THE FLORIDA STATE UNIVERSITY
COLLEGE OF ARTS AND SCIENCES

A DYNAMIC IMAGE QUERY SYSTEM USING PHP

BY

YU WANG

A project submitted to the
Department of Computer Science
in partial fulfillment of the
requirements for the degree of
Master of Science

Major Professor: Dr. Xiuwen Liu

Degree Awarded

Fall Semester, 2002

The members of the committee approve the Master's Project of **Yu Wang** defended on December 2, 2002

Dr. Xiuwen Liu
Major Professor

Dr. Hilbert Levitz
Committee Member

Dr. David Whalley
Committee Member

ACKNOWLEDGEMENTS

I thank my major professor, Dr. Xiuwen Liu, for his continuous guidance, inspiration, and enthusiasm. His timely suggestions and encouragement made the completion of this project possible. Also, I would like to thank Dr. Levitz and Dr. Whalley for their time and directions.

My whole heart thanks go to my former supervisor Dr. Wayne Sprague who has shown me the true professionalism, given me career guidance, constantly encouraged me in overcoming difficulties, and offered me unselfish supports. I would also like to thank my friends in the system group and the department. They made my graduate study here at FSU most memorable.

This work is dedicated to my parents and my wife for their great care and love.

TABLE OF CONTENTS

ABSTRACT

As the digital images and videos become an important source of information, effective ways of browsing large collections become a necessary tool for many applications. Based on a spectral histogram representation, a preliminary program was developed at Florida State University to query an image dataset based on perceptual similarity, which, however, does not provide a user-friendly interface.

The focus of this project is to provide an effective interface for the preliminary query program. The project was designed to provide such a dynamic image query system that consists of a number of online interfaces for researchers and users to interact with the image query system. These interfaces allow users to view and submit images to the image query system online either from collections or from their own images. The dynamic image query system then computes the image's representation and displays pages of matched images with the corresponding distance.

The dynamic image query system is written in PHP4 language. It adopts the server side implementation. All the interfaces and pages are generated dynamically.

The project is implemented with security in mind. Carefully designed security features made the system very robust.

1. INTRODUCTION

With the advances of digital sensor technologies and ever increasing popularity of the world wide web, digital images and videos become increasingly available and an important source of information in many application areas. As browsing large collections is inefficient and labor intensive, content-based image and video retrieval becomes an important tool for many applications. A typical content-based image retrieval system consists of two main modules. The first module is the representation of images and videos, where essentially a perceptually meaningful distance between images needs to be defined. The second module is the visualization of the results. As a typical dataset consists of thousands of images, visualizing and navigating the results becomes an important part of the system.

In this project, we used spectral histograms to represent images and distance between images is defined as the chi-square distance (or other distance measure) between the corresponding spectral histograms. A preliminary image query system based on the spectral histograms representation was built. Tests have shown the approach using the Spectral Histograms is effective. However, without an efficient visualization module, it is difficult to evaluate the results effectively, and thus difficult to carry the research further in the most effective way.

The focus of this project is to develop a visualization module for the preliminary content-based image retrieval system. The system can be accessed through the Internet using traditional web browsers any where in the world. It allows users to submit query images from displayed collections or from uploading their own images. When the image query system receives the submitted image, it computes the image's spectral histograms and returns users matched images with the corresponding distances. Side by side display makes it easy to do comparisons and evaluate the accuracy of the result. Thus, users can make more constructive suggestions to the research.

In order to make the system dynamic, a server side implementation has been used and PHP, a high performance programming language that was designed specifically for this purpose, was selected to build the system.

The system is designed with future expansion in mind. Current implementation has two independent modules: Color Images and B&W Texture Images. Each module uses its own program files and configuration files and runs independently. If in the future we want to add new image systems, we can simply copy a whole set of modules and change or add several configuration files and put a link into main page. At this point the new system is ready to run.

Accuracy of the query results and response time are two main factors to evaluate the performance of the dynamic image query system. The system was designed to have high performance to reflect the advancement of the research.

This report is organized as follows. Chapter 2 gives a very brief overview of spectral histograms. Chapter 3 summarizes the design principles and design choices while Chapter 4 provides implementation details and features of the system; screenshots are provided for illustration. Chapter 5 concludes the report with some suggestions on further improvement of the system.

2. BRIEF OVERVIEW OF SPECTRAL HISTOGRAMS

Based on psychophysical and neurophysiological data, it is widely accepted that the human visual system transforms a retinal image into a local spatial/frequency representation. Such a representation can be simulated by a bank of filters with tuned frequencies and orientations and finds applications in many areas. For texture modeling, filter responses themselves are not adequate. Based on psychophysical studies, histograms of filtered images are shown to be effective to characterize textures. This motivates a generic representation of images, which we call *spectral histograms*. A spectral histogram consists of marginal responses of a chosen set of filters, or histograms. As colors are responses at different wavelengths, visual patterns are responses of different filters. Analogous to color spectrum, we name the visual feature spectral histograms**.** Extensive studies have demonstrated that the spectral histogram exhibits desirable properties that are consistent with human visual perception. For example, it exhibits a nonlinearity which matches the psychophysical data on texture perception very well.

The following table shows four texture images along with its spectral histogram of eight filters. Each texture has a unique spectral histogram. Similar textures tend to have similar spectral histograms as shown in these examples.

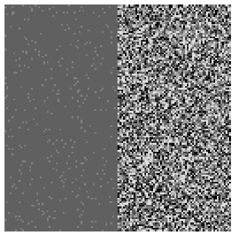
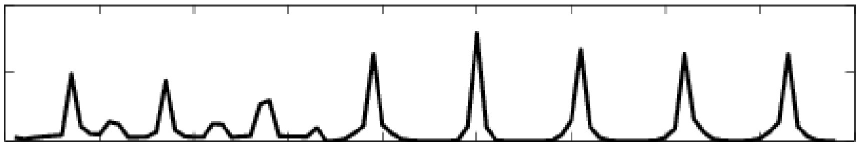
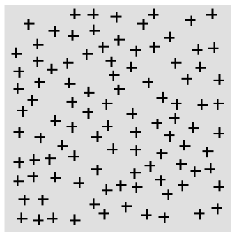
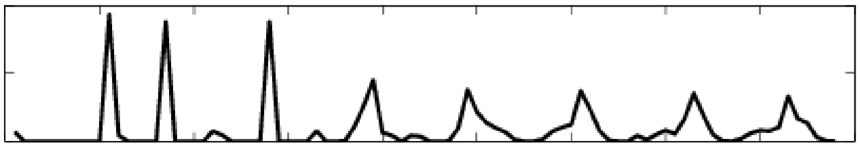
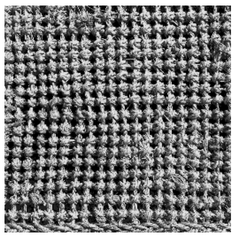
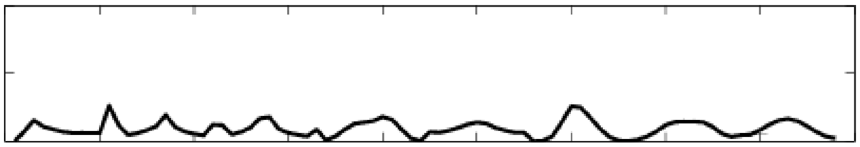
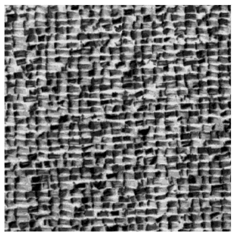
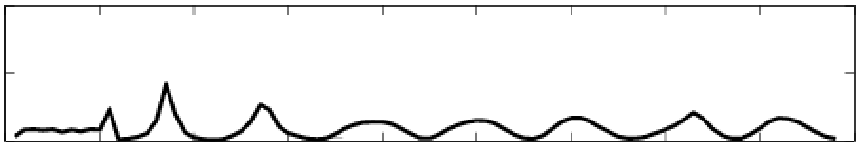
Texture Image	Spectral Histogram
	
	
	
	

Table 2.1

The spectral histogram provides a way to overcome some of the problems in texture modeling, classification, and segmentation. Besides texture modeling, spectral histograms have also been used for face recognition and 3D object recognition. The universality of the spectral histograms on various kinds of images makes it a good candidate as a representation in a content-based image retrieval. Please refer to [1] and references therein for further details.

3. DYNAMIC ONLINE IMAGE QUERY SYSTEM

3.1 System design principle

The main purpose of the dynamic online image query system is to provide a “window” to allow other researchers and users to “view” our research and to evaluate the Spectral Histograms representation in real time. We took advantage of the Internet and the World Wide Web to design and set up a web environment to access the image query system. Through the web environment, researchers and users can view and submit query images to the image query system and see the results computed and returned by the query system immediately. By interacting with the image query system in real time, other researchers and users can evaluate the performance of the system. It also provides useful feedback to help us to further improve the representation based on spectral histograms.

Since the online dynamic image query system is an important window to the world for our research, our design principle is to make the query system a professional appearance, a quick response, and a friendly environment with easy to use interfaces and highly secure mechanisms.

In order to fulfill the principle design, I have worked with my major professor, Dr. Liu, closely and have made a careful selection of the system structure and programming language.

3.2 Server side implementation

In the modern implementation of dynamic web programming (DHTML), there are two major methods: server side implementation and client side implementation. Server side implementation or programming is the kind of implementation that runs on the remote web server side, does web processing on the server side and outputs the web pages to the client’s web browser. Client side implementation is to put dynamic contents in the client side. For example, we can use JavaScript and Flash to make some dynamic effects.

