 Spatial information explained

Spatial information is the spatial relationship existing among properties characterizing image regions within the global image, and is commonly used to addresses the problem of discriminating similar images in homogeneous or non-diverse image database. Centroid and area of local image regions are local information features, and also the basis for deriving spatial layout or information. An example is shown in Fig.31. The two images have same size and same quantity of pixel colours. The two images will be seen to have same colour feature using colour histogram. However this is not the case if spatial information of the colours and shape of image regions is considered. In this case there is zero match because there is no overlap in colour or shape when there is perfect placement of one image on the other.

6.3 Spatial information representations

The use of symbolic images in CBIR system based on spatial information has been proposed [26], [27]. A symbolic image is a logical representation of the original image where each image objects or regions are uniquely labelled with symbolic names. The first step in deriving symbolic names for spatial information representation is to identify the local objects or regions in the image and their relative positions within the image. Thereafter a symbolic image is obtained by associating a name with each of the objects identified. Centroid coordinates of the image objects with reference to the image frame are also extracted. Most popular in these area are the use of 2-D strings [26], spatial-orientation graph [27], radon transform [28] and spatial quad-tree [29].

6.2 2-D Strings

2-D strings represent the spatial relationships among objects in an image by representing the projection of the objects along the x and y axes.

6.3 2-D String implementation

2-D strings representation of an image, for example, a human face shown in Fig. 32, is derived as follows.

- The image is first processed to obtain segmented objects followed by determination of the centroids of objects within the global image.
- Reference point of the segmented objects are the projection of the objects centroids on the x and y axes.
- Each image region is lexicographically labelled and indexed.
- Let the objects in the image (left eye, right eye, nose and mouth) be represented by a set of symbols

$$V = \{LE, RE, N, M\} \quad (59)$$

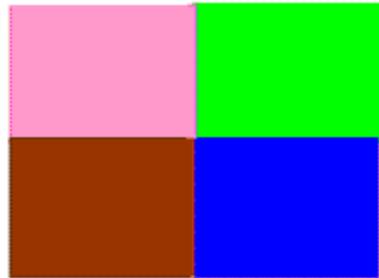


IMAGE 1

PINK	GREEN
BROWN	BLUE

SHAPE OF COLOR REGIONS IN IMAGE 1

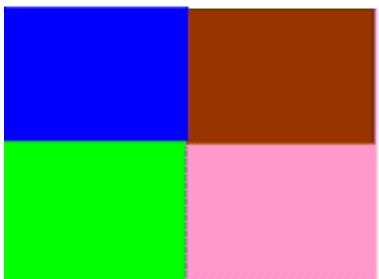


IMAGE 2

BLUE	BROWN
GREEN	PINK

SHAPE OF COLOR REGIONS IN IMAGE 2

Fig. 31 Demonstration of spatial information

IMAGE OF A SEGMENTED FACE

LE		RE
	N	
	M	

SYMBOLIC IMAGE OF THE FACE



{LE < N : M < RE , M < N < LE :RE}

2D-STRING REPRESENTATION

Fig. 32 2-D representation of the human face

- Let the spatial relationships between objects in the image be set of spatial relational operators

$$V_{SP} = \{<, =, :\} \quad (60)$$

where

< denote left/right or below/above relationship

: denote same grid position

= same projection on the x or y axis

The one-dimensional string representation of objects in the image is, from ***left -to-right***,

$$LE < N = M < RE \quad (61)$$

Interpreting the string,

The nose is **to the right** of the left eye. The nose and the mouth are on the **same x** projection. The right eye is **to the right** of the mouth.

From ***below -to -above***,

$$M < N < LE = RE \quad (62)$$

Interpreting again the nose is **above** the mouth. The left eye is **above** the nose, and the left and right eyes are on the **same y** projection

By viewing the ‘flow’ of the image regions from left to right and from bottom to top, and inserting the appropriate spatial relational operator two one-dimensional strings are merged to obtain the so called **normal** 2-D string representation

$$\{LE < N = M < RE, M < N < LE = RE\} \quad (63)$$

The first substrings describe the spatial relationship along the x -axis while the second substrings describe the relationship along the y -axis.

6.4 Similarity matching in 2-D strings

Before 2-D string matching, **normal** string description of each database image is pre-processed to translate the string description into a set of form called **expanded** 2-D strings. For the example image the expanded 2-D string is

$$\{x, r, s\} = \{\{LE, N, M, RE\}\{0,1,1,2\}, \{2,1,0,2\}\} \quad (64)$$

The expanded 2-D string is a three one-dimensional string where x contains the names of the objects as they appear in the 2-D string, r and s contain ranks of the objects along x and y directions respectively. The ‘rank’ of an object in a one-dimensional string equals to the number of ‘<’ preceding it in the string. This implies that the rank of an object in a string representing the projections of objects along the x -axis equal the number of objects on its left and the rank of an object in a string representing the projections of objects along the y -axis equal the number of objects which are below it.

In similarity matching the problem of image retrieval is transformed to the problem of 2-D string matching. For each image in the database the derived expanded 2-D string is stored in a different database called the spatial feature database. 2-D string representations corresponding to all stored images are individually compared with the 2-D string representation of the query image through a set of intersection operation. A non-empty intersection implies similarity among the images. Subsets of stored strings having more elements in the intersection sets are said to be more similar to the query image.

The disadvantage of 2-D string representation is that object centroid used to compute spatial relationships of all objects in the global image does not represent the complete picture of the spatial organization. According to [27], retrieval using 2-D strings suffers from exponential time complexity when computing the number of objects that are common to the query and the database images.

6.5 Spatial orientation graph explained

Spatial orientation graph is a technique for representing spatial relationships among objects used in symbolic representations of images by assigning values to edges or links between the objects constituting the symbolic image.

6.6 Deriving spatial orientation graph

The first step in constructing a spatial orientation graph is symbolic representation of the images. Then an ‘edge listing’ or sets of object ‘linking list’ for each symbolic image in database image and query image are constructed.

An edge in the spatial-orientation graph is a line connecting any two objects in the symbolic image. The edge lists is generated using a defined criteria. Consider an image represented by symbols as shown in Fig.33. Using the criteria adopted in [27] the objects or vertices of the symbolic image are lexicographically sorted in the spatial-orientation graph as P, Q, R, S, and T. Thereafter the vertex or object names are paired in such a way that the resulting edge names in the list remain in lexicographically sorted order. In the example shown the edge list generated using this criteria are

$$\{PQ, PR, PS, PT, QR, QS, QT, RS, RT, ST\}. \quad (65)$$

The number of edges in the list for any symbolic image is

$$\left(\frac{n(n-1)}{2} \right) \quad (66)$$

where n is the number of objects in the symbolic image.

The slope of each link in the graph is computed by considering the edges as directed line segments and applying coordinate translation such that their starting points are at the origin. The orientation between two links is the smaller of the two angles between the two line segments as shown in Fig.34. For each edge in the list, the object connected by the edge and the slopes of the edge are stored in the database

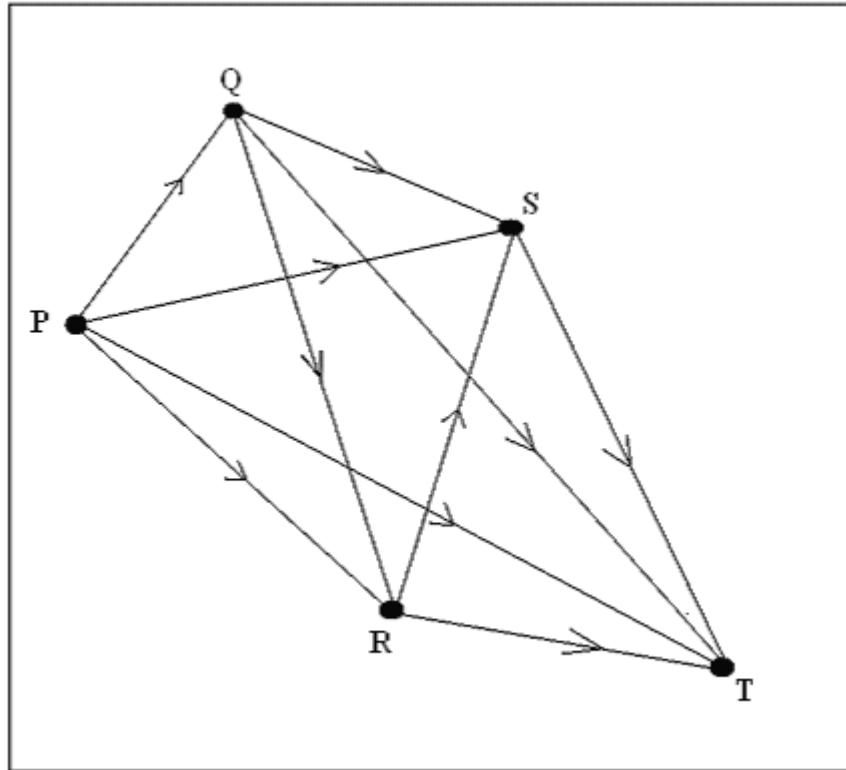


Fig. 33 Lexicographic ordering of edge list in spatial graph

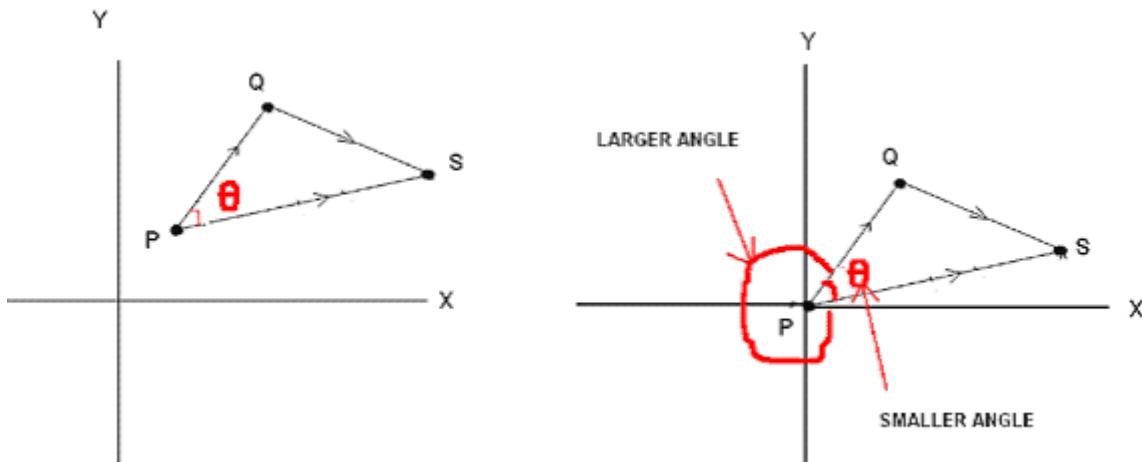


Fig. 34 Definition of edge orientation in a spatial graph

6.7 Similarity matching in spatial orientation graph

A similarity function between the symbolic query image and each symbolic image in the database is defined as

$$SIM : \{E^1, E^2\} \rightarrow R \quad (67)$$

E^1 represent the edge list of the first image, the query image, and E^2 represent the edge list of the second image, the database image.

The similarity function takes edge list corresponding to two symbolic images as arguments and produces a real number ranging from 0 to 100 indicating their similarity. Spatial similarity is quantified in terms of the number of, as well as to the extent to which the edges of the spatial orientation graph of the database image conform to the corresponding edges of the spatial-orientation graph of the query image.

Spatial-orientation graph is translation, scale and rotation invariant but the rotation invariance property is lost for large magnitude of rotation [27]. It has quadratic time complexity in terms of the number of objects in both the database and query images.

6.8 Geometrical transform

Geometrical transform techniques use the properties of radon transformation to develop image feature that contain both spatial and statistical information. Radon transform is named after J. Radon who showed how to describe a function in terms of its integral or projections. Radon transform of a function is the line integral of the function. The Radon transformation shows the relationship between a 2-D object and its projections. For an image it is the projection of the image intensity along a radial line oriented at a specific angle.

6.9 Theory of Radon transforms

Consider a 2-D function $f(x, y)$ shown in Fig.35. Integration along the line whose normal vector is in θ direction results in the function $g(s, \theta)$, which is the projection of the 2D function $f(x, y)$ on the axis s of θ direction. When s is zero, the g function has the value $g(0, \theta)$ which is obtained by integration along the line passing the origin of (x, y) -coordinate. The points on the line whose normal vector is in θ direction and passes the origin of (x, y) -coordinate satisfy the equation:

$$\frac{y}{x} = \tan(\theta + \frac{\pi}{2}) = \frac{-\cos \theta}{\sin \theta} \quad (68)$$

\Rightarrow

$$x \cos \theta + y \sin \theta = 0 \quad (69)$$

The integration along the line whose normal vector is in θ direction and which passes the origin of (x, y) -coordinate means the integration of $f(x, y)$ only at the points satisfying Eqtn. 69.

Using the unique property of the Dirac function δ , which is zero for every argument except at 0 where its integral is one, $g(0, \theta)$ is expressed as:

$$g(0, \theta) = \iint f(x, y) \delta(x \cos \theta + y \sin \theta) dx dy \quad (70)$$

The line with normal vector in θ direction and distance s from the origin is satisfying the equation:

$$(x - s \cos \theta) \cos \theta + (y - s \sin \theta) \sin \theta = 0 \quad (71)$$

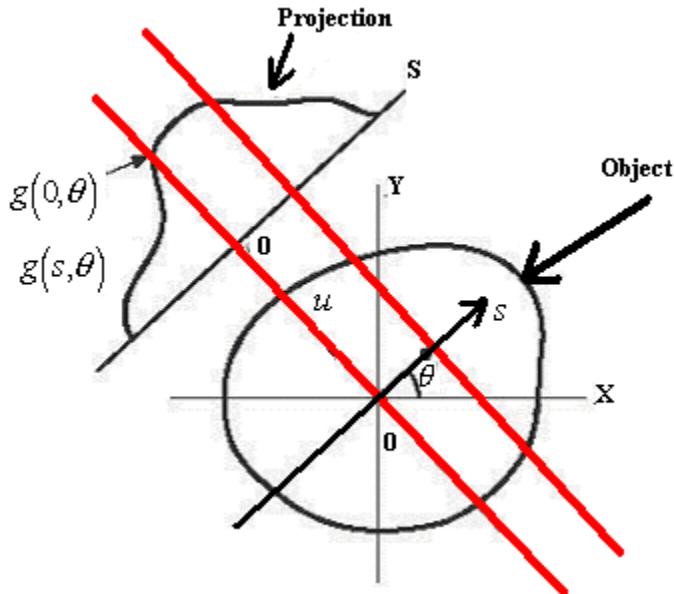


Fig. 35 Example of Radon transformation of an object

⇒

$$x \cdot \cos \theta + y \cdot \sin \theta - s = 0 \quad (72)$$

6.10 Radon transform representation

The unique properties of radon transform that is utilized in developing spatial information of image features are

1. **Translation.** Translation of $f(x, y)$ results in a shift of $g(s, \theta)$ in s

$$f(x - x_o, y - y_o) \Rightarrow g(s - x_o \cos \theta - y_o \sin \theta, \theta)$$

2. **Rotation.** Rotation of the $f(x, y)$ causes translation of $g(s, \theta)$ in θ

$$f(r, \phi + \theta_o) \Rightarrow g(s, \theta - \theta_o)$$

3. **Amplification.** Amplification of $f(x, y)$ causes amplification of $g(s, \theta)$ in s

The steps stated below [28] is used to derive the image feature vector

1. Each database image is transformed into $g(s, \theta)$ in the parameter space
2. The modal (s_0, θ_0) of $g(s, \theta)$ is determined.
3. Translate $g(s, \theta)$ into $g'(s, \theta)$ by wrapping around s and θ to make (s_0, θ_0) the new origin.
4. Translation of $g(s_0, \theta_0)$ to the origin and wrapping around s makes the feature translation invariant.
5. Translation of $g(s_0, \theta_0)$ to the origin and wrapping around θ makes the feature rotation invariant.

6. Unify $g'(s, \theta)$ by dividing by $g(s_0, \theta_0)$

The overall effect of the Radon transform is that spatial information within the image is preserved because the geometrical distributions of objects within the region is reflected in the image feature.

6.11 Similarity measure of Radon transforms

Similarity measure is carried out by computing the distance of the query image signature from corresponding signatures of images in the database.

6.12 Summary of the chapter

Research efforts in CBIR based on spatial information is yet to yield meaningful result because reliable segmentation of image regions is not feasible except in very limited applications [2]. Radon transform was proposed to address this problem. Though Radon transform compute image feature without complex segmentation research interest in this direction is yet to be appreciated in current image retrieval techniques. The reason is that its efficient retrieval performance is limited to only simple images on plain background. The technique performs poorly on images with fine details because complex images exhibiting fine details contain high frequency components that are suppressed by the low-pass filtering inherent in the algorithm.

7

SYSTEM DESIGN

7.1 CBIR system building block

Two main matlab programs processed the images in the database. They are **CBIR_create_image_database.m** and **CBIR_texture_shape_spatial_features_extract.m**. The matlab program, which operates the CBIR system is **CBIR_operation.m**

The database image processing programs and the CBIR operating program that are the building blocks in the design of the system are shown in Fig.36 and Fig.37, and explained below.

- **CBIR_create_image_database.m**

This program takes image text data file as input. It reads in all the database images in the matlab directory, resizes each image to a fixed dimension and stores them in **CBIR_image_database.mat**.

- **CBIR_texture_shape_spatial_features_extract.m**

This program receives the image database file as input and extracts texture, shape and spatial information features from each image in the database. It outputs three data files, **CBIR_texture_feature_database.mat**, **CBIR_shape_feature_database.mat** and **CBIR_spatial_feature_database.mat**

- **CBIR_operation.m**

This program receives three inputs, the three feature databases (texture, shape and spatial information). It processes the query image using the same algorithm that was used in extracting feature vectors from the database images. It compares each of the features extracted from the query image with corresponding features of each of the database images and ranks them in ascending order. Twelve most matched images are displayed according to each of texture, shape and spatial information.

7.2 Texture feature extraction

Texture feature was computed using Gabor wavelets. Gabor function was chosen as tool for texture feature extraction because of its widely acclaimed efficiency in texture feature extraction. B.S Manjunath and W.Y Ma [30] recommended Gabor texture features for retrieval after showing that Gabor features performs better than that using pyramid-structured wavelet transform features, tree-structured wavelet transform features and multiresolution simultaneous autoregressive model.

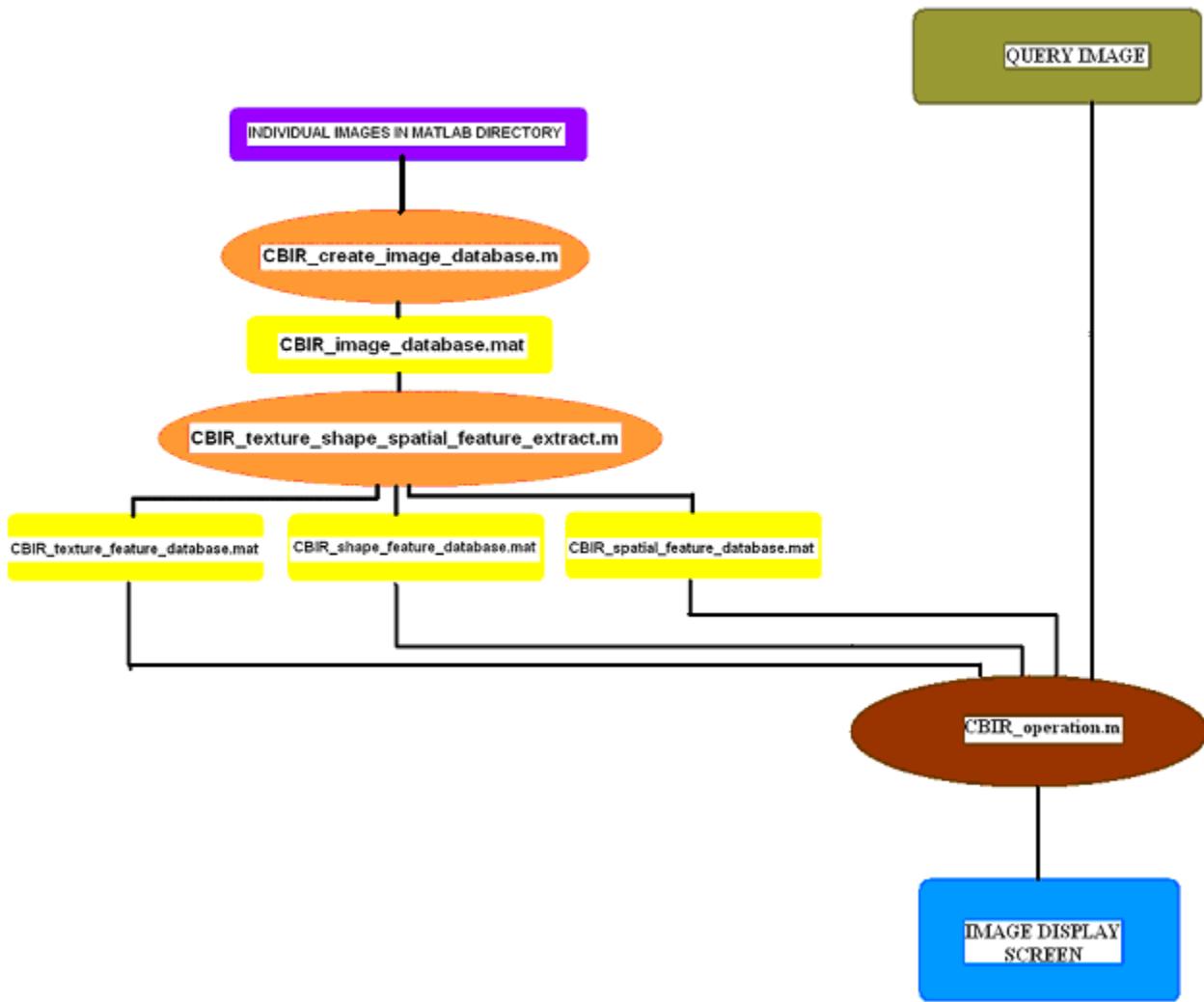


Fig. 36. Block diagram showing matlab programs used for building the CBIR system.

A total of twenty four wavelets were generated from each of six grid sizes of ‘mother’ Gabor function using four scales of frequency and six orientations. The six sets of each of the twenty four wavelets were obtained from grid sizes of 5×5 , 15×15 , 25×25 , 35×35 , 45×45 and 55×55 . Fig.38 shows the 2-D view of the wavelets (channel) that constitute the Gabor filter bank of size 25×25 . Moving horizontally from right to left of the page the exponential scale of frequency increase while moving vertically from top to bottom the orientation increase.

Redundancy, which is the consequence of the nonorthogonality of Gabor wavelets, was addressed by choosing the parameters of the filter bank to be set of frequencies and orientations that cover the entire spatial frequency space so as to capture texture information as much as possible in accordance with filter design in [30]. The lower and upper frequencies of the filters were set at 0.04 octaves and 0.5 octaves respectively, the orientations were at intervals of 30 degrees, and the half-peak magnitudes of the filter responses in the frequency spectrum are constrained to touch each other as shown in Fig. 39. Note that because of the symmetric property of the Gabor function as explained in section 4.9, wavelets with

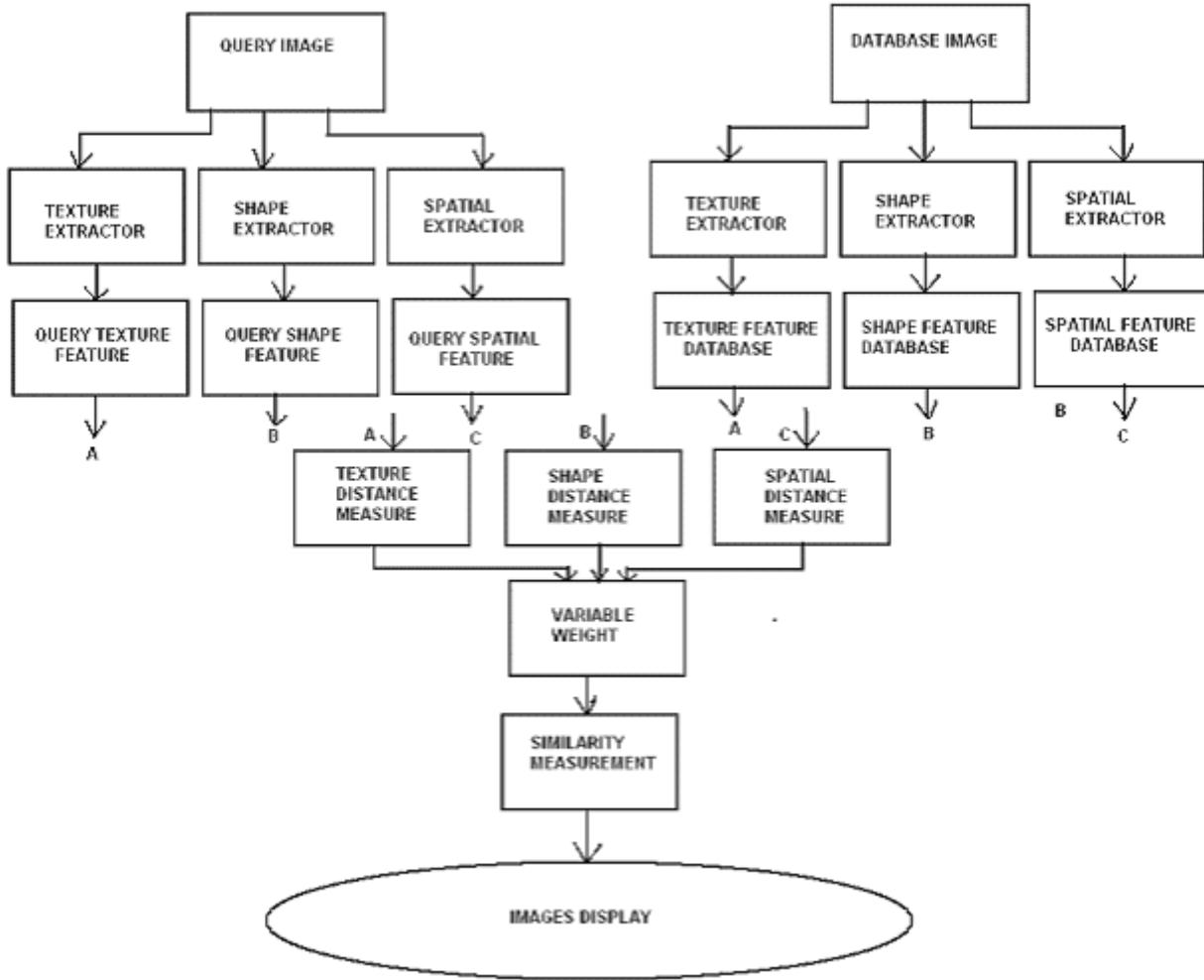


Fig. 37 Block diagram of sub-unit programs used in designing the CBIR system

center frequencies and orientation covering only half of the frequency spectrum $\left(0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{3\pi}{2}, \frac{5\pi}{6}\right)$ are generated.

The following filter parameters relating to earlier discussion on Gabor wavelet in section 4.9 and 4.11 were used in the filter dictionary design

$$\alpha = \left(\frac{U_h}{U_l} \right)^{\frac{1}{S-1}} \quad (73)$$

$$\sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2 \ln 2}} \quad (74)$$

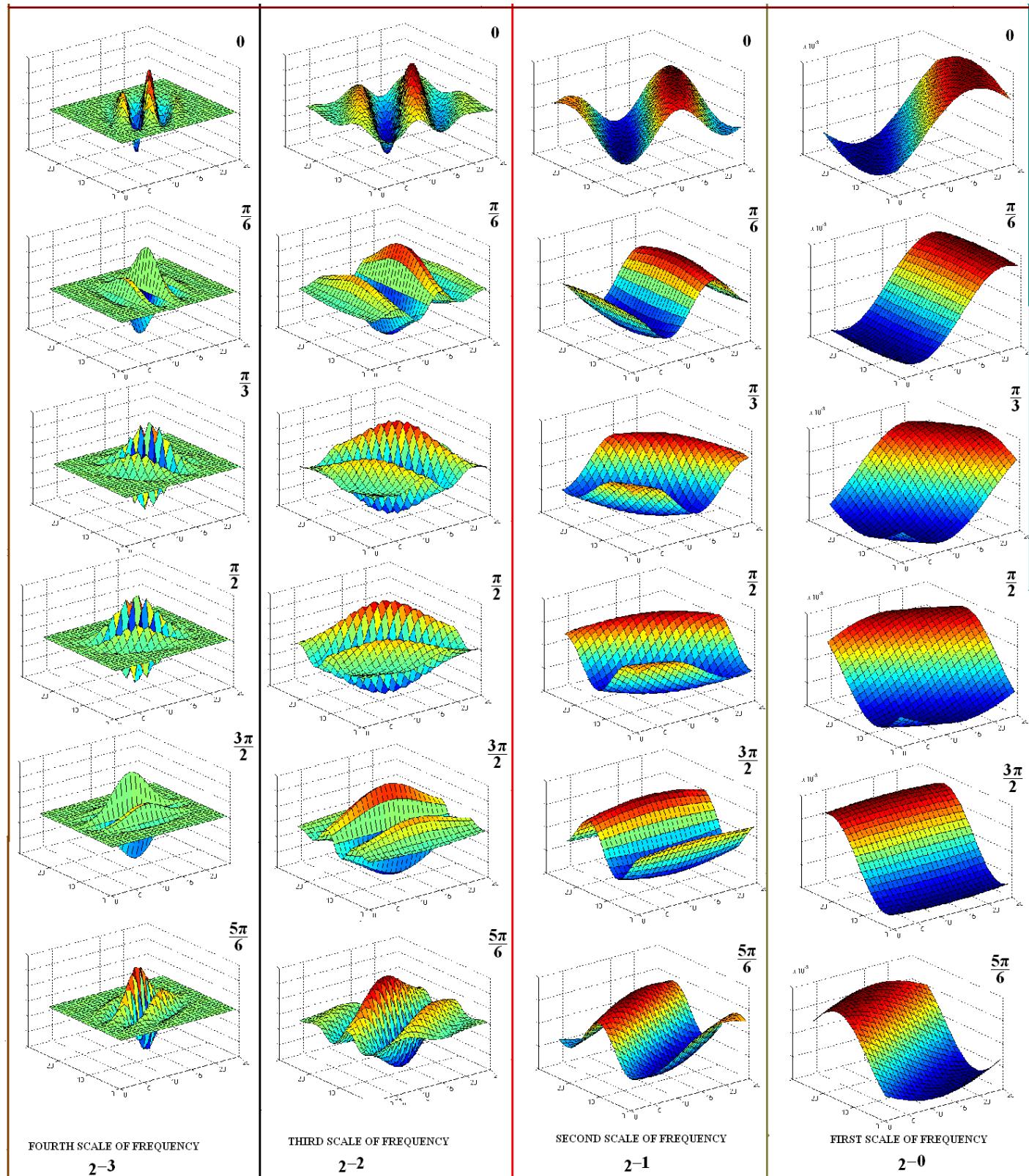


Fig. 38. The twenty four wavelets (channels) that constitute the 25×25 grid size Gabor filter bank

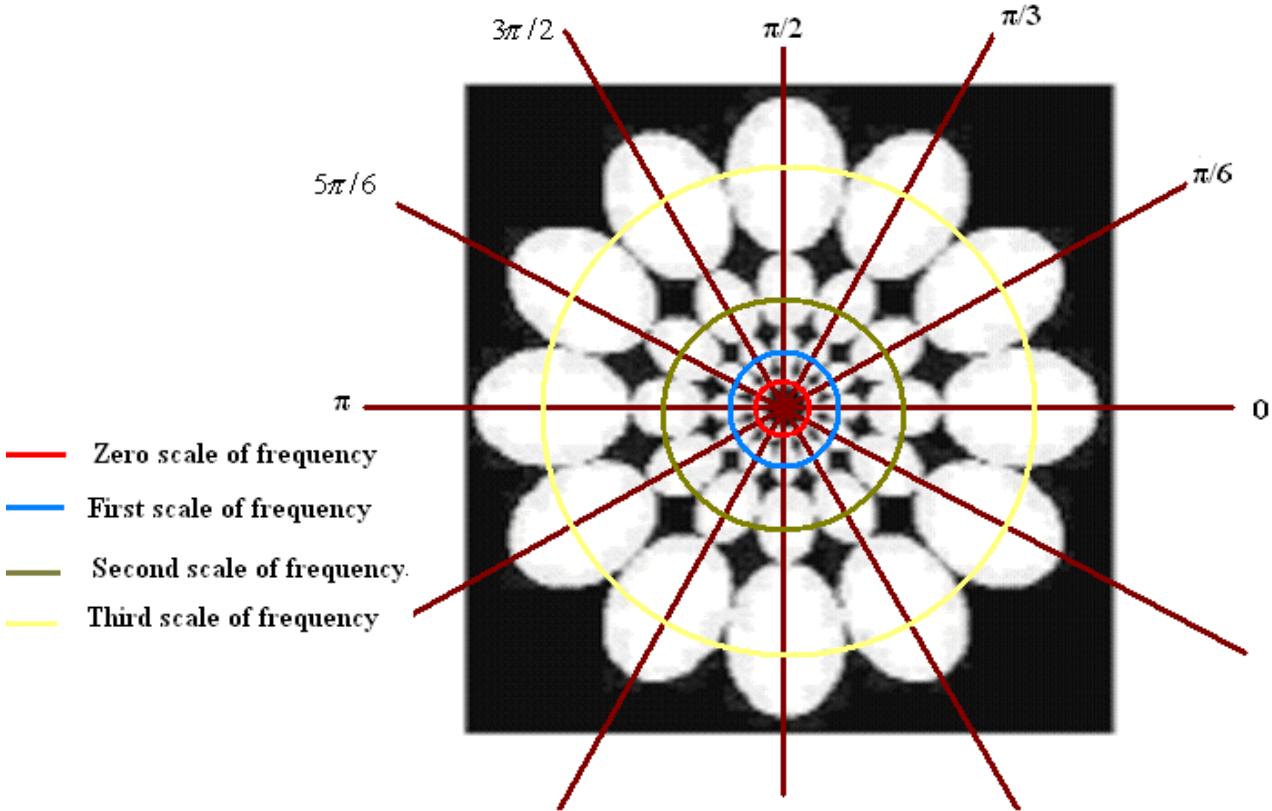


Fig 39. 2-D frequency spectrum view of Gabor filter bank designed to cover frequency spectrum as much as possible

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right) \left[U_h - 2 \ln\left(\frac{\sigma_u^2}{U_h}\right) \right] \left[2 \ln 2 - \frac{(2 \ln 2)^2 \sigma_u^2}{U_h^2} \right]^{-\frac{1}{2}} \quad (75)$$

k, s are the respective number of orientations and scales of frequencies.

U_l, U_h are the upper and lower frequencies of interest.

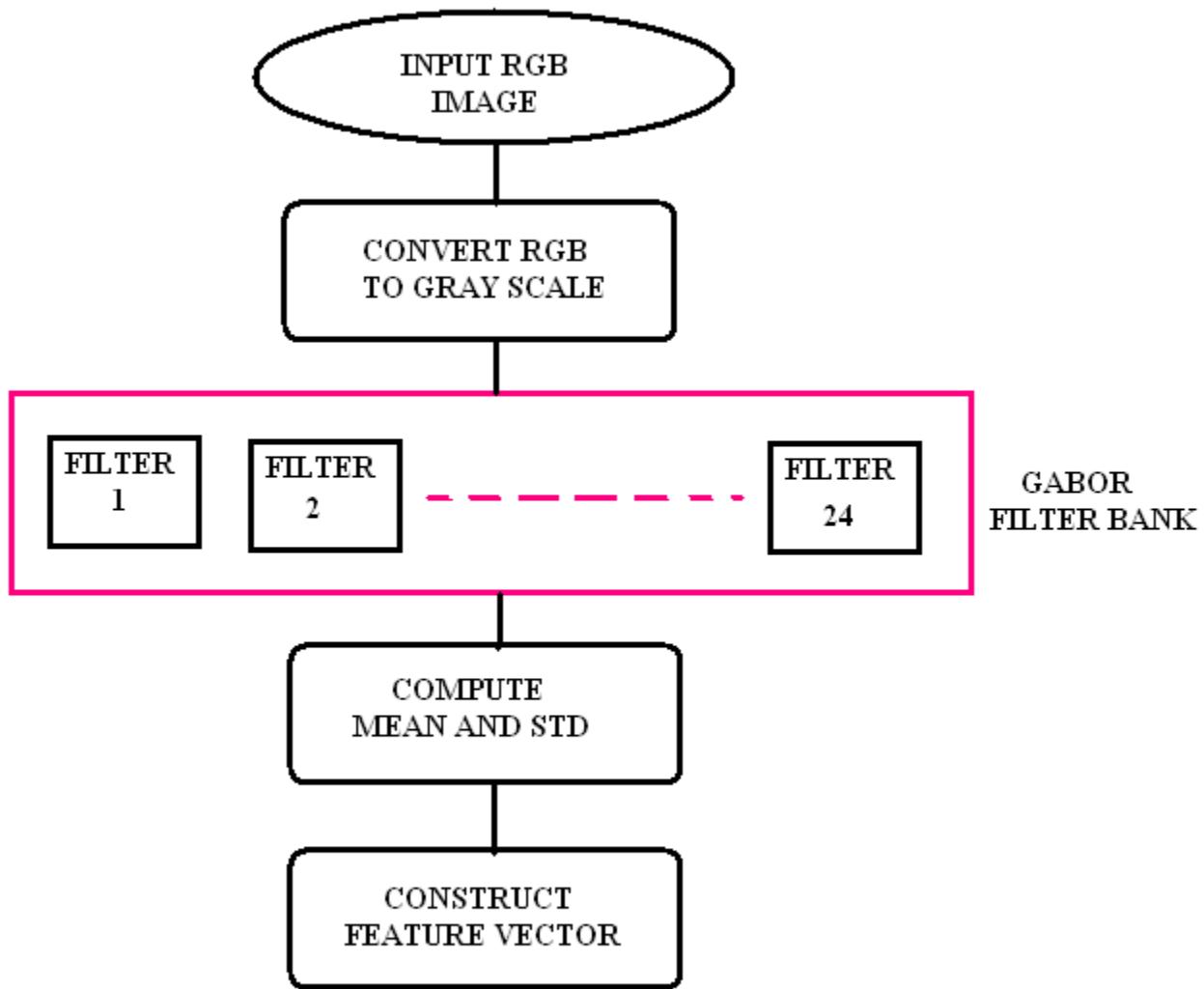
Each database image $I(x, y)$ was convolved with each wavelet in the filter bank according to the convolution equation

$$G_{mn}(x, y) = \int I(x-s, y-t) \varphi_{mn}^* (s, t) \quad (76)$$

s, t are the dimensions of the filter and φ_{mn}^* is the complex conjugate of the Gabor wavelet.

By assuming spatial homogeneity of texture regions, the texture features were computed as the mean and standard deviation of the magnitude of the transformed coefficients according to the formula

$$\mu_{mn} = \iint |G_{mn}(x, y)| dx dy \quad (77)$$

**Fig. 40 Block diagram of texture feature extraction method**

and

$$\sigma_{mn} = \sqrt{\iint (G_{mn}(x, y) - \mu_{mn})^2 dx dy} \quad (78)$$

Texture feature vector was constructed using the computed mean μ_{mn} and standard deviation σ_{mn} as feature components according to the formula

$$CBIR_texture_feature = [\mu_{00} \sigma_{00} \mu_{01} \sigma_{01} \dots \mu_{35} \sigma_{35}]$$

Block diagram of the algorithm is shown in Fig. 40

7.3 Shape feature extraction

Shape feature extraction takes the following steps

- Local mean and standard deviation (5×5 neighbourhood) of each database image pixel was computed according to the convolution equation

$$\mu_I(x, y) = \sum_{g=1}^5 \sum_{h=1}^5 I(x-5, y-5) * Z(5,5) \quad (79)$$

and

$$\sigma_I(x, y) = \sum_{g=1}^5 \sum_{h=1}^5 I(x-5, y-5) * K(5,5) \quad (80)$$

Z, K are the respective impulse responses of the mean and standard deviation filters.

Since texture is assumed to be homogeneous the local mean and standard deviation represent the texture feature of the pixel within the neighbourhood.

- Each database image was fed into a 5×5 grid size Gabor filter bank. Twenty four output images were obtained.
- Like the pre-filtered image the local mean and standard deviation of each output of the filter bank was also calculated using 5×5 neighbourhood according to the equation

$$\mu_{pmn}(x, y) = \sum_{g=1}^5 \sum_{h=1}^5 G_{pmn}(x-5, y-5) * Z(5,5) \quad (81)$$

and

$$\sigma_{pmn}(x, y) = \sum_{g=1}^5 \sum_{h=1}^5 G_{pmn}(x-5, y-5) * K(5,5) \quad (82)$$

$$p = 1, 2, \dots, 24$$

Thus for each pixel in the database image there are twenty four reference pixels as shown in Fig. 41

- Consider a pixel, $I(x, y)$ in the database image and the twenty four corresponding pixels $G_{pmn}(x, y)$ in the output of the filter bank, the distance between the database image feature vector and any of the corresponding pixels feature vector is computed as

$$D\{I(x, y), G_{pmn}(x, y)\} = |\mu_{pmn} - \mu_I| + |\sigma_{pmn} - \sigma_I| \quad (83)$$

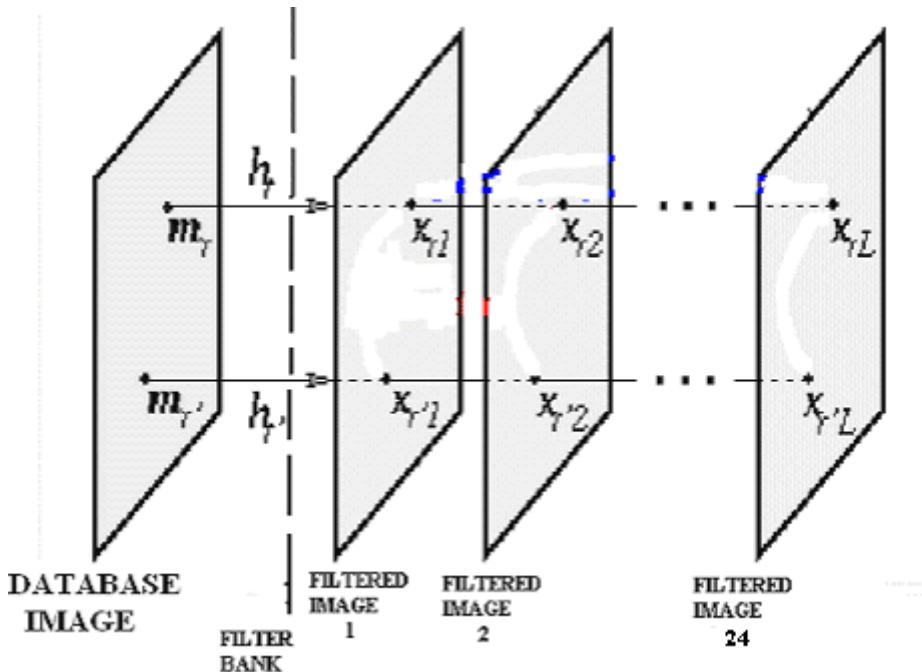


Fig. 41 Illustration of pixel correspondence between each pixel in database image and the twenty four reference pixels in the output of the filter bank

The computed distance is a measure of similarity of texture between pixel in the original image and corresponding pixel at the output of the Gabor filter bank.

For every pixel in the database image and for each distance measure $D\{I(x, y), G_{pmn}(x, y)\}$ with corresponding pixels in the output of the Gabor filter, the pixel whose texture is most likely similar to that of the database image is

$$\{G_{pmn}(x, y) = I(x, y)\} \vee \{\min\{D\{I(x, y), G_{pmn}(x, y)\}\}\} \quad (84)$$

This results in a filtered image called ‘texture classified image’. Its texture is most similar to the texture of the database image

- The texture-classified image was first divided into 2×2 blocks followed by computing its mean and standard deviation. Texture segmented image was obtained by finding the absolute value of the square root of the sum of squares of the mean and standard deviation.
- The texture-segmented image was grey-thresholded by computing global image threshold using Otsu's method before converting image to binary.
- Simple morphological processing was carried out on the binary image.
- Each object or region in the binary image were identified and labelled using connected component analysis.
- Thereafter five region properties namely area, eccentricity, minor axis length, major axis length and equivalent diameters were computed.

The number of elements for each region property was normalized to 100. Block diagram of the algorithm is shown in Fig. 42

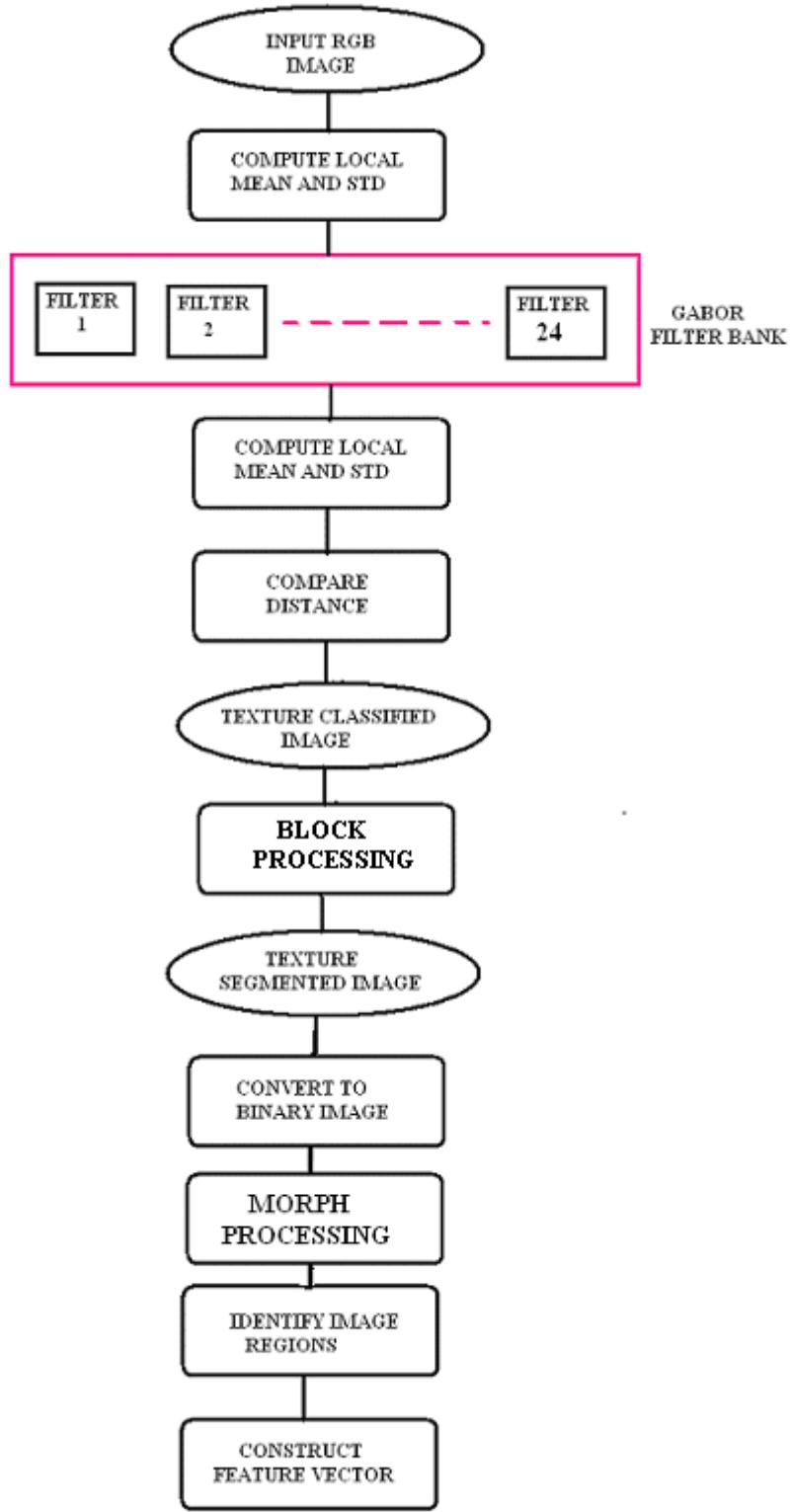


Fig. 42 Block diagram for shape feature extraction

7.4 Spatial information extraction

The two region properties adopted to describe the spatial features are elementary spatial feature descriptors, centroid and spatial extent. This is because the image database is complex and heterogeneous. Though elementary features, the spatial descriptors are invariant to rotation and translation.

The steps taken in spatial information extraction are as follows

- Compute the centroid distances of regions in the binary image obtained from the texture segmented image

To obtain translation and rotation invariant feature vector the centroid distances from the geometrical center (x_c, y_c) of the image were computed according to the equation,

$$C = \sqrt{[x_{Ti} - x_c]^2 + [y_{Ti} - y_c]^2} \quad (85)$$

(x_{Ti}, y_{Ti}) are centroids of the binary image

- Thereafter the spatial extents of the texture regions are also separately computed according to the equation

$$SE = \left(\frac{A_{Ti}}{A} \right) \quad (86)$$

A , total area

A_{Ti} , texture region area

$i = 1, 2, \dots, M$

M , number of regions in the binary image

The number of elements for each region property was normalized to 100.

7.5 Similarity calculation

The similarity measure between the query image and each image in the database for each of texture, shape and spatial information features is carried out in the Euclidean space according to the equation

$$D_{ik} = \sqrt{\{J_{ik}\}^2 - \{Q_i\}^2} \quad (87)$$

D is the Euclidean distance

J is the database image feature vector

Q is the query image feature vector

$k = 1, 2, \dots, P$

P is the number of images in the database

$i = 1, 2, 3$

$i = 1$ is the index for texture

$i = 2$ is the index for shape

$i = 3$ is the index for spatial information

The computed Euclidean distance between query image and database images for each of the feature vectors are normalized so it lies between 0 and 1. Normalization becomes necessary because the feature vectors have to be weighted in similarity distance calculation. Weighting of the feature vectors is necessary for a diverse database because since a particular feature cannot adequately describe an image a weighted combination will give an optimal description. In a diverse database the features suitable for retrieval of the images similar to a query image varies with the class of images with which the database is modelled or constituted. Explanation had been given in section 3.3.

Weight assignment is the degree of relevance the similarity matching process assigns to a particular visual feature. For example if the query image is the silhouette of machine parts and the database is diverse but including images similar to the query image, it is more suitable to assign high weight to shape feature for retrieval as this is the most suitable feature for retrieval. It is useless assigning high weight to texture or spatial information except the database is narrow such that the other features particularly spatial information can be used for discrimination.

This system is designed so that the user can retrieve images using flexible weight combinations. Flexible weight in a sense provides the much needed retrieval refinements and robust characteristics to the system. By flexibly combining the different weight features the retrieval process can be refined several times to satisfy the user's demand.

For a query image Q , and k th number of images in the image database, if i numbers of visual feature vectors are considered for retrieval, i different distances $d_{ik}(Q(x, y), I_k(x, y))$ will be obtained.

The effective feature distance obtained from the weighted sum of each feature distance is given by

$$D_k = \sum_p w_p d_{ik}(Q(x, y), I_k(x, y)) \quad (88)$$

$$\sum_p w_p = 1 \quad (89)$$

Eqtn. 89 imply that the sum of the weights must be equal to unity. A weight assignment of 1 is the highest degree of relevance that can be assigned to a particular feature while assignment of zero is the highest degree of irrelevance.

The retrieved images are the twelve most matched images whose feature distances are the first twelve in ascending order according to texture, shape. Spatial information and weighted features.

8

SYSTEM OPERATION

8.1 Program codes

The following twenty one matlab files are required to operate the system

1. **CBIR_interface.m**
2. **CBIR_interface.fig**
3. **CBIR_operation.m**
4. **CBIR_texture_feature_database_5_D_mat**
5. **CBIR_texture_feature_database_15_D_mat**
6. **CBIR_texture_feature_database_25_D_mat**
7. **CBIR_texture_feature_database_35_D_mat**
8. **CBIR_texture_feature_database_45_D_mat**
9. **CBIR_texture_feature_database_55_D_mat**
10. **CBIR_shape_feature_database.mat**
11. **CBIR_spatial_feature_database.mat**
12. **gabor_filter.m**
13. **Gabor_features.m**
14. **mean_std_filter.m**
15. **Euclid_dist.m**
16. **filter_size_5.mat**
17. **filter_size_15.mat**
18. **filter_size_25.mat**
19. **filter_size_35.mat**
20. **filter_size_45.mat**
21. **filter_size_55.mat**

The CBIR system was tested on three image databases. They are the university of Washington image database [31], the Pennsylvania state university image database [32] and the Stanford university image database [33].

The codes for each database is stated below

UW - University of Washington image database.

PSU - Pennsylvania state university image database.

SU - Stanford university image database.

Note that six sets of Gabor filter banks were used in the system. Program codes # 1, 2, 3, 10 11, 12, 13, 14, 15 are common to all the six sets of filter banks. Program codes # 4, 5, 6, 7, 8 and 9 are distinguished from each filter bank by attaching the filter grid size and database name to the code name. For example

CBIR_texture_feature_database_mat for a 25 x 25 filter bank and university of Washington image database is written as **CBIR_texture_feature_database_25_UW_mat**.

There are three main matlab directories The matlab directories are named

- **CBIR_UW**

- **CBIR_PSU**
- **CBIR_SU**

Each matlab directories contain the images for each of the database and each of the sixteen matlab codes are loaded into these directories

8.2 Summary of system operation.

The system operation, block diagram is shown in Fig. 43, begins by typing **CBIR_Interface** at the matlab command window. Pressing the return key displays the graphical user interface (GUI). Typing **CBIR_Interface** as **CBIR_interface** in the matlab command window gives warning message

Warning: Function call CBIR_interface invokes inexact match H:\CBIR_PSU\CBIR_Interface.m.

because of the lower case letter ‘i’

The physical features of the GUI are shown in Fig. 44.

The sequences of operation of the system are as follows

- Load system data - Filter size (**A**), texture feature database (with appropriate filter size code) (**B**), shape feature database (**C**) and spatial feature database (**D**). Click on any of the buttons displays a dialog box that prompts the user to select appropriate data. If appropriate system data is not selected and loaded the following error message results when the system is operated.

Error in ==> CBIR_Operation at 31

eval (loadstr);

Error in ==> CBIR_Interface>btnSubmit_Callback at 353

ImgOut = CBIR_Operation(w_Texture, w_Shape, w_Spatial, Gbfiltersize, imgtxtdbname, imgshdbname, imgspdbname);

Error in ==> gui_mainfcn at 75

feval (varargin {:});

Error in ==> CBIR_Interface at 70

gui_mainfcn(gui_State, varargin{:});

??? Error while evaluating uicontrol Callback.

- Load query image (**J**). A dialog box opens and prompts the user to select query image from the file directory.
- Choose and set feature weights - texture weight (**E**), shape weight (**F**) and spatial (**G**) using the slider.

- Validate the chosen weights by depressing the validation button (**I**). Validation ensures that weights sum up to unity. Error message is displayed when the sum of weights is not equal to 1.
- The query image is submitted to the system for processing by depressing the submit query button (**H**)
- After processing, twelve images according to similarity ranks are displayed.

The following operating features are available on the GUI

- Display sets of twelve most similar images according to texture (**L**), shape (**M**), spatial information (**N**) and weighted features (**O**).
- Display sets of images by varying the filter grid sizes. The system operates with six Gabor filter sizes.
- By the immediate left side of the query image, are four buttons, the three feature buttons corresponding to the three features and the weighted features. The system design enables all three buttons when weight other than zero is assigned to any of the feature. If the user assigns a feature weight of 1 to any of the feature only that feature is enabled on the set of feature buttons. Any feature assigned value of zero is not enabled on the feature button
- User can view detail of the query image in an enlarged window. The ‘image detail’ button (**K**), which is located, by the ‘load image’ button invokes matlab imtool function and open a new window to allow user have detailed view of the image to further help in using visual cues to assess retrieval result.
- Help file (**P**) is provided to assist the user

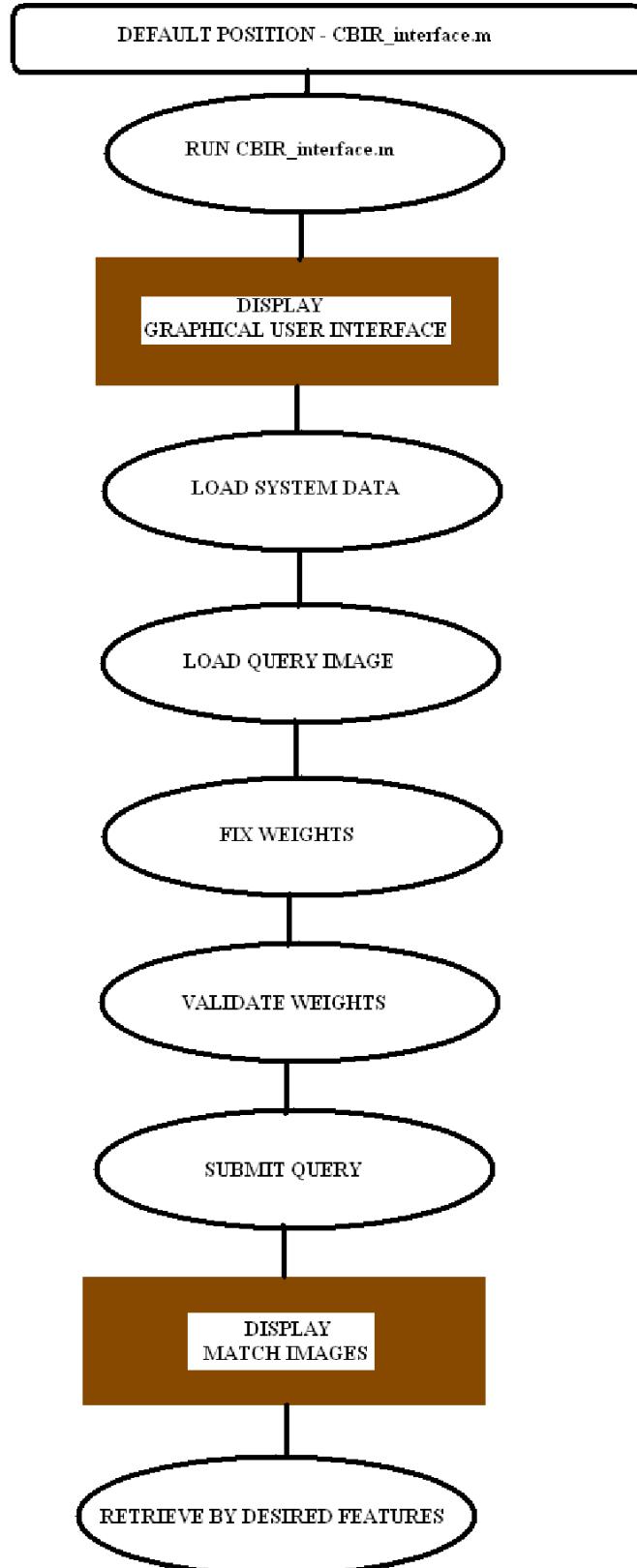


Fig. 43 Block diagram of the system operation

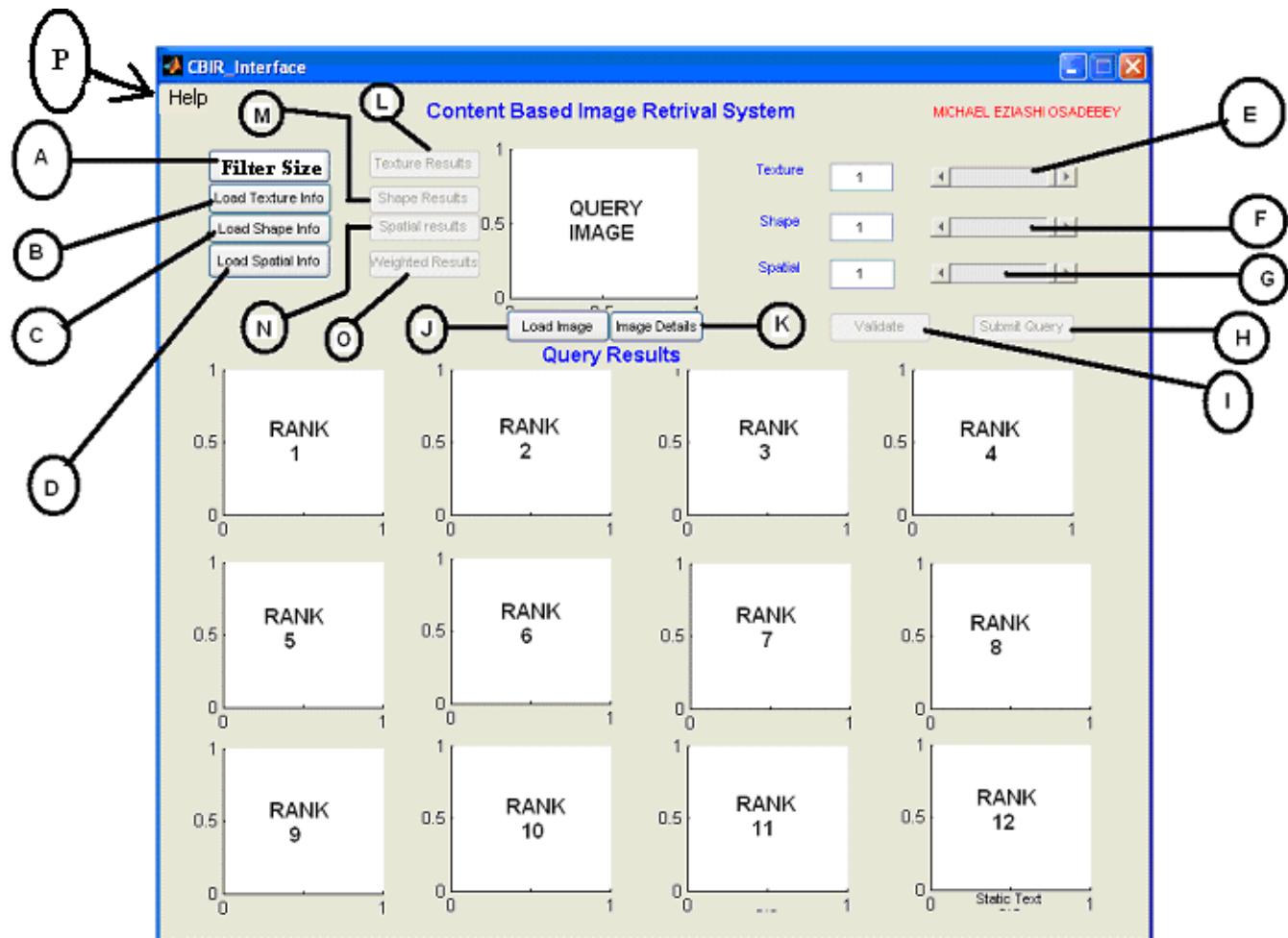


Fig. 44. Layout of the graphical user interface

8.3 System specification

The system's specification is given in the table below.

S/N	FEATURE	SYSTEM SPECIFICATION		
		U W	PSU	SU
1	DATABASE SIZE	161 MB	187 MB	465MB
2	IMAGE FORMAT	JPG	JPG	JPG
3	TEXTURE FEATURE SIZE	306 KB	355 KB	3.44 MB
4	SHAPE FEATURE SIZE	102 KB	114 KB	951 KB
5	SPATIAL FEATURE SIZE	31.2 KB	38.1 KB	311 KB
6	SYSTEM DATA LOADING TIME	Real time	Real time	Real time
7	QUERY IMAGE PROCESSING TIME	30 s	30 s	Real time
8	SIMILARITY MATCH TIME	60 s	60 s	30 s
9	NUMBER OF SIMILARITY RANKINGS	12	12	12
10	AVERAGE PRECISION	0.8 (First operation)	0.8(First operation)	0.8
11	RECALL	N/A	N/A	N/A
12	QUERY TYPE	Example	Example	Example
13	# OF IMAGES	860	1000	10,000
14	FEATURE	WEIGHT VALIDATE BUTTON		
15	FEATURE	VIEW QUERY IMAGE DETAILS		
16	FEATURE	HELP FILE		

9

CONCLUDING REMARKS

9.1 Evaluation and Results

The common evaluation measures used in CBIR systems are

precision, defined as

$$\text{precision} = \frac{\text{No. of relevant images retrieved}}{\text{Total number of images retrieved}} \quad (89)$$

and

recall, defined as

$$\text{recall} = \frac{\text{No. of relevant images retrieved}}{\text{Total number of relevant images in the database}} \quad (90)$$

This method has been criticised as not containing the desired information required for proper evaluation of an information retrieval system such as CBIR [34]. Several evaluation methods also based on precision and recall have been adopted by the Text retrieval conference (TREC) [35]. Among them are $P(10)$, $P(30)$, $P(N_R)$, the precision after the first 10, 30 and N_R documents have been retrieved. N_R is the number of relevant images. This method was used to evaluate this CBIR system.

Ten images representing different classes and complexities were chosen from each of the database as query images.

During precision- recall tests one hundred and eighty images were retrieved in succession in fifteen operations (15×12 retrieved images per operation). For the first operation the precision was computed while for the later operations the cumulative precision was computed. The retrieval result was for only texture retrieval. There was neither weighting nor relevance feedback.

Fig. 45 show the plot of cumulative precision versus number of images retrieved and Fig. 46 shows the plot of cumulative precision after retrieval of numbers of relevant images, both for the university of Washington image database and Gabor filter size of 25×25 . Fig. 47 shows the plot of cumulative precision versus number of images retrieved and Fig. 48 shows the plot of cumulative precision after retrieval of numbers of relevant images, both for the Pennsylvania state university image database and Gabor filter size of 25×25 . Fig. 49 shows the plot of cumulative precision versus number of images retrieved and Fig. 50 shows the plot of cumulative precision after retrieval of numbers of relevant images, both for the Stanford state university image database and Gabor filter size of 25×25 .

Test results for texture and weighted feature retrieval are shown in appendix B, for the university of Washington image database, appendix C, for the Pennsylvania state university image database and appendix D, for the Stanford state university image database. All the test results shown in the appendix are results for the first twelve images retrieved from the system, hence the tag '**first operation**'. The filter size, feature weights are indicated on the results.

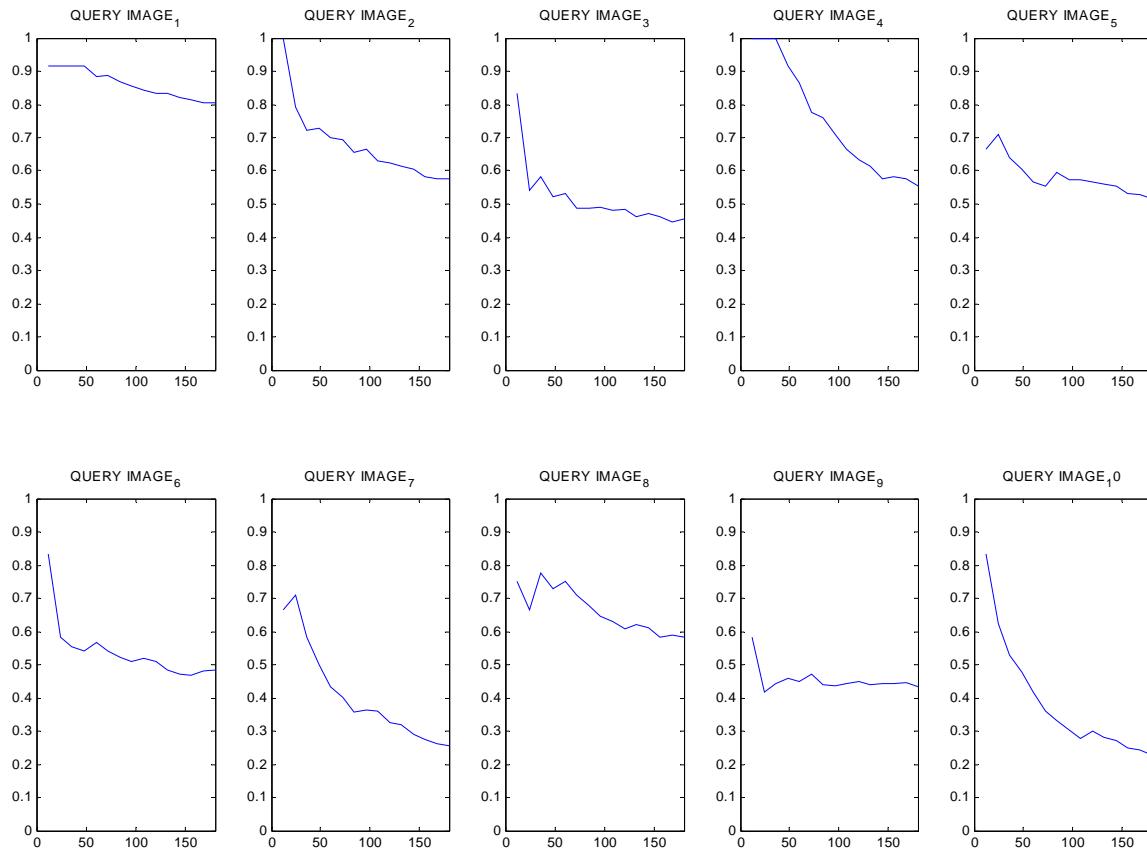


Fig. 45 Plot of cumulative precision versus number of images retrieved (25 x 25 texture only) -UW

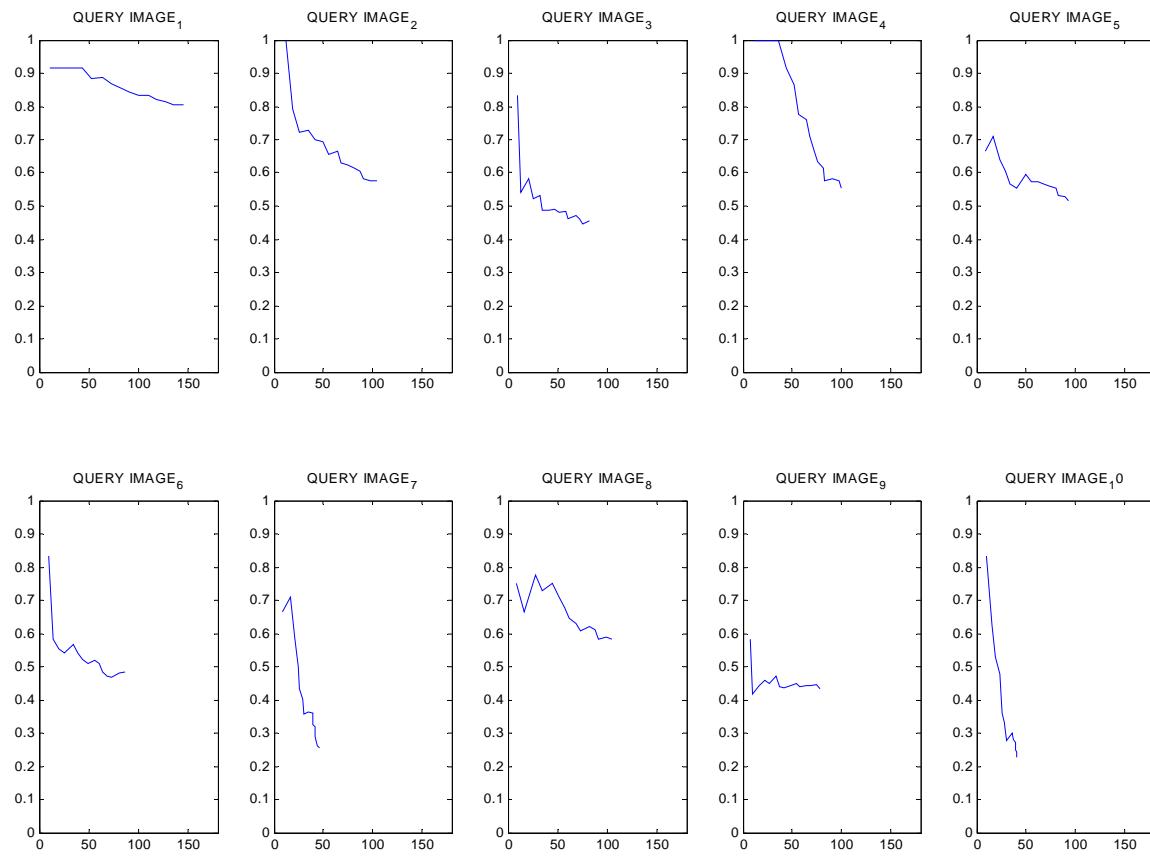


Fig. 46 Plot of cumulative precision versus retrieved relevant images (25 x 25 texture only) –UW

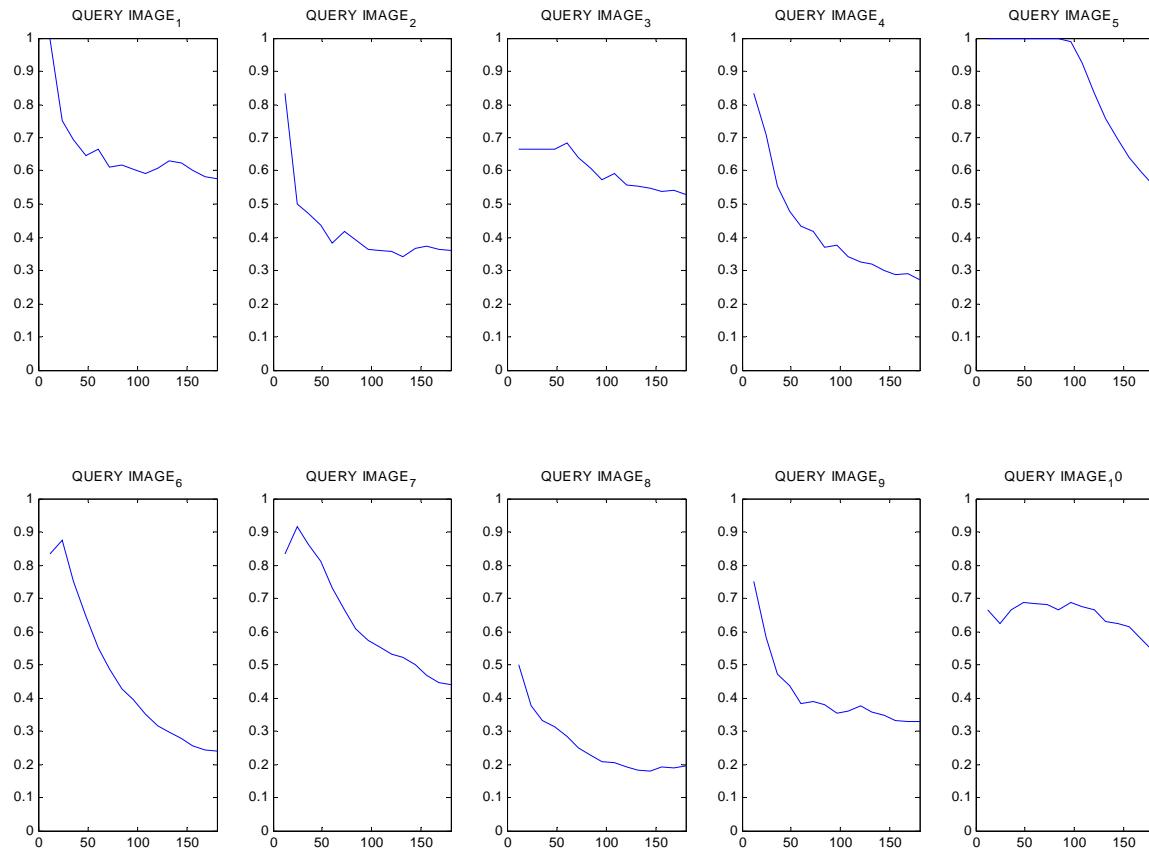


Fig. 47 Plot of cumulative precision versus number of images retrieved (25 x 25 texture only)–PSU

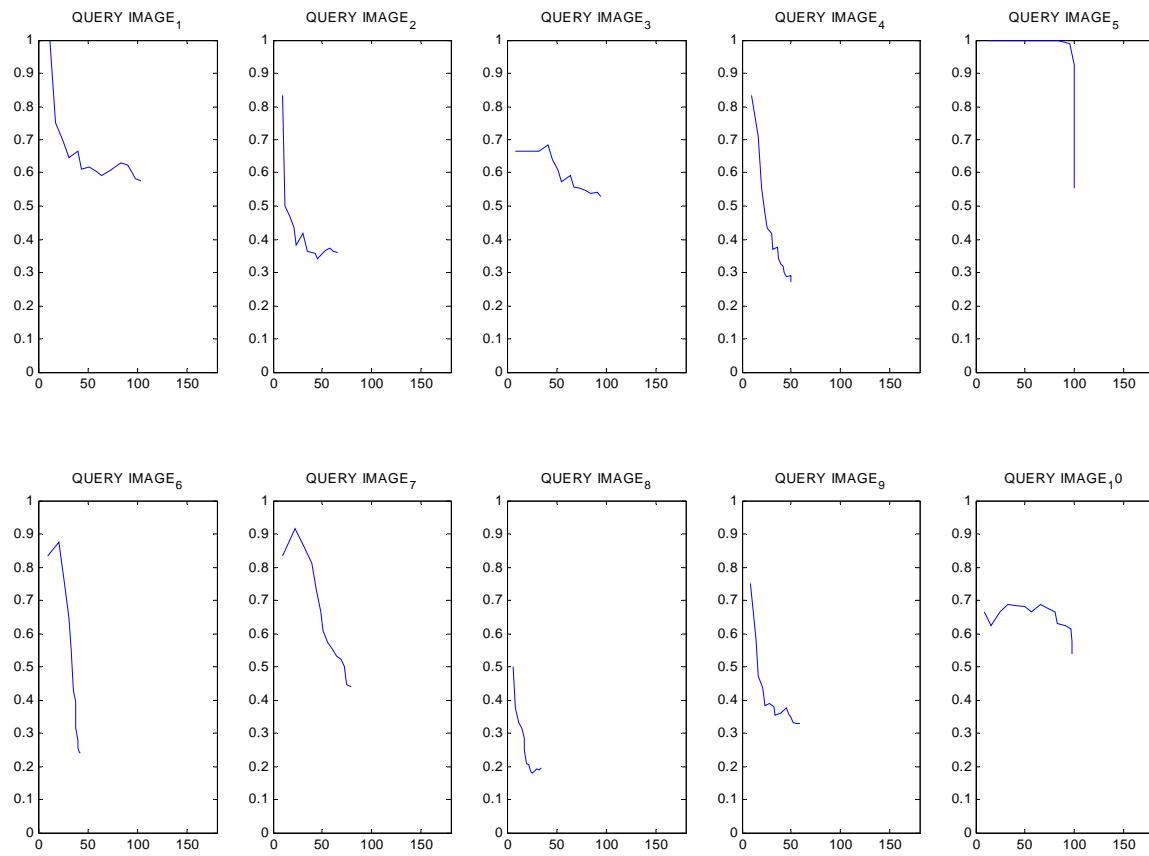


Fig. 48 Plot of cumulative precision versus retrieved relevant images (25 x 25 texture only) -PSU

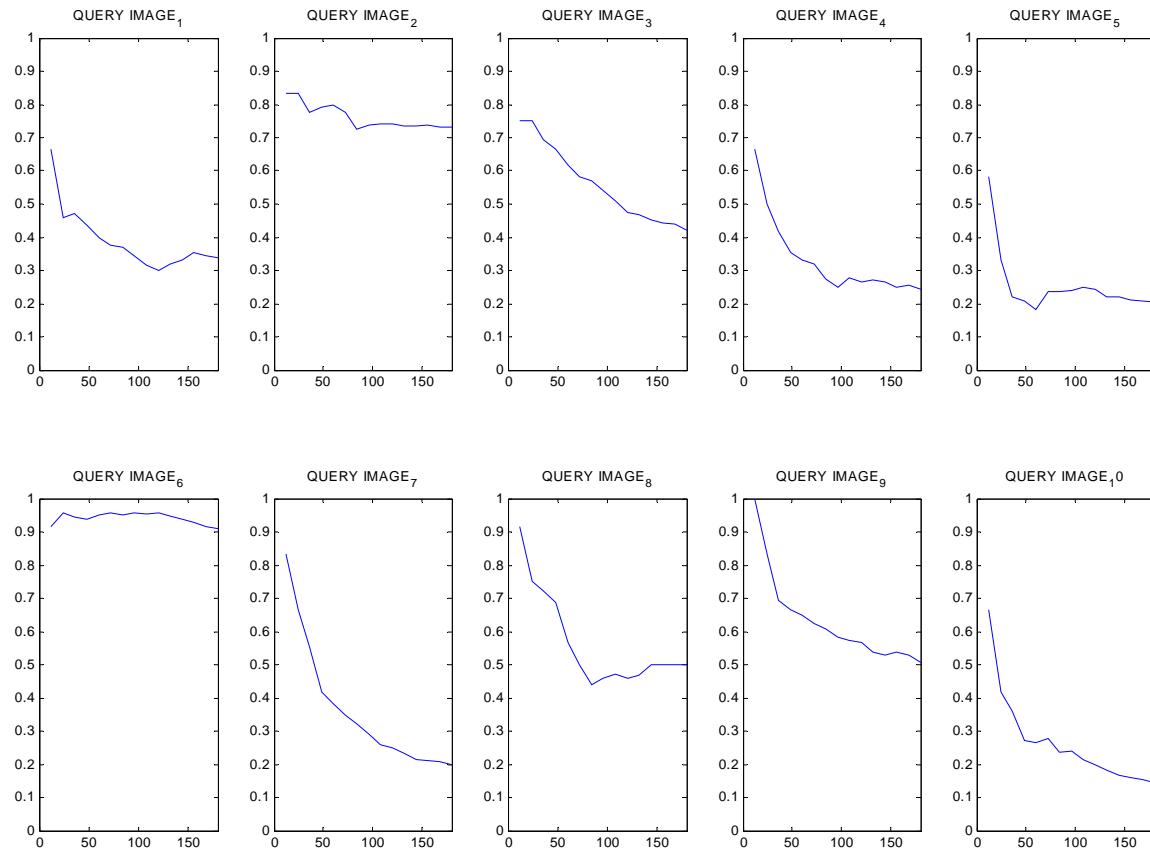


Fig. 49. Plot of cumulative precision versus number of images retrieved (25 x 25 texture only)–SU

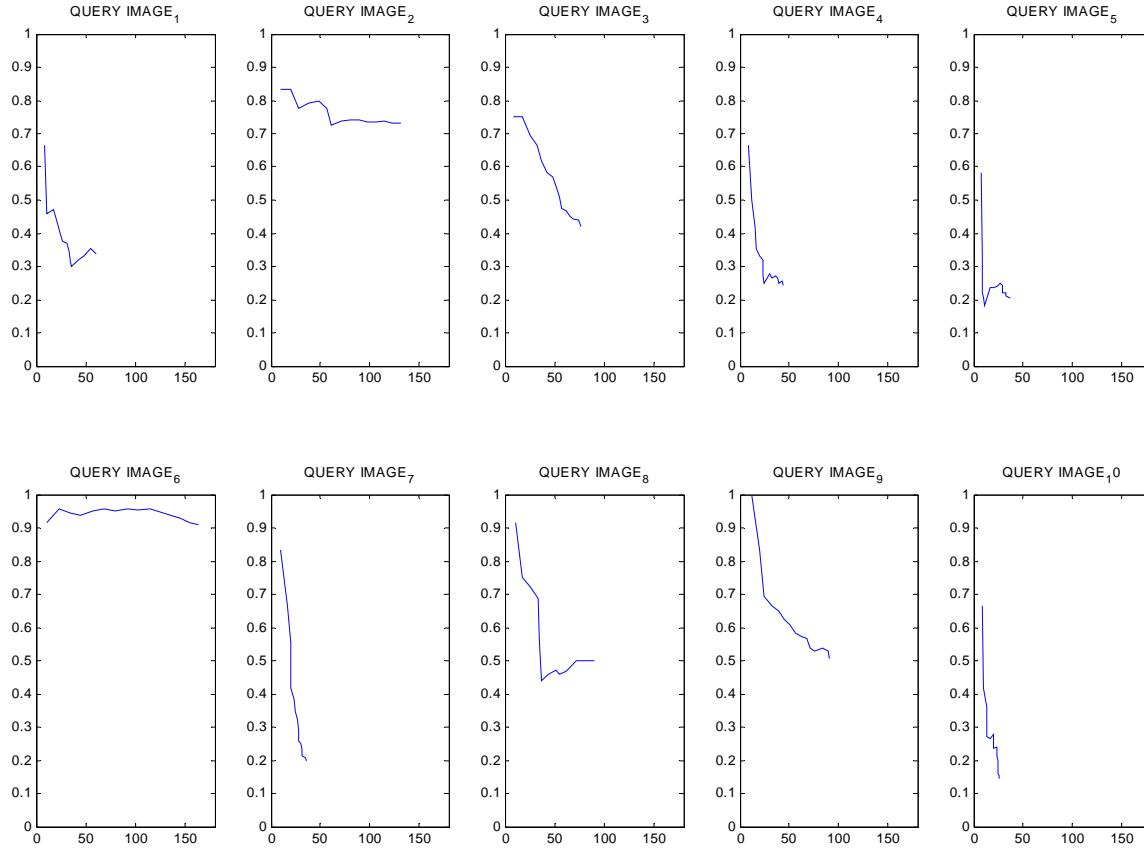


Fig. 50. Plot of cumulative precision versus retrieved relevant images (25 x 25 texture only) –SU

9.2 Discussion

In the first two image databases, each image was resized to 256 x 256 dimension because some of the database image dimensions are as much as 1200 x 1200. Resizing them below 256 x 256 may lead to pixelization of the images and thus give poor retrieval result. On the other hand the images need not be resized beyond 256 x 256 because the system is expected to be computationally time-efficient. In the case of the third image database, the average size of the images is 85 x 85. The maximum dimension of images in the database is 128 x 100, hence resizing each image to a 128 x 128 dimension was found suitable.

Implementation of the CBIR system is a painstaking process. The reason is that the Gabor filter dictionary adopted for the system design indicated the frequency of operation and the number of filters for optimal performance but did not have a ready-made answer for the filter grid size that gives optimal performance. This aspect was taken care of by computing texture feature of images in the three databases using Gabor filter of grid sizes 5 x 5, 15 x 15, 25 x 25, 35 x 35, 45 x 45 and 55 x 55. By using six different grid sizes of Gabor filter the coverage of the frequency spectrum will look like that in Fig. 51. The sketch shown in Fig. 51 is better frequency coverage than that of Fig. 39 because its coverage of the frequency spectrum is optimal.

In [30] it was recommended that the mean of the real part of the Gabor filter be set to zero so that the filter will show no response to constant intensities. When the mean of the real part of the filters are set to zero the filters are said to be ‘flagged’ down. ‘Unflagged’ implies that the mean parts are not set to zero.

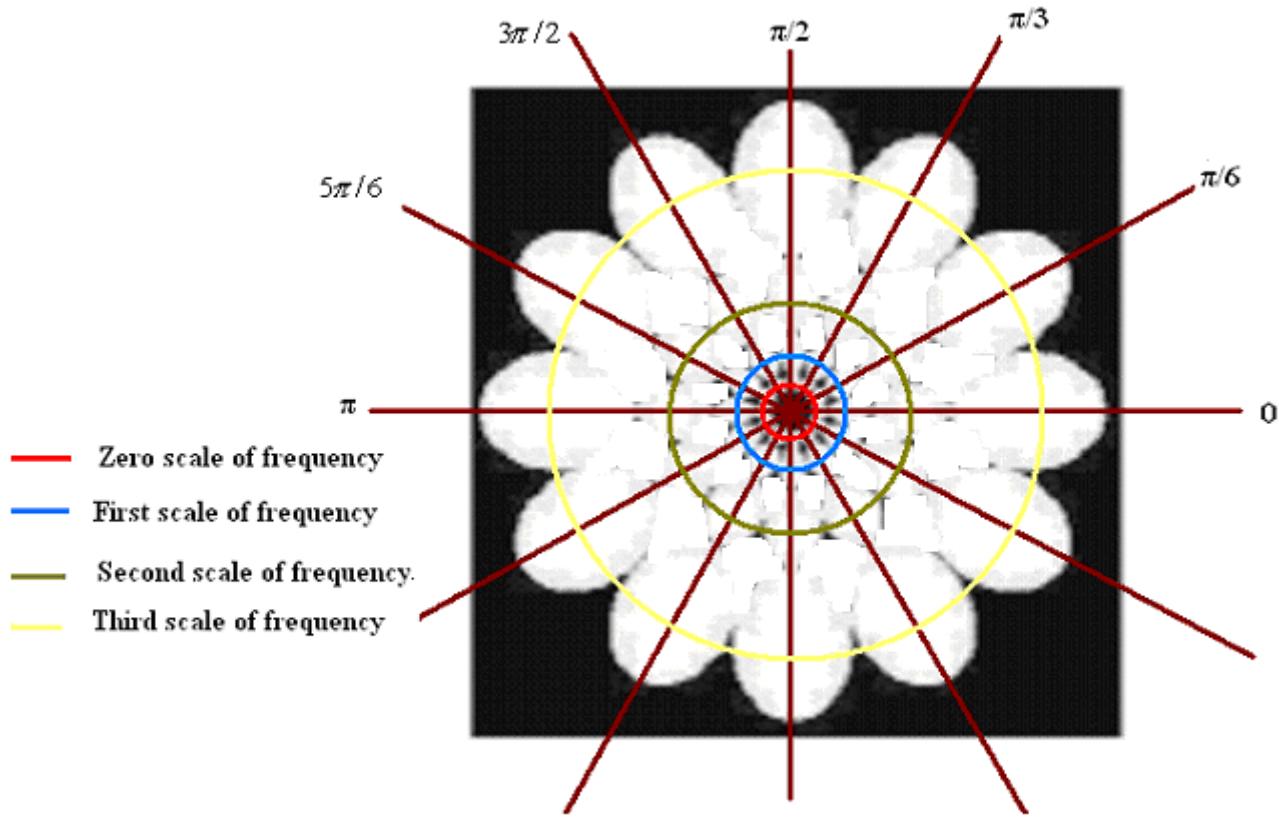


Fig 51. Sketch of 2-D frequency spectrum view using six grid sizes of Gabor filter bank designed for optimal coverage of frequency spectrum.

In the specific area where the author in [30] applied Gabor filter, the Gabor filter need not be flagged because the database images are mainly pure textures in plain background. Flagging is to ensure the filter captures only pure texture and not the plain background. In this thesis the reverse is the case. The databases are real-world scenes. Real-world scenes consist of appreciable size of different grey levels of constant intensities, but within the global image they constitute texture. There is need to capture these areas as texture within the global image, hence it is not necessary to flag down the filter channels.

During system design flagged and unflagged Gabor filters of the various sizes earlier mentioned were used for texture feature extraction. In the first two databases lower grid sizes of 3×3 , 5×5 , 7×7 and 9×9 showed comparably equal retrieval performance. At higher grid sizes flagged Gabor filters perform poorly while unflagged Gabor filter improved and showed better retrieval performance than at lower grid size.

The superior performance of the unflagged Gabor filters over the flagged Gabor filters can be explained using three key points.

First, the first two image databases consist mainly of real-world images. Close examination of pixels located in seemingly constant grey levels areas of real-world images show that there is reasonable variation in the pixel intensities, and therefore constitutes texture. Example is a natural scene like the sky as shown in Fig 3(4). It

seems like constant blue sky but close examination of its pixels show remarkable variation in intensities – light blue and deep blue. Flagging the filter will prevent the filter from capturing these slowly varying grey levels.

Second, by the design of the Gabor filter a larger grid size implies high spectral selectivity as explained in section 4.11. Thus a larger Gabor grid is capable to capture slowly varying levels than a lower grid size filter. This assertion is demonstrated using only texture feature and Gabor filter sizes of 5 x 5, 15 x 15, 25 x 25, 35 x 35, 45 x 45 and 55 x 55, in Fig. 52 (a) to Fig. 52(f), for university of Washington image database, Fig. 53(a) to Fig. 53(f) for Pennsylvania state university image database, and Fig. 54(a) to Fig. 54(f) for Stanford state university image database. A cursory and perceptual view of the images show that retrieval performance of the system, in general, was boosted by the increasing Gabor filter size. ***The implication of this assertion is that Gabor filter sizes can play the same role as weighting of feature vectors for improved retrieval performance.***

Third, It is necessary to mention that the dimension of 256 x 256, of each image in these two databases is quite large enough for variations in pixel intensities to be detected.

Experimental result during design process show that the flagged Gabor filter was most suitable for retrieval in the case of the third image database because it outperform the unflag filter. The reason can be said to be how the database was modelled. The third database unlike the first two databases consists of low resolution images. Moreover it is a mix of artificial images and real-world images. Most of the artificial images are on plain background. The average size of the images is 85 x 85. Resizing the image to 128 x 128 may have resulted in pixelization and this size is too small to detect slowly varying grey levels.

In [30] it was recommended that for texture feature vector computation, the standard deviation of the database feature vectors be normalized. In this work I did not see the need for normalization because non-normalization did not affect the retrieval performance of the system.

In this thesis, the normalization that was carried out was to rescale each of the feature distances for texture, shape and spatial information to lie between 0 and 1. This is to adapt the feature vectors to weight during retrieval.

Retrieval using only texture feature showed high and reliable precision-recall compared to shape and spatial information features. This did not come as a surprise because most of the database images are real world and complex images that are better described by texture feature. Despite the high performance of the system using only texture for retrieval, there are isolated cases as shown in Fig. 55(a) to Fig. 55(c) where texture retrieval failed and the system recovered its retrieval efficiency with shape and weighted features. Since no particular feature such as texture, as stated in section 3.2, can completely describe an image, texture feature alone cannot solely retrieve all relevant images in response to a query. Test results shown in the appendix attest to this fact. In effect the three features complement each other, and in particular shape and spatial information complement texture retrieval efficiency to improve the CBIR system's overall efficiency.

In this thesis I tried to apply the Gabor tuning properties as applicable in physiology to retrieval, and strictly follow the principle of scale-space theory by recognizing the filer (probe) size as an important variable. Gabor texture features computed for grid sizes of 5 x 5, 15 x 15, 25 x 25, 35 x 35, 45 x 45 and 55 x 55 were inserted into the system as texture feature vectors and coded with the filter size. During operation the user selects a particular filter size as well as its corresponding texture feature. Since different filter size have different frequency selectivity and images in the database can be said to contain different localized frequencies, it implies that each Gabor filter size is particularly suited to capture particular set of localized frequency-images in a diverse database. This assertion can be deduced from results shown in Fig. 52, Fig.53 and Fig. 54. The system is

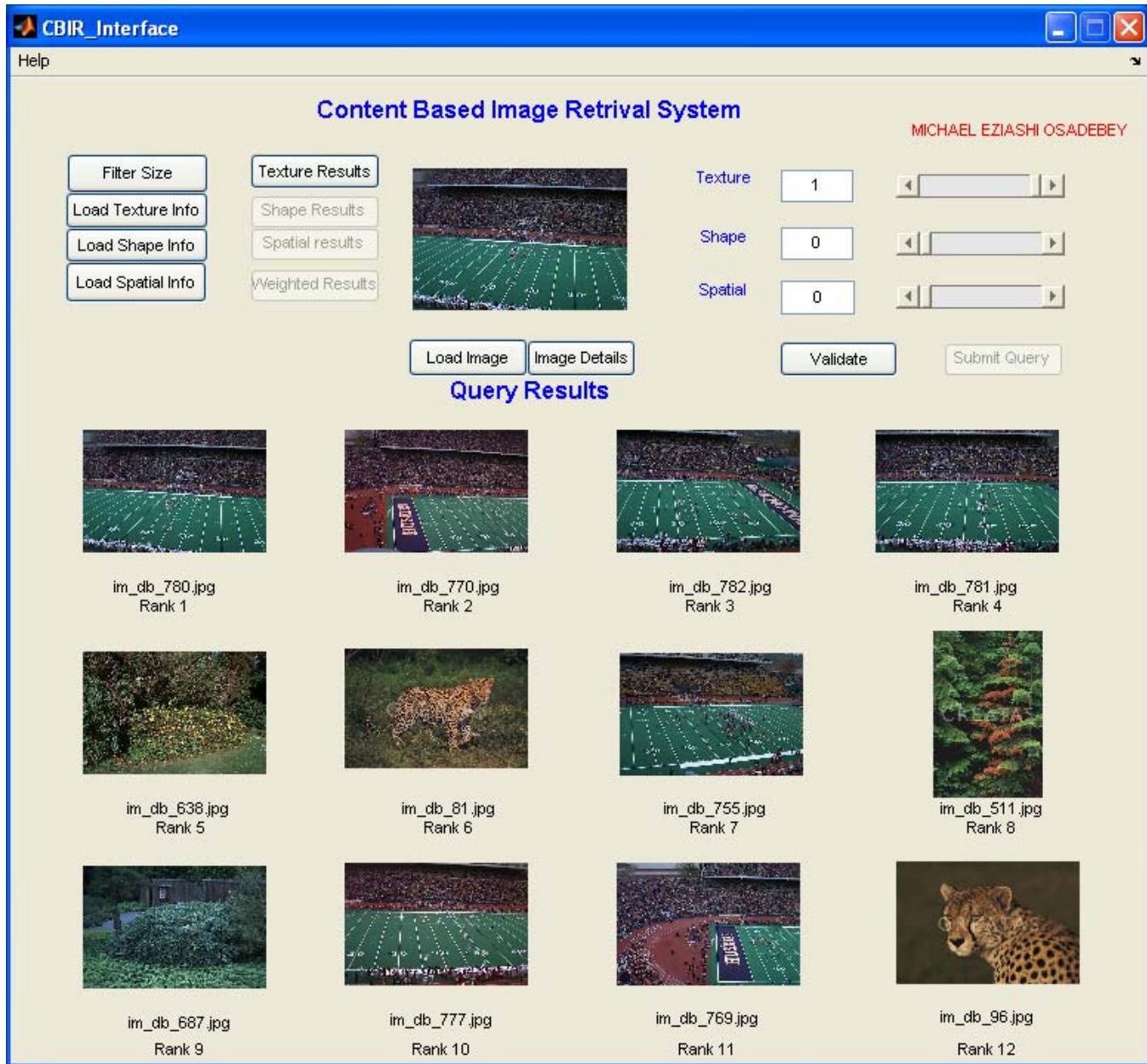


Fig. 52(a). Retrieval using only texture. Gabor filter size is 5 x 5. Precision = 7/12. Query image file name is im_db_780. Database is UW.

designed such that users can choose any of the six different Gabor filter sizes in conjunction with shape and spatial information feature for retrieval. By so doing Gabor frequency-tuning property is explored to maximum advantage.