This online image query system mainly used the server side implementation. The benefits of the server side implementation are:

- On the server we can make our programs read through large amount of data without the client having to download them first.
- Only the output is sent to the client. This means we can make our program invisible from the end-user and prevent them from looking at the source code to get some sensitive data.
- We can make multiple programming languages work together to take advantage of each programming language.

The drawback is that once the output is sent to the end user it cannot do anything more. That is where client side implementation kicks in. Many designers combine the two to maximize the dynamic effects. Since our purpose is to dynamically query the image query system, we did not exploit client side implementation features heavily. We only created several Flash buttons and embedded them into some interfaces to give the system some better interfaces.

3.3 Programming language selection

ASP (Active Server Page), PHP (PHP Hypertext Preprocessor), and JSP (Java Server Page) are three main programming languages used by current server side implementation. ASP runs mainly on a Windows platform while PHP and JSP can run on any platform. For JSP the web server must have JVM (Java Virtual Machine) and Tomcat installed; for PHP the Zend engine must be installed.

We selected PHP as the programming language to build the system for the reasons listed below:

- PHP is incorporated with Apache so it is multi-threaded, which means the program loaded into memory can be shared by multiple requests to decrease the use of system resources.

- PHP is created to manage images and web applications. PHP has rich build in functions that make image manipulation much easier. For instance, the getimagesize() function in PHP can be used to check if a file is an image file. If it is an image file, the function can also determine what type of image file it is. This gives us great convenience when we check user uploaded image file types.
- PHP has already been implemented in our department's system. This is one of the important factors that made us decide to use PHP. We do not have to put extra work load on our system group and they will take care of upgrades and security fixes.
- PHP is a scripting language. There is no compilation required. That means we do not have to worry about our program's platform compatibility issue.
- PHP is based on Perl and adopted C program style. Programmers who have C program experience or training can learn it quickly. PHP also has Perl's flexibility in string manipulation.
- PHP has wide user bases. People from all over the world help each other. I ran into several technical problems and was able to solve them through exchanging ideas with others.
- Finally, database support is one of the PHP's most significant features. With PHP, creating web pages with dynamic content from a database is remarkably simple. Although at this time we do not have a database implemented in this system, PHP gives us the possibility to move in that direction when we need it in the future.

4. IMPLEMENTATION DETAILS OF THE QUERY SYSTEM

4.1 System structure

The dynamic image query system consists of a number of web interfaces. Each web interface's action depends on the input (action) sent from another interface. Each interface is also capable of passing its data to another interface. The logical relationship among these interfaces and actions are as shown in Figure 4.1:

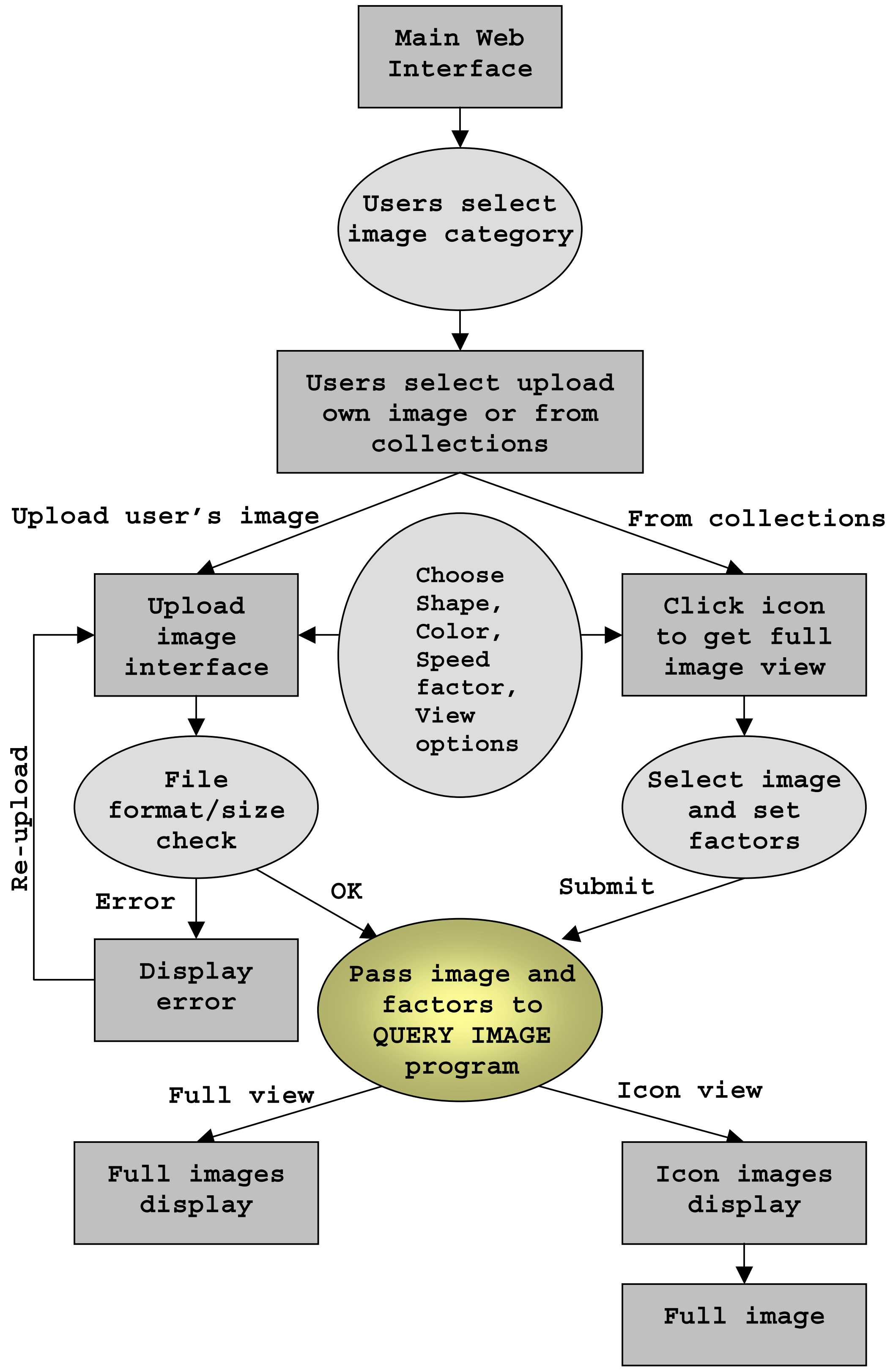
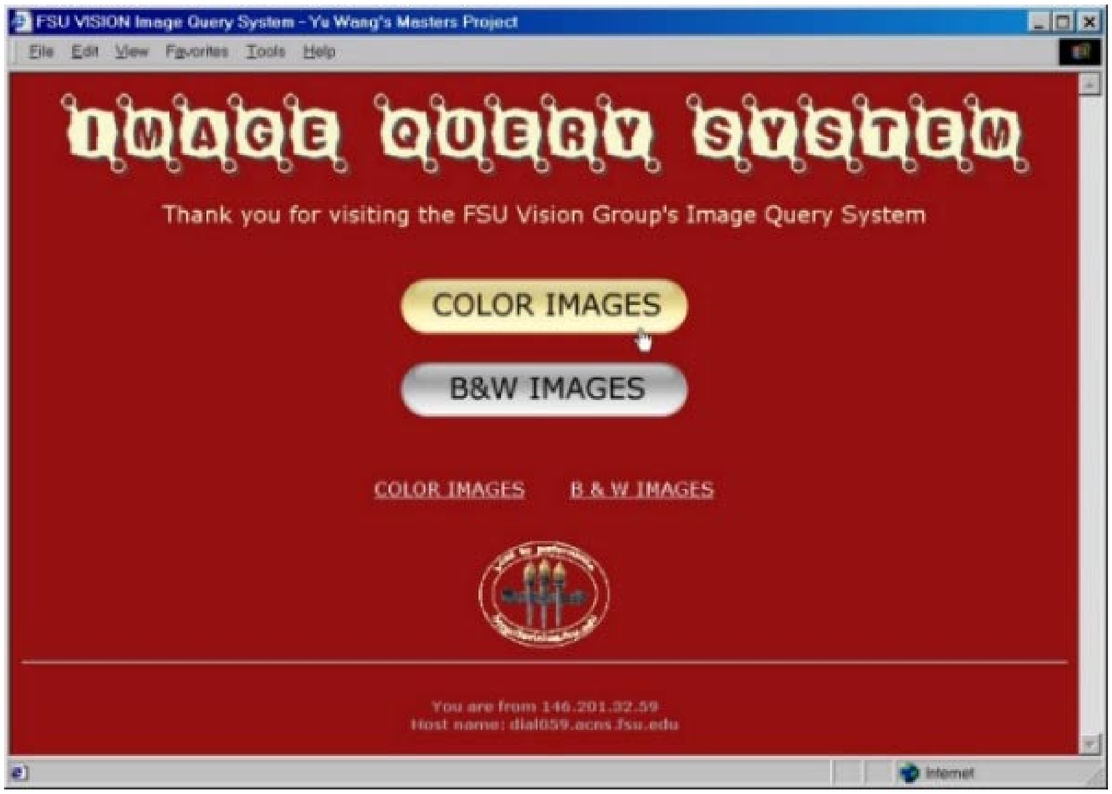

Main Web Interface
Users select image category
Users select upload own image or from collections
Upload user's image
From collections
Upload image interface
Choose Shape, Color, Speed factor, View options
Click icon to get full image view
Re-upload
File format/size check
Select image and set factors
Error
OK
Submit
Display error
Pass image and factors to QUERY IMAGE program
Full view
Icon view
Full images display
Icon images display
Full image

Figure 4.1 Diagram of interfaces and actions relationships

Details of implementations of interfaces and actions are described as follows.