The texture features derived from the Gabor filter sizes, and hence the output of the system can be combined or weighted using appropriate mathematical function, just as feature vectors are weighted, to produce an integrated CBIR system that can retrieve images based on semantic attributes.

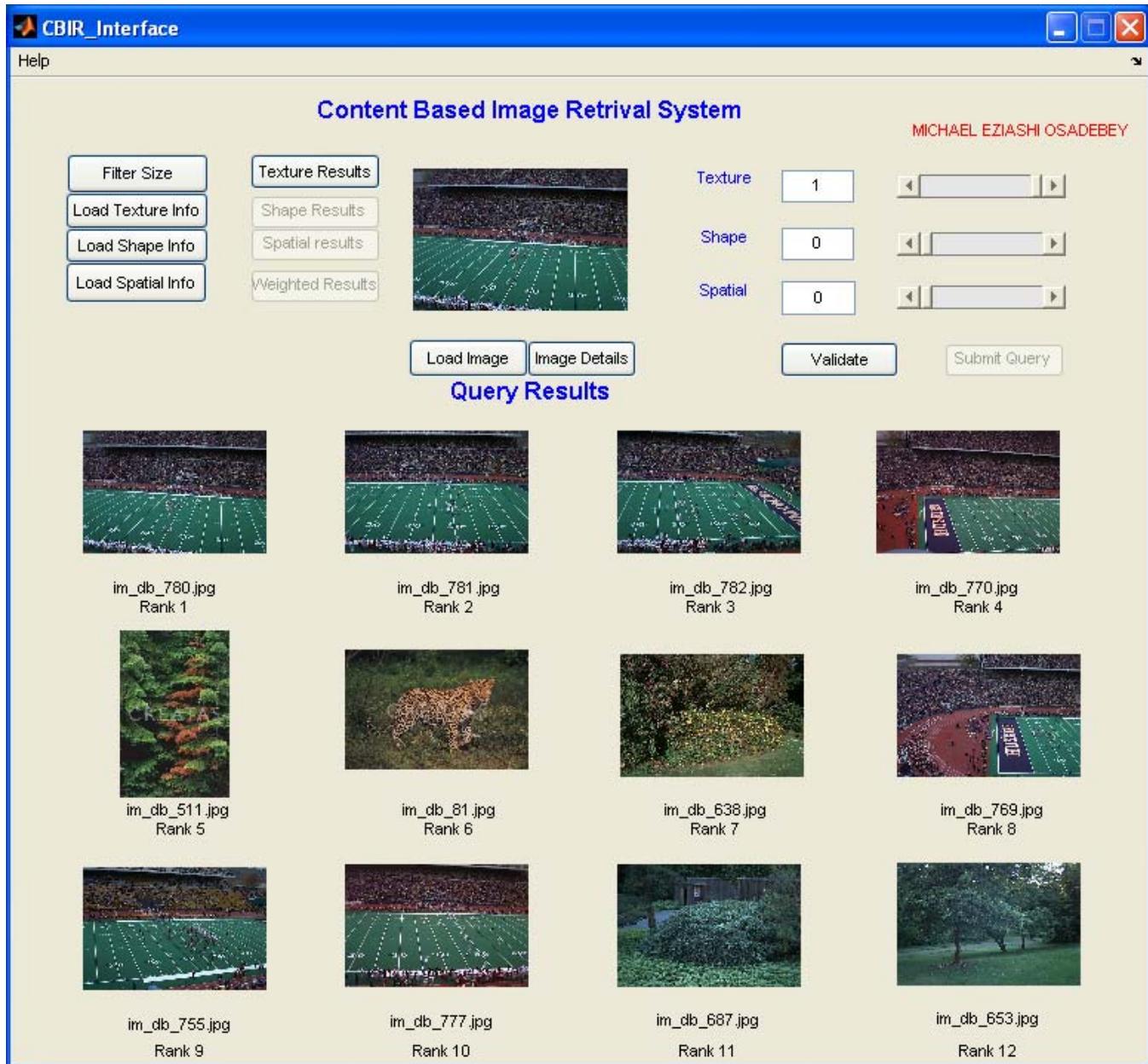


Fig. 52(b). Retrieval using only texture. Gabor filter size is 15 x 15. Precision = 7/12. Query image file name is im_db_780. Database is UW.

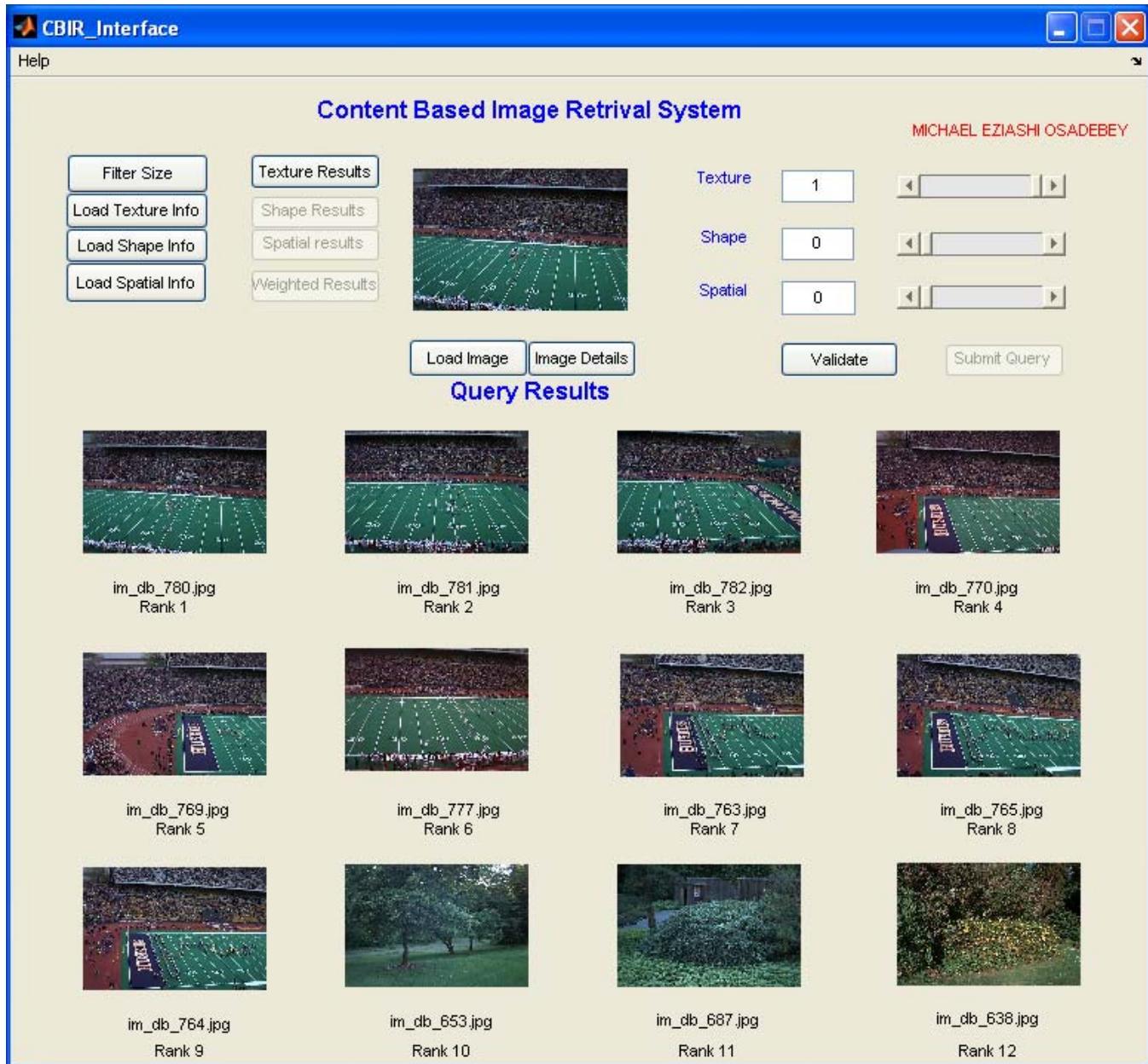


Fig. 52(c). Retrieval using only texture. Gabor filter size is 25 x 25. Precision = 9/12. Query image file name is im_db_780. Database is UW.

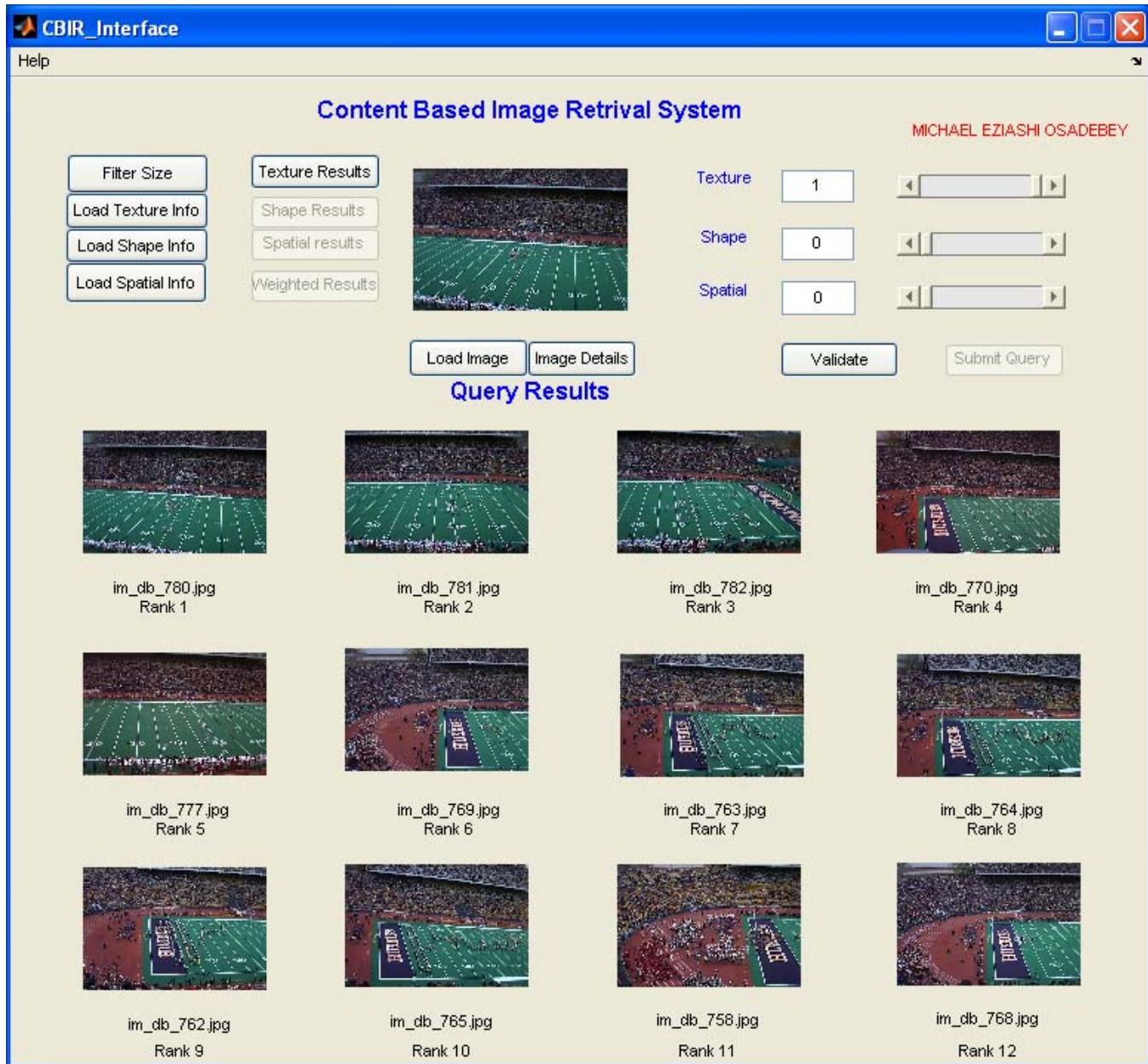


Fig. 52(d). Retrieval using only texture. Gabor filter size is 35 x 35. Precision = 12/12. Query image file name is im_db_780. Database is UW.

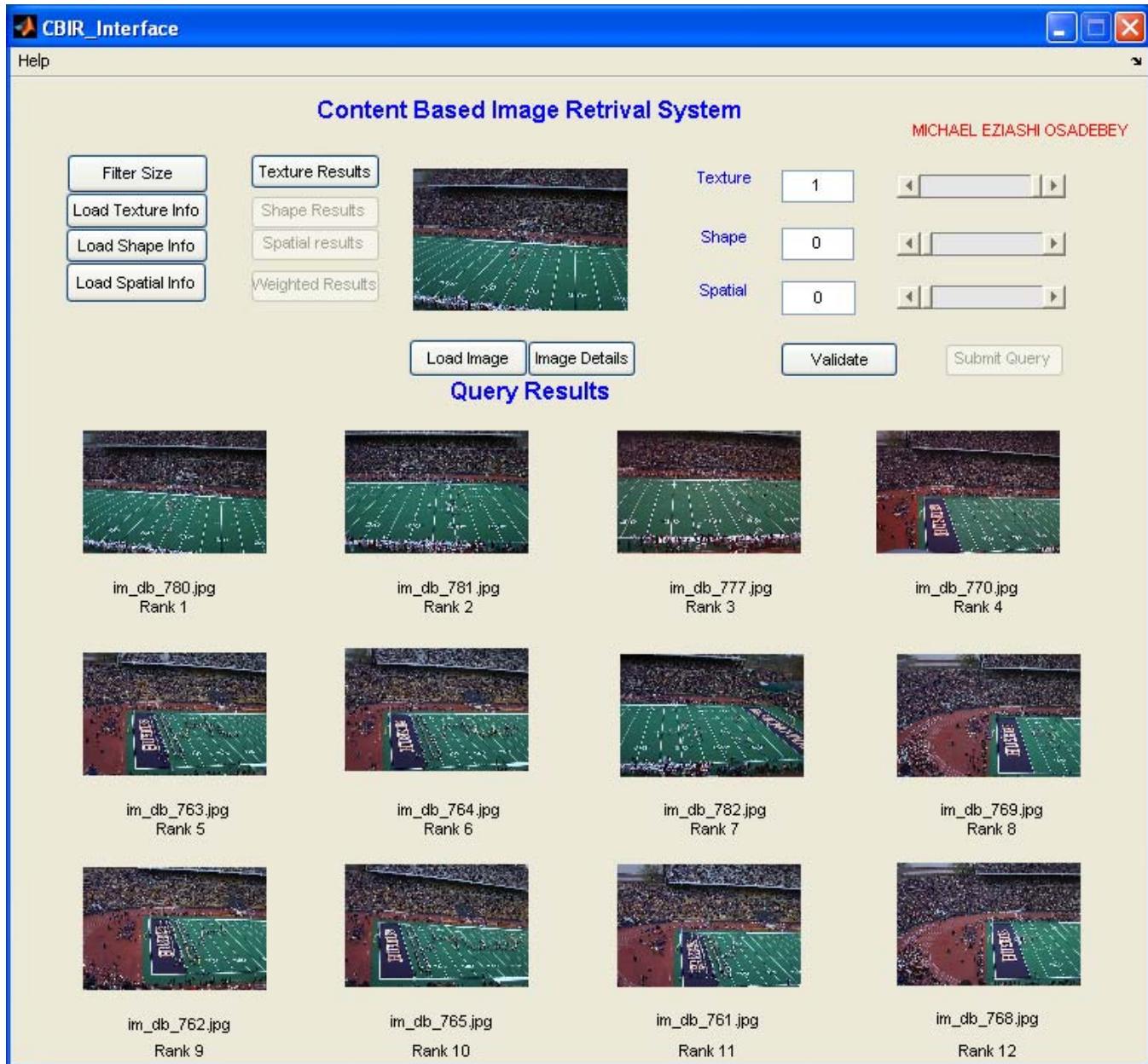


Fig. 52(e). Retrieval using only texture. Gabor filter size is 45 x 45. Precision = 12/12. Query image file name is im_db_780. Database is UW.

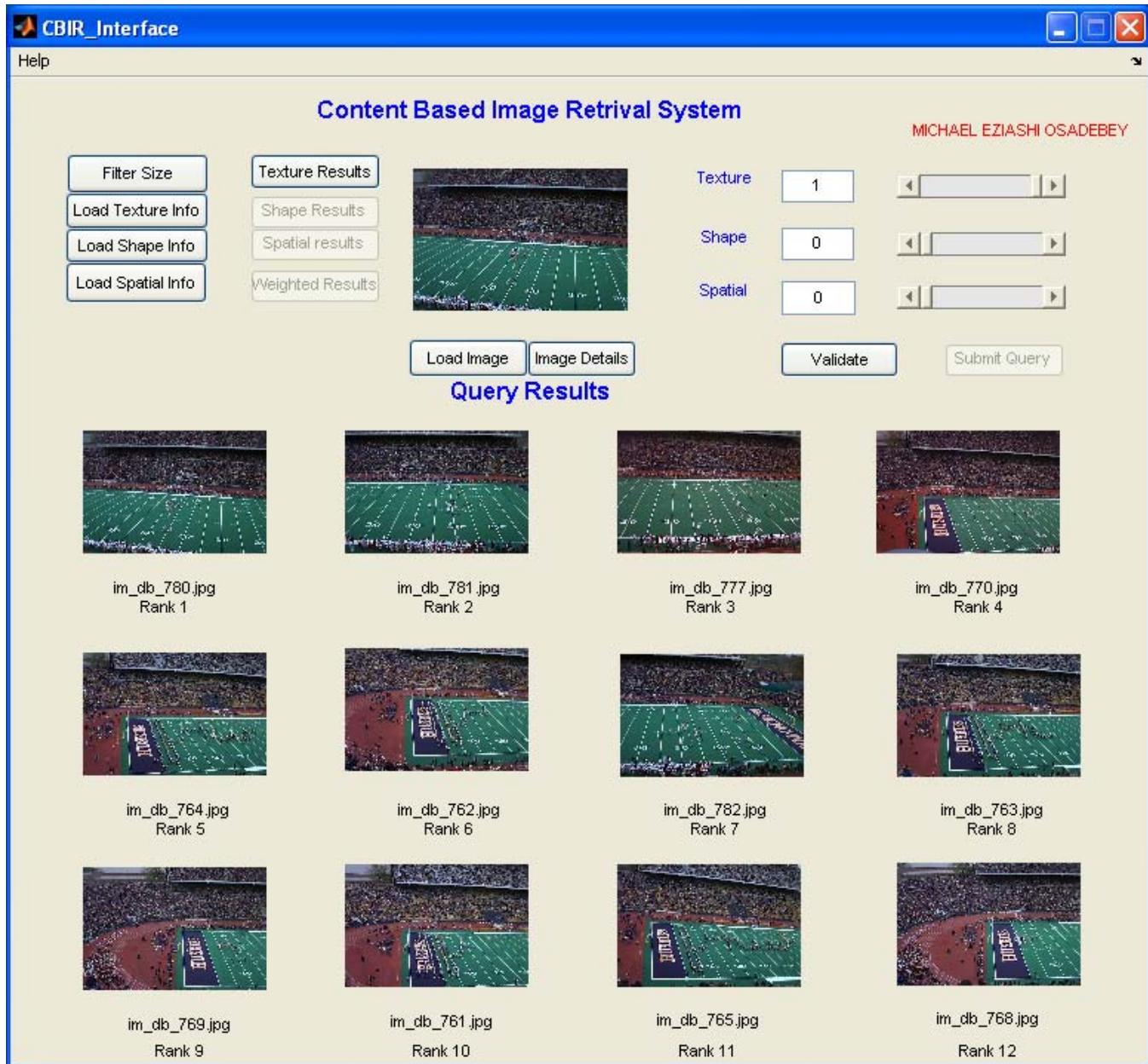


Fig. 52(f). Retrieval using only texture. Gabor filter size is 55 x 55. Precision = 12/12. Query image file name is im_db_780. Database is UW.

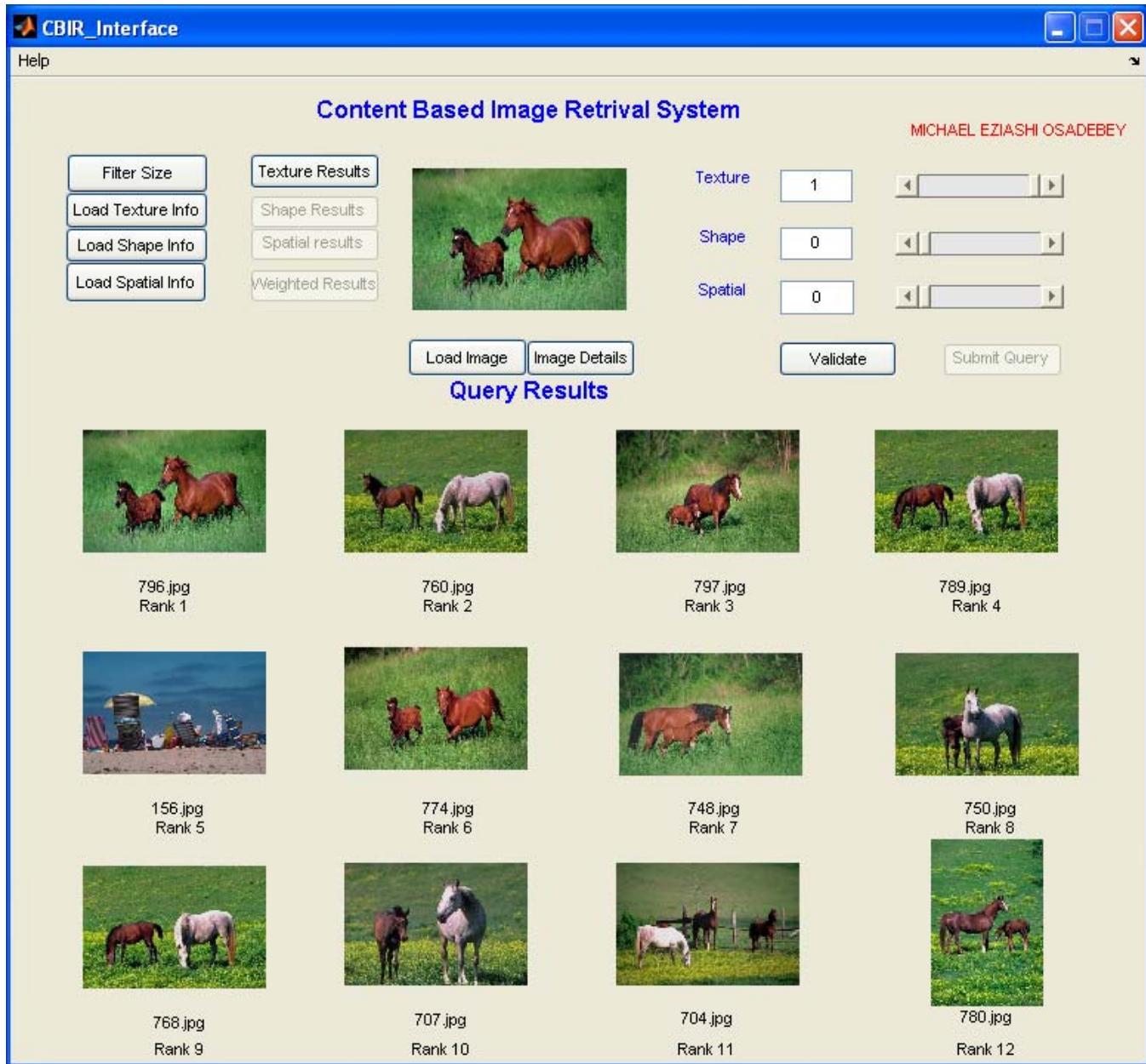


Fig. 53(a). Retrieval using only texture. Gabor filter size is 5 x 5. Precision = 11/12. Query image file name is 796. Database is PSU.

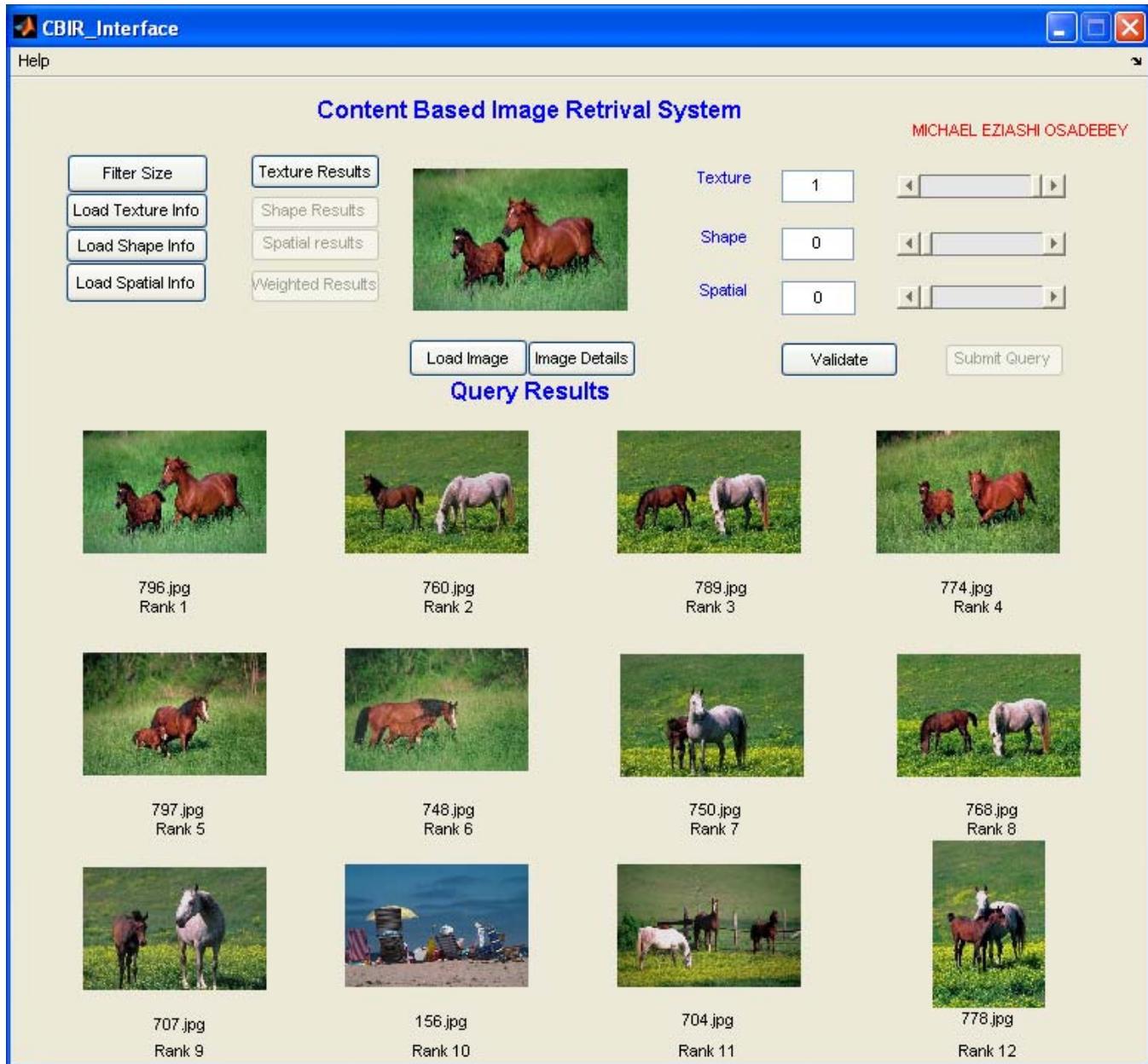


Fig. 53(b). Retrieval using only texture. Gabor filter size is 15 x 15. Precision = 11/12. Query image file name is 796. Database is PSU.

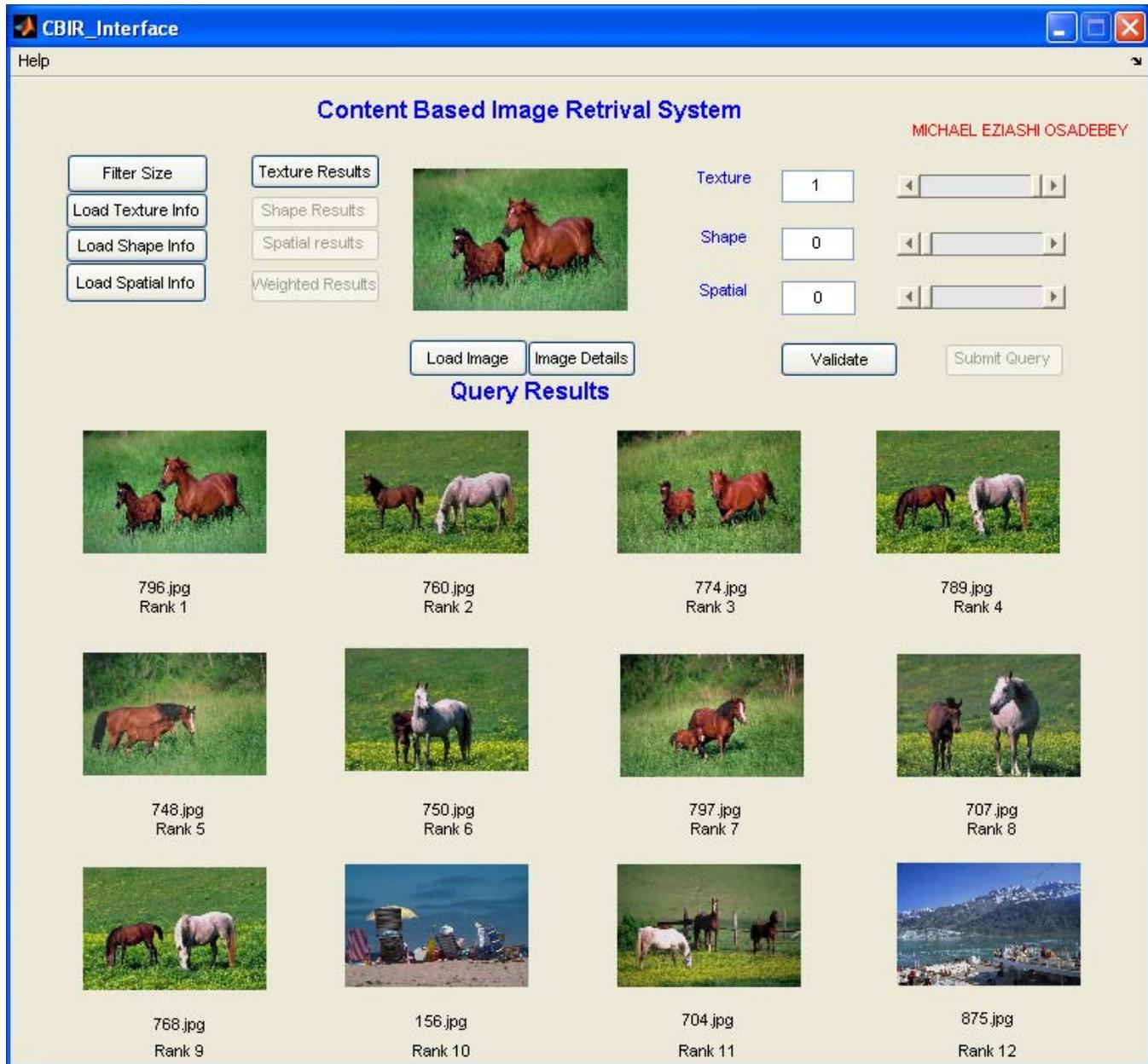


Fig. 53(c). Retrieval using only texture. Gabor filter size is 25 x 25. Precision = 10/12. Query image file name is 796. Database is PSU.

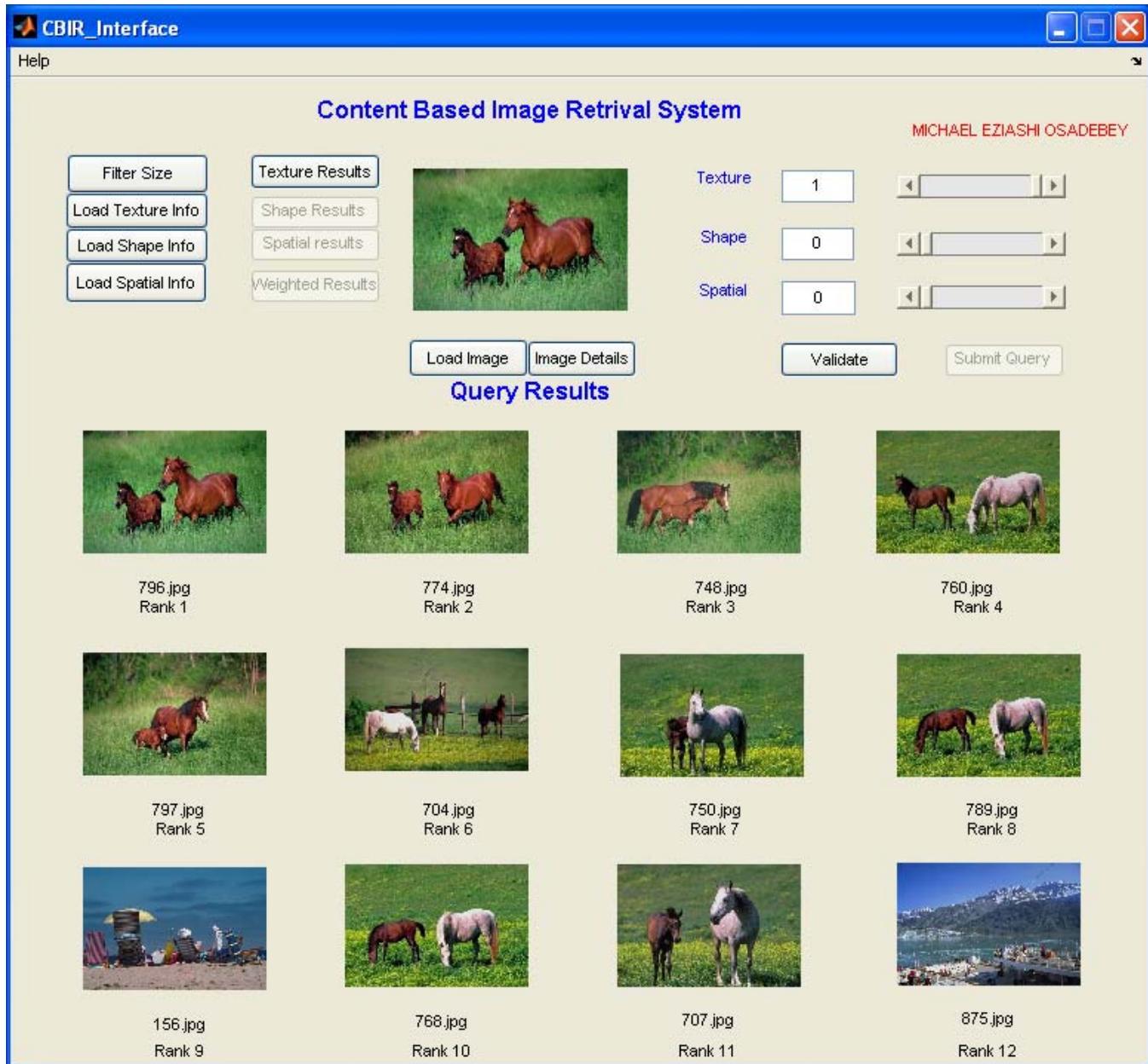


Fig. 53(d). Retrieval using only texture. Gabor filter size is 35 x 35. Precision = 10/12. Query image file name is 796. Database is PSU.

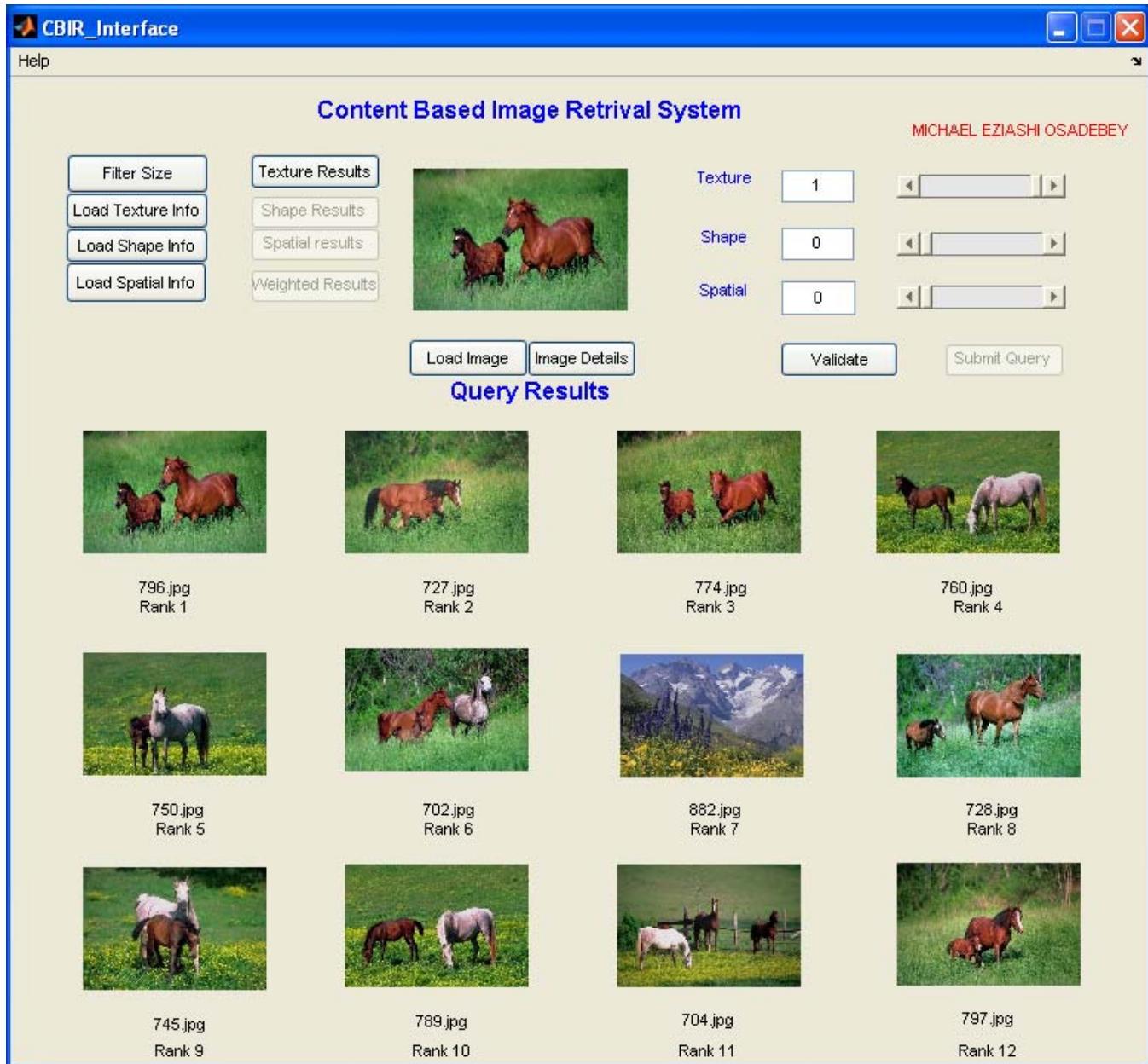


Fig. 53(e). Retrieval using only texture. Gabor filter size is 45 x 45. Precision = 11/12. Query image file name is 796. Database is PSU.

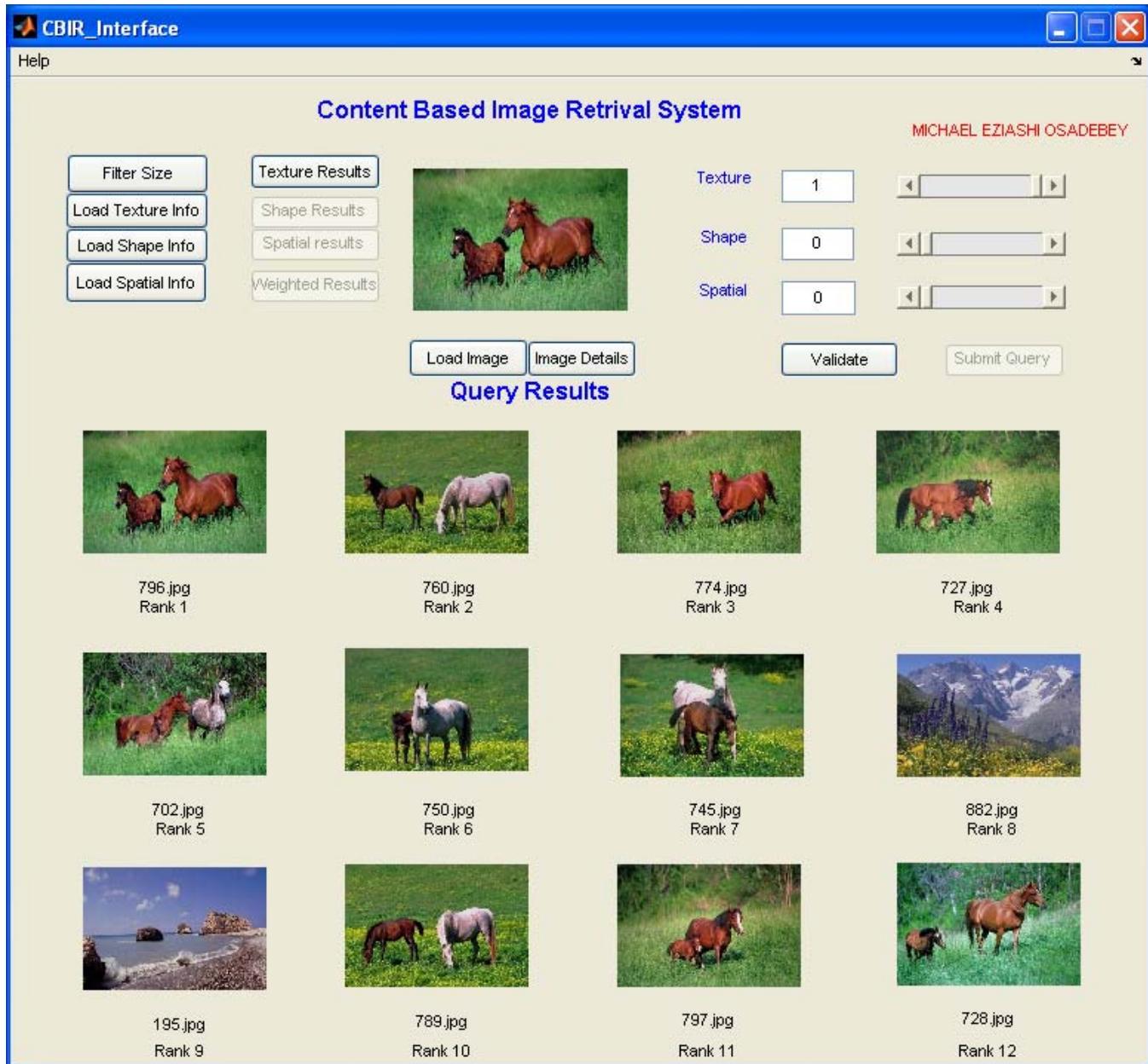


Fig. 53(f). Retrieval using only texture. Gabor filter size is 55 x 55. Precision = 10/12. Query image file name is 796. Database is PSU.

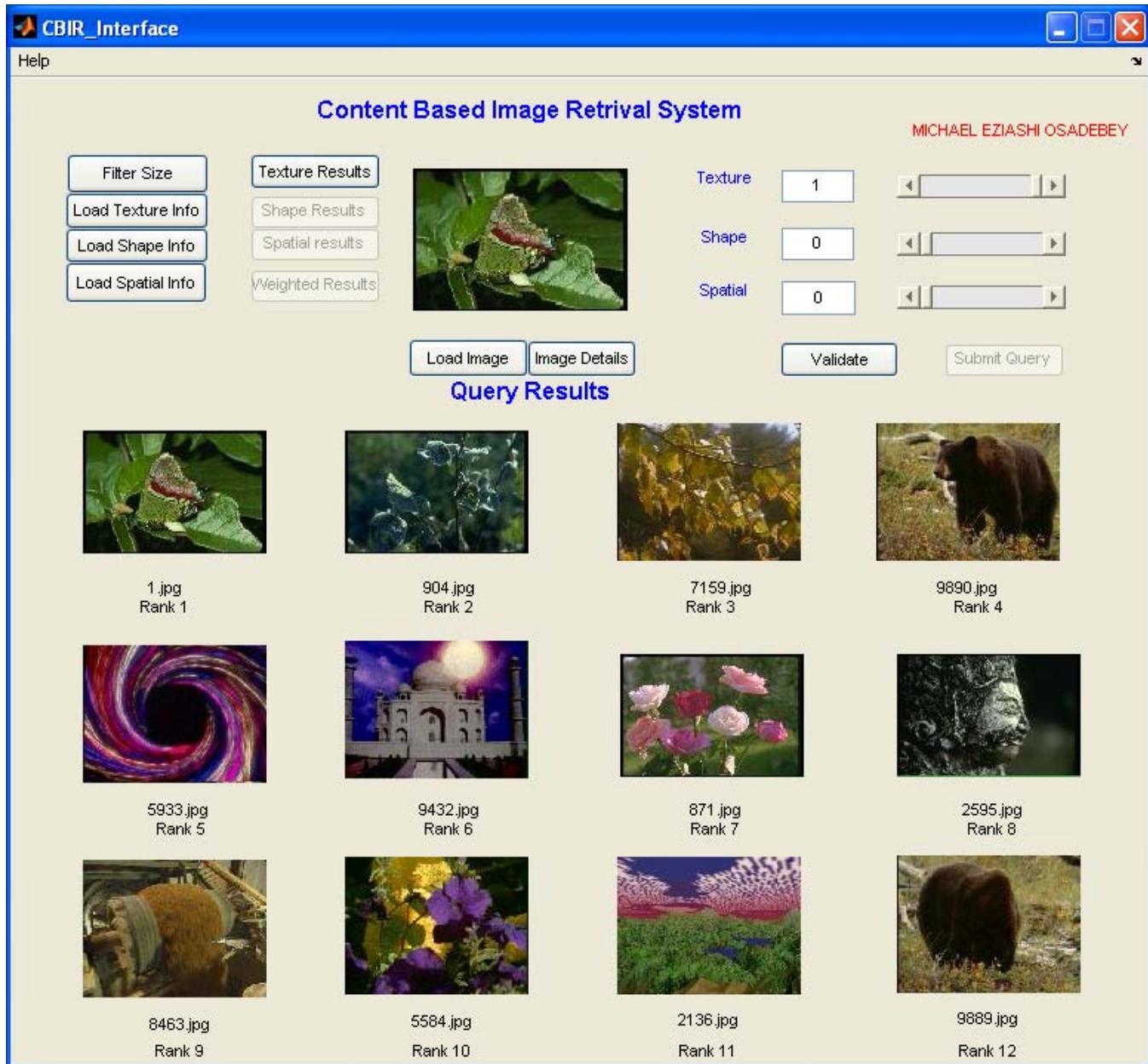


Fig. 54(a). Retrieval using only texture. Gabor filter size is 5 x 5. Precision = 4/12. Query image file name is 1. Database is SU.

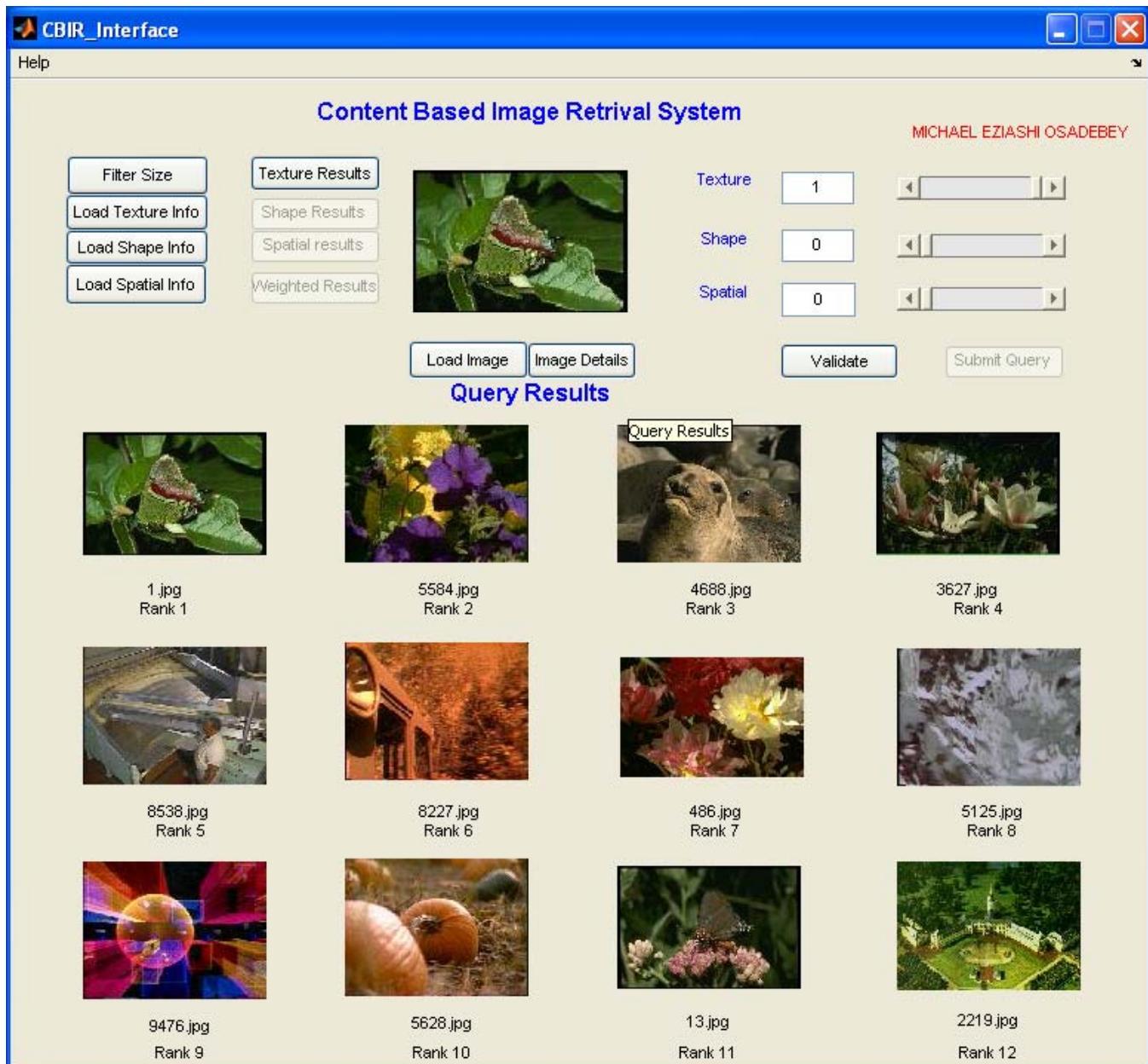


Fig. 54(b). Retrieval using only texture. Gabor filter size is 15 x 15. Precision = 5/12. Query image file name is 1. Database is SU

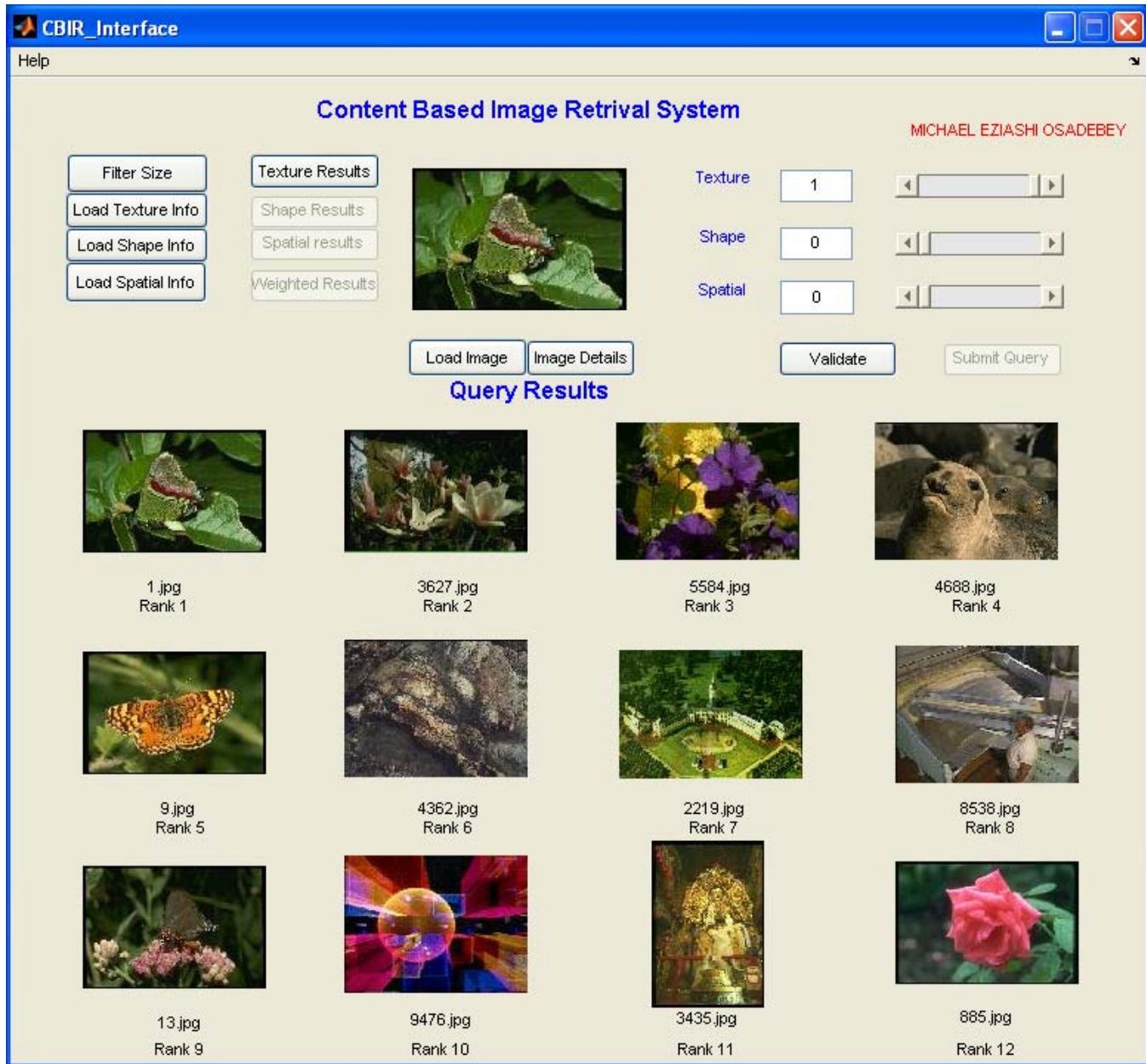


Fig. 54(c). Retrieval using only texture. Gabor filter size is 25 x 25. Precision = 6/12. Query image file name is 1. Database is SU

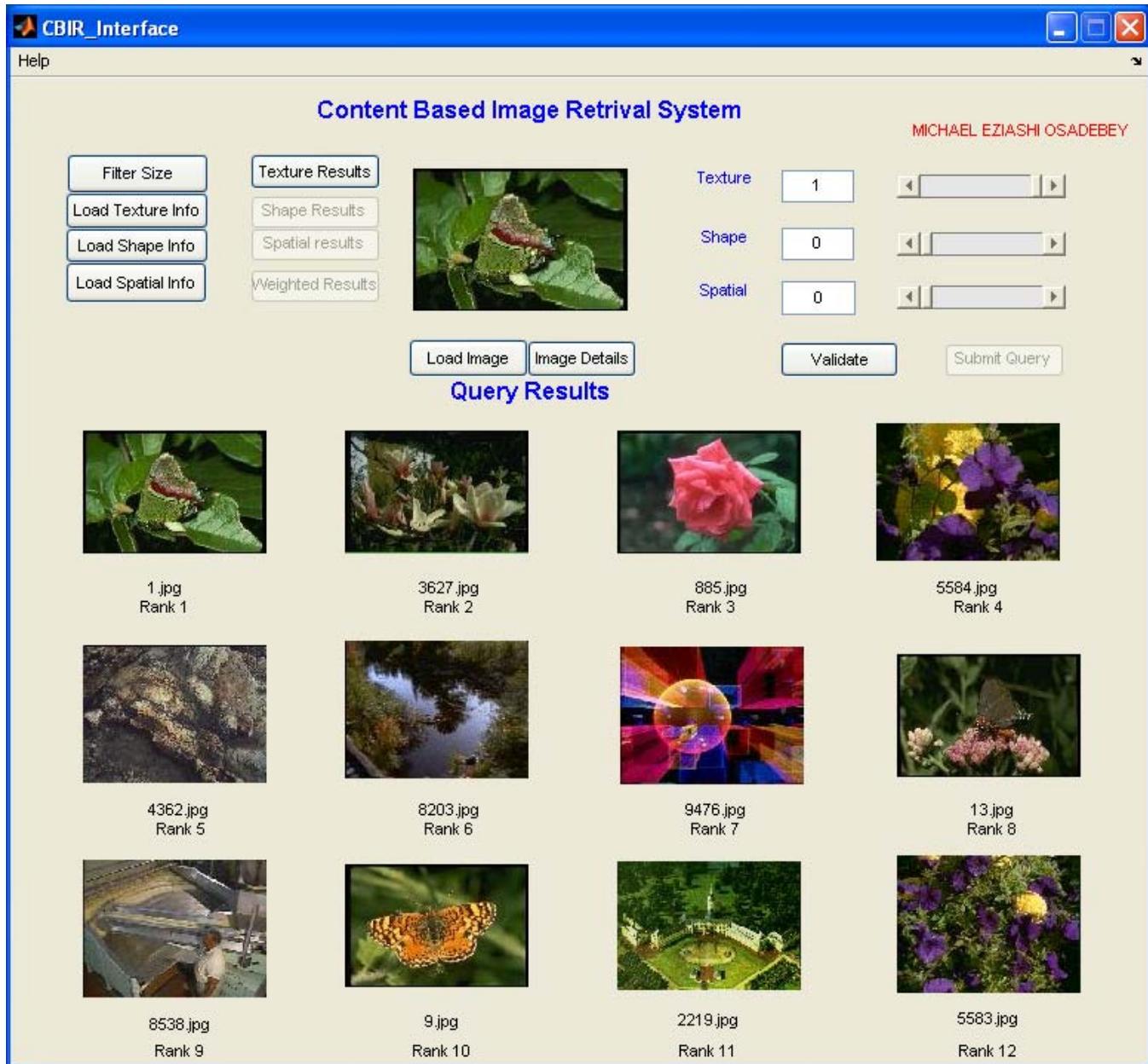


Fig. 54(d). Retrieval using only texture. Gabor filter size is 35 x 35. Precision = 7/12. Query image file name is 1. Database is SU

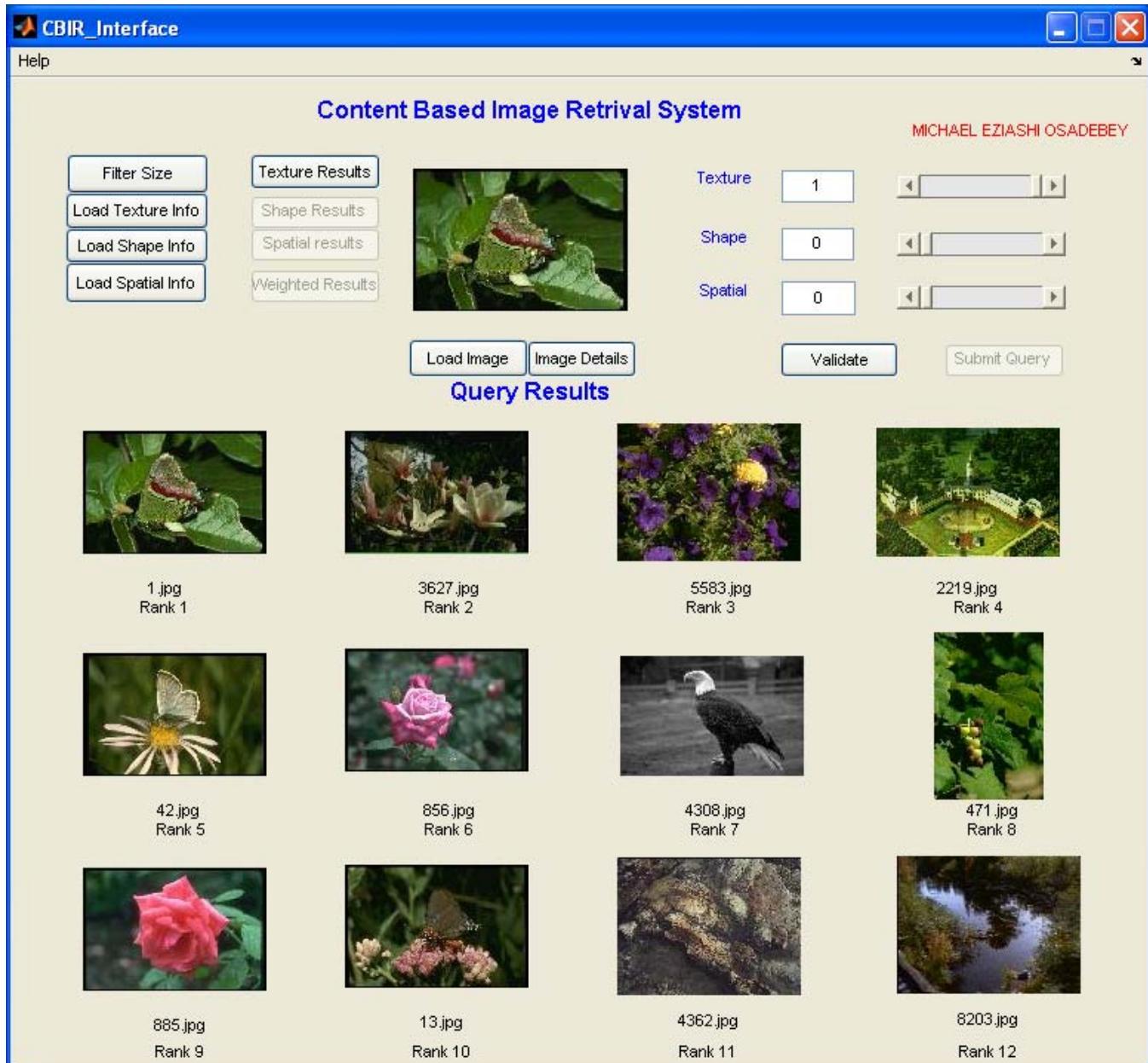


Fig. 54(e). Retrieval using only texture. Gabor filter size is 45 x 45. Precision = 8/12. Query image file name is 1. Database is SU

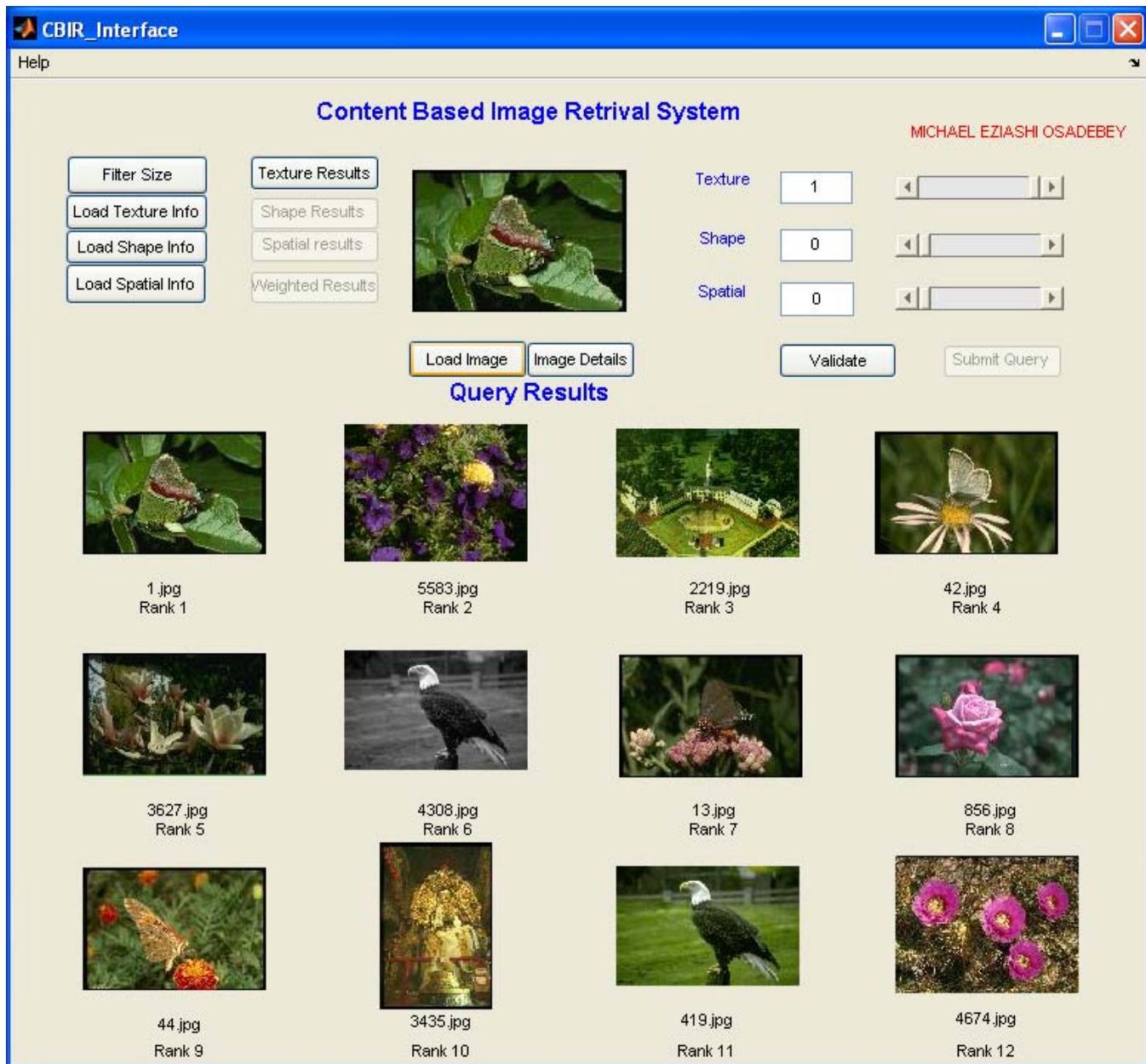


Fig. 54(f). Retrieval using only texture. Gabor filter size is 55 x 55. Precision = 10/12. Query image file name is 1. Database is SU

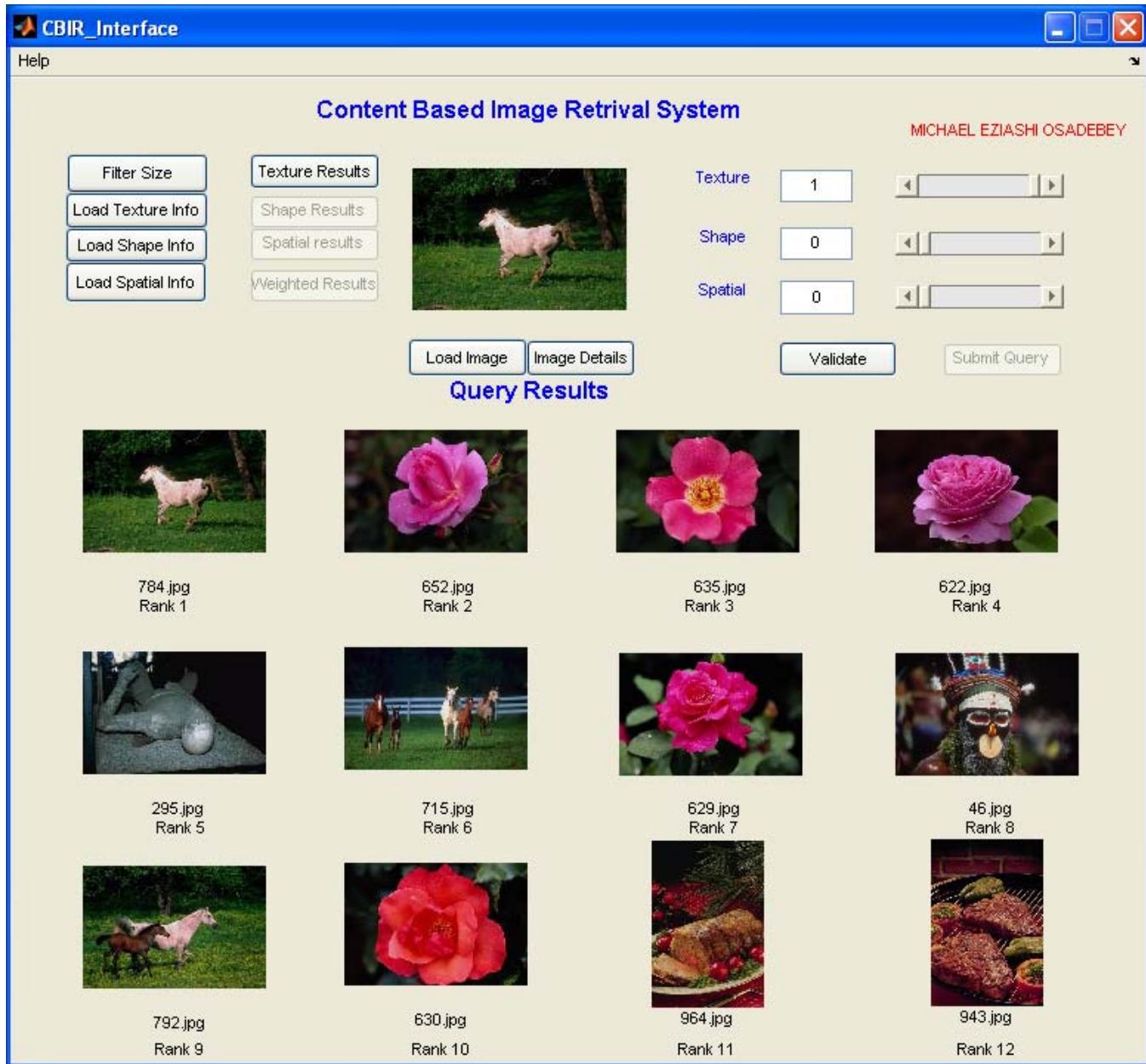


Fig. 55(a). Retrieval using only texture. Gabor filter size is 5 x 5. Precision = 3/12. Query image file name is 784. Database is SU

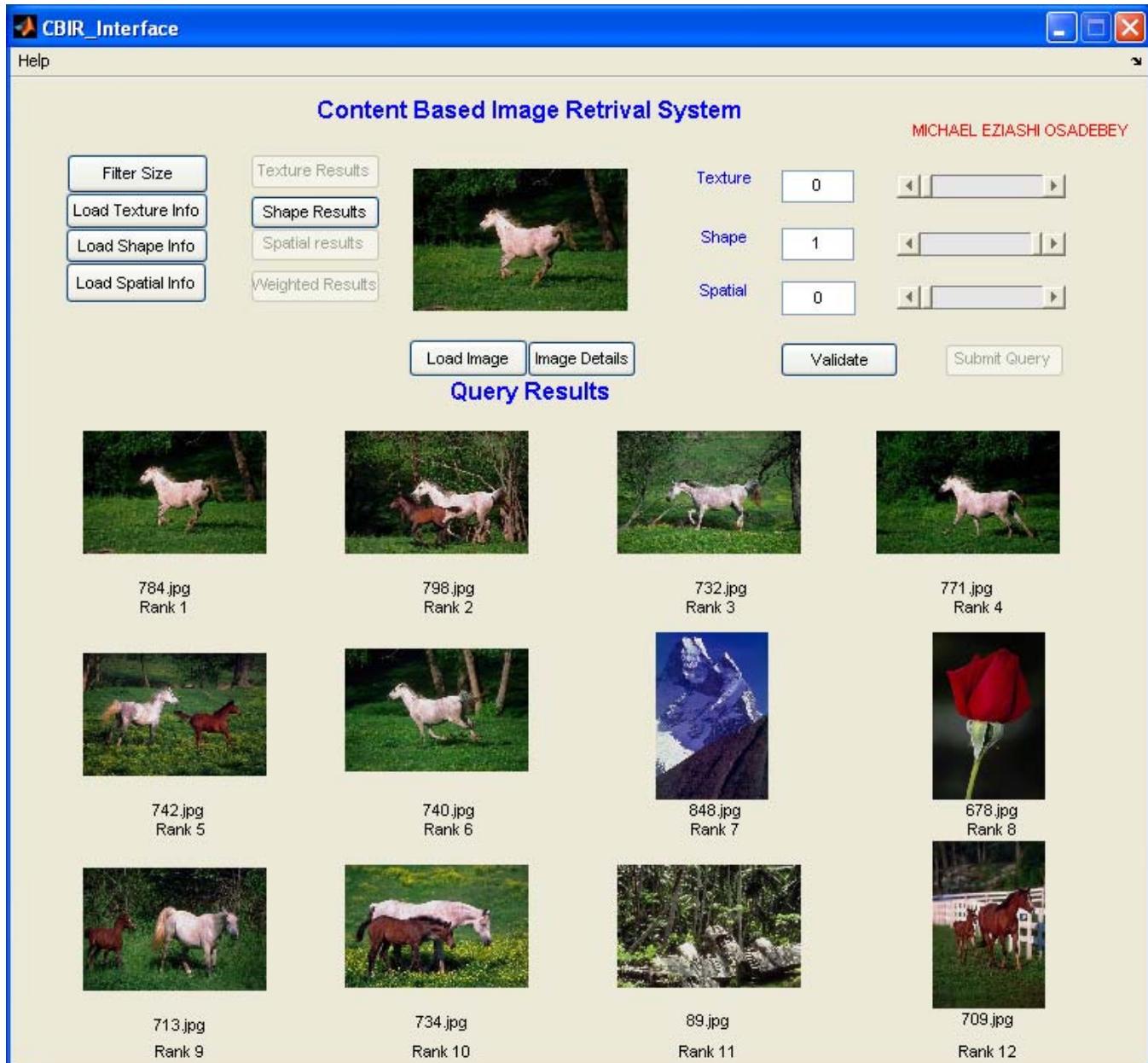


Fig. 55(b). Retrieval using only shape. Gabor filter size is 5 x 5. Precision = 8/12. Query image file name is 784. Database is SU

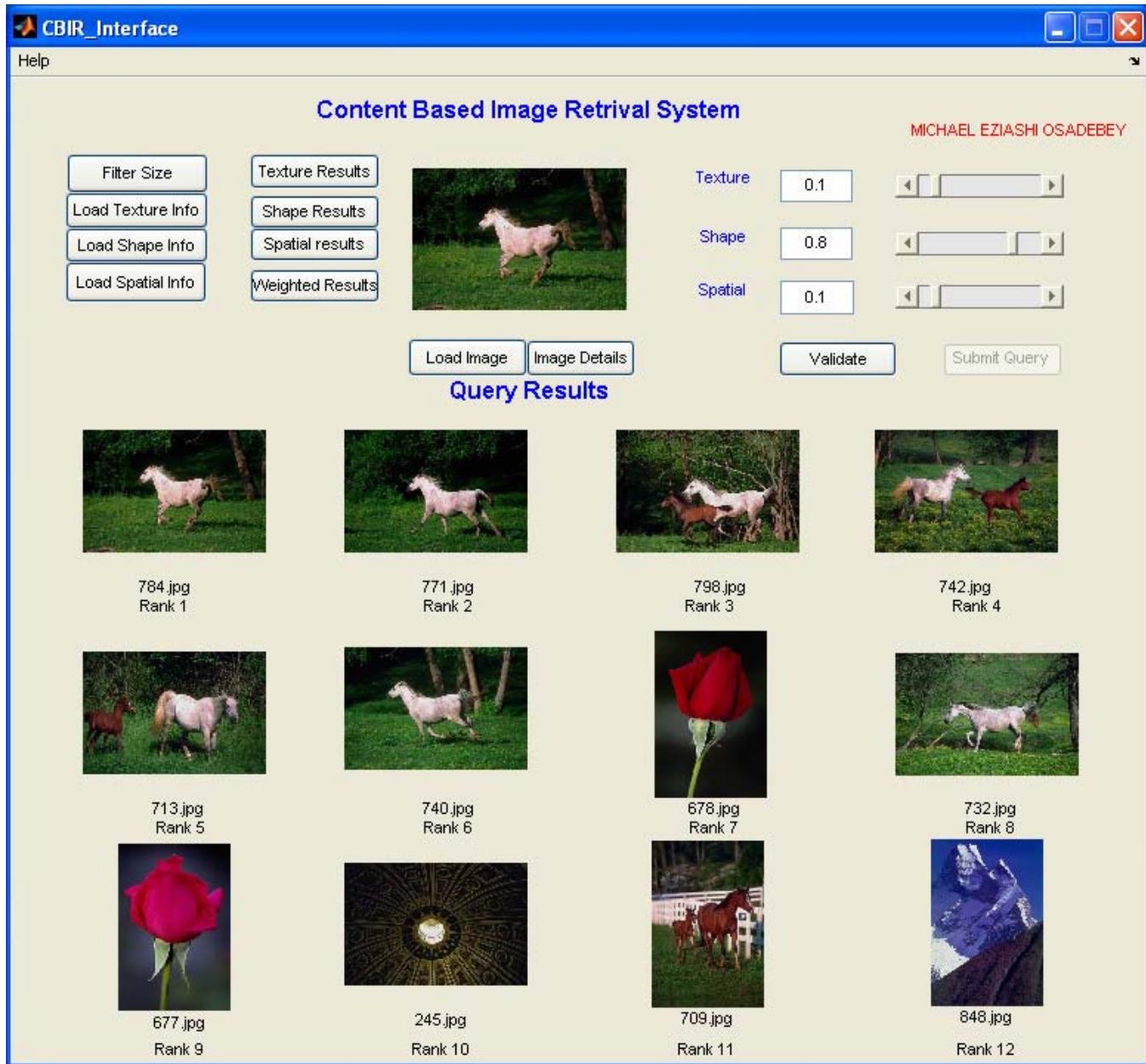


Fig. 55(c). Retrieval using weighted features texture (0.1), shape (0.8) and spatial (0.1). Gabor filter size is 5 x 5. Precision = 8/12. Query image file name is 784. Database is SU

9.3 Future direction

The following are suggested direction for future work

- Reconstruct the texture feature algorithm to extract translation, rotation and scale invariant texture features. The use of circular Gabor filter [36], [37], [38] and [39] may be useful in this direction.
- Since the system gave encouraging result in retrieving objects and scenes the interface may be redesigned to accept semantic in addition to content-based queries.
- The algorithm needs reprogramming to achieve near real-time retrieval process for higher Gabor filter sizes.
- The system may need to be incorporated with multiple algorithms that allow the user to freely choose algorithm most suitable for a particular query image. This is necessary because since the database is heterogeneous a particular algorithm will not achieve all-round efficiency.
- The texture features derived from the Gabor filter sizes can be combined using appropriate mathematical model to produce an integrated system that can retrieve images based on semantic attributes.

9.4 Conclusion

Current and most CBIR systems focus on colour as primary feature for retrieval. In this thesis colour is de-emphasized. Focus is on texture as primary feature. Shape and spatial information were secondary features.

The space/spatial-frequency tuning property of the neurons in the visual cortex as observed in physiological study is replayed by Gabor filters in images in diverse databases.

Texture features derived from six grid sizes of independent and different Gabor filter banks were incorporated into the CBIR system by taking advantage of the fact that each grid size of filter is suited to capture particular set of localized frequency-images in diverse database. This design enable the Gabor filter to optimally cover the frequency space, and gives the system the artificial intelligence to ‘scroll’ locally and globally through the database and retrieve images based on high level features.

It is shown that Gabor filters can replay their efficient texture feature extraction in pure texture images, in complex and real-world images, because these images, though constituted by constant grey levels, the various constant grey levels within the global image constitute texture that can be captured by the tuneable characteristics of Gabor filters.

An integrated but simple, robust, flexible and effective image retrieval system using weighted combination of integrated Gabor texture features, shape features and spatial information features is hereby proposed.

The shape and spatial features are quite simple to derive and effective, and can be extracted in real time.

The system is integrated because it incorporates Gabor filters of six grid sizes namely 5 x 5, 15 x 15, 25 x 25, 35 x 35, 45 x 45, and 55 x 55.

The system is simple because of the ease with which the system can be operated and display results.

The system is flexible because the feature weights can be adjusted to achieve retrieval refinement according to user's need.

It is robust because the system's algorithm is applicable to retrieval in virtually all kinds of image database.

The system can successfully retrieve not only visual-content based queries but also queries based on semantics such as objects or scenes, hence it is a contribution towards the current research in semantic image retrieval.

In current CBIR systems the common method of improving retrieval performance is by weighting the feature vectors. In this thesis a new and reliable method of improving retrieval performance, and which complement feature weighting is proposed. Since the system use Gabor filter for texture feature extraction, the proposal is weighting the output (features) of the system as derived from various sizes of Gabor filter.

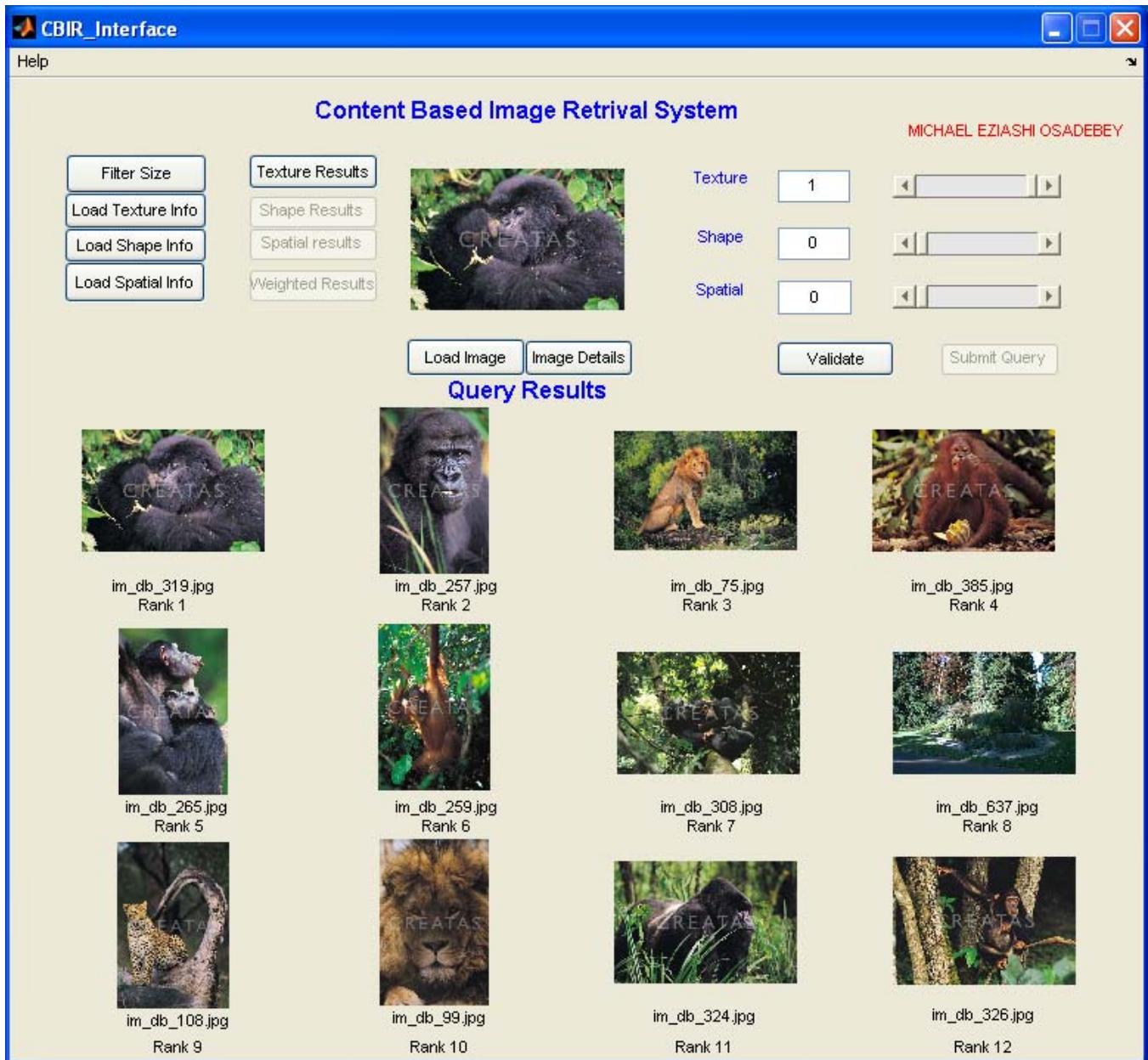
The system has the potential of developing into a semantic based CBIR system by proper mathematical modelling of the texture features obtained from the six grid sizes of Gabor filter and the output of the system.

APPENDIX A: REFERENCES

- [1]. John P. Eakins and Margaret E. Graham, Content-based image retrieval, a report to the JISC technology applications programme, *Institute for image database research, University of Northumbria at Newcastle, U.K*, January 1999.
- [2]. Hideyuki Tamura and Naokazu Yokoya. Image Database Systems: A Survey, *PatternRecognition*, 17(1):29–49, 1984.
- [3]. Fahui Long, Hongjiang Zhang and David Dagan Feng, Fundamentals of content-based image retrieval: *Microsoft corporation research articles*, 2003.
- [4]. Tobias Weyand and Thomas Deselaers, Combining Content-based Image Retrieval with Textual Information Retrieval , *Department of Computer Science, RWTH, Aachen*, October 2005
- [5] Remco C. Veltkamp, Mirela Tanase, Content-based image retrieval systems, a survey, *Technical report, department of computer science, Utrecht University, October 2000*.
- [6]. Rong Zhao and William I. Grosky , Bridging the Semantic Gap in Image Retrieval, *Wayne State University*, USA, Idea group publishing, 2002
- [7]. Zoran Pecenovic, Final year graduate thesis, Image Retrieval Using latent semantic indexing , *Department of Electrical Engineering, Swiss Federal University of Technology, Lausanne*
- [8]. Geoffrey Montgomery, The visual pathway, *A report from the Howard Hughes Medical institute*, 4000 Jones Bridge Road,Chevy Chase, MD 20815-6789, U.S.A
- [9]. J.G. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters", *J. Opt. Soc. Am.*, Vol. 2, 1985.
- [10]. D.A. Pollen and S.F. Ronner, "Phase Relationships Between Adjacent Simple Cells in the Visual Cortex, *Science* Vol. 212, 1981.
- [11]. Zijun Yang and Jay Kuo, Survey on content-based analysis, indexing and retrieval techniques and status report of MPEG-7, *Tamkang journal of science and engineering*, vol.2, No.3 pp.101-118, 1999
- [12]. T. Ojala and M. Pietikäinen, Texture Classification, *Machine vision and Media signal Processing Unit*, University of Oulu, Finland.
- [13]. Kuo, Chung-Feng Jeffrey Su, Te Li, Gray relational analysis for recognizing fabric defects, *Textile research Journal*, May , 2003
- [14]. Mari Partio, Bogdan Cramariuc, Moncef Gabbouj and Ari Visa, Rock texture retrieval using gray level co-occurrence matrix, *Tampere University of Technology*, P.O. Box 553, FIN-33101 Tampere, Finland
- [15]. Leena Lepistö, Iivari Kunttu, Jorma Autio, and Ari Visa , Rock Image Classification Using Non-Homogenous Textures and Spectral Imaging, *WSCG SHORT PAPERS proceedings WSCG'2003, February 3-7, 2003, Plzen, Czech Republic*.published by UNION Agency – Science Press
- [16]. D. Gabor. Theory of communication. *J. IEE (London)*, 93(III):429-457, November 1946.
- [17]. Javier R. Movellan. Tutorial on Gabor Filters, mplab.ucsd.edu/tutorials/tutorials.html, 2002
- [18]. S. Marcelja, "Mathematical Description of the Responses of Simple Cortical Cells, *J. Opt. Soc. Am.*, Vol. 70, 1980.
- [19]. J.G. Daugman: Uncertainty relations for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, *Journal of the Optical Society of America A*, 1985, vol. 2, pp. 1160-1169.
- [20]. J. Jones and L.Palmers, An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex: *Neurophysiology*, pp. 1233--1258, 1987.

- [21]. M. Porat and Y.Y Zeevi. The generalized Gabor scheme of image representation in biological and machine vision. *IEEE Trans. PAMI*, 10(4):452-468, 1988.
- [22] Tony Lindeberg, Scale-Space Theory in Computer Vision, KTH (Royal Institute of Technology) Stockholm, Sweden
- [23]. Dengsheng Zhang, and Guojun Lu, Content-based shape retrieval using different shape descriptors *Grippsland school of computing and information technology, Monash university*, Churchill, Victoria, Australia.
- [24]. Rafael C. Gonzales and Richaerds E. Woods, *Digital image processing* second edition, 2002
- [25]. Jean-Marc Pelletier, Computer vision documentation, www.iamas.ac.jp/~jovan02/cv/jit_cv_doc.pdf
- [26]. S.K Chang, Q.Y.Shi, and C.Y. Yan, Iconic indexing by 2-D strings, *IEEE transactions on pattern analysis. Machine intelligence. Volume 9, No.3, pp.413-428, May 1987.*
- [27]. V.N Gudivada and V.V Raghavan, Design and evaluation of algorithms for image retrieval by spatial similarity, *ACM transactions on information systems, Vol 13, No. 2, pp. 115-144, April 1995*
- [28]. H. Wang, F. Guo, D. Feng and J. Jin, A signature for content-based image retrieval using geometrical transforms, *ACM Multimedia '98*, Bristol , UK
- [29]. H. Samet, The quadtree and related hierarchical data structures, ACM computing surveys, volume 16, No. 2, pp.187-260, 1984
- [30]. B.S Manjunath and W.Y. Ma, Texture features for Browsing and retrieval of image data, *IEEE transactions on pattern analysis and machine intelligence*, vol, 18. No. 8, August 1996
- [31]. Index of /research/imagedatabase/groundtruth, www.cs.washington.edu/research/imagedatabase/groundtruth/
- [32]. test1.tar, www.wang.isu.psu.edu
- [33]. image.vary.jpg.tar, www-db.Stanford.edu
- [34]. Salton, G.: The state of retrieval system evaluation. *Information Processing and Management* 28(4) (1992):441—450
- [35]. TRE(1999). Text Retrieval Conference (TREC), web page: <http://trec.nist.gov>
- [36]. A.C Bovik, M.Clark and W.S. Geisler, Multi-channel texture analysis using localized spatial filters: *IEEE transaction on pattern analysis and machine intelligence*, 1990.
- [37]. Moon-Chuen Lee , Chi-Man Pun, Rotation and scale invariant wavelet feature for content-based texture image retrieval, *Journal of the American Society for Information Science and Technology*, Volume 54, Issue 1, 2003. Pages 68-80
- [38]. Jainguo Zhang, Tieniu Tan, Li Ma, Invariant Texture Segmentation Via Circular Gabor Filters. *16th International Conference on Pattern Recognition (ICPR'02) - Volume 2*, 2002.
- [39]. George M. Haley and B.S Manjunath , Rotation-invariant texture classification using a complete space-frequency model, *IEEE transactions on image processing*, vol, 8. No. 2, February 1999