4.1.1 Main Page – index.php

The index.php script displays the main page which greets visitors as shown in Figure 4.2. It provides users two major image query systems: Color Image query system and B&W Texture Image query system.


FSU VISION Image Query System - Yu Wang's Masters Project

Thank you for visiting the FSU Vision Group's Image Query System

You are from 146.201.32.59
Host name: dial059.acns.fsu.edu

Figure 4.2 Main Interface

By clicking the flash links, the user enters the corresponding image query system. Since Color Image and B&W Texture Image have similar system implementations and the Color Image system is slightly more complicated than B&W Text Image system, we will use the color system to do the illustration.

The main page also displays link to FSU Vision Group's main page. It shows the visitor's IP address and host name for greeting and for tracking.

4.1.2 Image submission selection system

The imgformcolor.php and imgformbw.php scripts give users two choices for submitting images to the image query system: upload their own images or submit from displayed collections (Figure 4.3)

Click the UPLOAD IMAGES link will enter the image file uploading system. If user does not want to upload their own images, they can choose images from displayed collections.

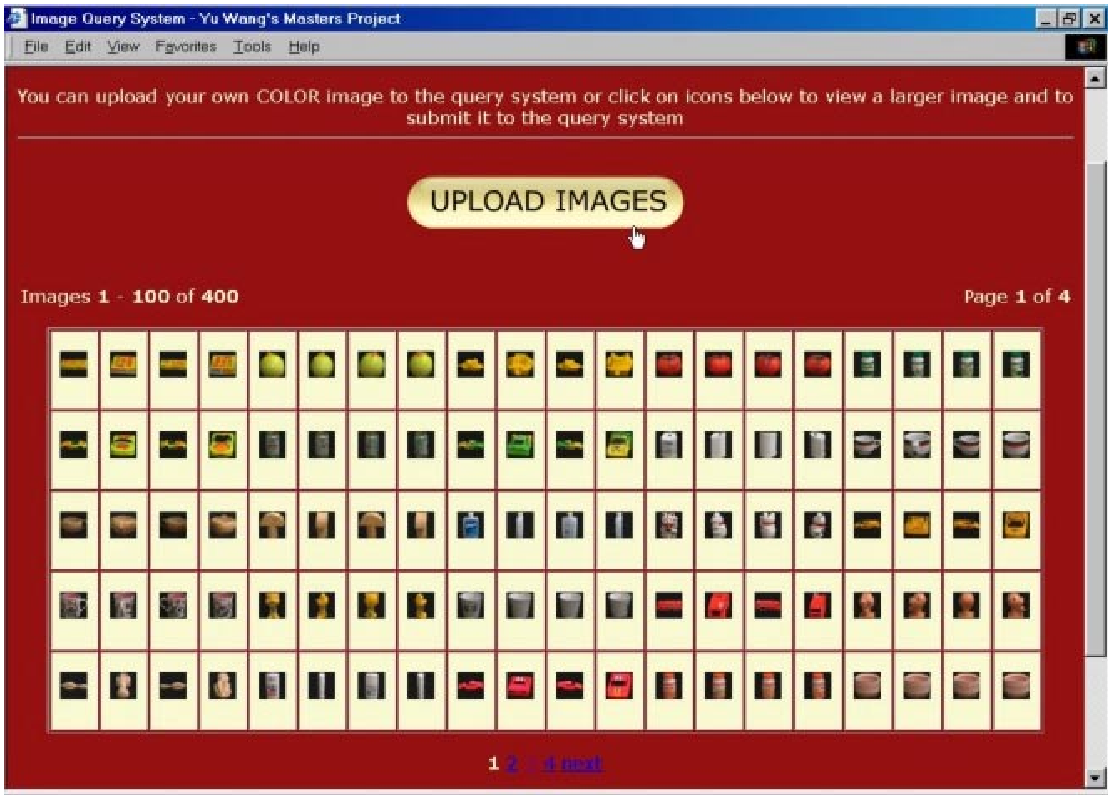


Image Query System - Yu Wang's Masters Project
File Edit View Favorites Tools Help
UPLOAD IMAGES
1 2 3 4 next

You can upload your own COLOR image to the query system or click on icons below to view a larger image and to submit it to the query system

Images 1 - 100 of 400

Page 1 of 4

Figure 4.3 Image submission selection interface

There are hundreds of images in the query system's database. In order for users to comfortably view the collections, we display them in multiple pages. In each page, we display a limited number of images. The limit can be set by changing the value in the configuration file.

In considering that users may have slow network connection and we also want to display as many images per screen as possible, the system displays icon images in its collections. If users want to submit an image, they can click the icon image to view a full size version image.

4.1.3 Image uploading system

If users want to submit their own images, the Flash button link (Figure 4.3) will bring them to an interface (Figure 4.4) to allow them to upload their images (procolorfile.php and probwfile.php).

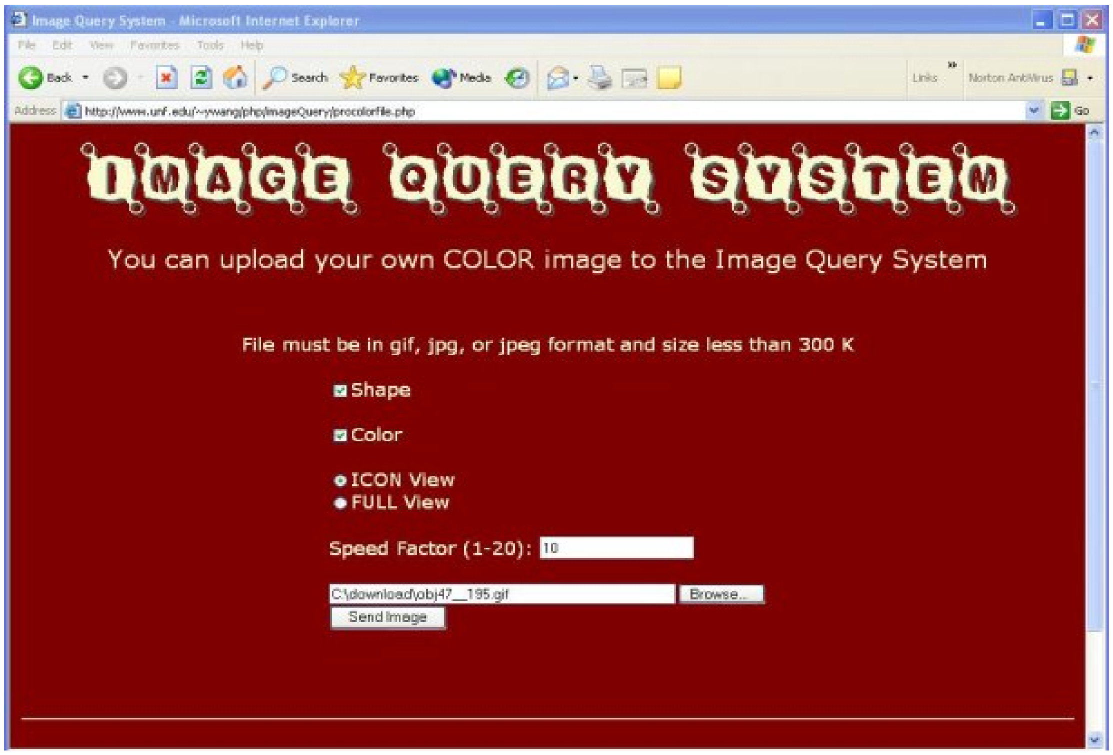


Image Query System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.unf.edu/~ywang/php/imageQuery/procolorfile.php
IMAGE QUERY SYSTEM

You can upload your own COLOR image to the Image Query System

File must be in gif, jpg, or jpeg format and size less than 300 K

✓ Shape

✓ Color

○ ICON View
● FULL View

Speed Factor (1-20): 10

C:\download\obj47__195.gif Browse...

Send Image

Figure 4.4 File uploading interface

The two scripts provide users interfaces to upload their images. Each interface displays the image system identification, Color or B&W Texture, and specifies what kind of image type the system will accept and the size limitation (Figure 4.4).

User can browse their file system to choose the image they want to submit (Figure 4.5). If the user clicks the “send image” button without supplying any file name, an error message will be displayed (Figure 4.6).