APPENDIX B1.1

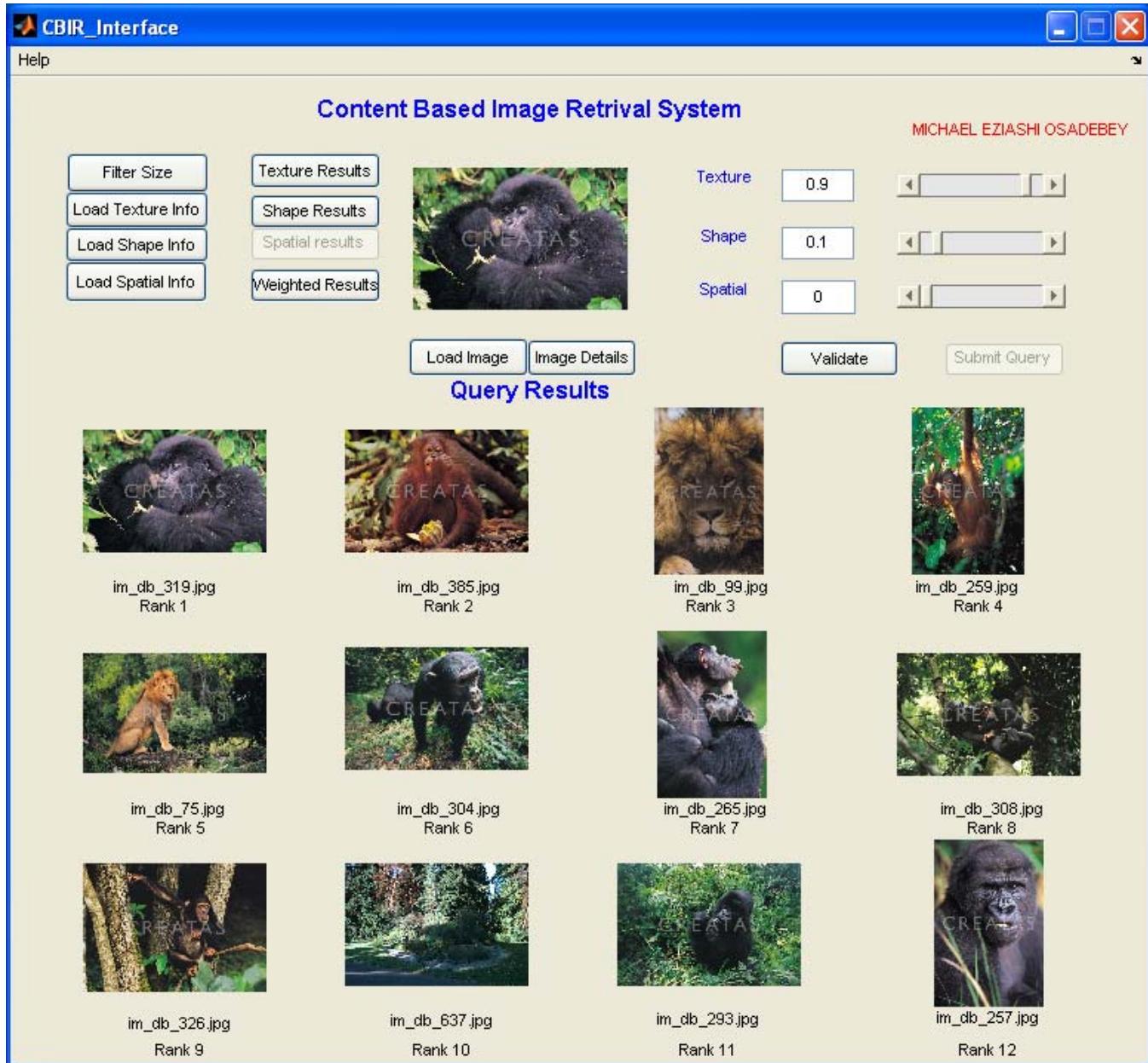


UW - 25 X 25 - First operation

ANNOTATION - Monkey /green plant

PRECISION – 11/12

APPENDIX B1.2



UW - 25 X 25 - First operation

ANNOTATION - Monkey /green plant

PRECISION - 11/12

APPENDIX B2.1

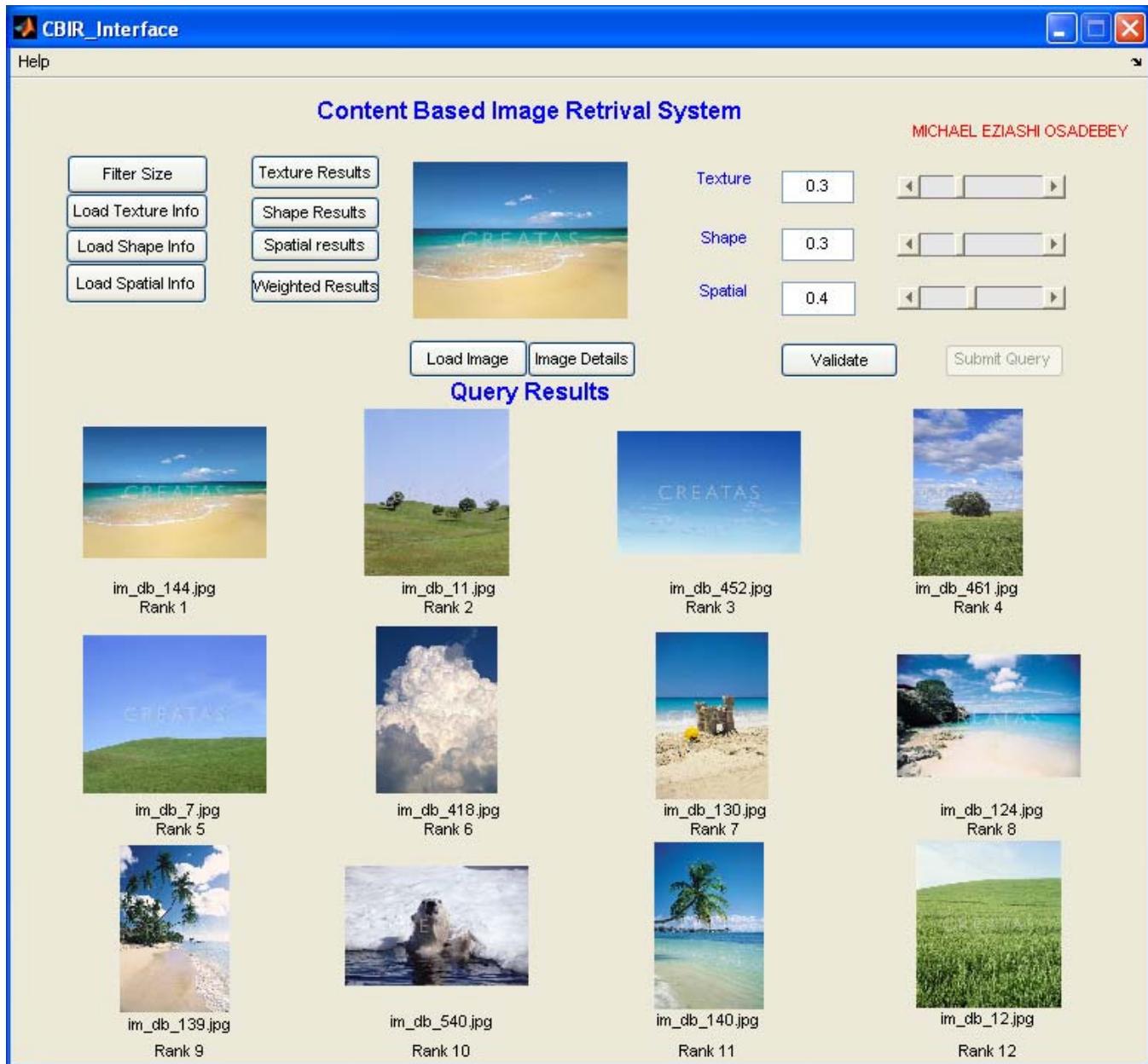


UW – 25 X 25 - First operation

ANNOTATION - Blue sky/cloud/bluish-green sea/yellow-brown land

Precision - 8/12

APPENDIX B2.2

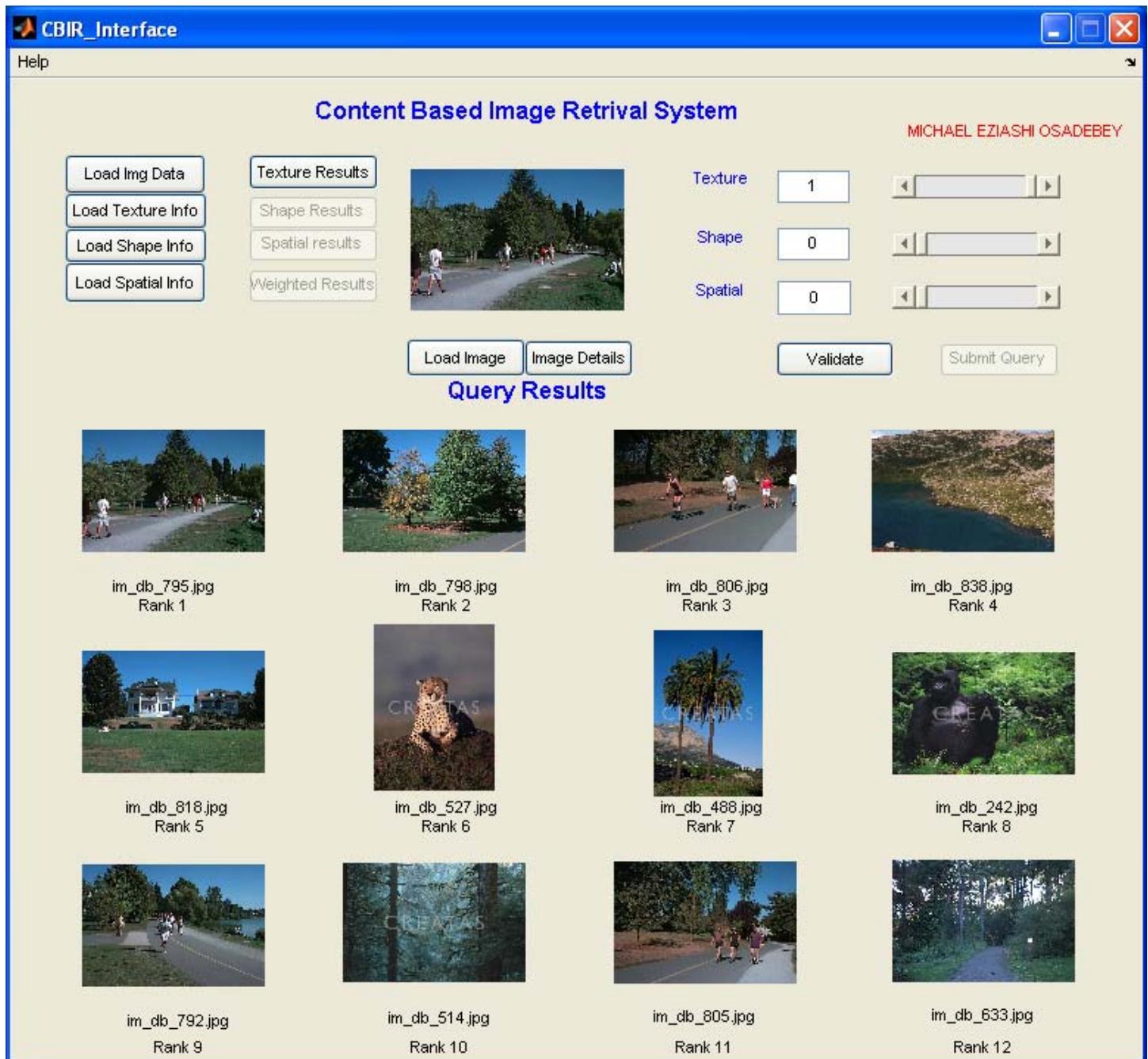


UW – 25 X 25 - First operation

ANNOTATION=Blue sky/cloud/bluish-green sea/yellow-brown land

Precision=12/12

APPENDIX B3.1

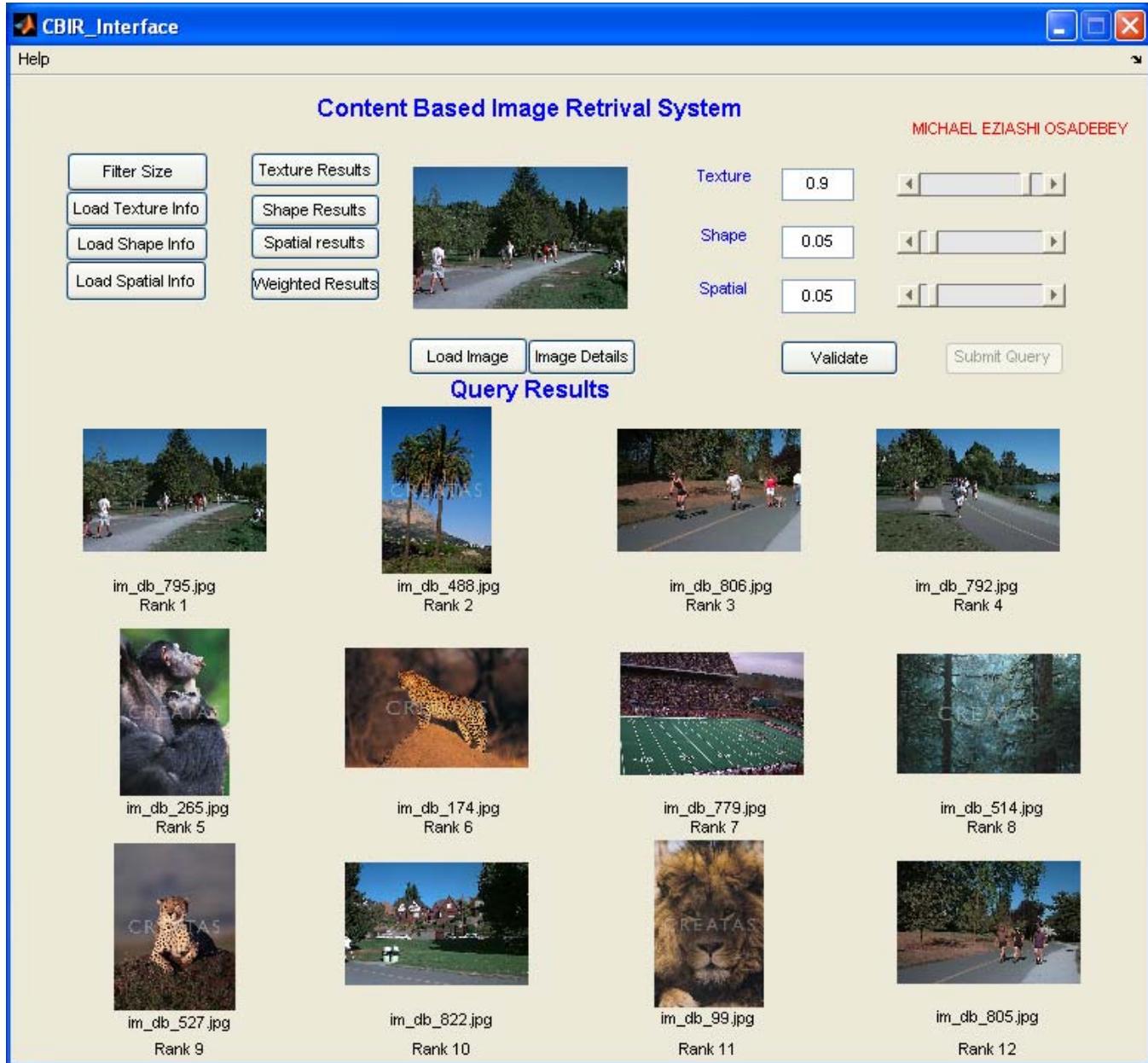


UW – 25 X 25 - First operation

ANNOTATION - Blue sky/road/people/green grass/green forest

Precision - 10/12

APPENDIX B3.2

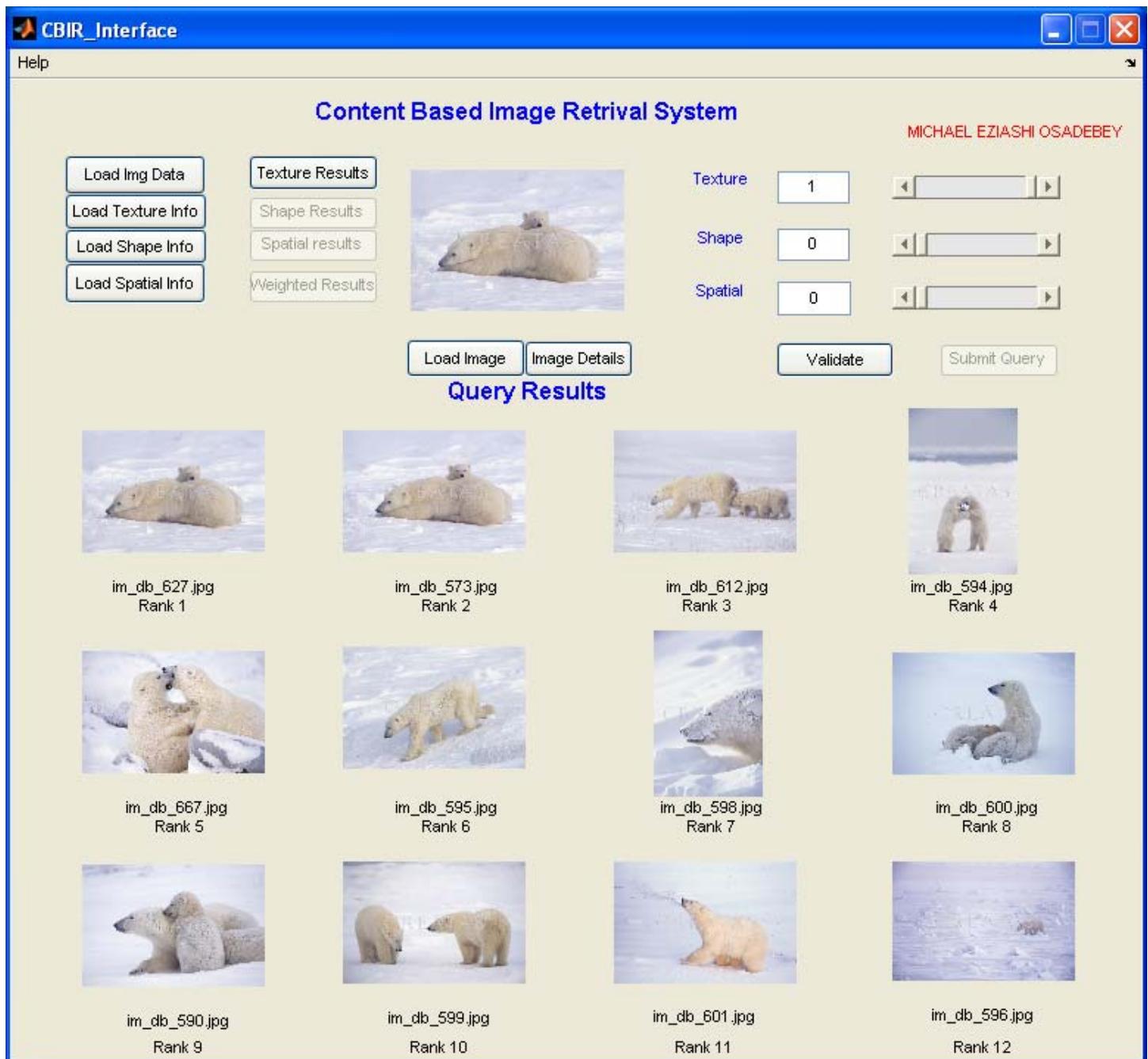


UW – 25 X 25 - First operation

ANNOTATION - Blue sky/road/people/green grass/green forest

Precision - 8/12

APPENDIX B4.1

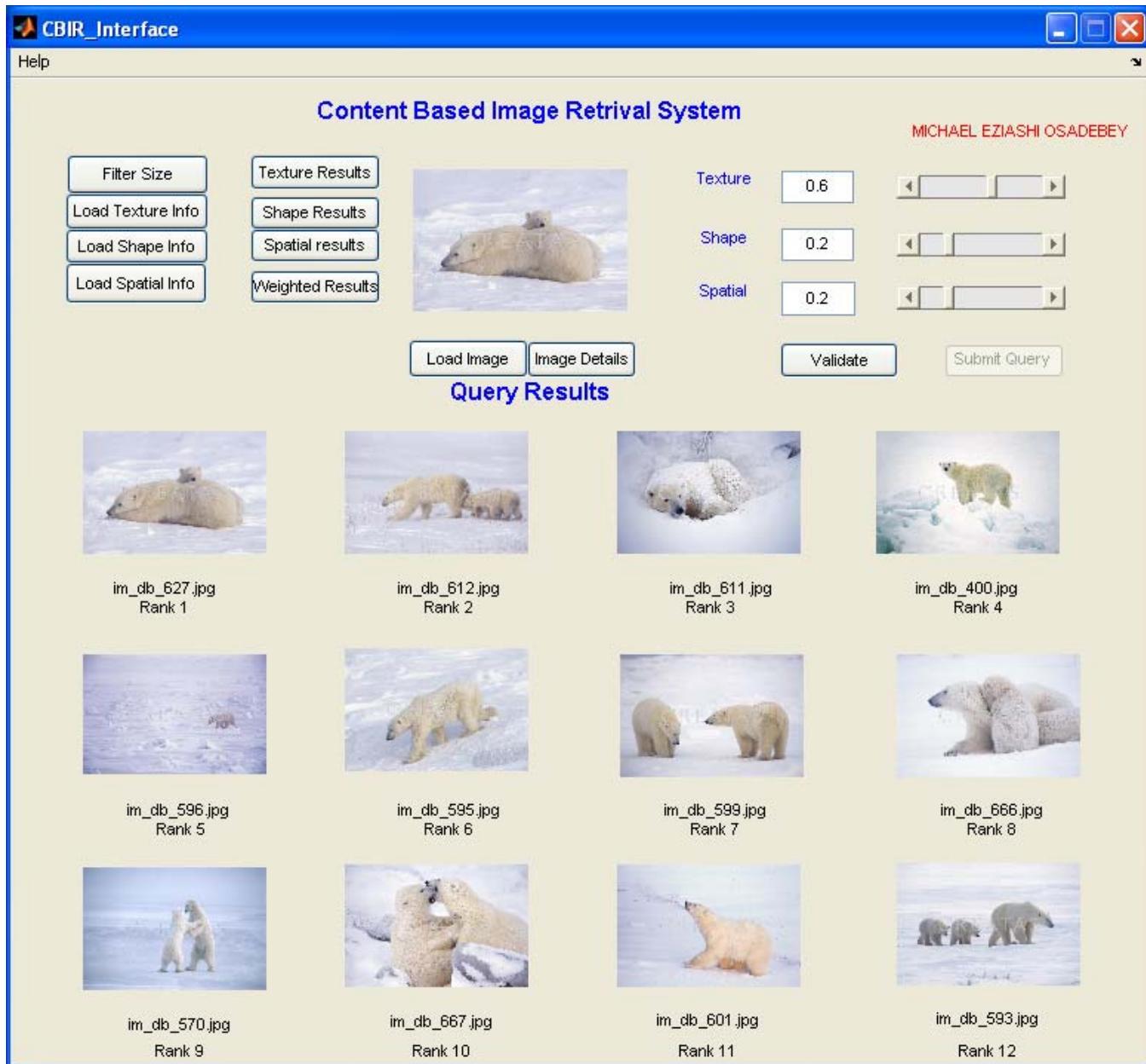


UW - 25 X 25 - First operation

ANNOTATION - Polar bear/snowy background

Precision - 12/12

APPENDIX B4.2

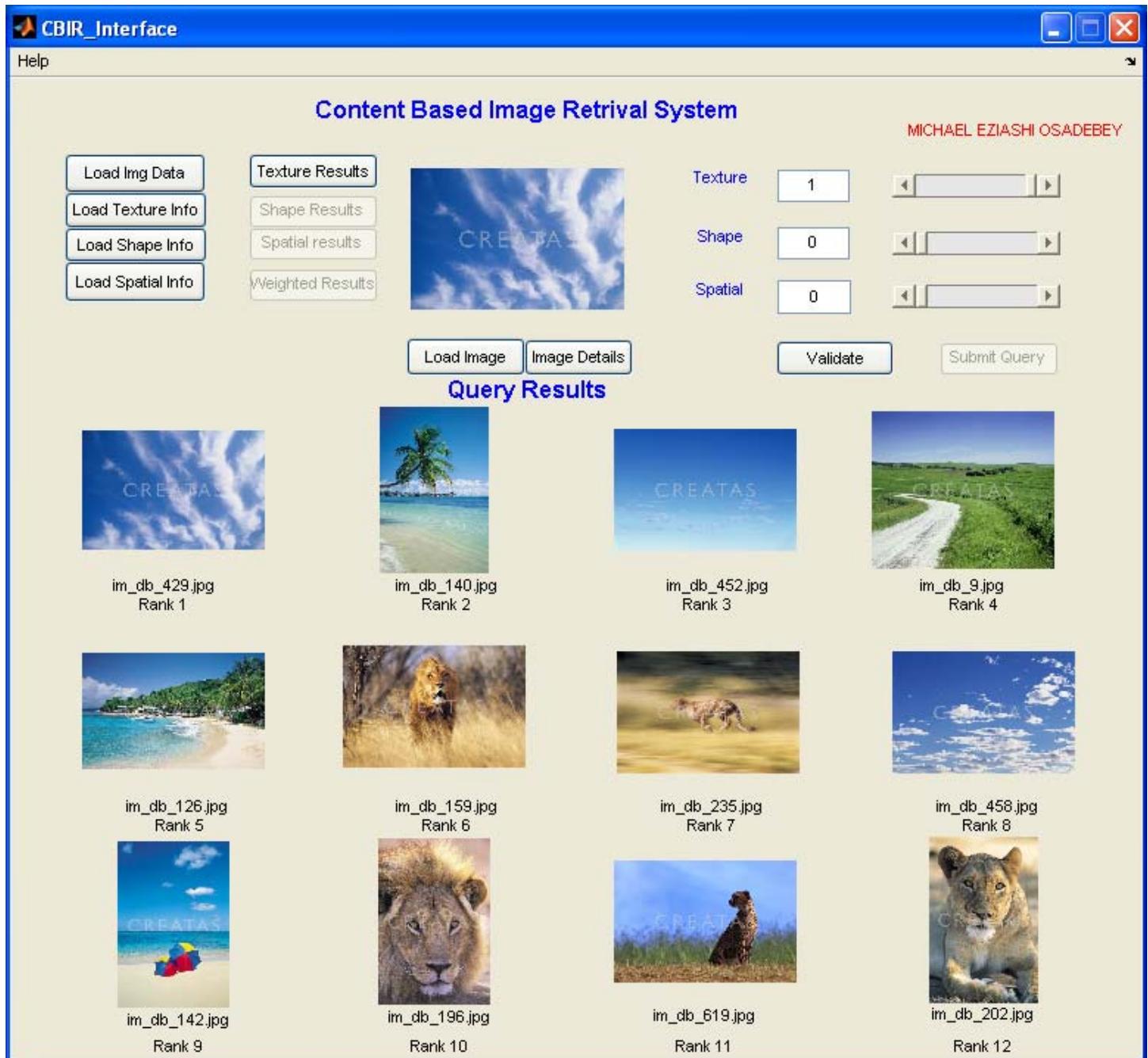


UW – 25 X 25 - First operation

ANNOTATION - Polar bear/snowy background

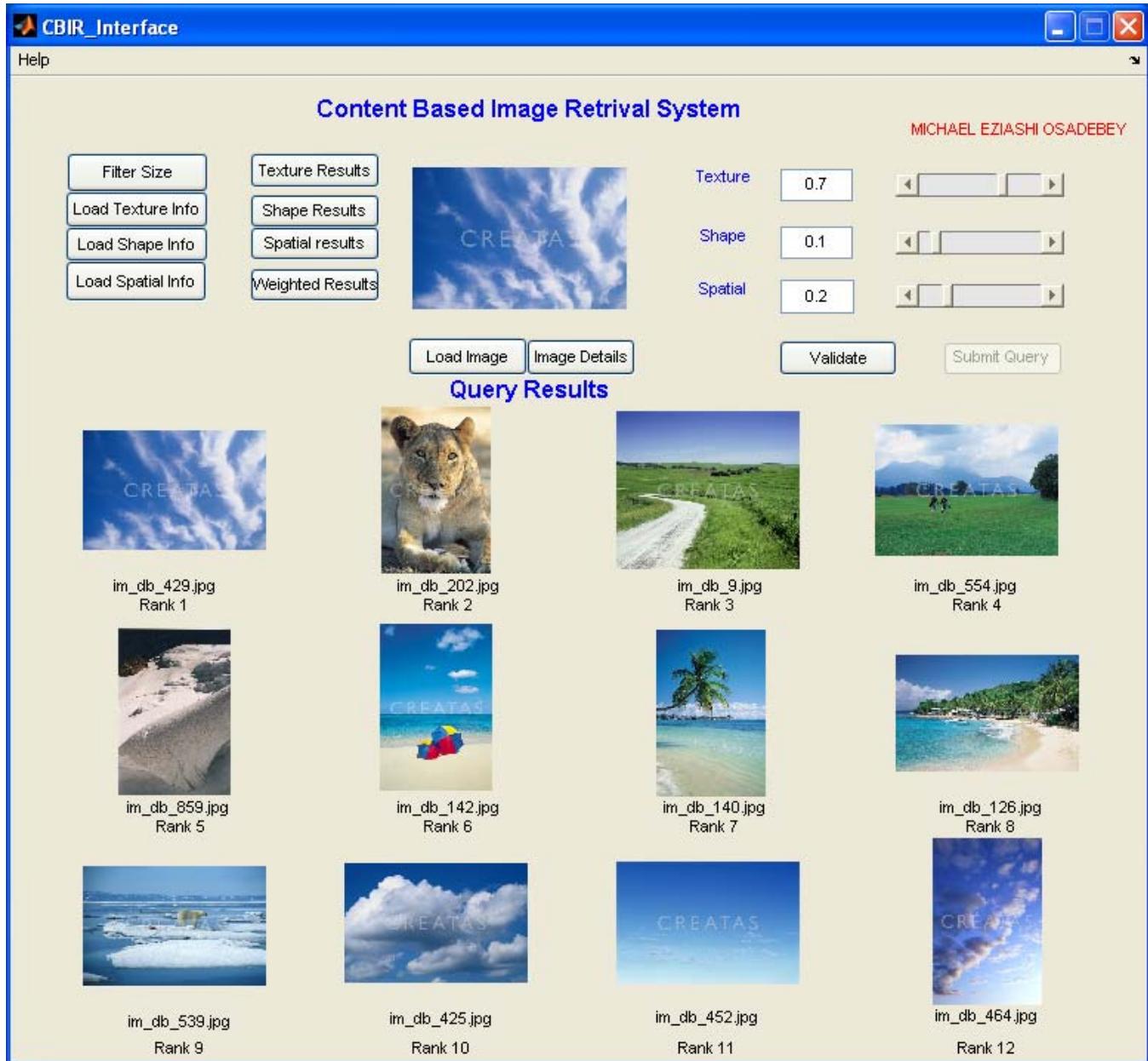
Precision - 12/12

APPENDIX B5.1



UW – 25 X 25, First operation
ANNOTATION - Blue snowy sky
Precision - 8/12

APPENDIX B5.2

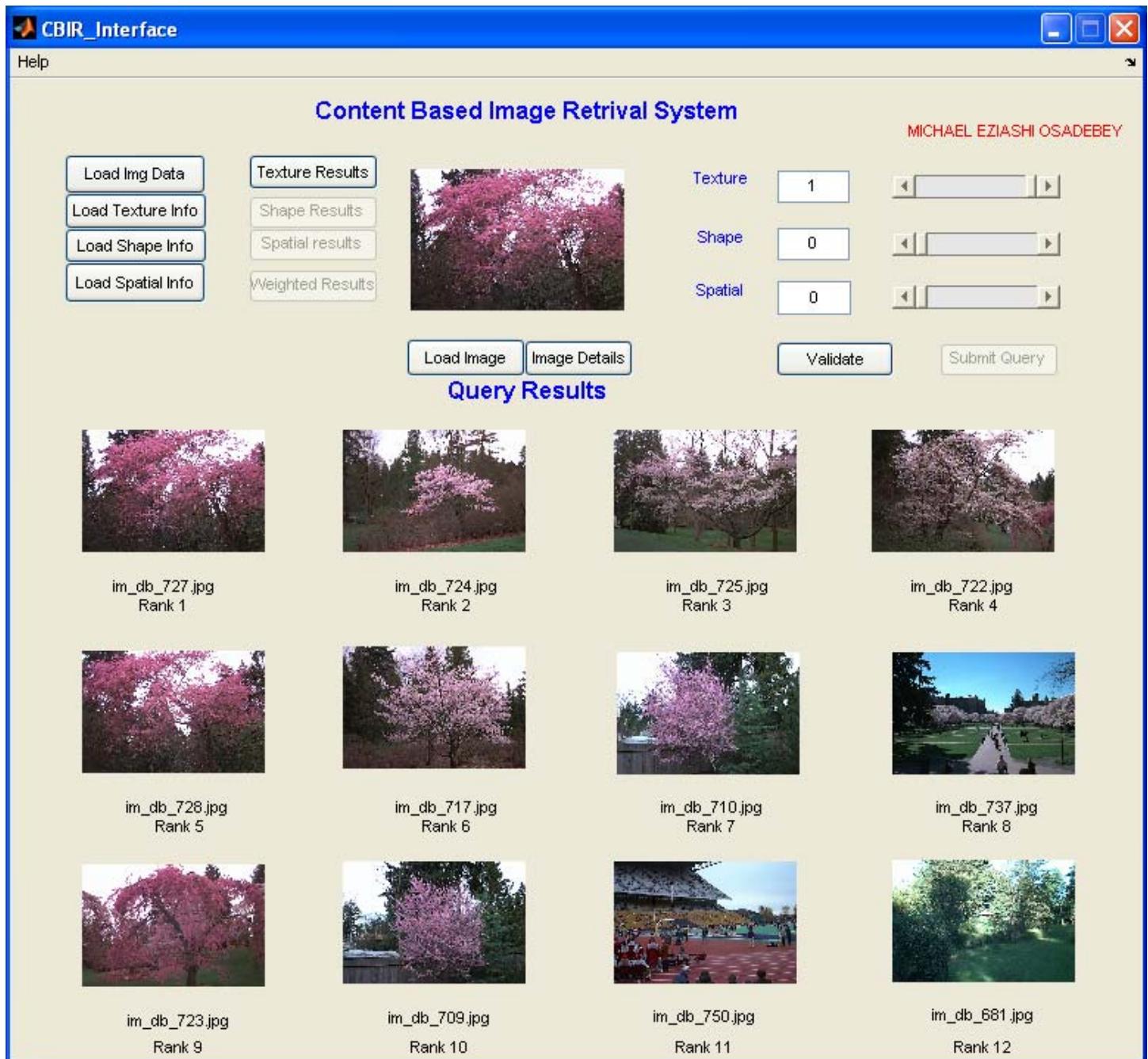


UW - 15 X 15 – First operation

ANNOTATION - Blue snowy sky

Precision - 10/12

APPENDIX B6.1

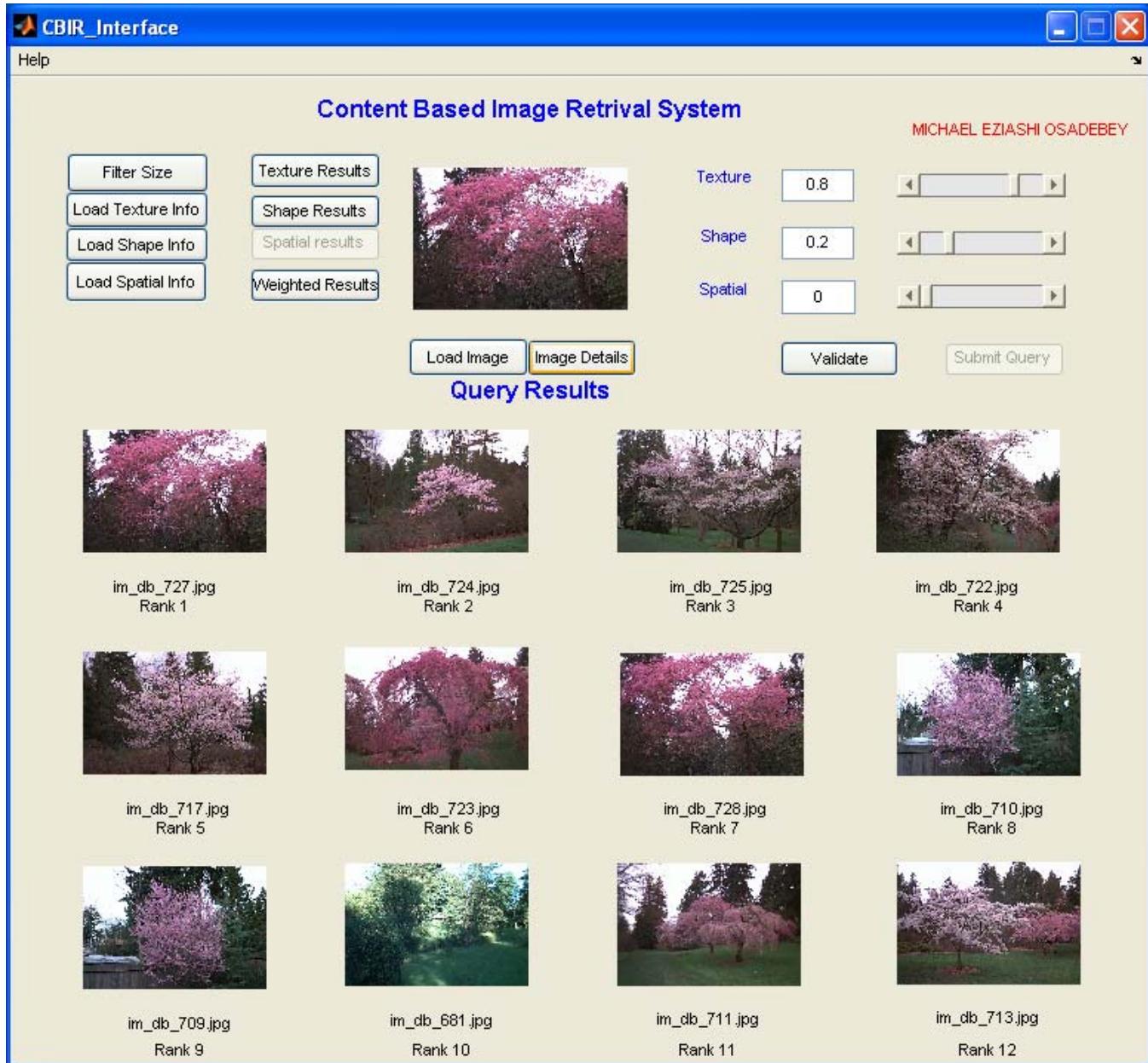


UW - 25 X 25 – First operation

ANNOTATION - Reddish cherry/green plant/white sky

Precision - 10/12

APPENDIX B6.2

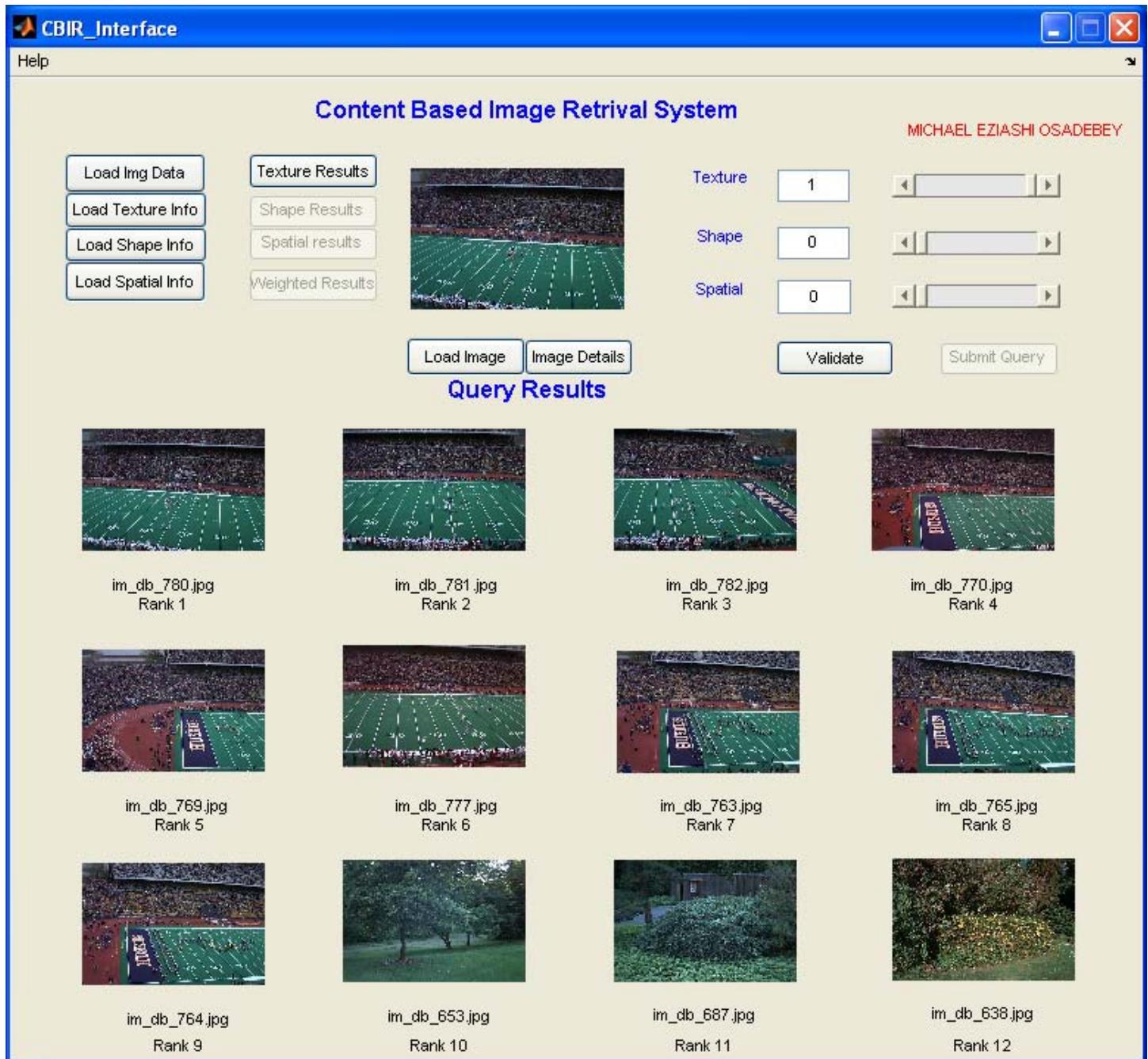


UW - 25 X 25 – First operation

ANNOTATION - Reddish cherry/green plant/white sky

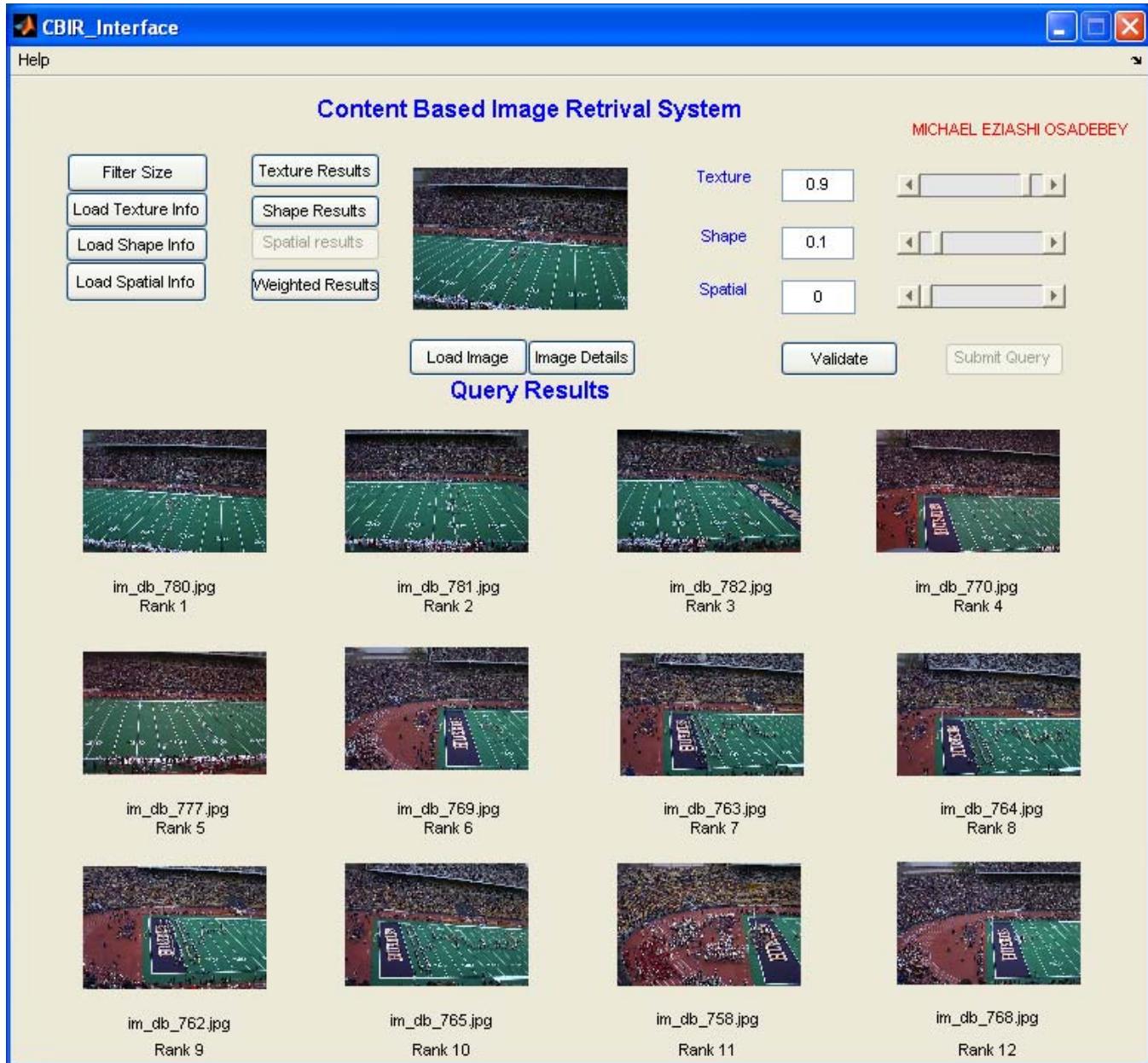
Precision - 12/12

APPENDIX B7.1



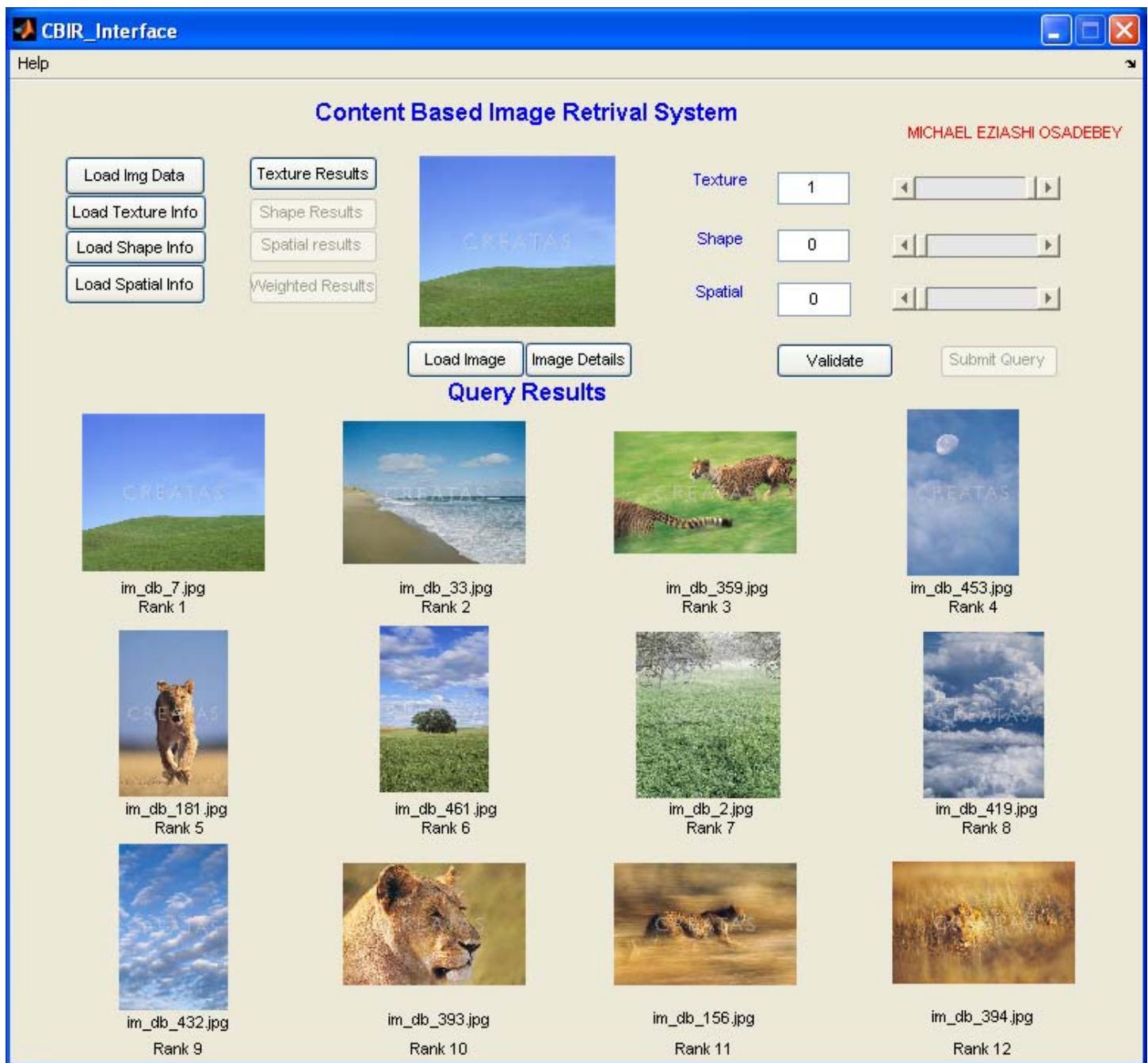
UW - 25 X 25 – First operation
ANNOTATION - stadium/people
Precision - 9/12

APPENDIX B7.1



**UW - 35 X 35 – First operation
ANNOTATION - stadium/people
Precision - 9/12**

APPENDIX B8.1

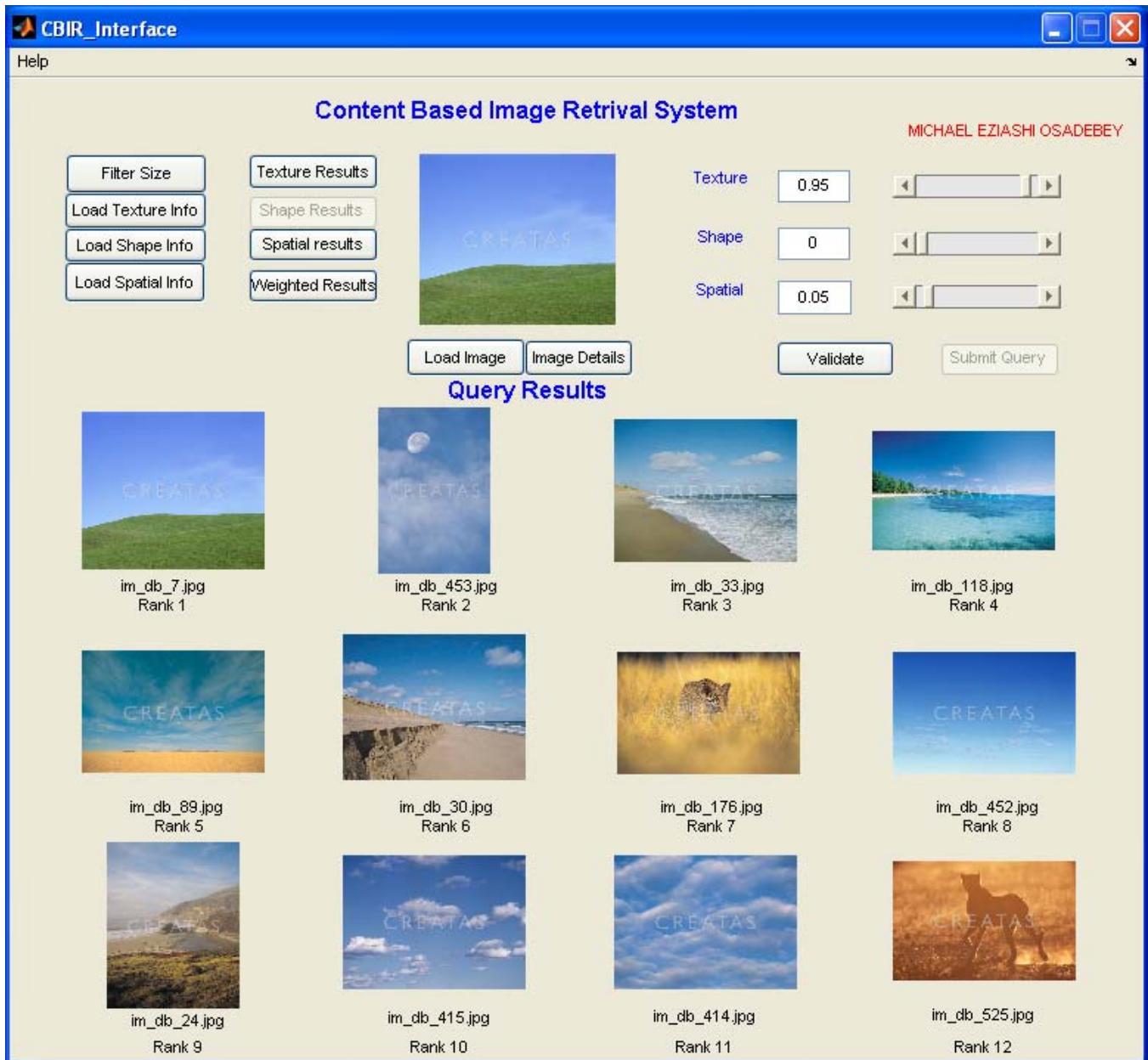


UW - 25 X 25 – First operation

ANNOTATION - green grass/blue sky

Precision - 9/12

APPENDIX B8.2

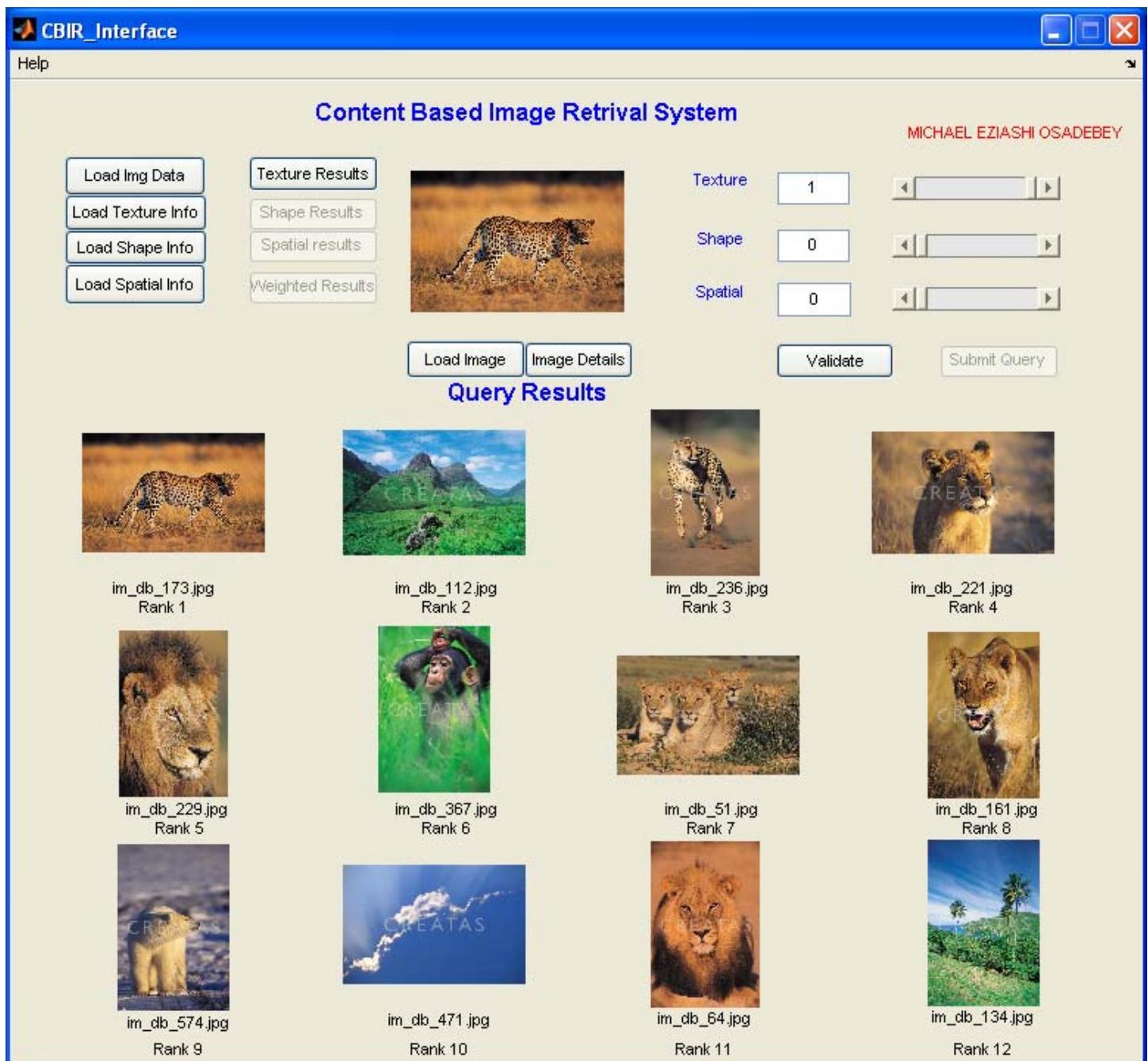


UW - 55 X 55 – First operation

ANNOTATION - green grass/blue sky

Precision - 10/12

APPENDIX B9.1

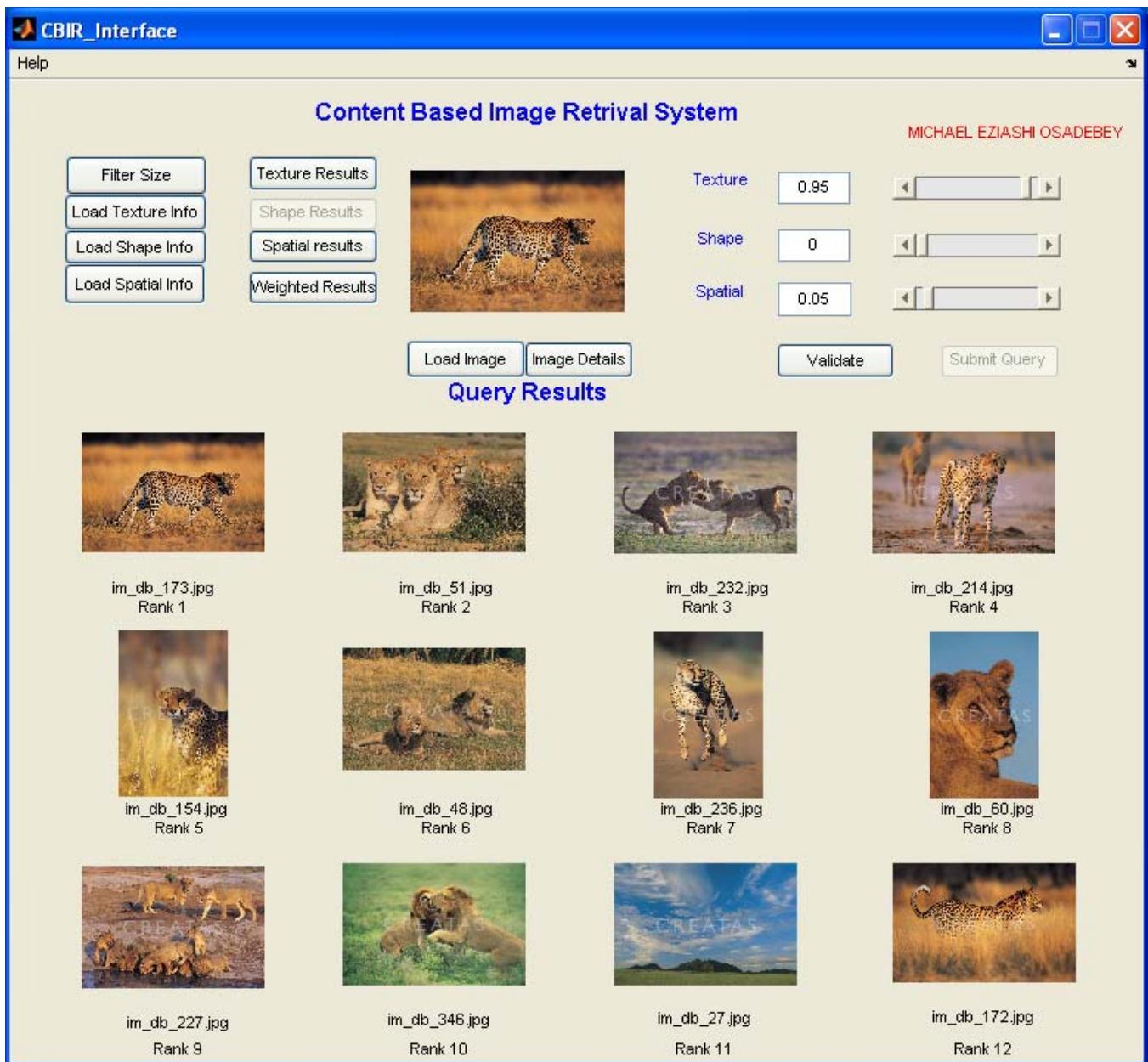


UW - 25 X 25 –First operation

ANNOTATION - leopard/lion/red-brown textured grass

Precision - 7/12

APPENDIX B9.2

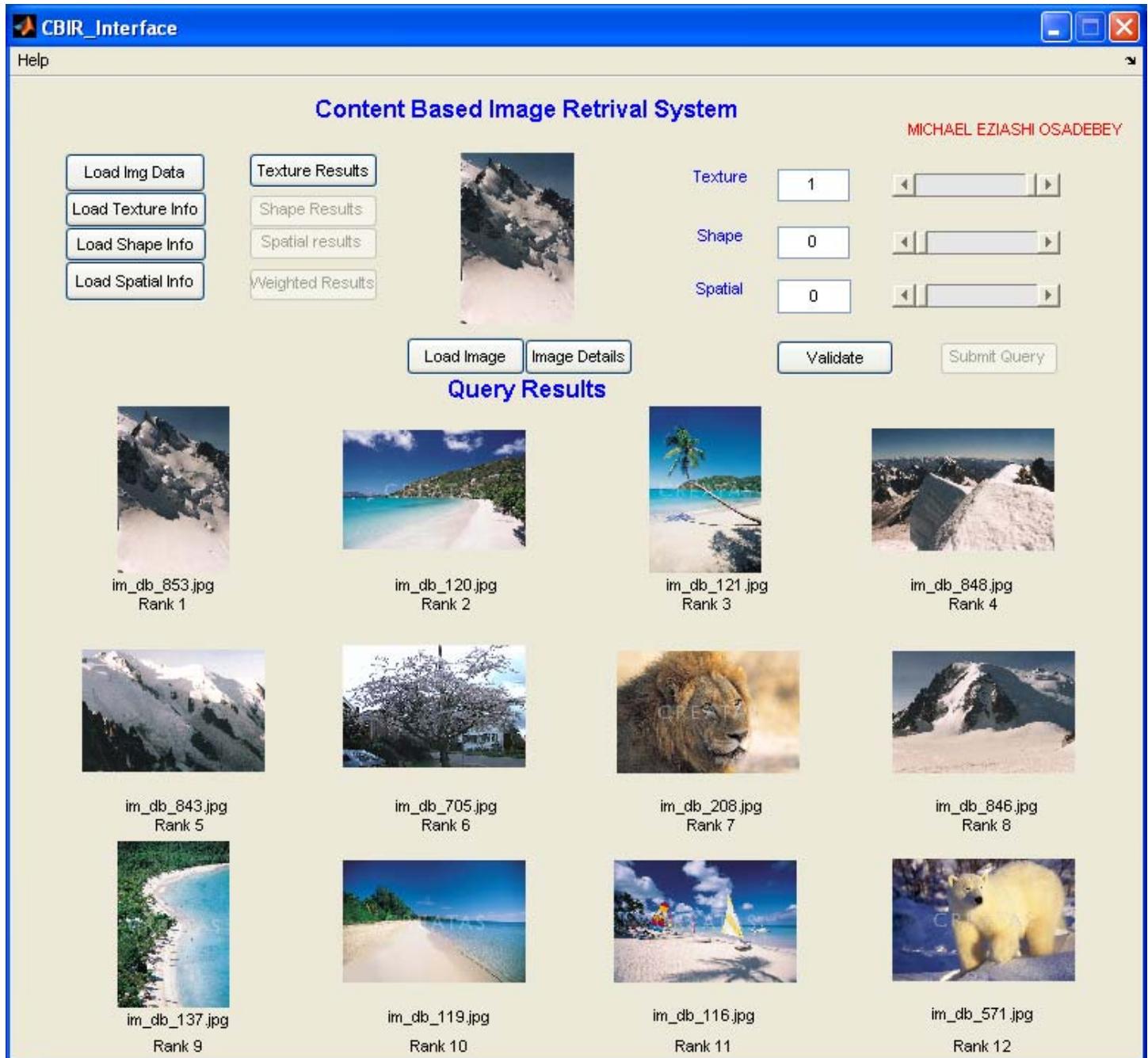


UW - 55 X 55 –First operation

ANNOTATION - leopard/lion/red-brown textured grass

Precision - 11/12

APPENDIX B10.1

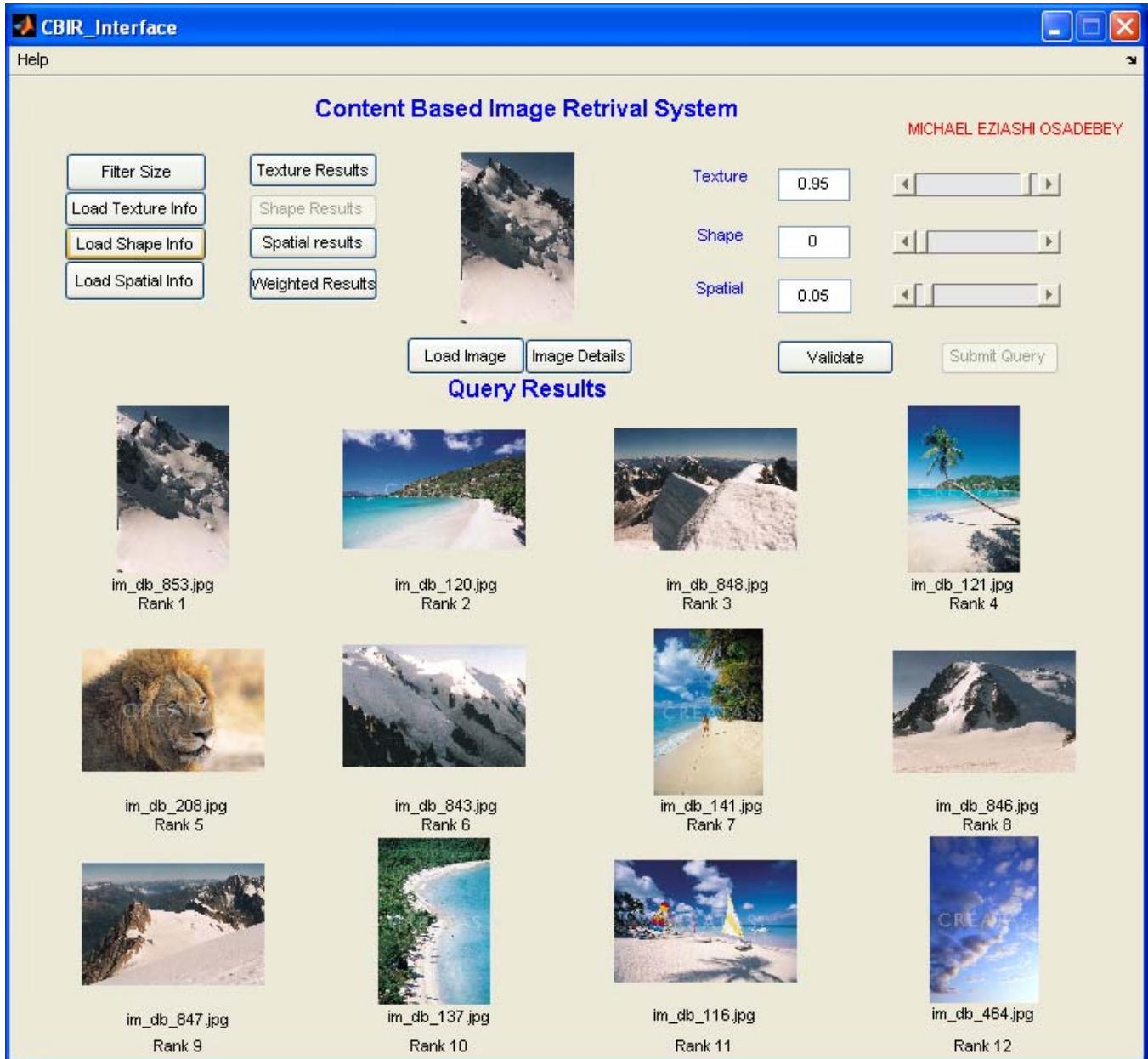


UW - 25 X 25 – First operation

ANNOTATION - Mountain/milk-colour sandy terrain

Precision - 9/12

APPENDIX B10.2

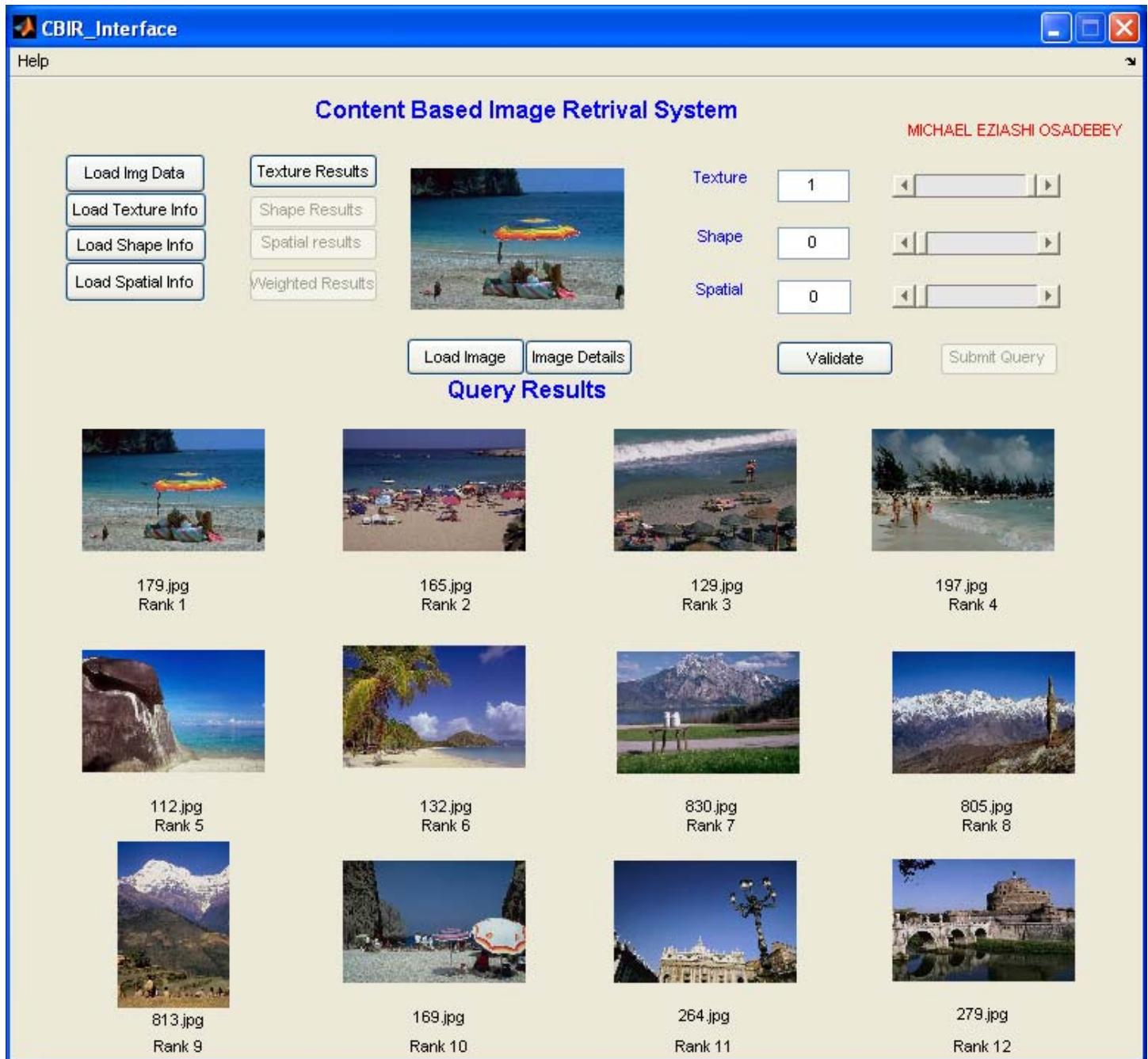


UW - 35 X 35 – First operation

ANNOTATION - Mountain/milk-colored sandy terrain

Precision - 11/12

APPENDIX C1.1

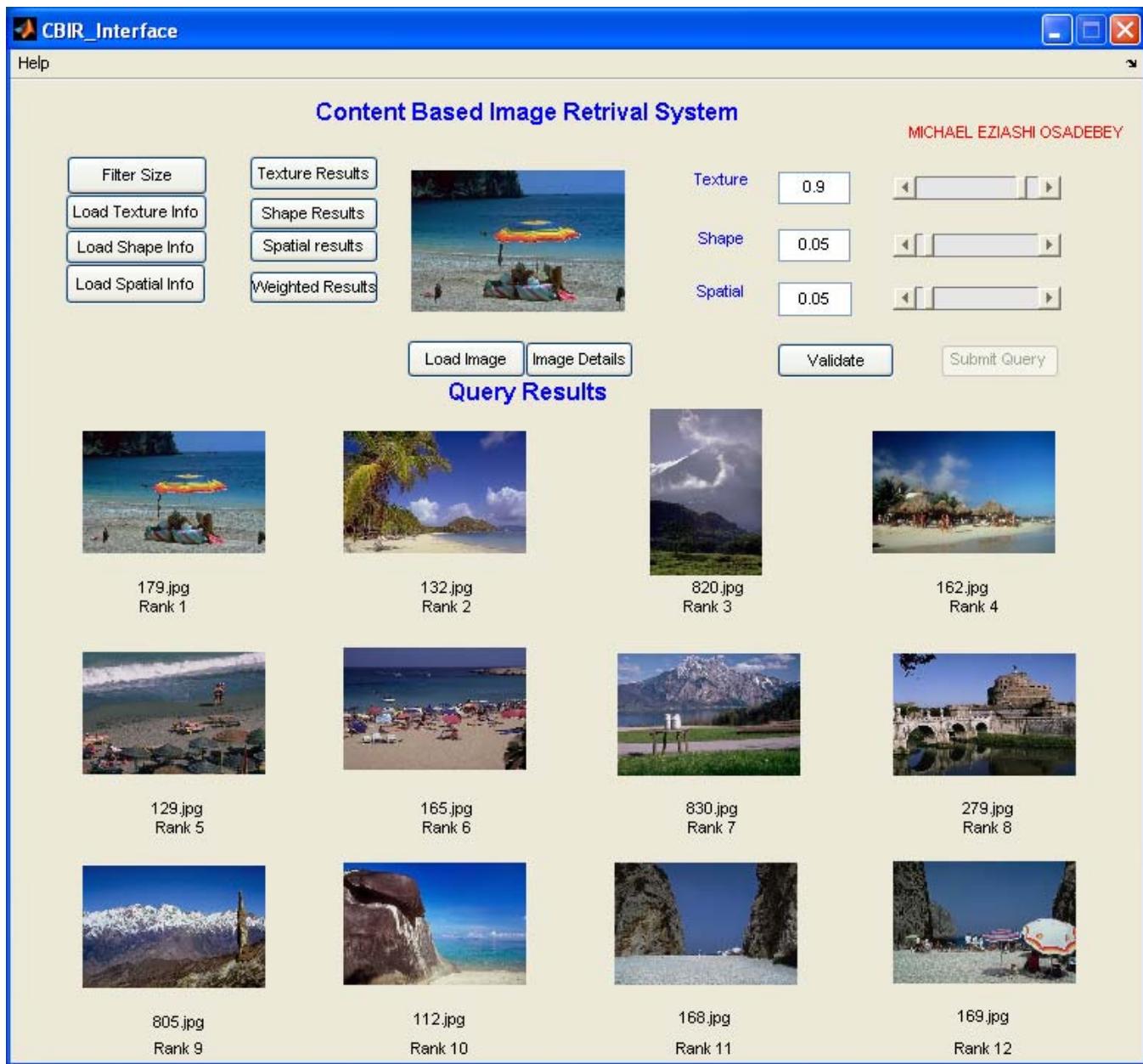


PSU - 25 X 25 –First operation

ANNOTATION - Blue river/blue sky/brown beach/canopy/ridge

Precision - 12/12

APPENDIX C1.2



PSU - 35 X 35 –First operation

ANNOTATION - Blue river/blue sky/brown beach/canopy/ridge

Precision - 10/12

APPENDIX C2.1



PSU - 25 X 25 – First operation
ANNOTATION - Bus/road/green leaves
Precision - 8/12