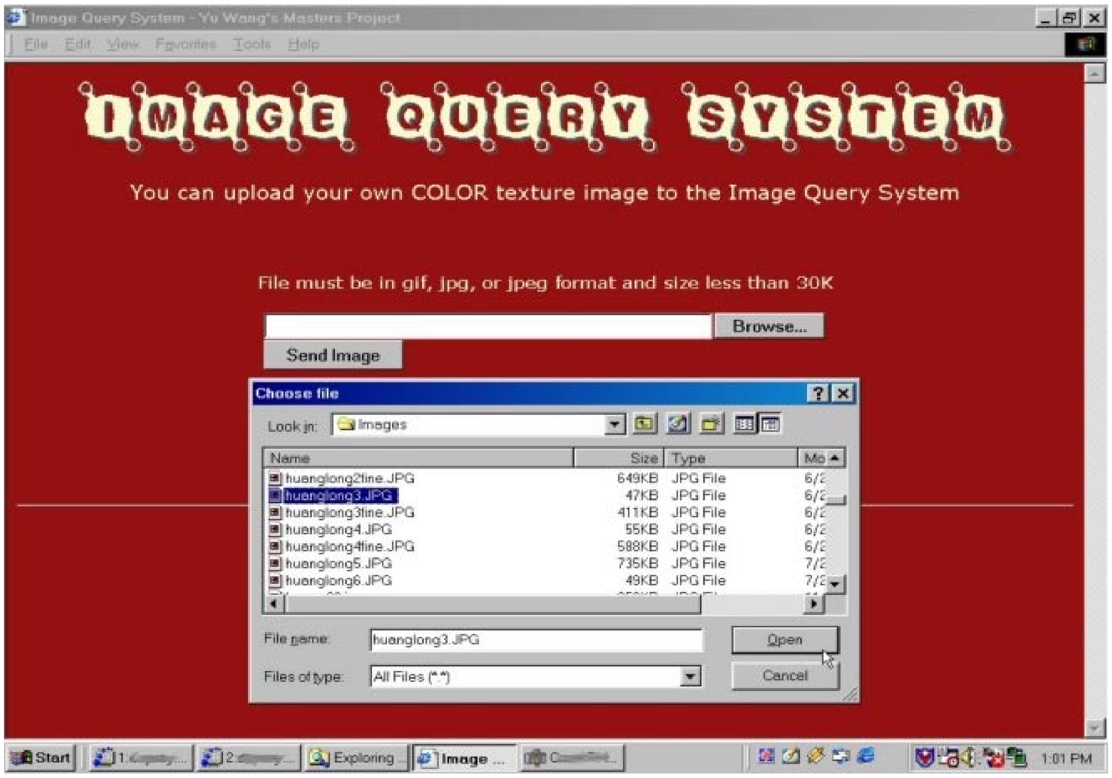


Image Query System - Yu Wong's Masters Project

You can upload your own COLOR texture image to the Image Query System

File must be in gif, jpg, or jpeg format and size less than 30K

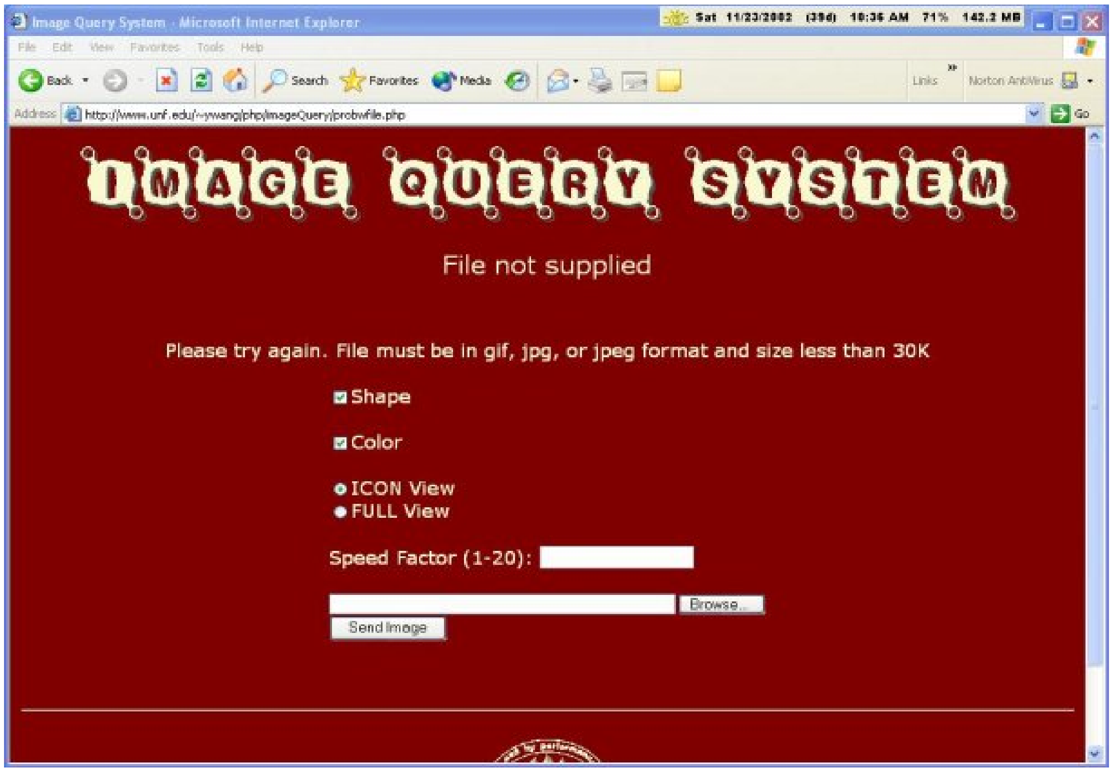
Look in: Images

Name	Size	Type	Mo
huanglong2fine.JPG	649KB	JPG File	6/2
huanglong3.JPG	47KB	JPG File	6/2
huanglong3fine.JPG	411KB	JPG File	6/2
huanglong4.JPG	55KB	JPG File	6/2
huanglong4fine.JPG	588KB	JPG File	6/2
huanglong5.JPG	735KB	JPG File	7/2
huanglong6.JPG	49KB	JPG File	7/2

File name: huanglong3.JPG

Files of type: All Files (*.*)

Figure 4.5 File browsing window



File not supplied

Please try again. File must be in gif, jpg, or jpeg format and size less than 30K

☑ Shape

☑ Color

○ ICON View
● FULL View

Speed Factor (1-20):

Image Query System - Microsoft Internet Explorer
http://www.unf.edu/~ywang/php/imageQuery/probwfile.php
IMAGE QUERY SYSTEM
Browse...
Send Image

Figure 4.6 File not supplied error

For security reason, the image size checking is performed to prevent malicious users from uploading big files to our file system (Figure 4.7). The file format is then checked. Only gif and jpeg files are accepted. If a user tries to upload different format files, an error message will be displayed. Due to the PHP's built in function we are able to distinguish the file format accurately even a user changes an .exe file name to .gif or .jpg. The secret is that the function reads the file's signature only and does not care about the file name and its extension (Figure 4.8)

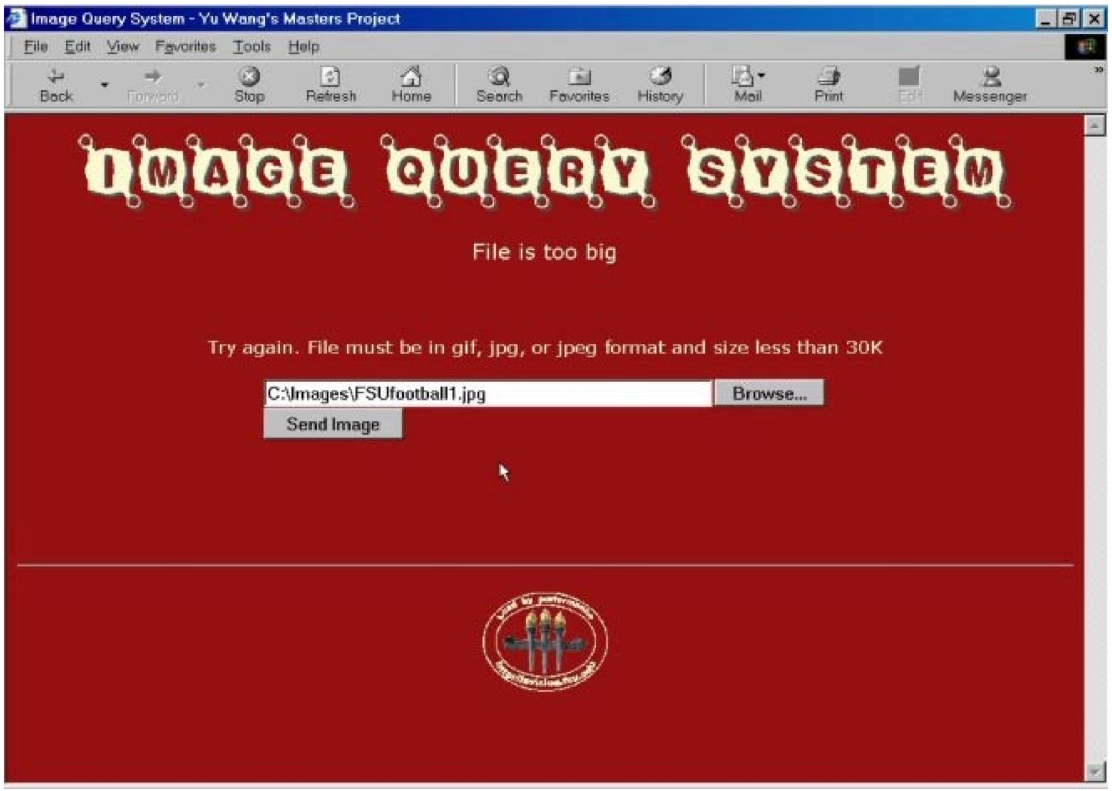


Image Query System - Yu Wang's Masters Project
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Messenger
IMAGE QUERY SYSTEM

File is too big

Try again. File must be in gif, jpg, or jpeg format and size less than 30K

C:\Images\FSUfootball1.jpg

Figure 4.7 File too big error

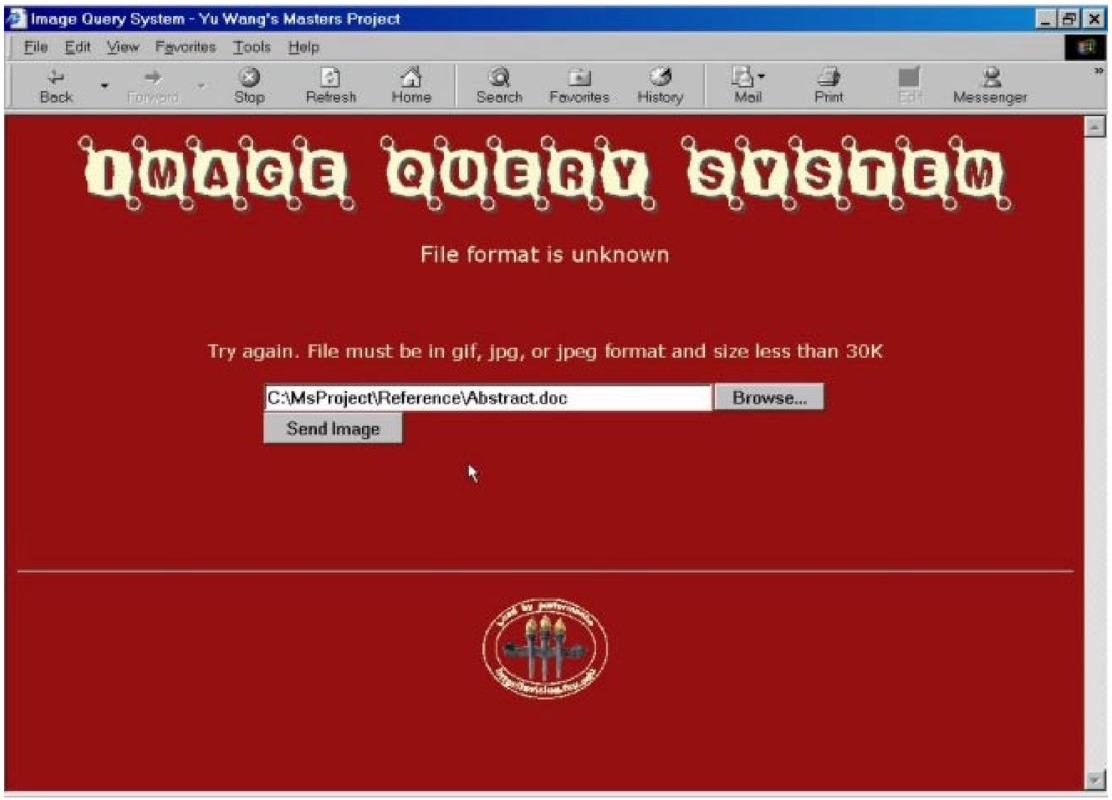


Image Query System - Yu Wang's Masters Project
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Messenger
IMAGE QUERY SYSTEM

File format is unknown

Try again. File must be in gif, jpg, or jpeg format and size less than 30K

C:\MsProject\Reference\Abstract.doc

Figure 4.8 File format unknown error

If a user submits an image file with a format other than gif or jpg, for example, bmp format, the system also displays error message and asks the user to upload a correct image file (Figure 4.9).

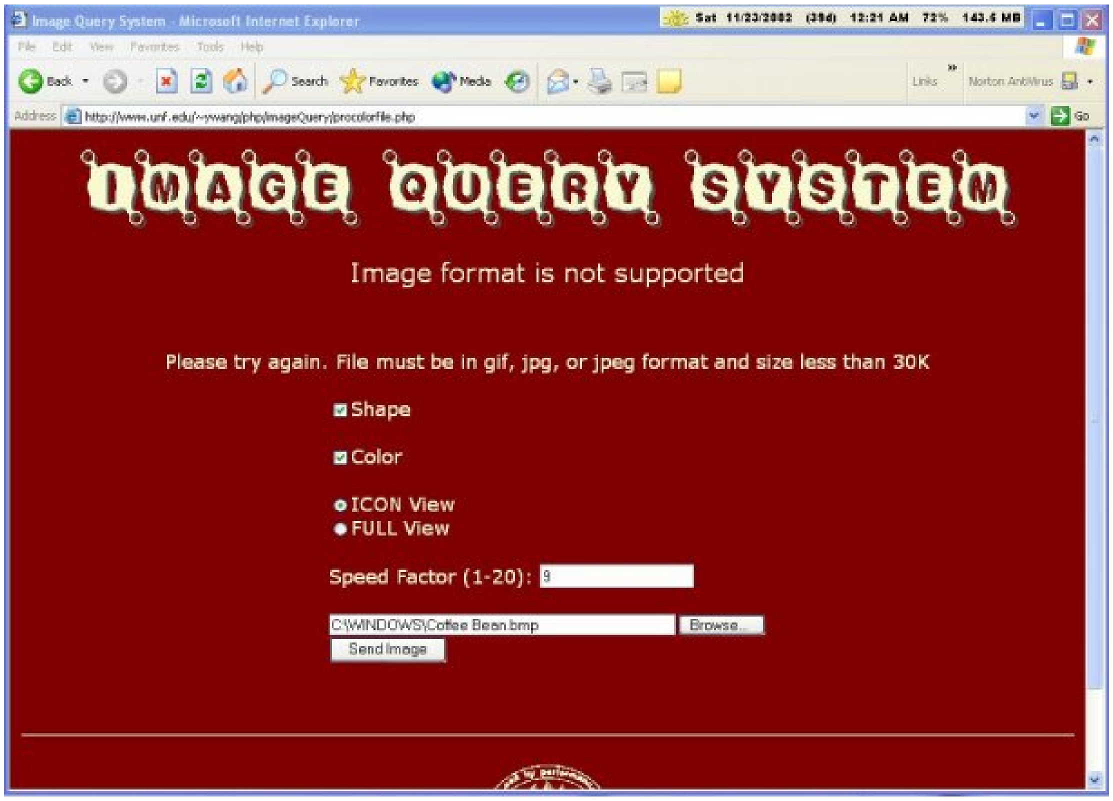


Image Query System - Microsoft Internet Explorer
http://www.unf.edu/~ywang/php/imageQuery/procolorfile.php
IMAGE QUERY SYSTEM
Image format is not supported

Please try again. File must be in gif, jpg, or jpeg format and size less than 30K

☑ Shape

☑ Color

○ ICON View
● FULL View

Speed Factor (1-20): 9

C:\WINDOWS\Coffee Bean.bmp

Figure 4.9 File format not supported error

While users must submit valid image format with a correct image size, users need to decide to select Shape and Color factors, which tells the image query program to do shape matching or color matching or both.

A user also needs to choose either icon view or full view. Icon view displays the results in icon image first and then the user can click each icon image to view a full sized image. This will yield display results fast. Full view, otherwise, displays the full sized image directly but the user will see fewer number of images per page and loading may take a little longer depends on the user's network connection speed.

A user then needs to choose a speed factor. The speed factor is used to tell the query system to run slower (better quality) or faster (worse quality) and provides a compromise between speed and quality. The allowed range is 1 – 20 with the 1 indicating the slowest speed and 20 the fastest speed. If a user's input value is out of the range, then an error message will be displayed (Figure 4.10).

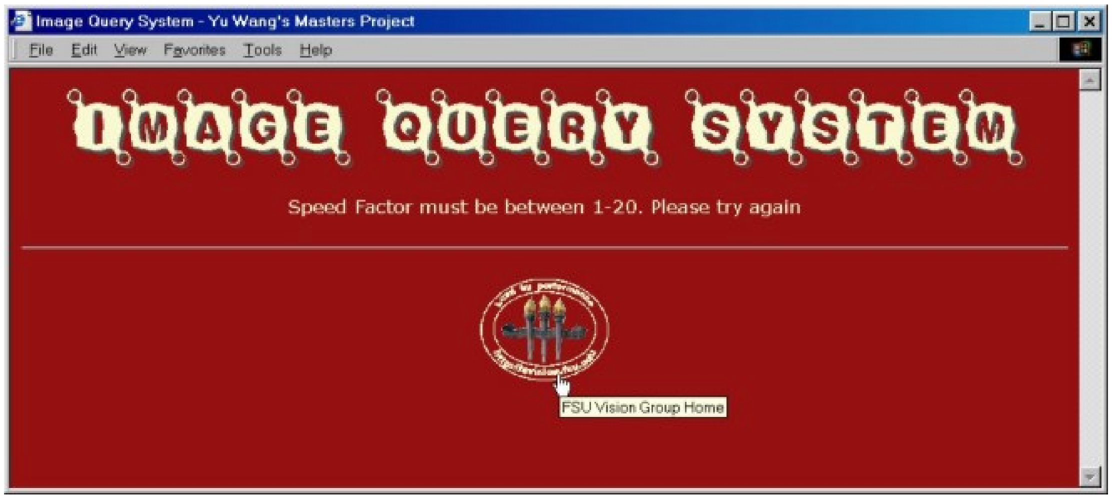


Image Query System - Yu Wang's Masters Project
File Edit View Favorites Tools Help
IMAGE QUERY SYSTEM
FSU Vision Group Home

Speed Factor must be between 1-20. Please try again

Figure 4.10 Speed factor out of range error

After the user submits the correct format and sized image, the system does a self-processing call to upload the image to a designated directory. For security reason, the image is renamed. The system then converts the image from either gif or jpeg into ppm (color) or pgm (B&W) format and passes it to the image query program. The image query program computes the image's spectral histograms, retrieves images from database, and displays the results (Figure 4.11 – 4.12) through the script procolorfile.php or probwfile.php. If users select icon view, they can click any icon image to view its full sized image displayed by upcolordisp.php or upbwdisp.php.