APPENDIX C2.2

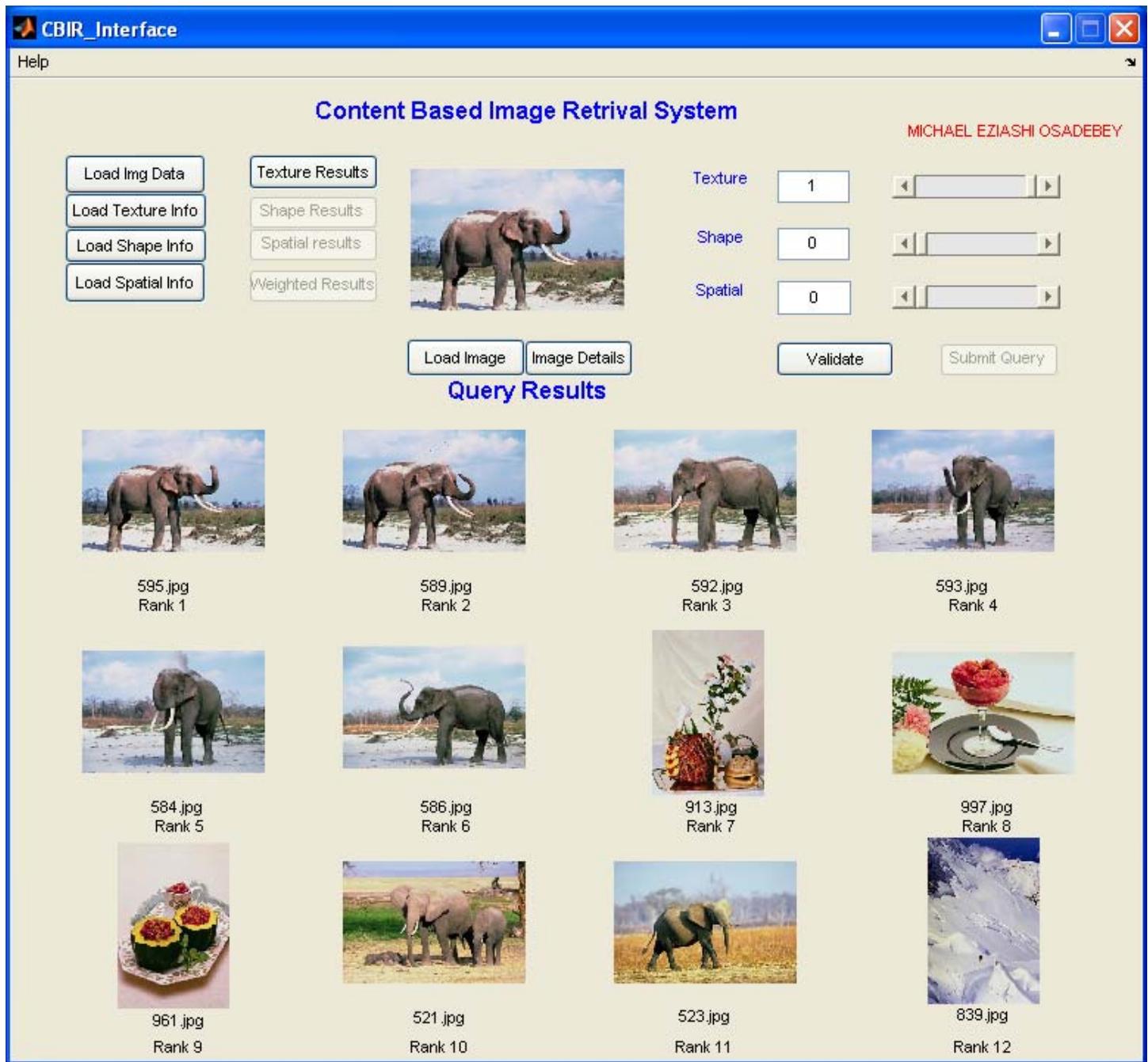


PSU - 55 X 55 – First operation

ANNOTATION - Bus/road/green leaves

Precision - 9/12

APPENDIX C3.1

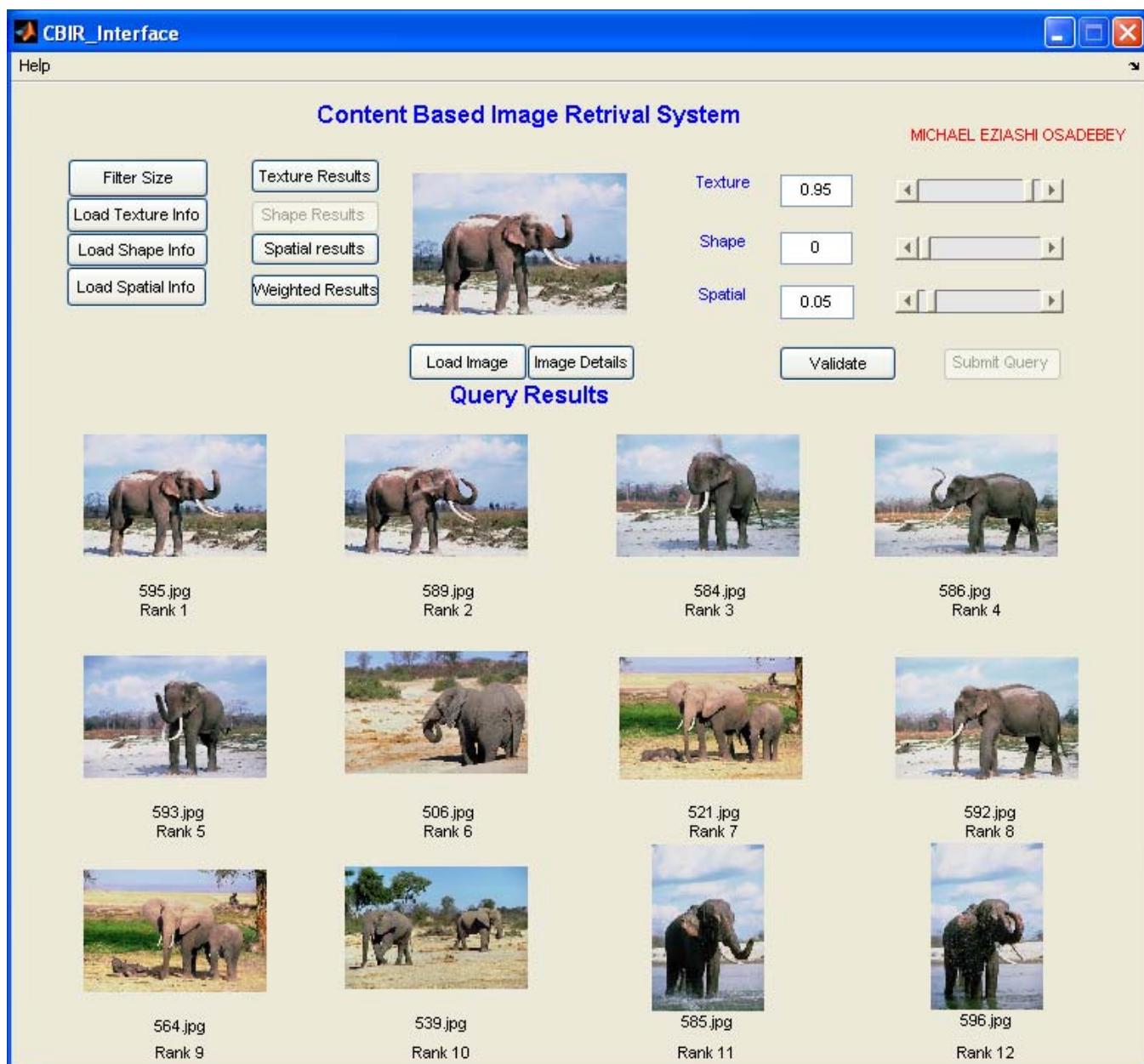


PSU - 25 X 25 – First operation

ANNOTATION - Elephant/blue sky/grass/sandy ground

Precision - 8/12

APPENDIX C3.2

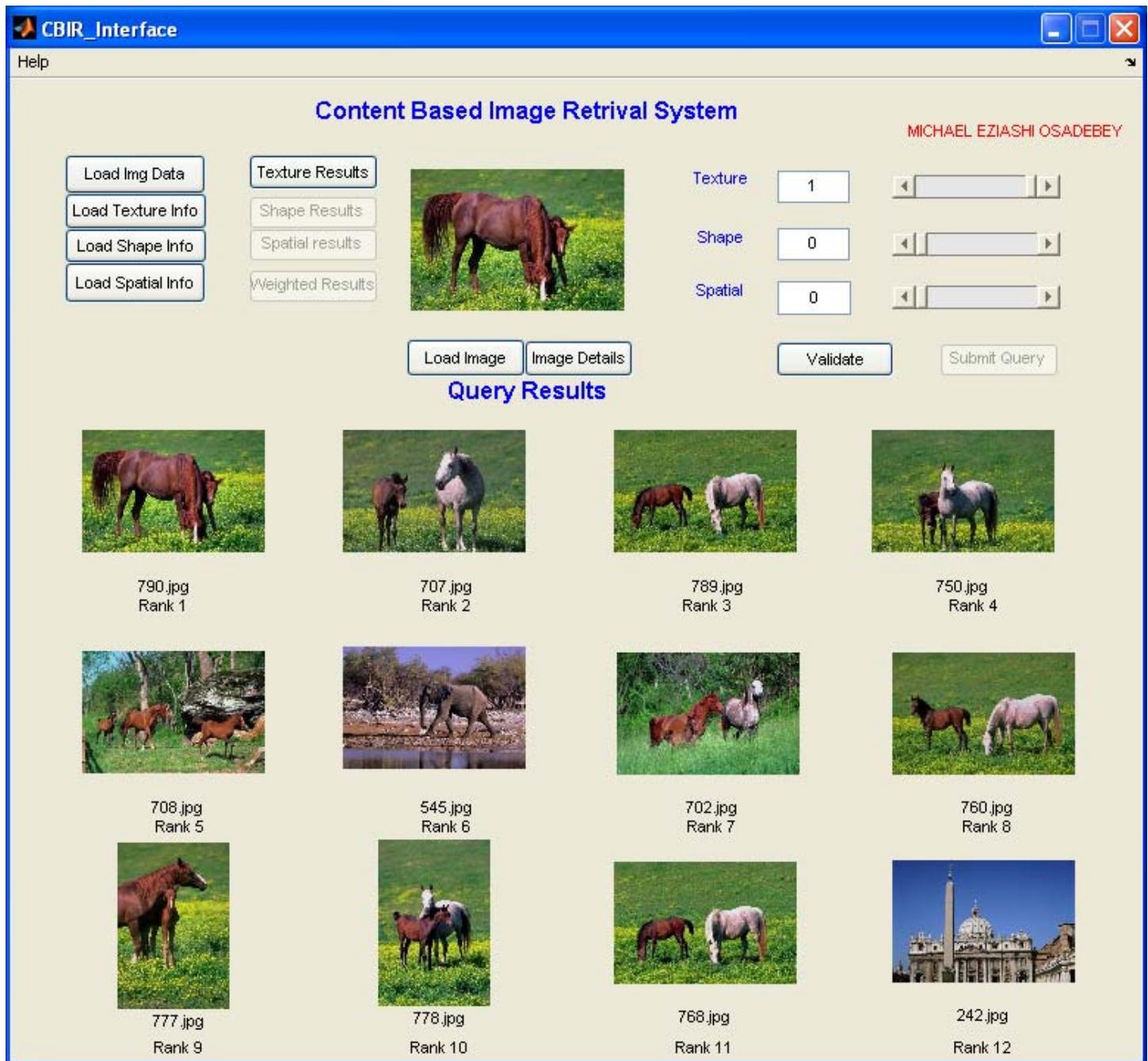


PSU - 55 X 55 – First operation

ANNOTATION - Elephant/blue sky/grass/sandy ground

Precision - 12/12

APPENDIX C4.1

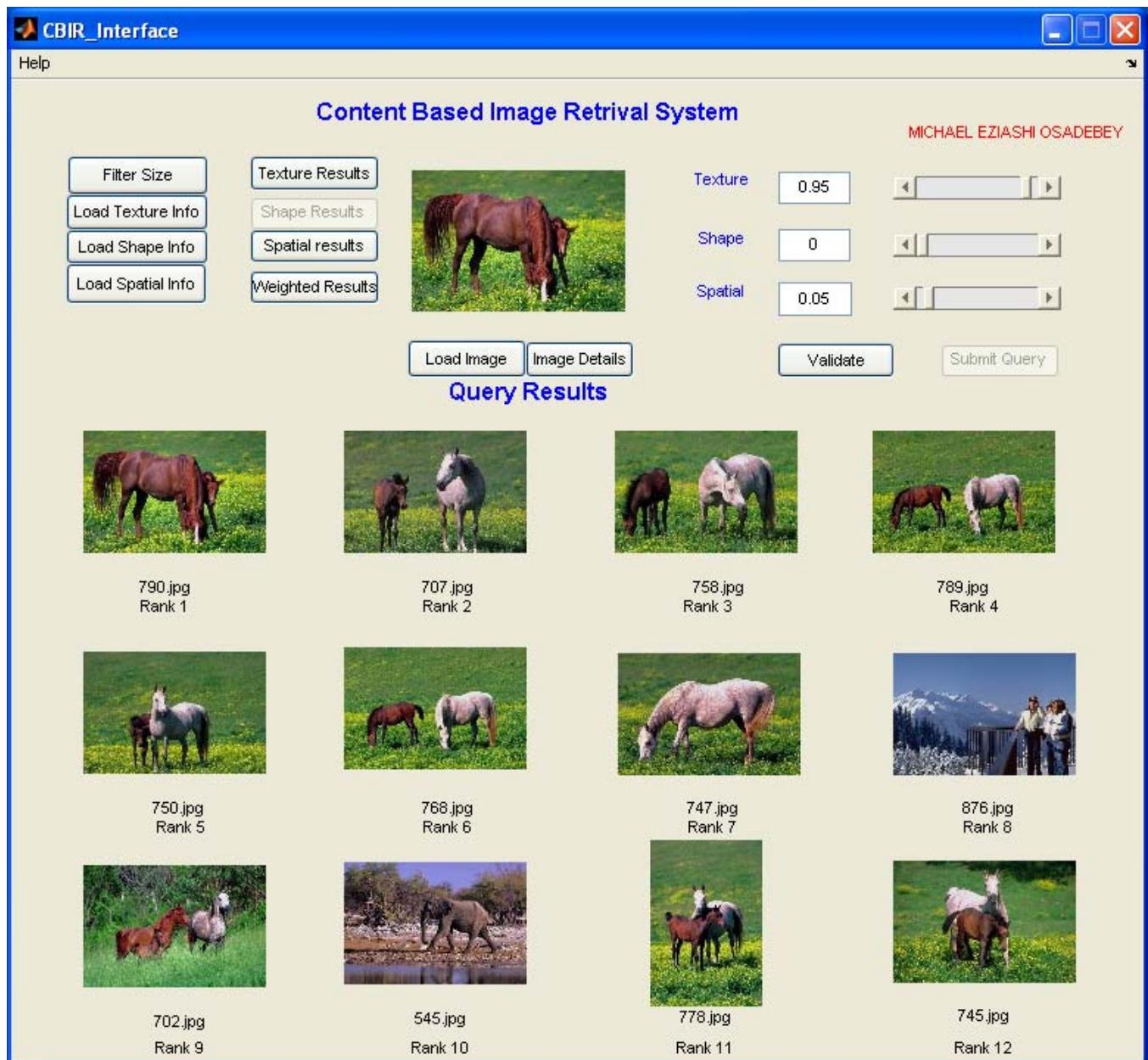


PSU - 25 X 25 – First operation

ANNOTATION - Horse/green field

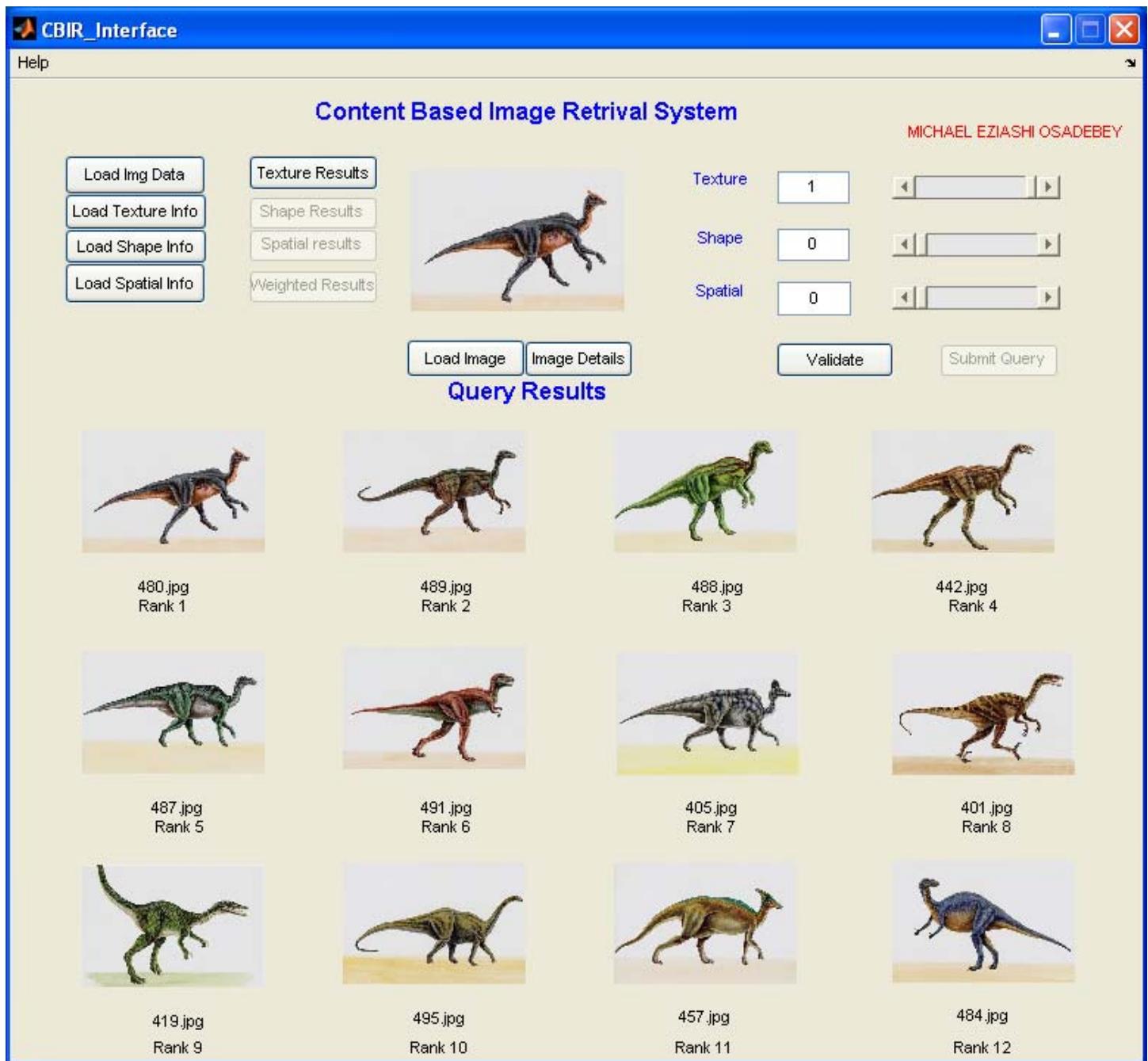
Precision - 10/12

APPENDIX C4.2



PSU - 35 X 35 – First operation
ANNOTATION - Horse/green filed
Precision - 11/12

APPENDIX C5.1

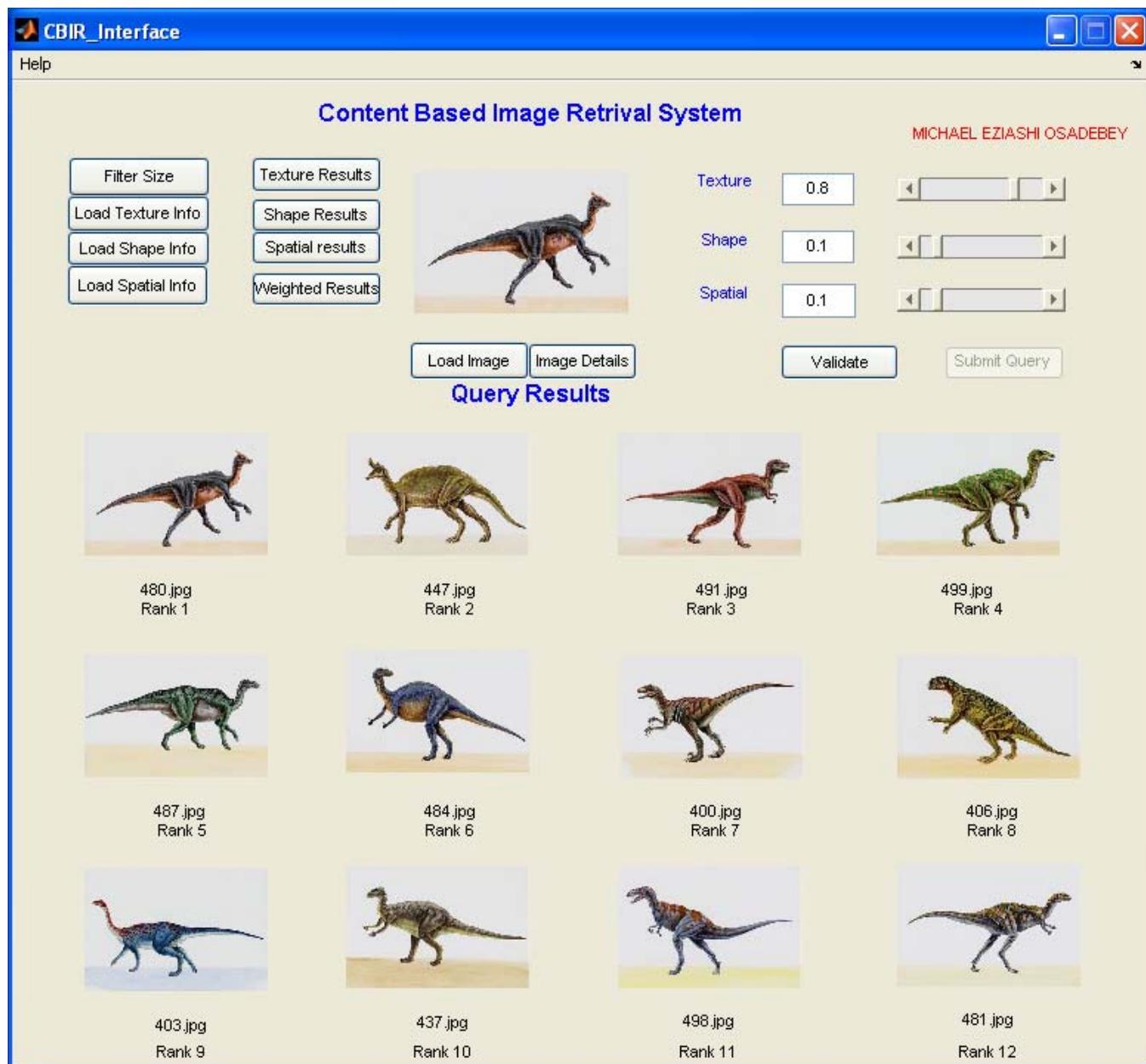


PSU - 25 X 25 – First operation

ANNOTATION - kangaroo

Precision - 12/12

APPENDIX C5.2

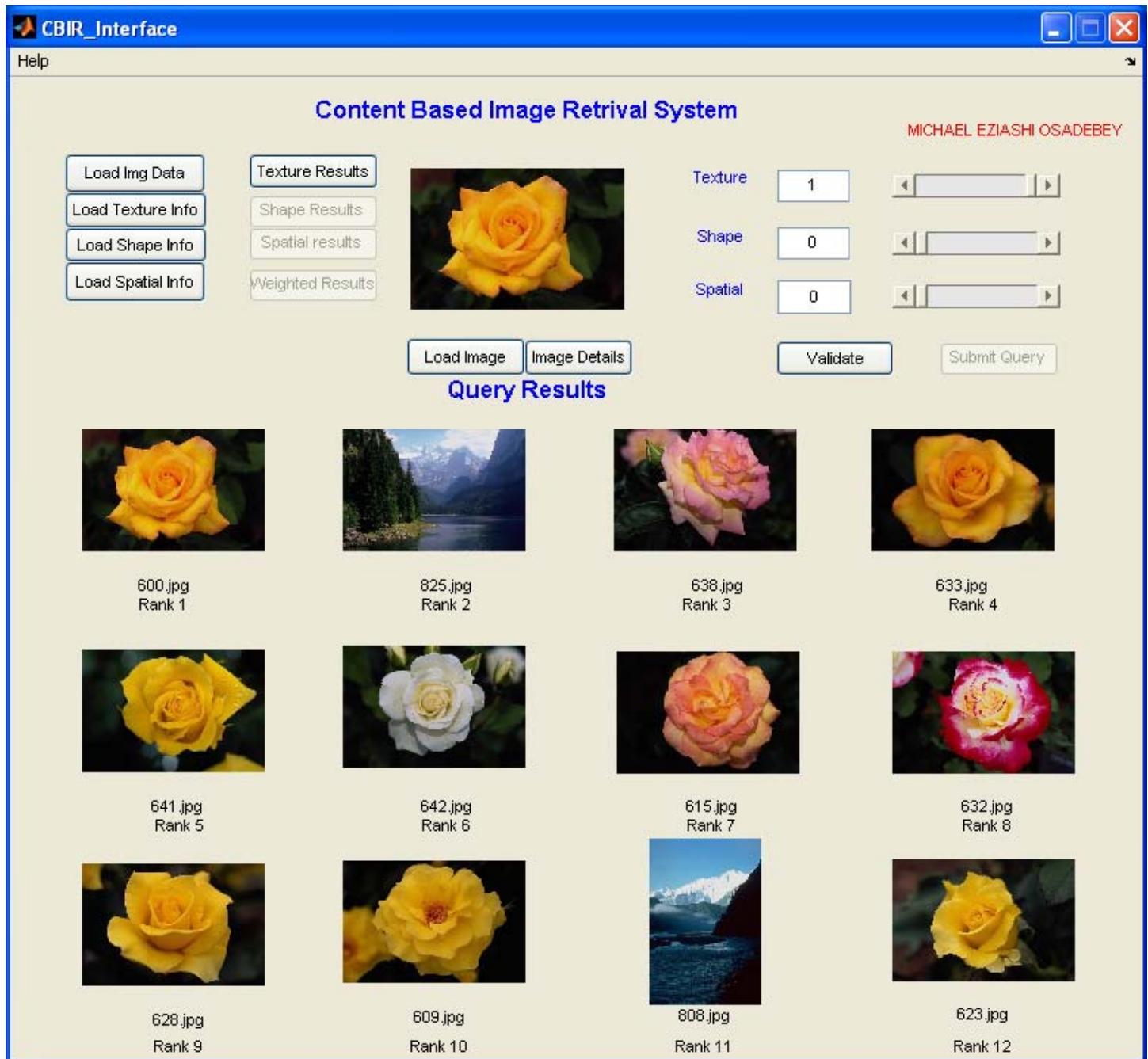


PSU - 25 X 25 – First operation

ANNOTATION - kangaroo

Precision - 12/12

APPENDIX C6.1

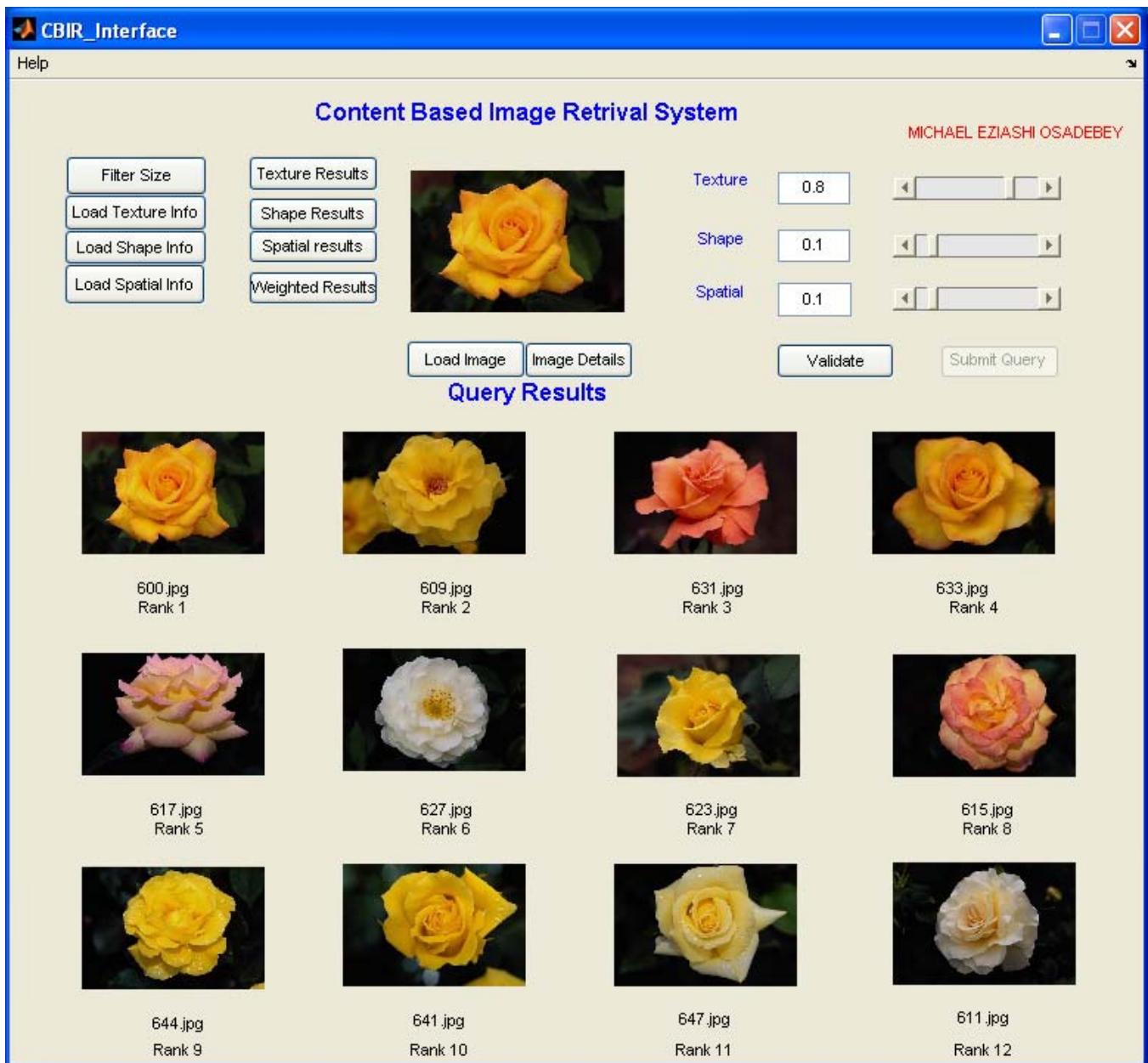


PSU - 25 X 25–First operation

ANNOTATION - red-yellow textured Flower

Precision - 10/12

APPENDIX C6.2

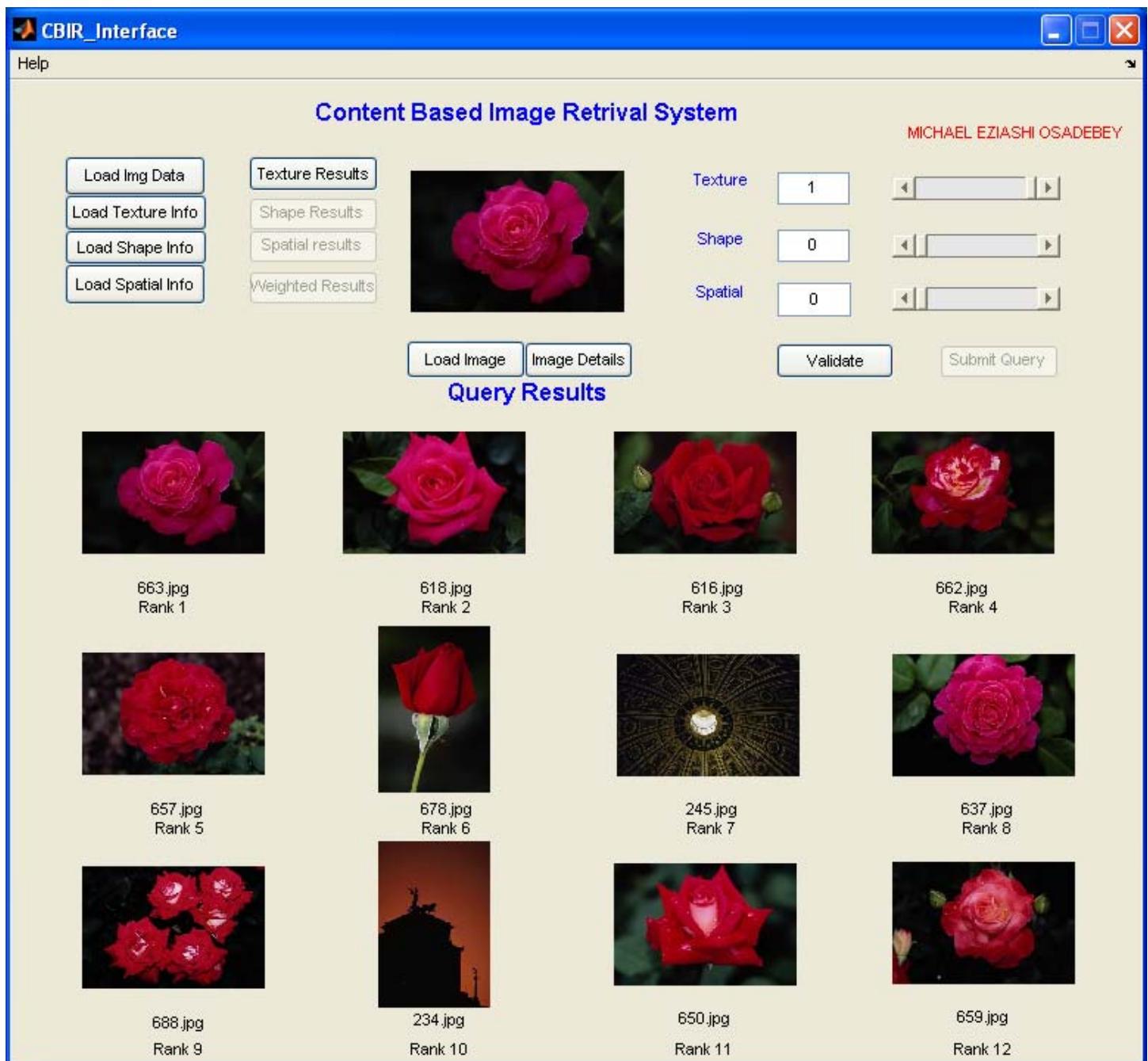


PSU - 45 X 45–First operation

ANNOTATION - red-yellow textured Flower

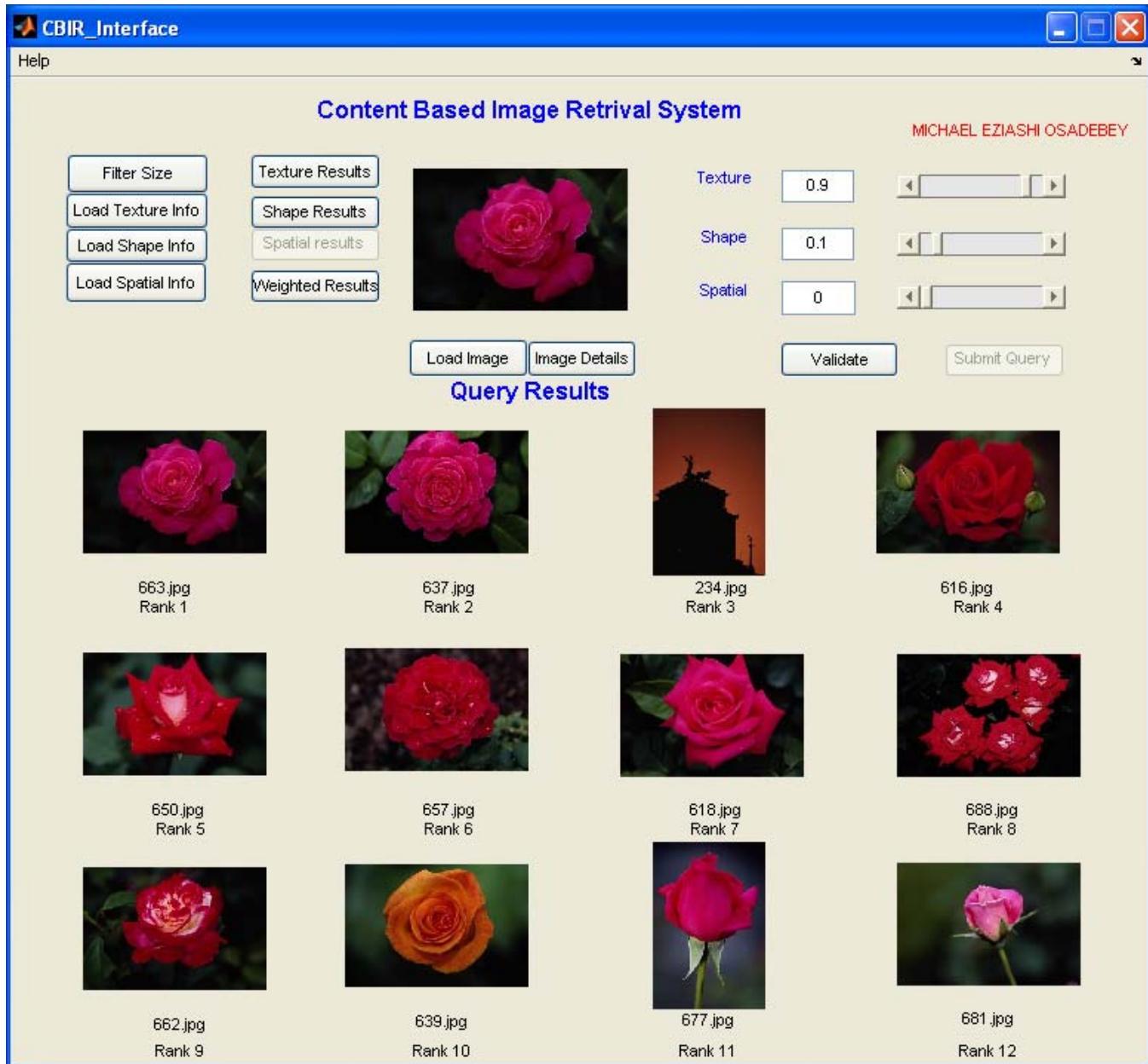
Precision - 12/12

APPENDIX C7.1



PSU- 25 X 25-First operation
ANNOTATION - Red textured flower
Precision =10/12

APPENDIX C7.2

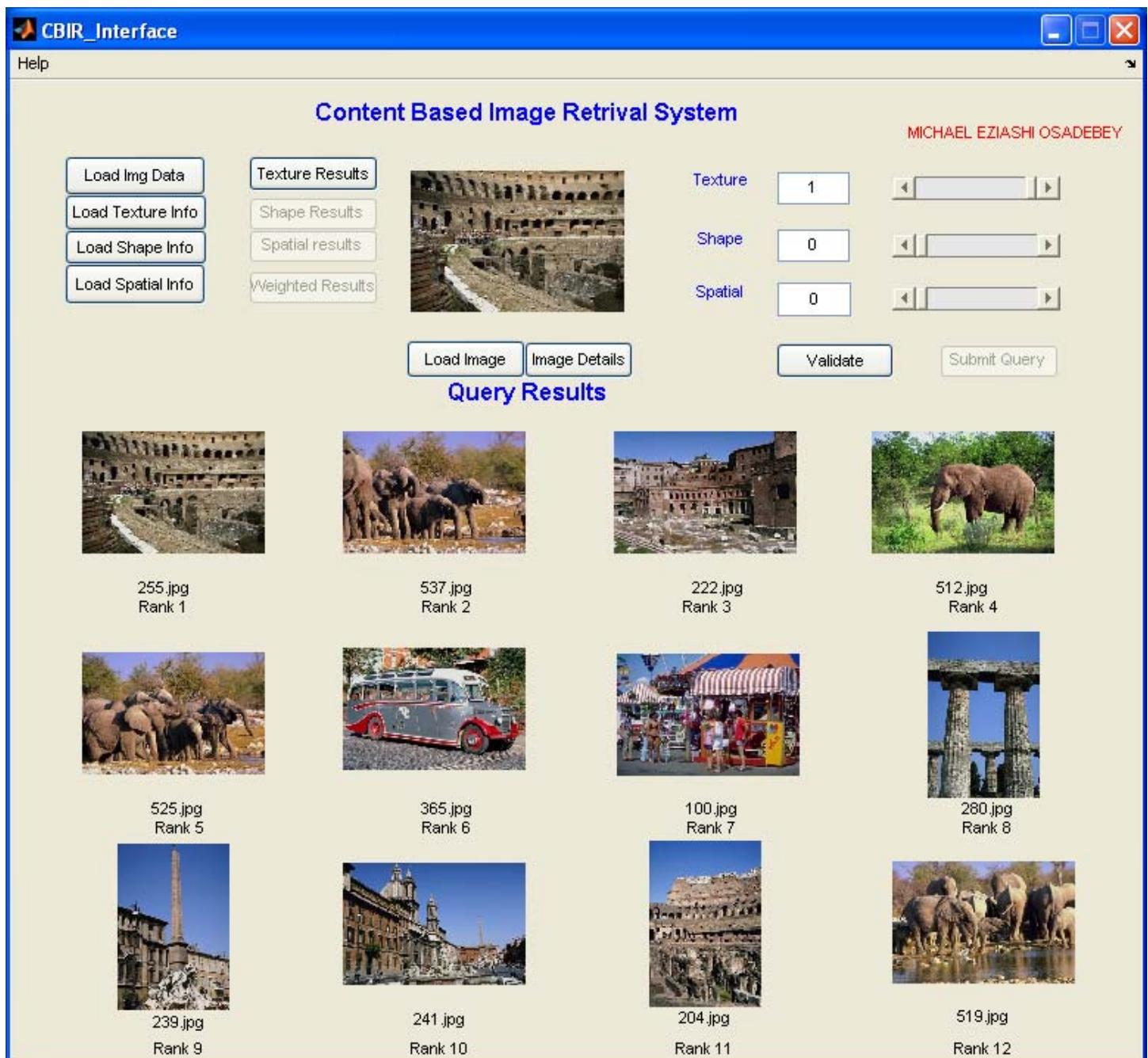


PSU - 45 X 45–First operation

ANNOTATION - Red textured flower

Precision - 11/12

APPENDIX C8.1

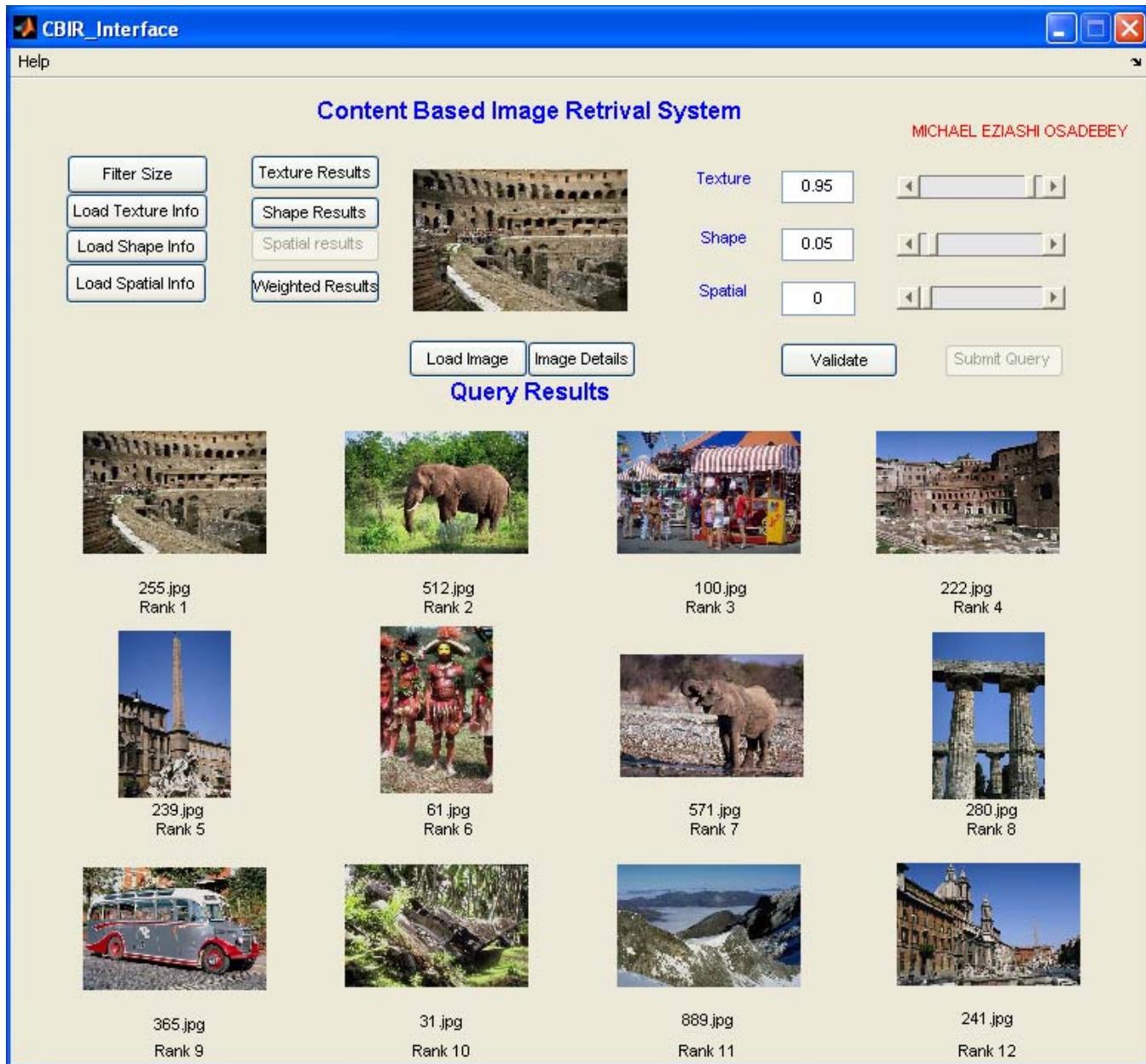


PSU - 25 X 25–First operation

ANNOTATION - Building

Precision =6/12

APPENDIX C8.2



PSU - 25 X 25-First operation

ANNOTATION - Building

Precision - 5/12

APPENDIX C9.1



PSU - 25 X 25–First operation

ANNOTATION - Bus

Precision - 9/12

APPENDIX C9.1

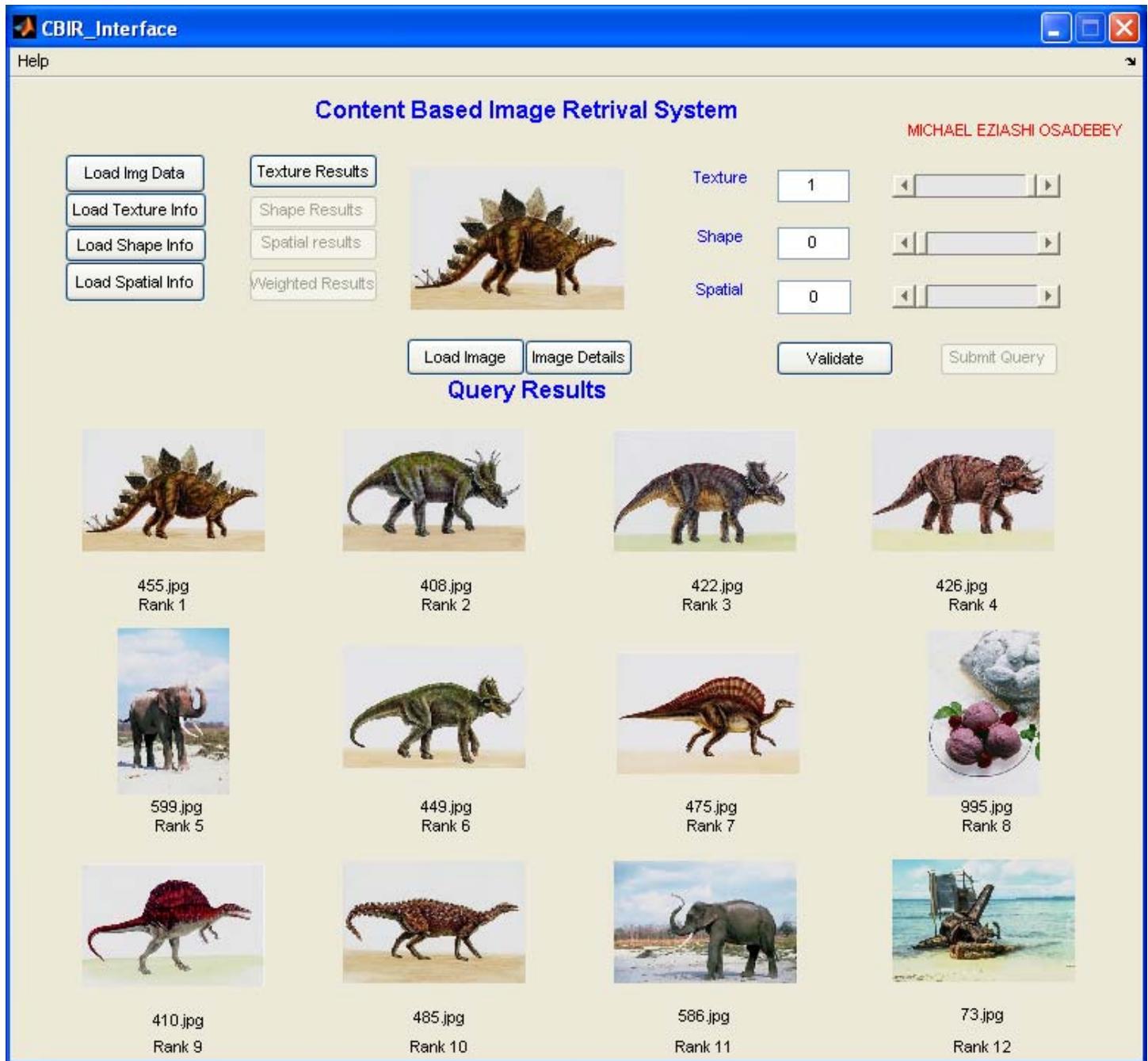


PSU - 45 X 45-First operation

ANNOTATION - Bus

Precision - 10/12

APPENDIX C10.1

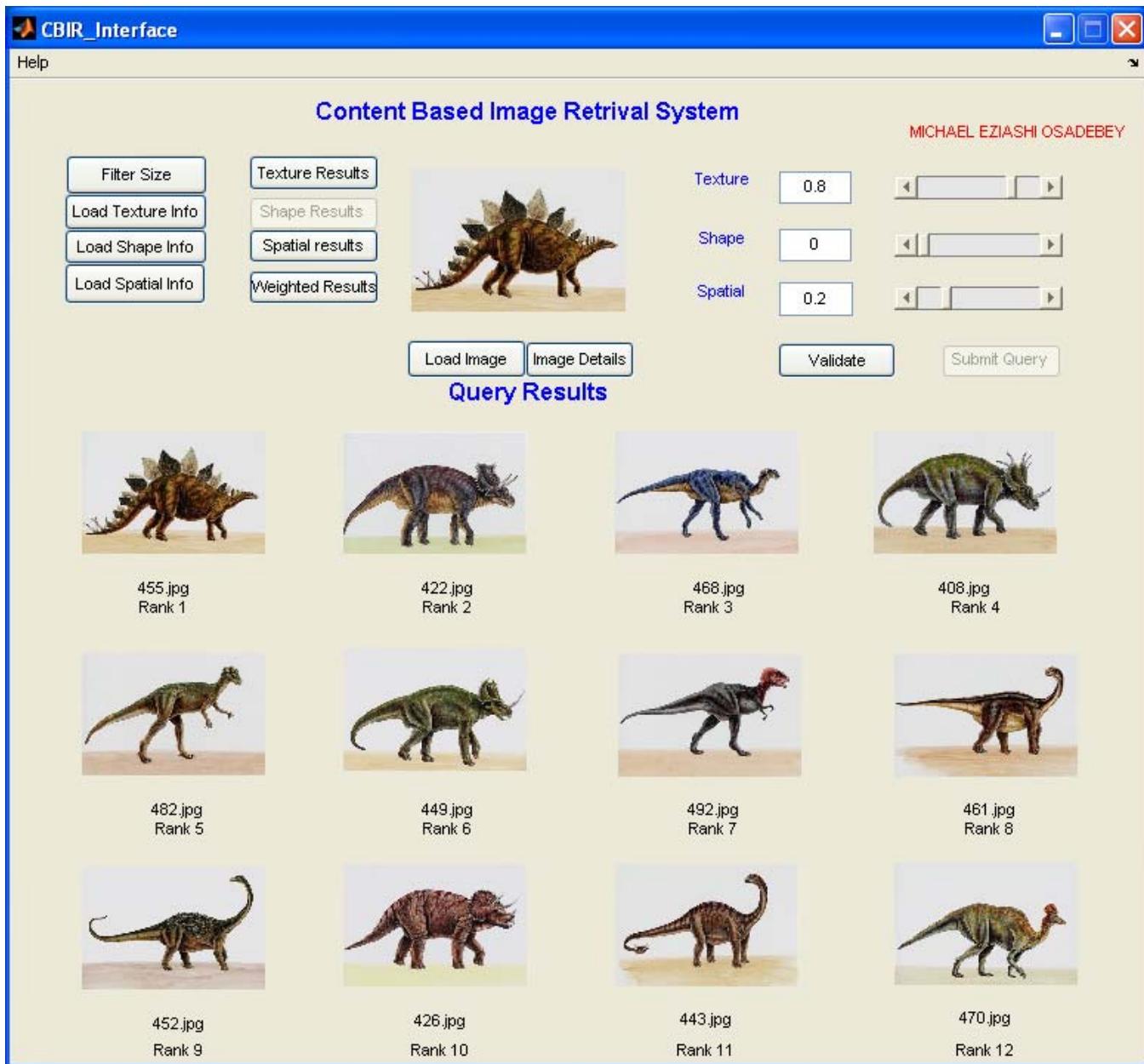


PSU 25 X 25–First operation

ANNOTATION: Kangaroo

Precision =8/12

APPENDIX C10.2



PSU - 45 X 45—First operation
ANNOTATION - Kangaroo
Precision - 12/12

APPENDIX D1.1

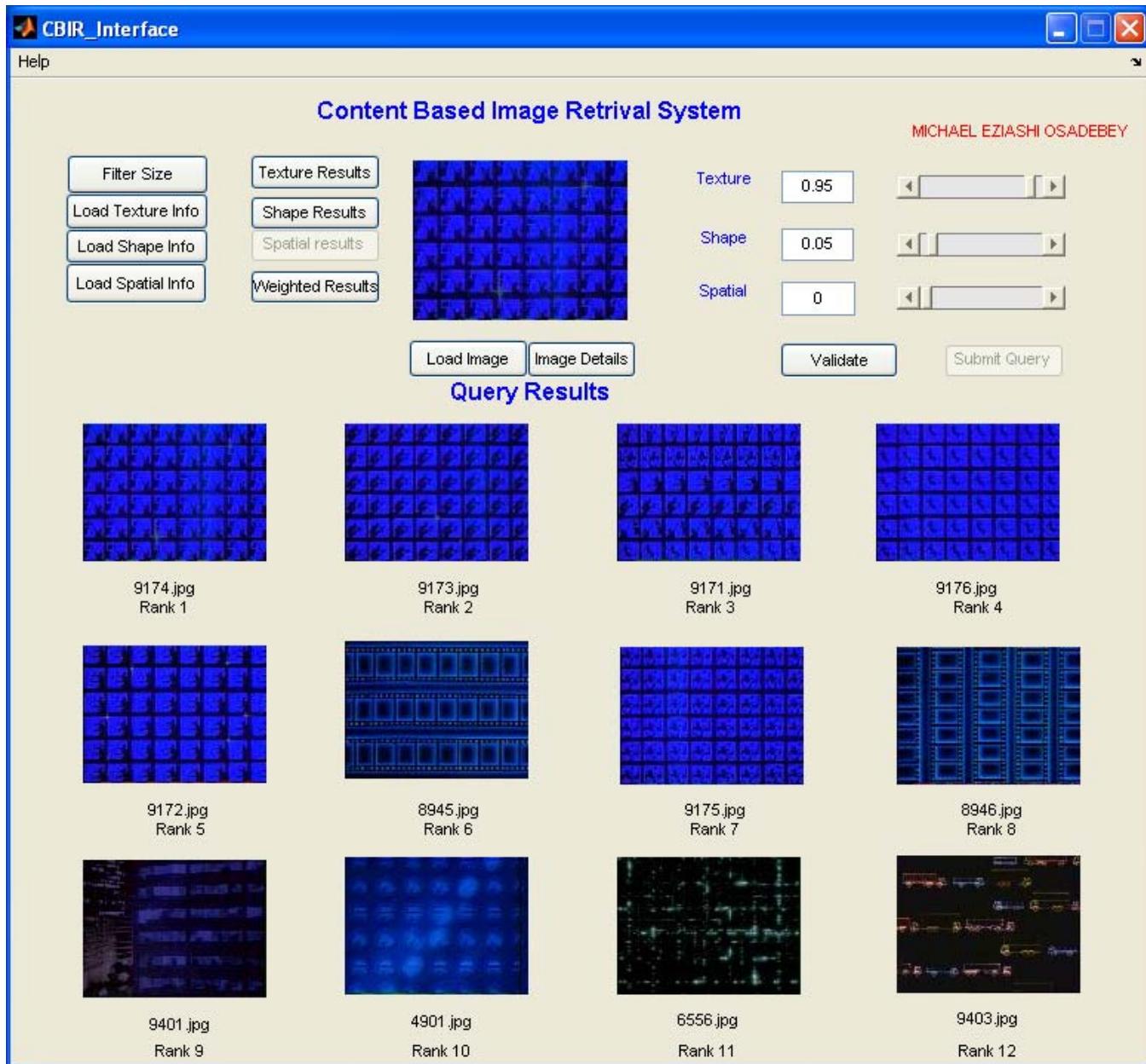


SU - 25 X 25–First operation

ANNOTATION - Blue-textured surface

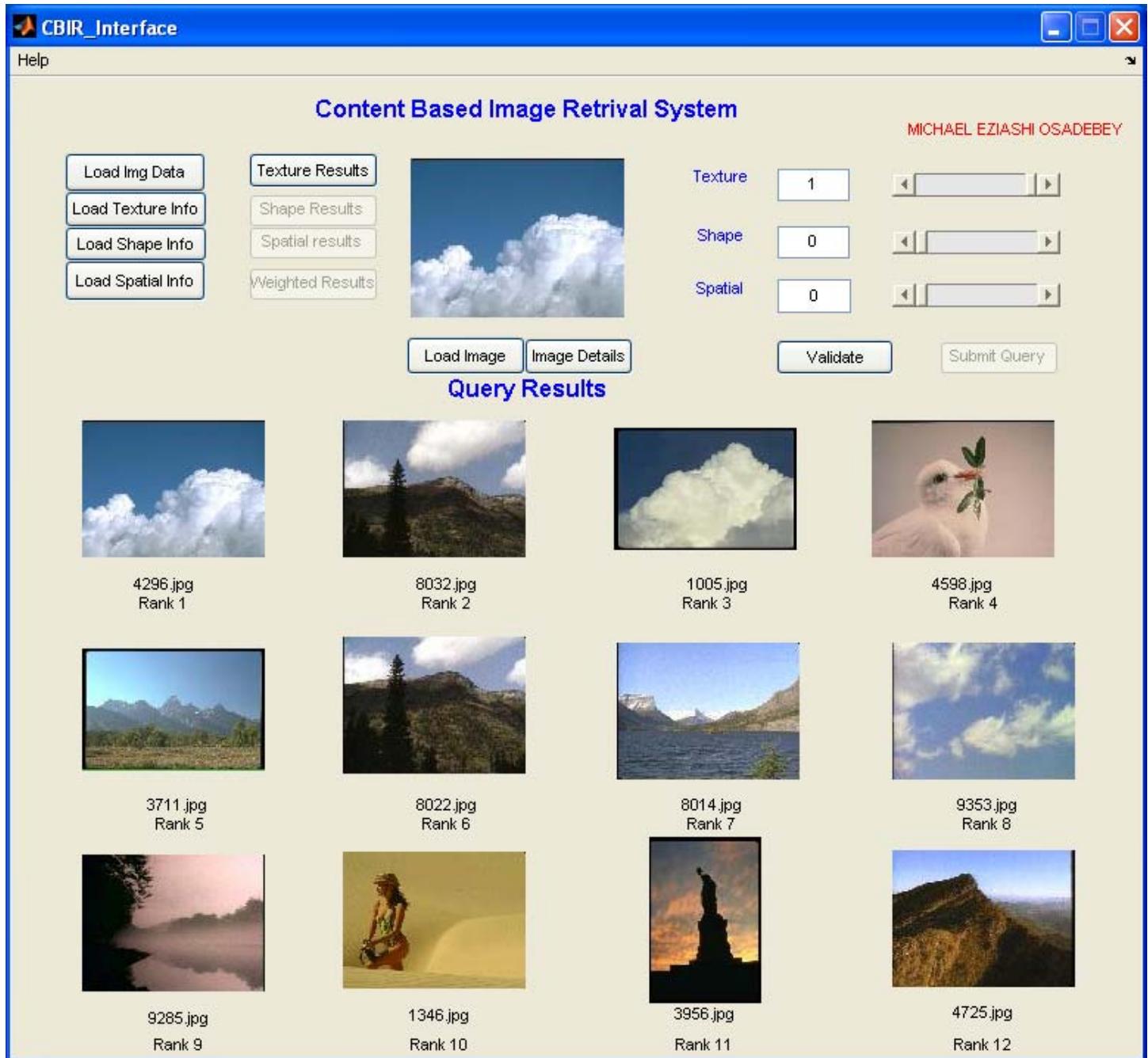
Precision - 8/12

APPENDIX D1.2



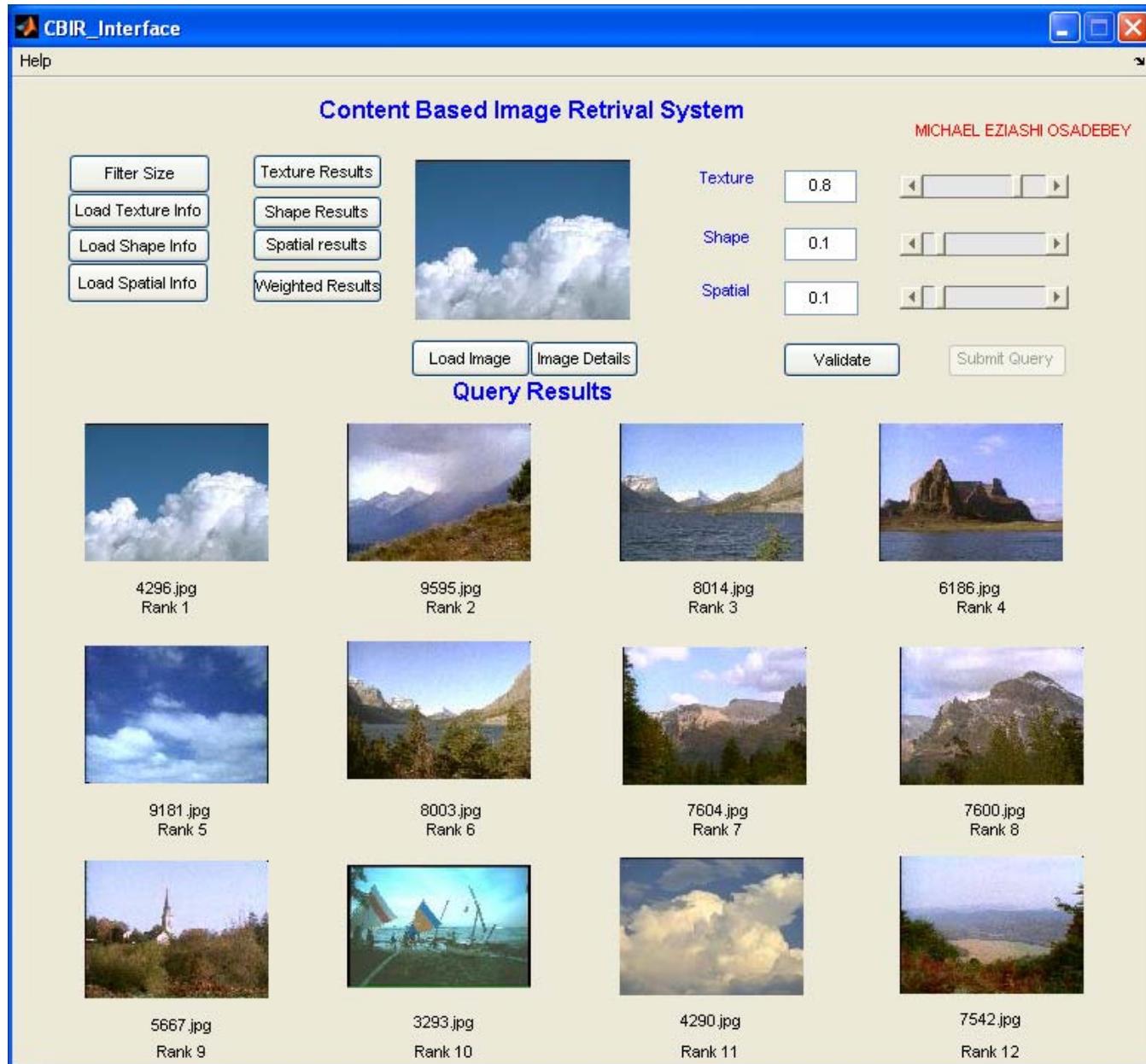
SU - 35 X 35–First operation
ANNOTATION - Blue-textured surface
Precision - 10/12

APPENDIX D2.1



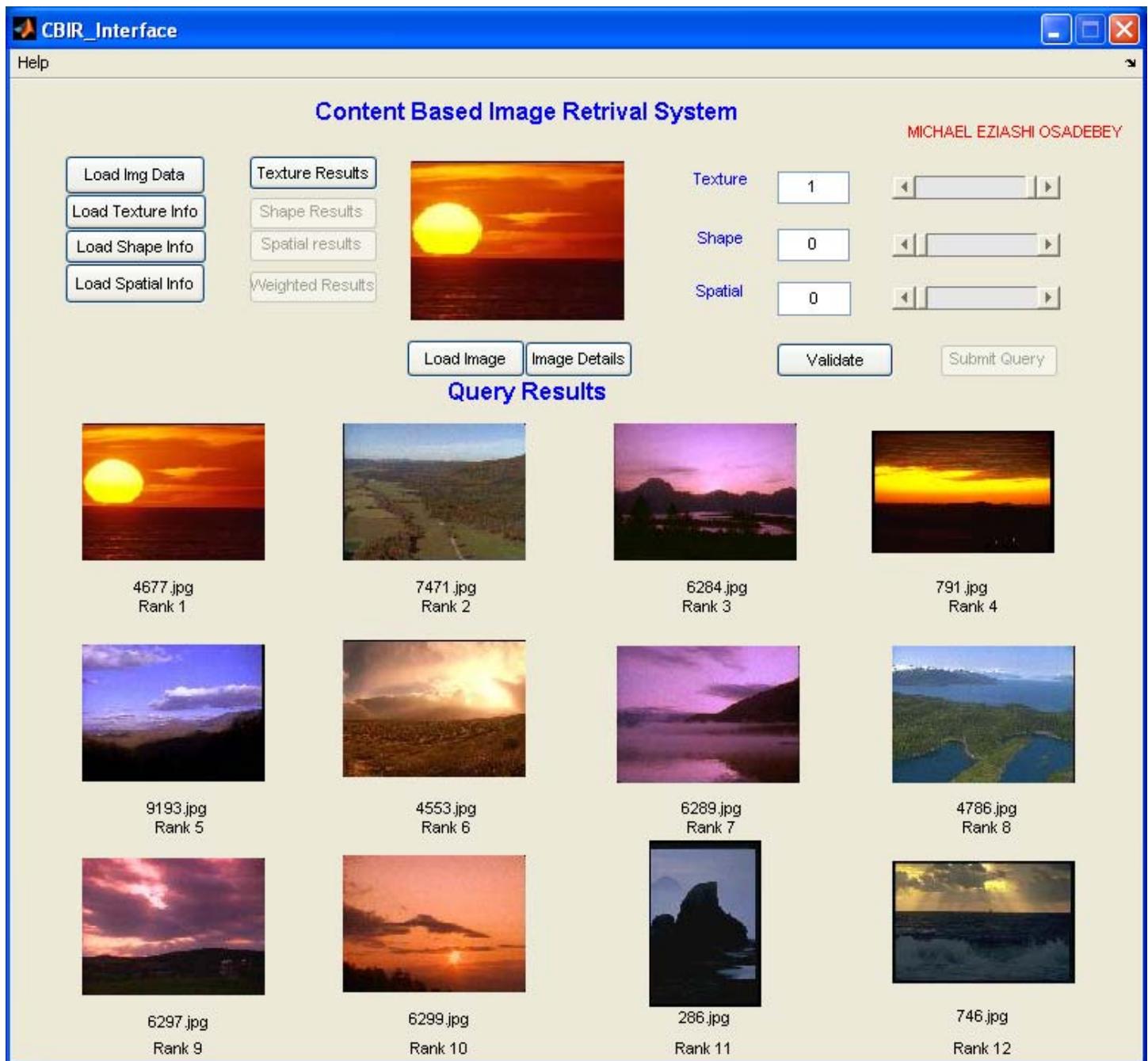
SU - 25 X 25–First operation
ANNOTATION - Blue-sky/cloud
Precision - 9/12