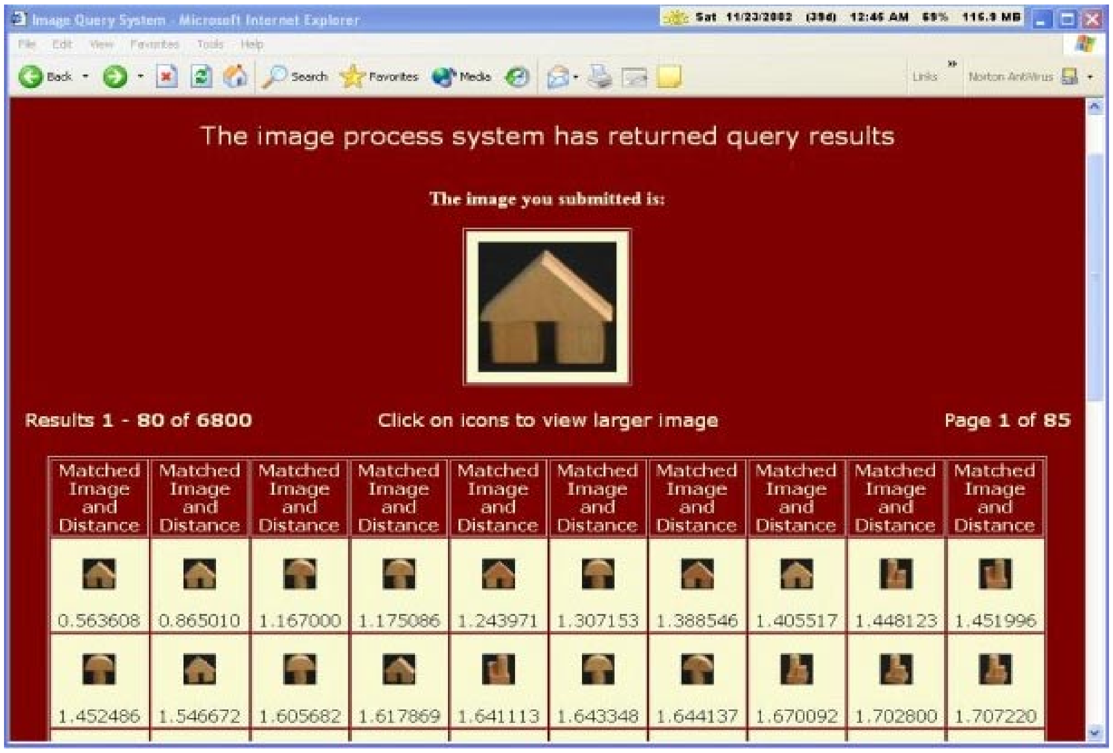


Image Query System - Microsoft Internet Explorer
Sat 11/23/2002 (39d) 12:45 AM 69% 116.9 MB
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Norton AntiVirus

The image process system has returned query results

The image you submitted is:

Results 1 - 80 of 6800 Click on icons to view larger image Page 1 of 85

Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance
0.563608	0.865010	1.167000	1.175086	1.243971	1.307153	1.388546	1.405517	1.448123	1.451996
1.452486	1.546672	1.605682	1.617869	1.641113	1.643348	1.644137	1.670092	1.702800	1.707220

Figure 4.11 Query results in icon view

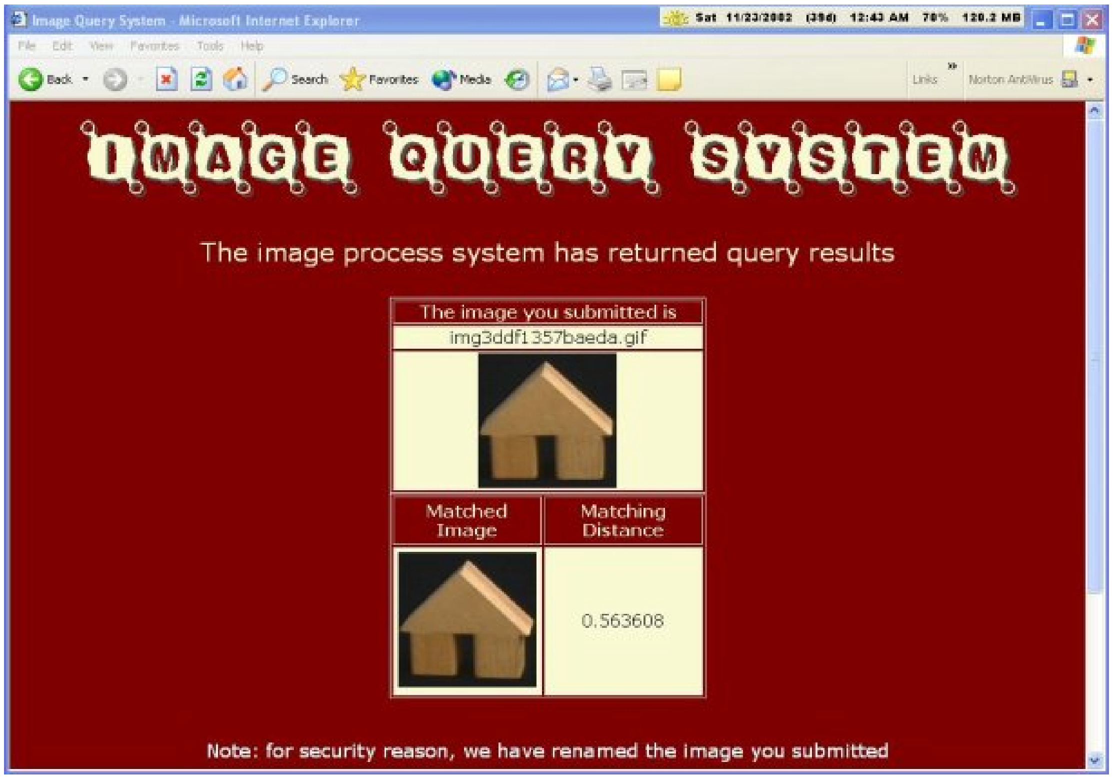


Image Query System - Microsoft Internet Explorer
Sat 11/23/2002 (39d) 12:43 AM 78% 120.2 MB
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Norton AntiVirus
IMAGE QUERY SYSTEM

The image process system has returned query results

The image you submitted is
img3ddf1357baeda.gif

Matched Image	Matching Distance
	0.563608

Note: for security reason, we have renamed the image you submitted

Figure 4.12 Full size image display from icon view with uploaded image renamed

4.1.4 Image submission from collections

A user can select images from displayed collections by calling imgformcolor.php or imgformbw.php. Since the system has many collections, icon images and multiple page displays have been designed for better performance and for viewing convenience. The user clicks one of the icons shown in Figure 4.3. A full size image will be shown (displayed by imgcoloraction.php or imgbwaction.php). If users decide to submit their own image, as before they need to choose if they want to view result in icon view or full view, check or uncheck the Shape flag and Color flag. They also need to supply a speed factor ranging from 1 to 20 (Figure 4.13). A range check is performed here too as shown in Figure 4.9.

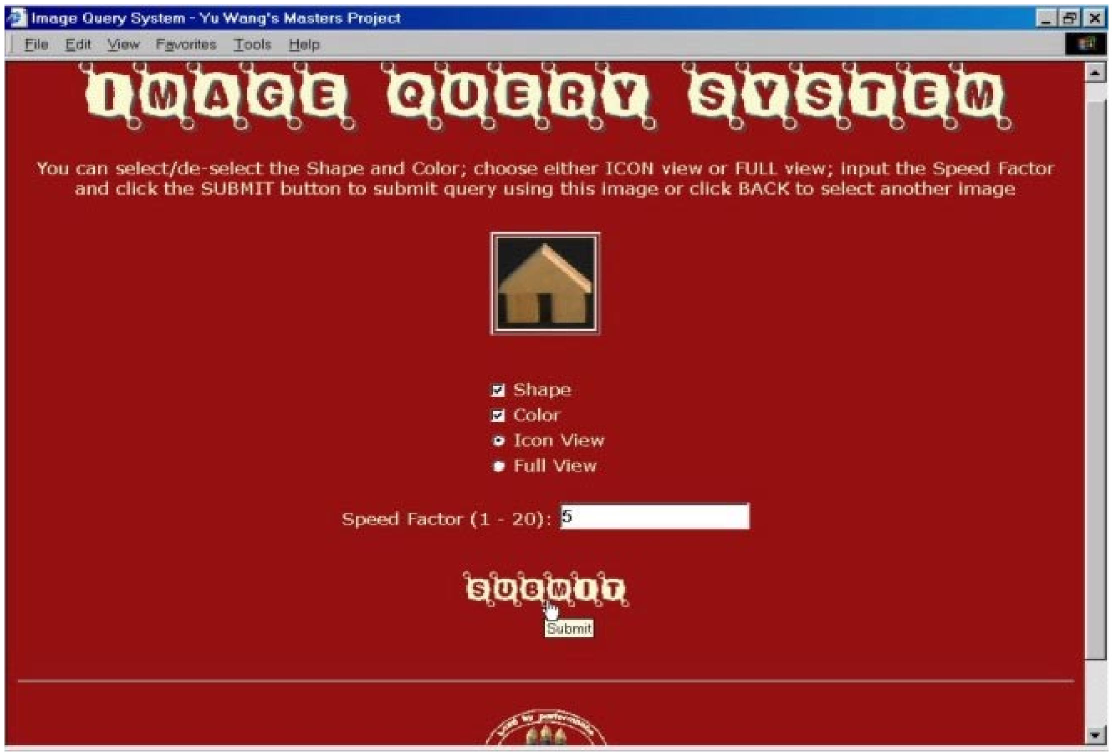


Image Query System - Yu Wang's Masters Project
File Edit View Favorites Tools Help
IMAGE QUERY SYSTEM
You can select/de-select the Shape and Color; choose either ICON view or FULL view; input the Speed Factor and click the SUBMIT button to submit query using this image or click BACK to select another image

☑ Shape
☑ Color
○ Icon View
● Full View

Speed Factor (1 - 20): 5

Figure 4.13 Submit image from collections

4.1.5 Results display

After the image is submitted to the query image program, the imgcolormultact.php or imgbwmultact.php accepts the inputs and calls the convert program to convert the gif or jpeg image into ppm or pgm format for Color image system or B&W Texture image system, respectively. The ppm or pgm image is then passed to the image query program, query_image. The query_image program computes the supplied image, compares against its database, and displays the matched images and matching distances (Figure 4.14).

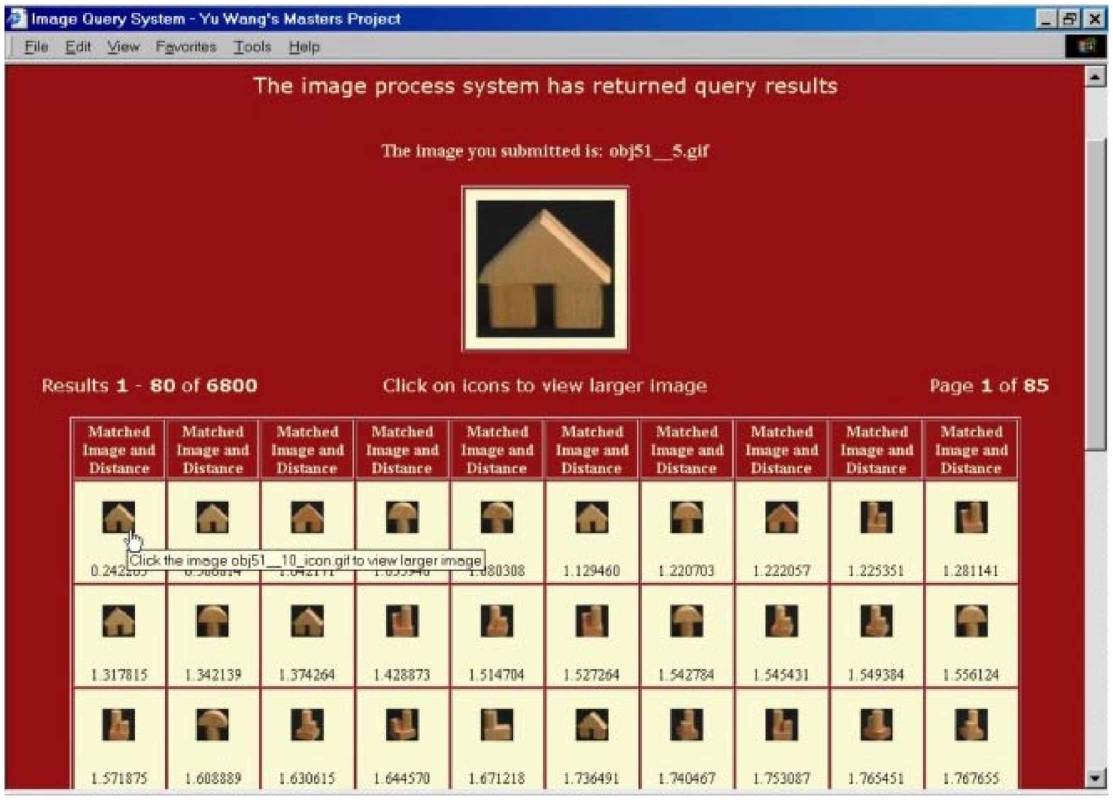


Image Query System - Yu Wang's Masters Project
File Edit View Favorites Tools Help

The image process system has returned query results

The image you submitted is: obj51__5.gif

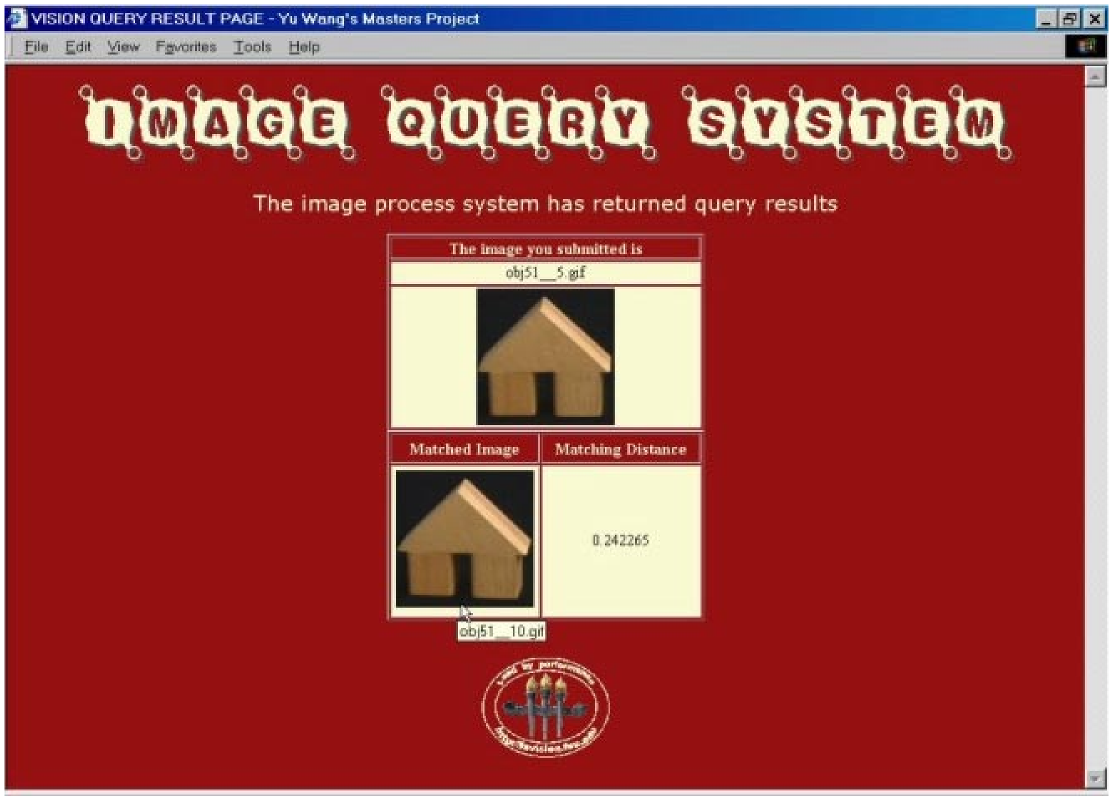
Results 1 - 80 of 6800 Click on icons to view larger image Page 1 of 85

Click the image obj51__10_icon.gif to view larger image

Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance
0.242[illegible]	[illegible]	[illegible]	[illegible]	1.080308	1.129460	1.220703	1.222057	1.225351	1.281141
1.317815	1.342139	1.374264	1.428873	1.514704	1.527264	1.542784	1.545431	1.549384	1.556124
1.571875	1.608889	1.630615	1.644570	1.671218	1.736491	1.740467	1.753087	1.765451	1.767655

Figure 4.14 Results of submission from collection

If users choose icon view, all results are displayed in icon images first by imgcolormultact.php or imgbwmultact.php. The user can click any of the icons to view a full size image view (figure 4.15) displayed by the imgcolordisp.php or imgbwdisp.php script.


VISION QUERY RESULT PAGE - Yu Wang's Masters Project
File Edit View Favorites Tools Help
IMAGE QUERY SYSTEM

The image process system has returned query results

The image you submitted is	
obj51__5.gif	
Matched Image	Matching Distance
obj51__10.gif	0.242265

Figure 4.15 Result display from icon view and from collection submission

If users choose full view, all full size images will be displayed in the result page (Figure 4.16).