APPENDIX D2.2



SU - 35 X 35–First operation
ANNOTATION - Blue-sky/cloud
Precision - 9/12

APPENDIX D3.1

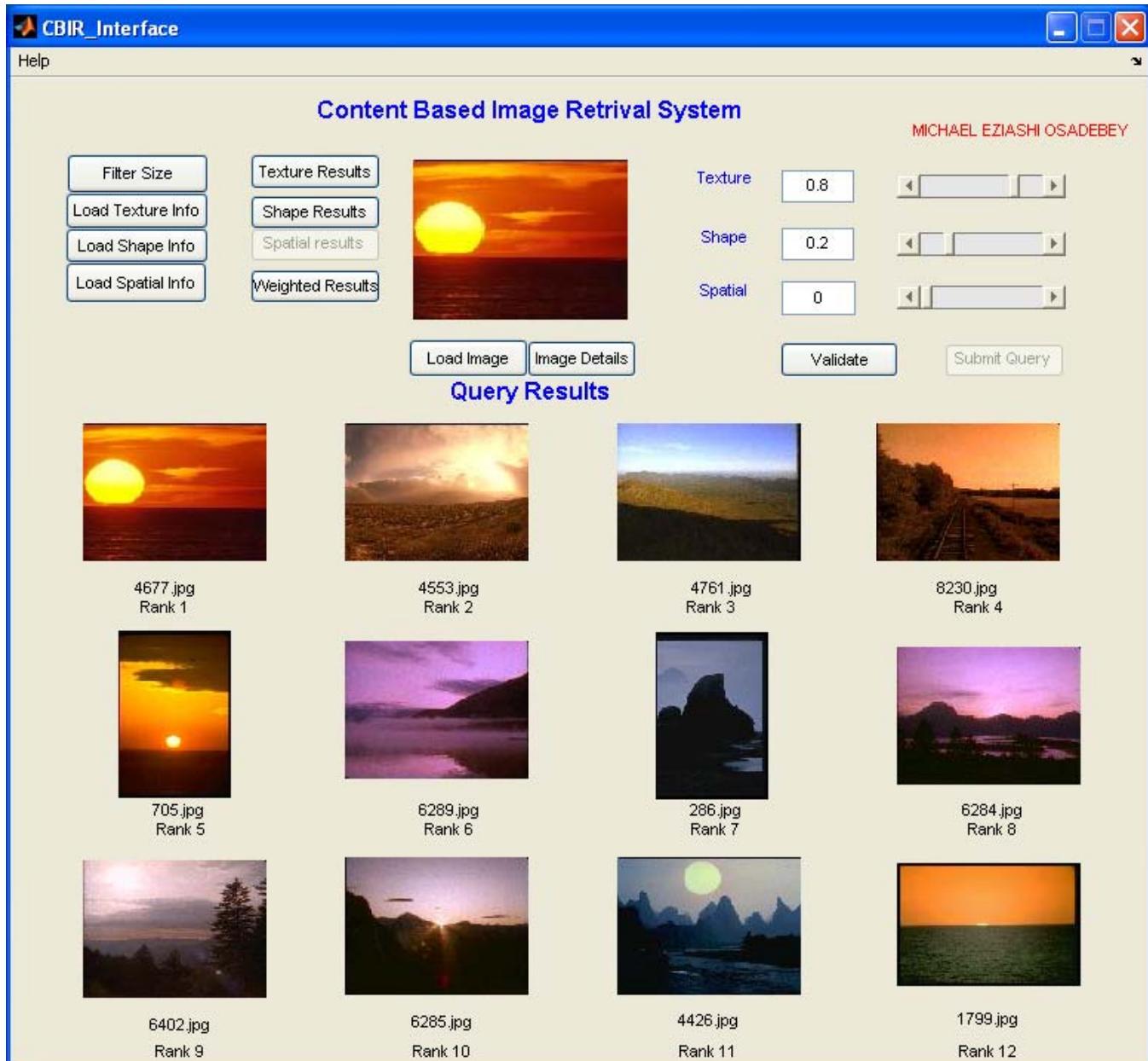


SU - 25 X 25—First operation

ANNOTATION - Reddish-yellow sky/circular object

Precision - 9/12

APPENDIX D3.2

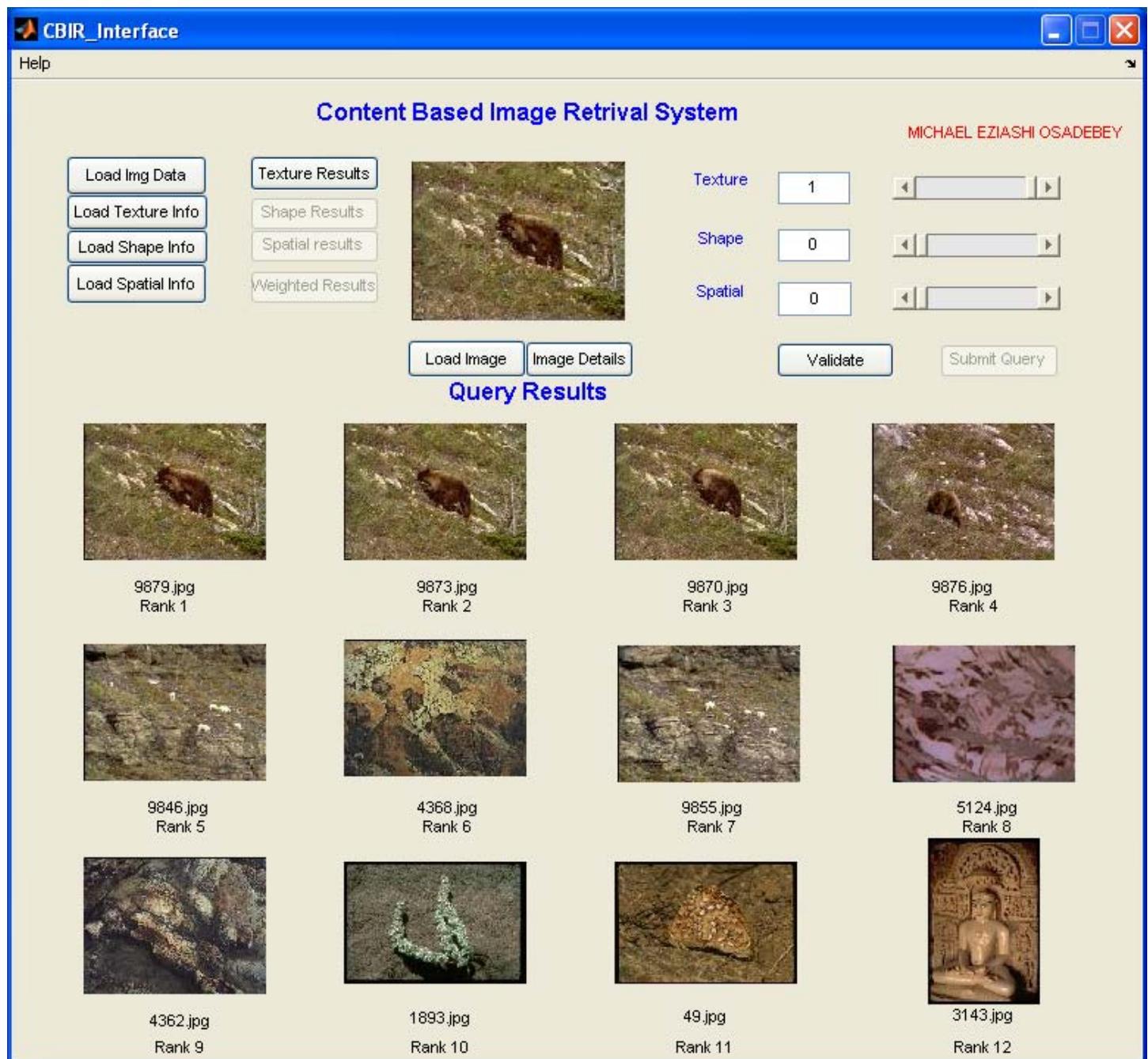


SU - 15 X 15–First operation

ANNOTATION - Reddish-yellow sky/circular object

Precision - 10/12

APPENDIX D4.1

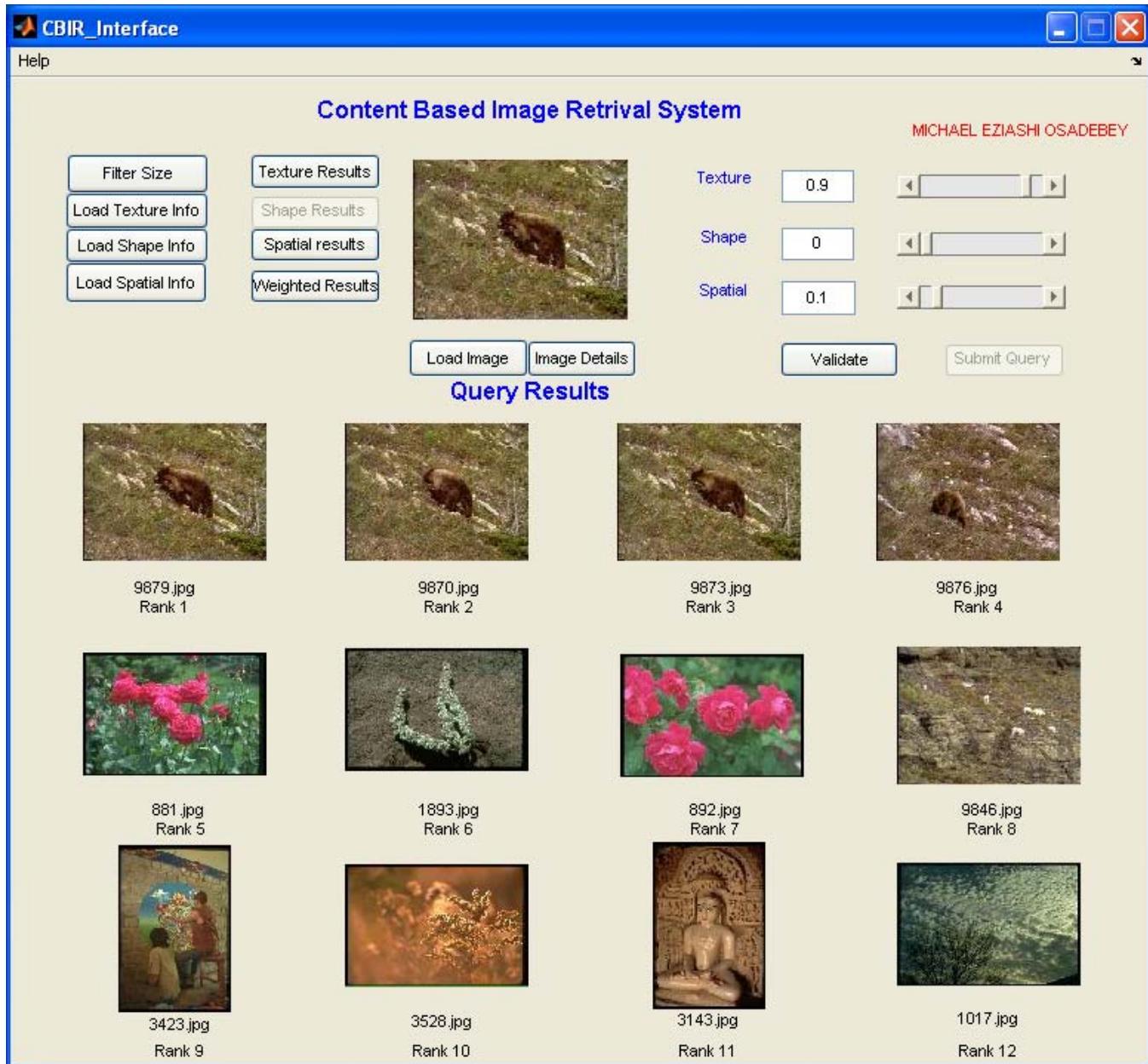


SU - 25 X 25–First operation

ANNOTATION - Brown animal/green-brown field

Precision - 9/12

APPENDIX D4.2

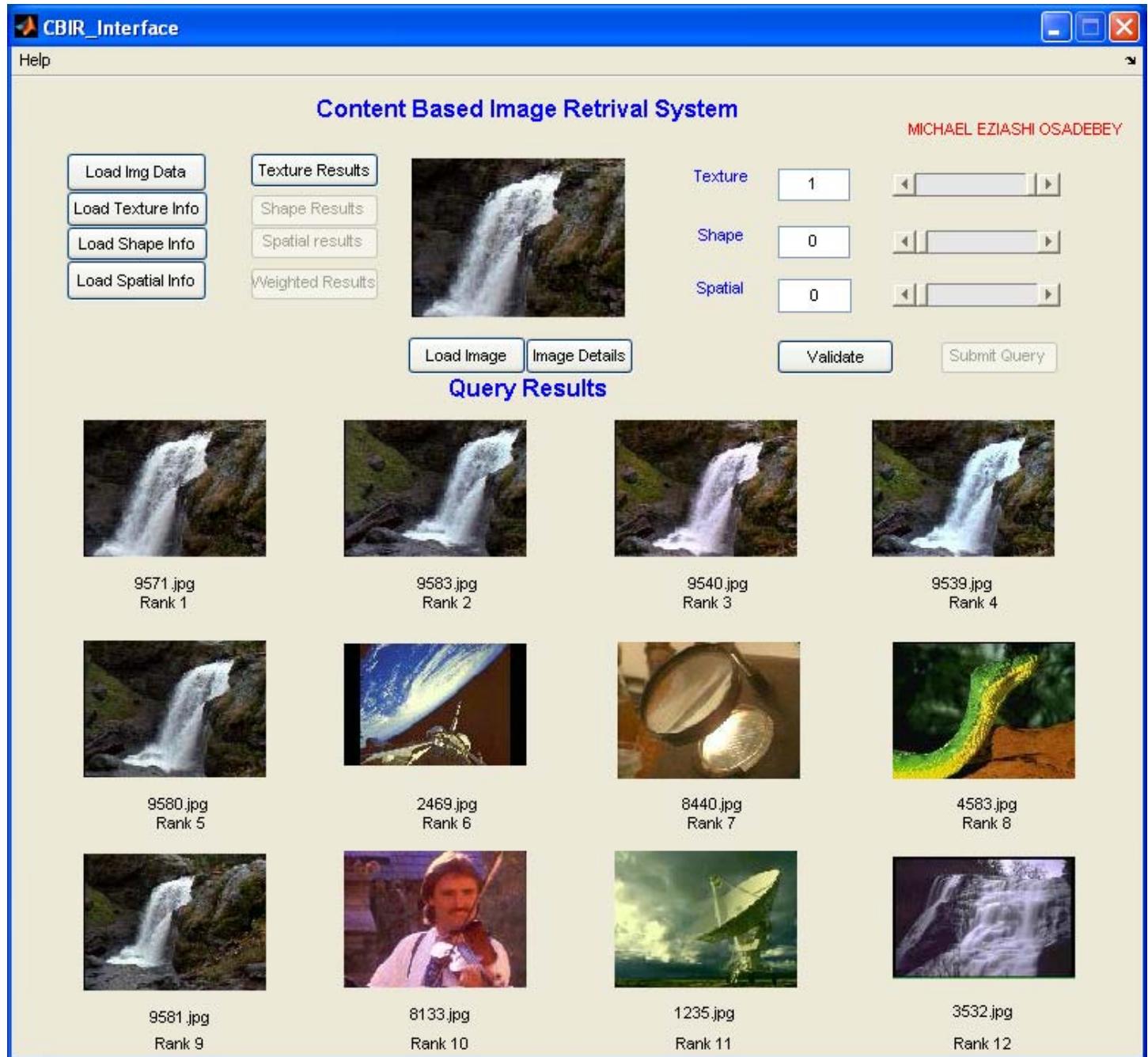


SU - 25 X 25–First operation

ANNOTATION - Brown animal/green-brown field

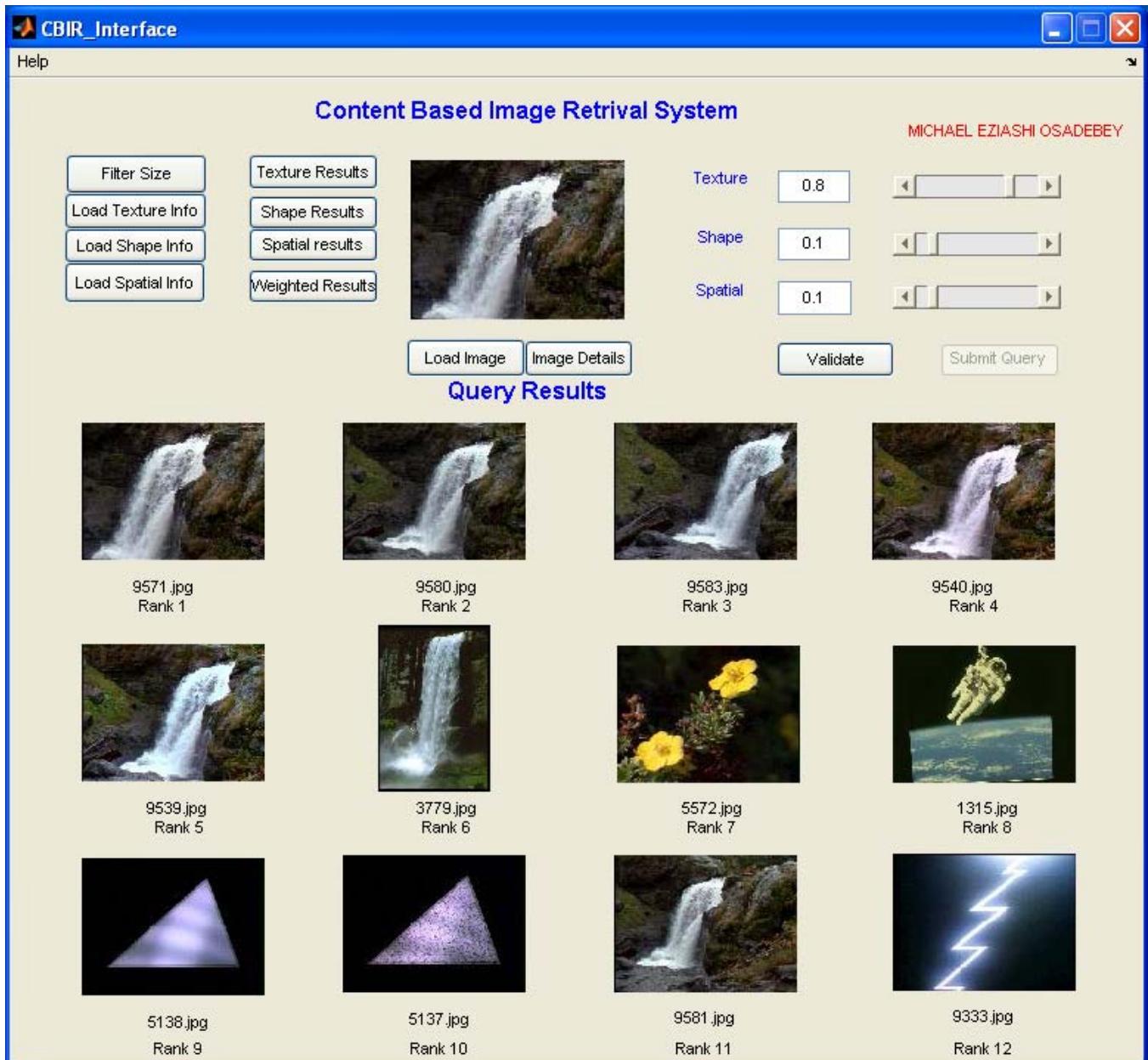
Precision - 6/12

APPENDIX D5.1



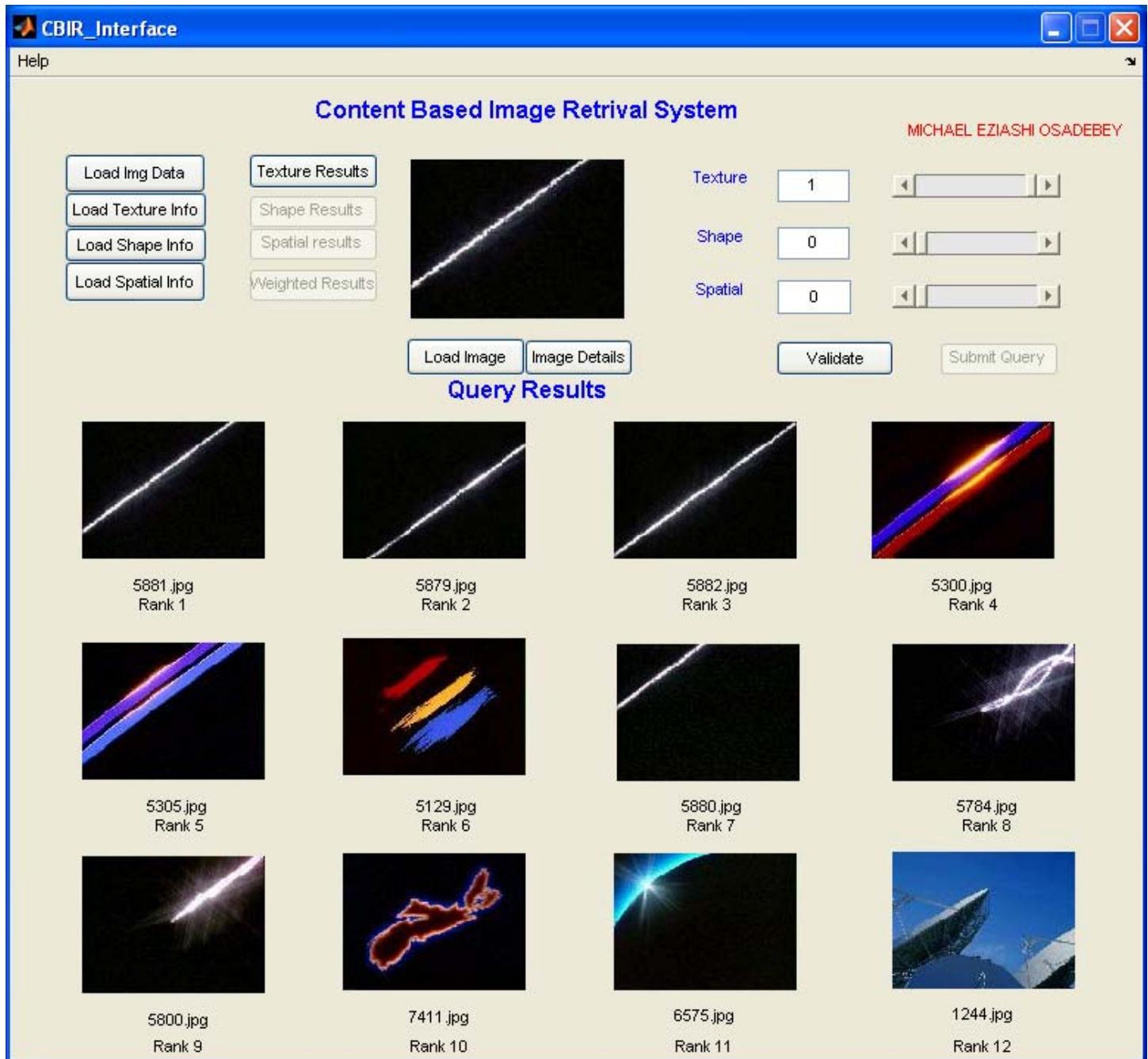
SU - 25 X 25–First operation
ANNOTATION - Water/Brown ridge
Precision - 7/12

APPENDIX D5.2



SU - 55 X 55–First operation
ANNOTATION - Water/Brown ridge
Precision - 7/12

APPENDIX D6.1

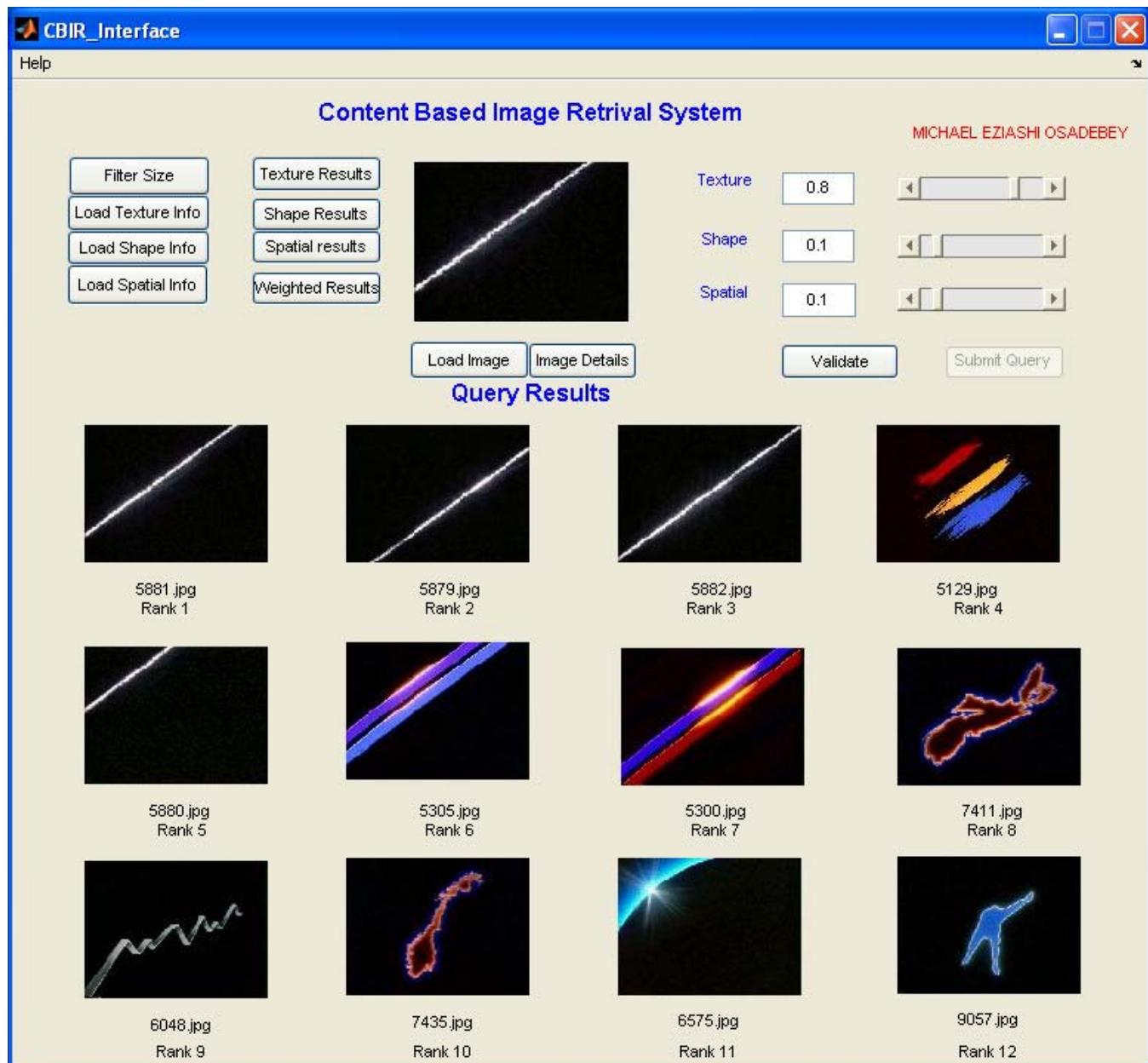


SU - 25 X 25–First operation

ANNOTATION - Black background stripe

Precision - 11/12

APPENDIX D6.2

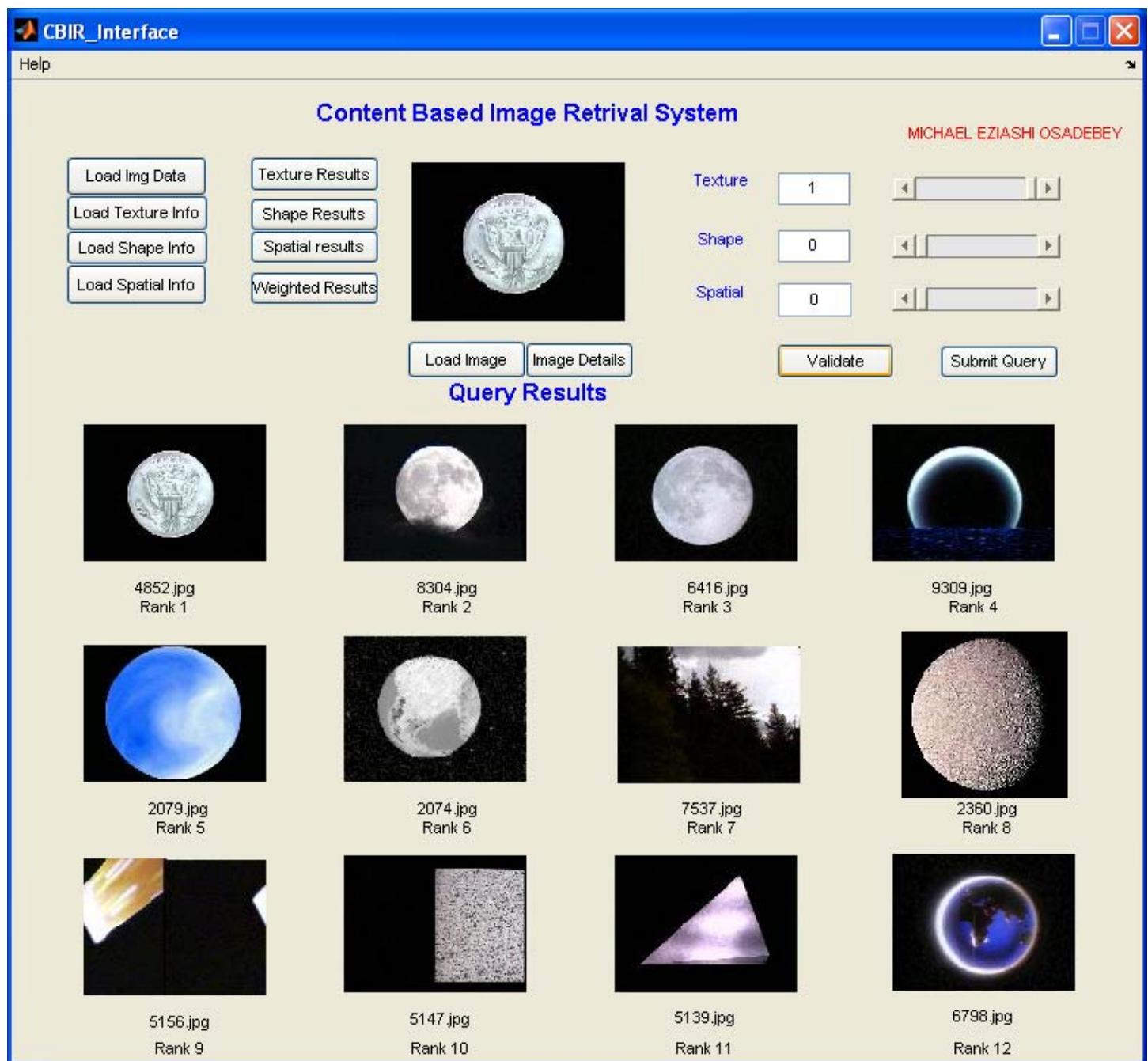


SU - 55 X 55–First operation

ANNOTATION - Black background stripe

Precision - 12/12

APPENDIX D7.1

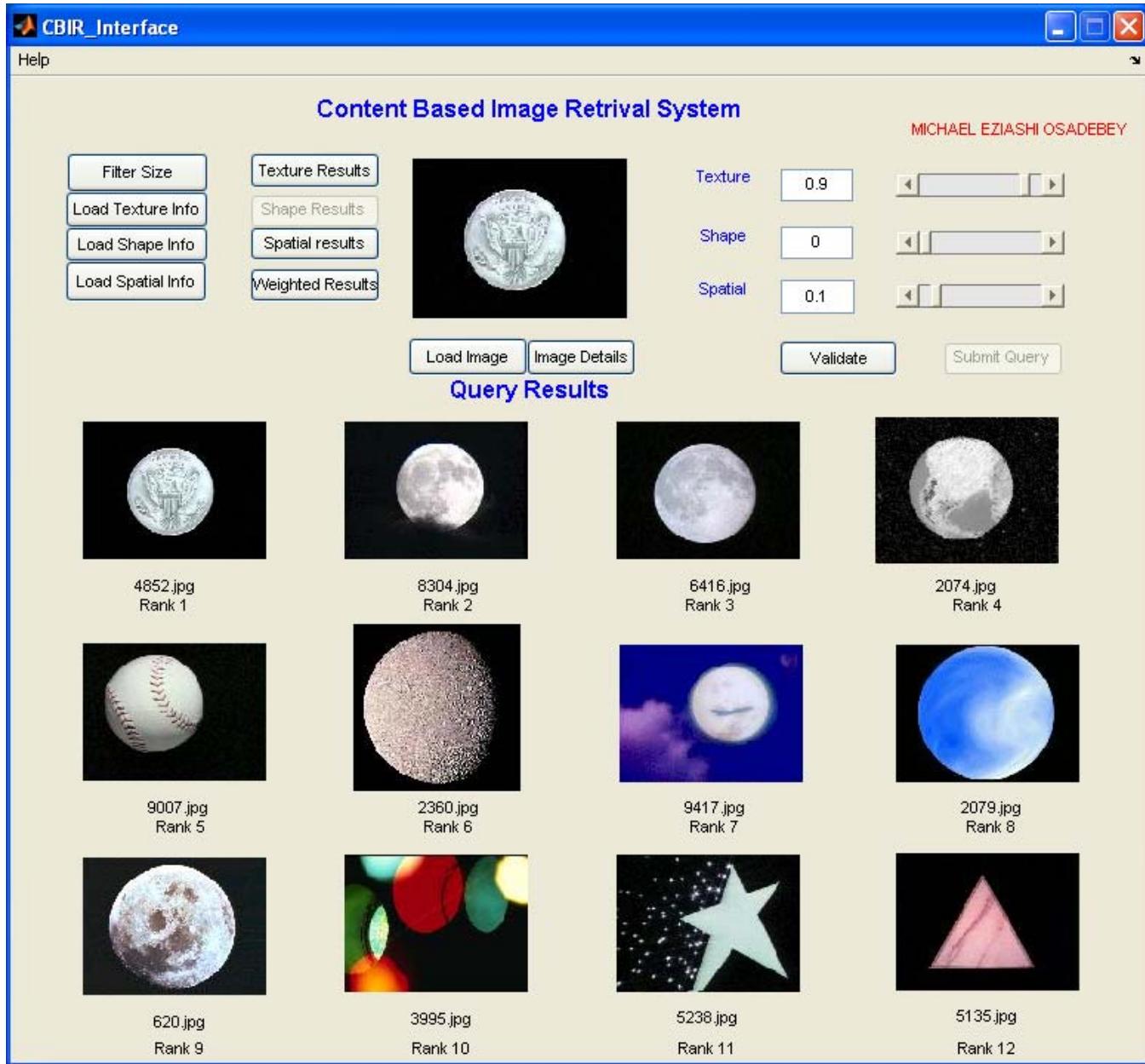


SU - 25 X 25–First operation

ANNOTATION - Circular object/White object

Precision - 8/12

APPENDIX D7.2

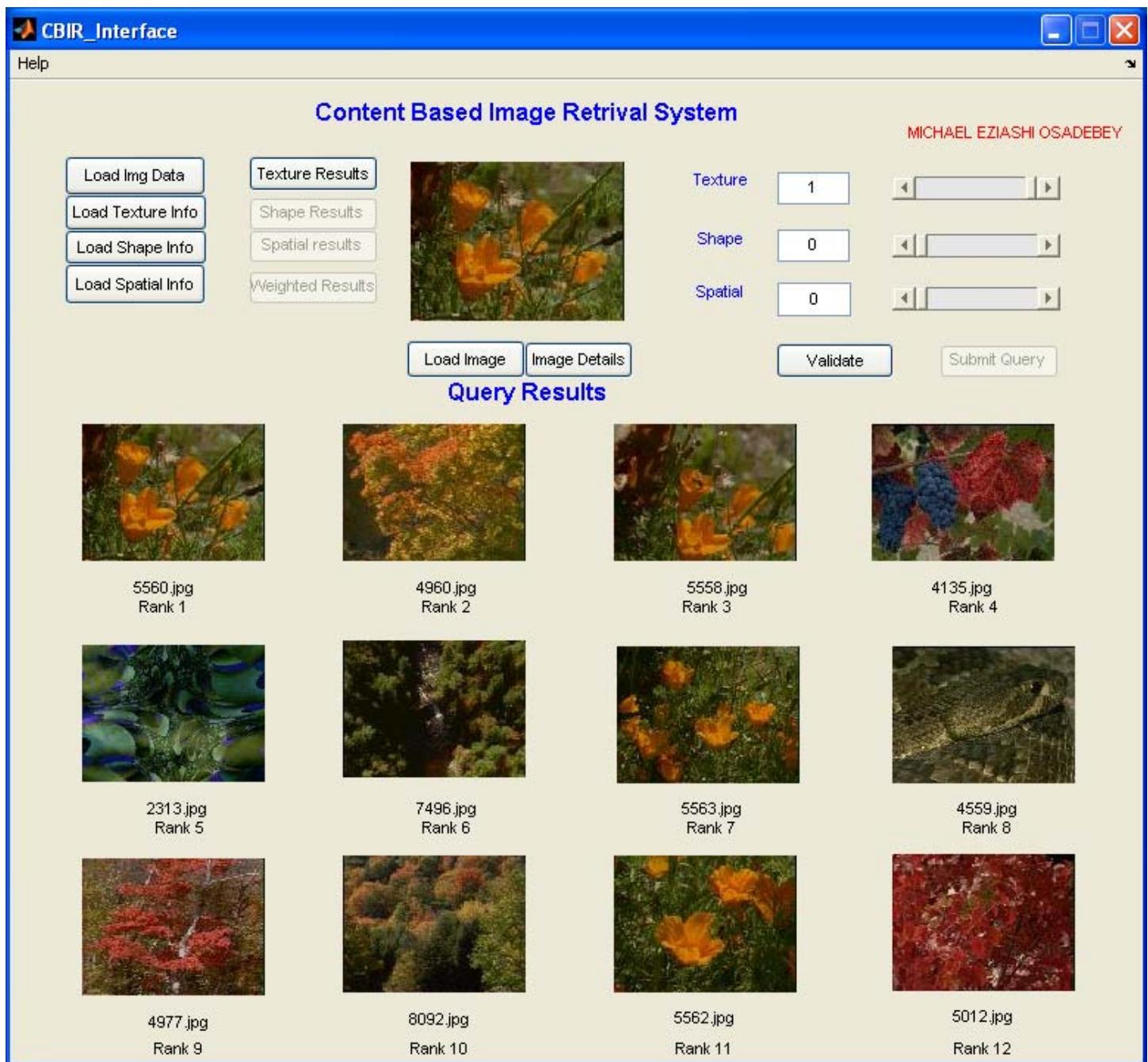


SU - 55 X 55–First operation

ANNOTATION - Circular object/White object

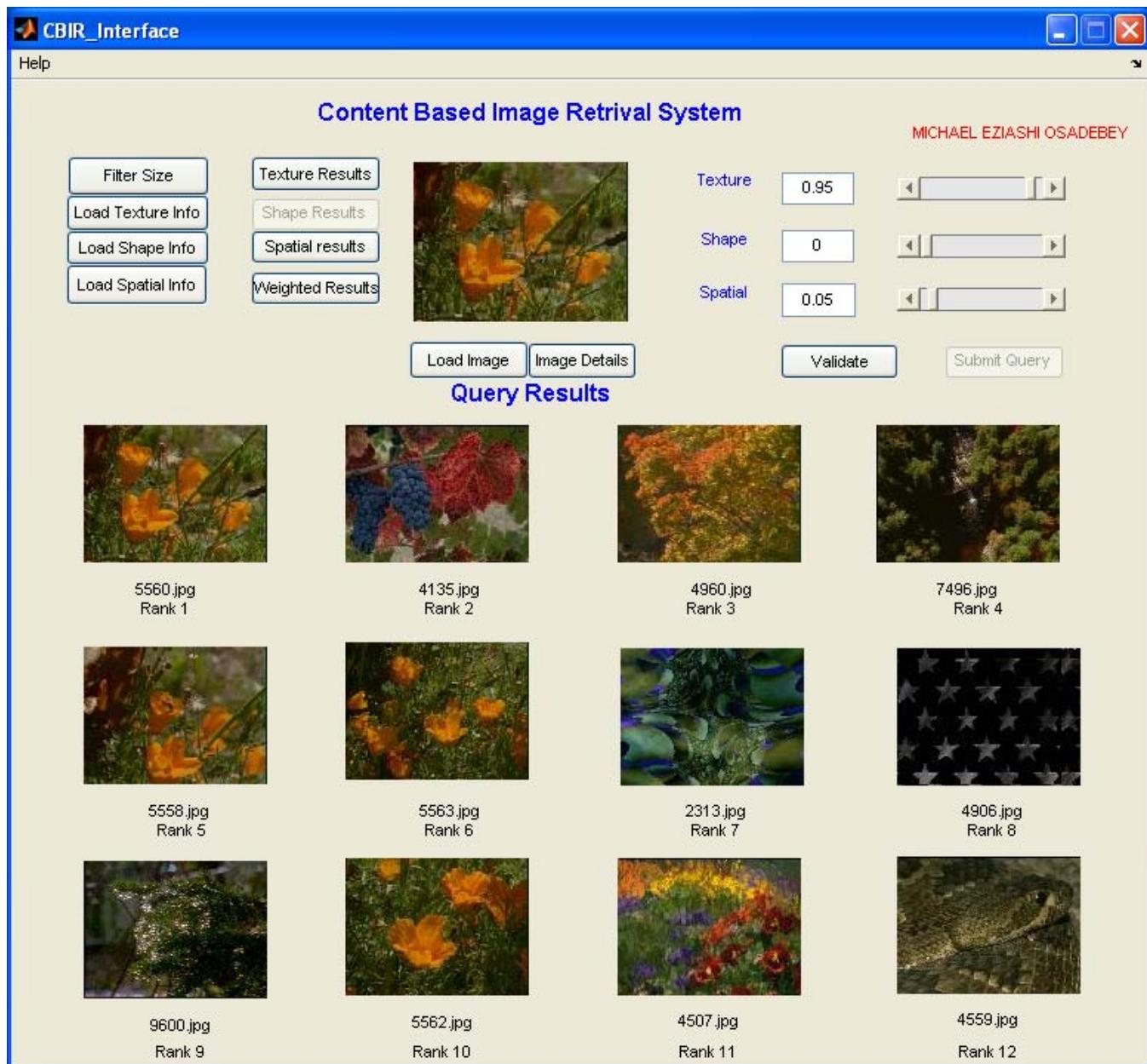
Precision - 10/12

APPENDIX D8.1



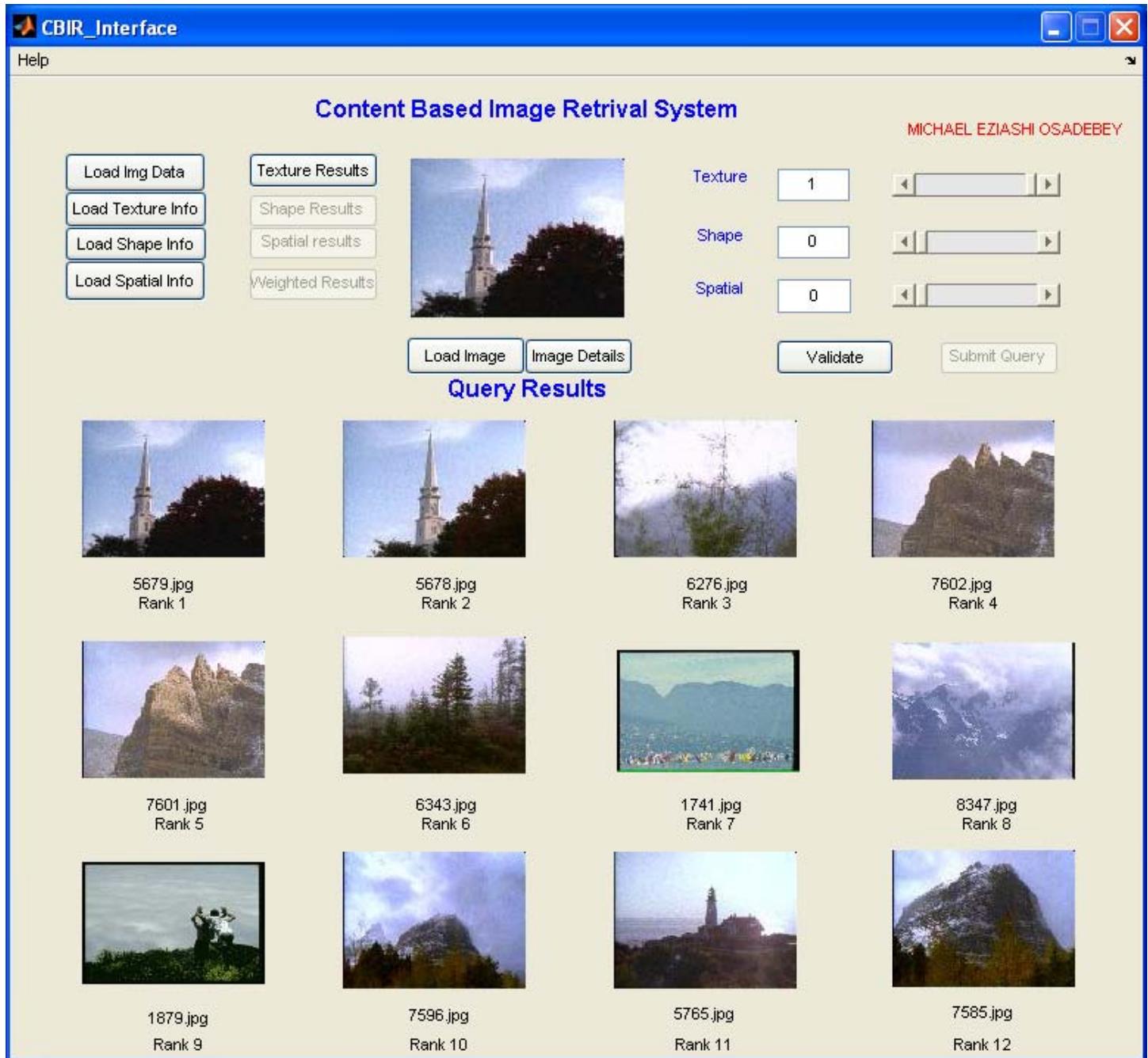
SU - 25 X 25–First operation
ANNOTATION - Flower/green field
Precision - 10/12

APPENDIX D8.2



SU - 15 X 15–First operation
ANNOTATION - Flower/green field
Precision - 8/12

APPENDIX D9.1

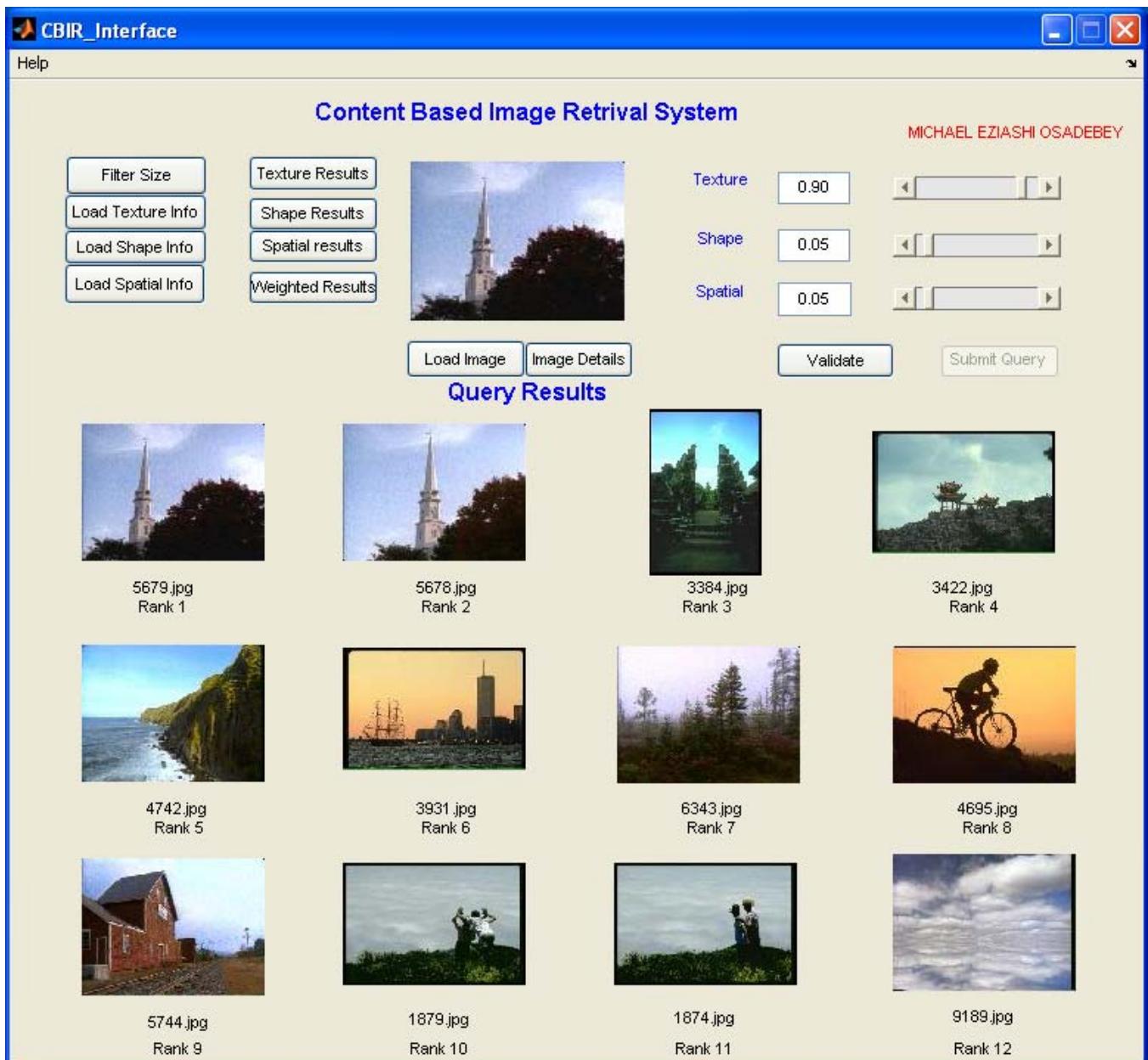


SU - 25 X 25–First operation

ANNOTATION - Building/sky/black forest

Precision = 10/12

APPENDIX D9.2



SU - 45 X 45–First operation

ANNOTATION - Building/sky/black forest

Precision - 12/12

APPENDIX 10.1



SU - 25 X 25 –First operation

ANNOTATION - Stars/Red stripes/ stripe object

Precision - 4/12

APPENDIX D10.2



SU – 25 X 25 First operation
ANNOTATION - Stars/Red stripes/ stripe object
PRECISION - 4/12