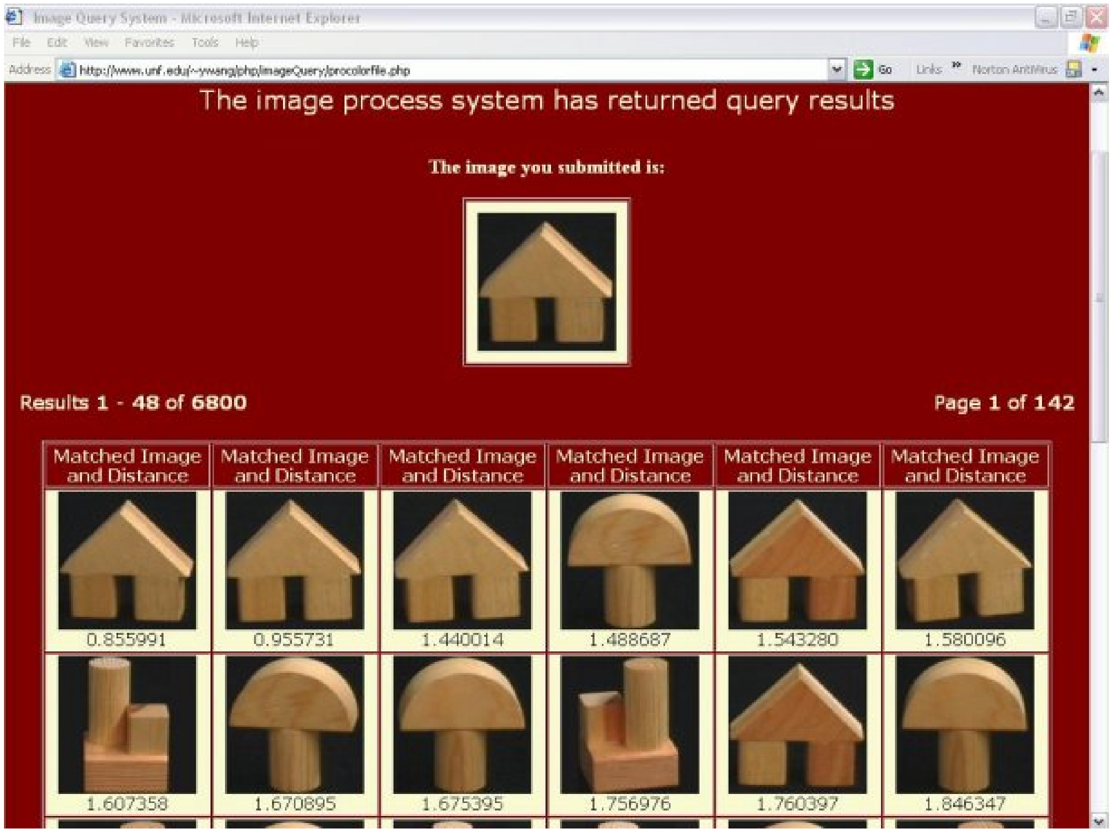


Image Query System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
Links
Norton AntiVirus

http://www.unf.edu/~ywang/php/imageQuery/procolorfile.php

The image process system has returned query results

The image you submitted is:

Results 1 - 48 of 6800

Page 1 of 142

Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance	Matched Image and Distance
0.855991	0.955731	1.440014	1.488687	1.543280	1.580096
1.607358	1.670895	1.675395	1.756976	1.760397	1.846347

Figure 4.16 Result in full view of submission from collections

The results may contain too many images to display in a single page. Thus, multiple pages are needed to display the results. The system is designed to be able to handle this kind of situation. As shown in Figure 4.16, the total results are 6800. We can display 48 full size images per page and we therefore have 142 pages. The system displays page links at the bottom of the page to allow user to navigate the results (see Figure 4.17).

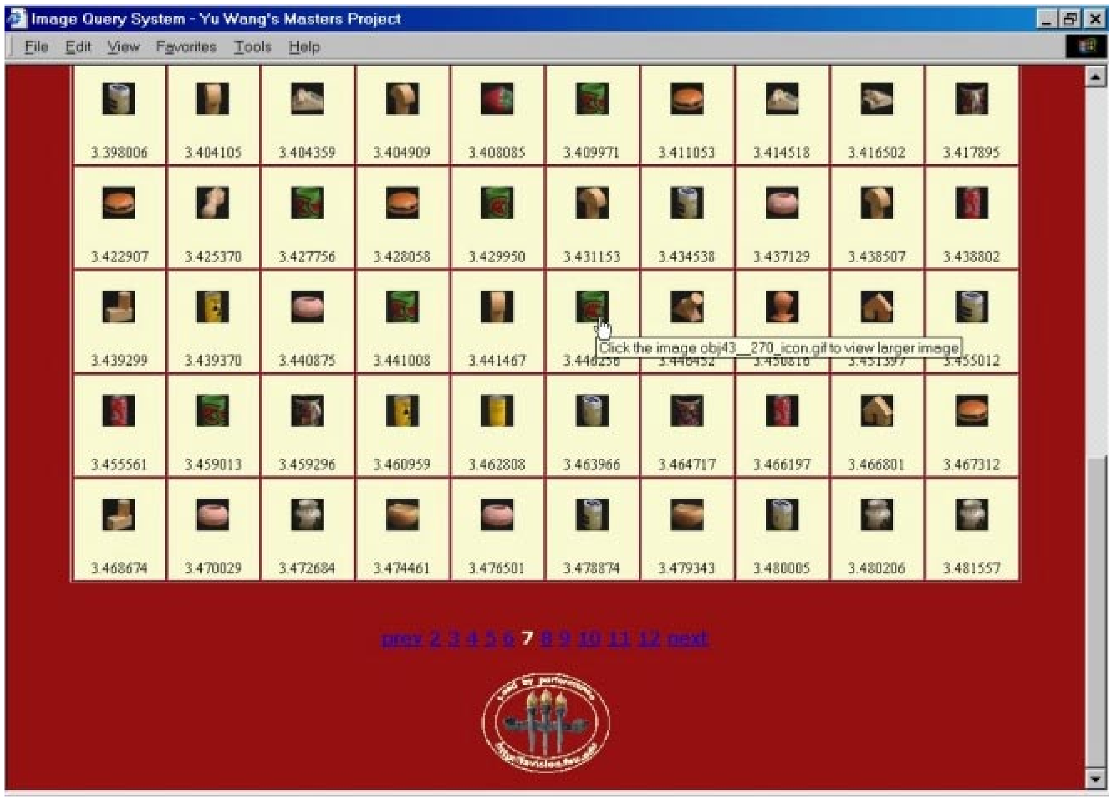


Figure 4.17 Page navigation links

4.2 System security

As the amount of hacking and denial of service (DOS) has been increased dramatically in recent years, system security becomes an important factor that must be considered into the system design and implementation. We have built several security mechanisms in the query system.

4.2.1 Email notification

When a user uploads an image file to the system, the system sends an email message which contains the user's basic information such as IP and hostname to the image system's administrator. The administrator can then check the uploaded file.

4.2.2 Access control

If we believe a certain user or host is not friendly, we can put their IP into the blacklist file. Each time when a visitor tries to access the image uploading page, the blacklist is checked. If the visitor's IP is found in the blacklist, the visitor will not be able to access the file uploading system.

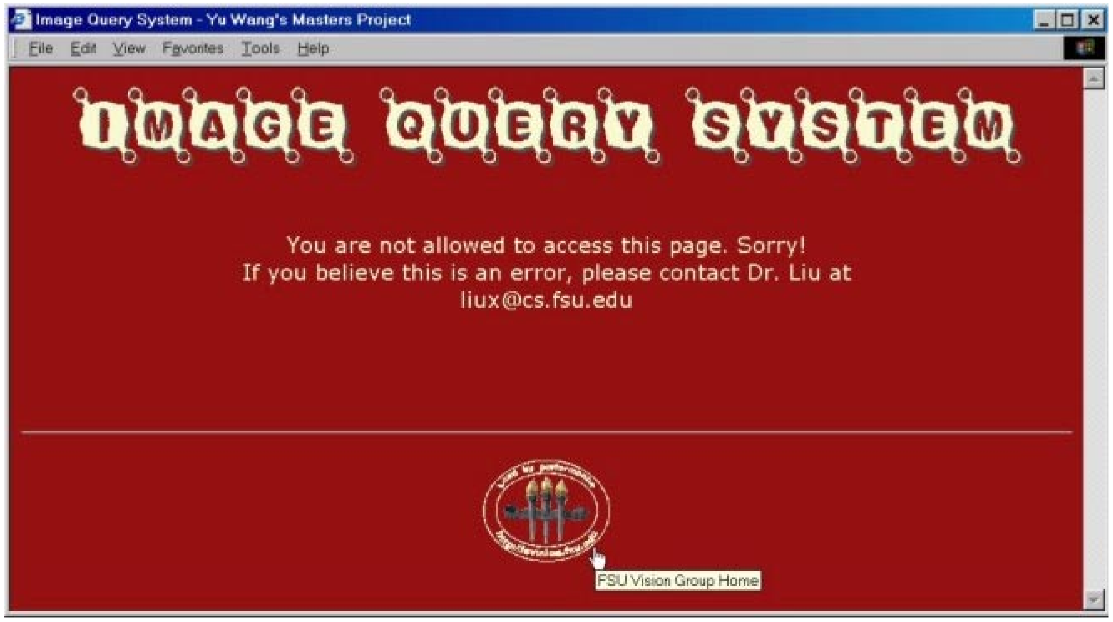


Image Query System - Yu Wang's Masters Project
File Edit View Favorites Tools Help
IMAGE QUERY SYSTEM
FSU Vision Group Home

You are not allowed to access this page. Sorry!
If you believe this is an error, please contact Dr. Liu at
liux@cs.fsu.edu

Figure 4.18 Access denied from checking the blacklist

4.2.3 File system automatic clean

Periodically, a Perl script, cleanfile.pl, runs to check the file uploading directory. If the number of files or the total size in the directory exceeds a predefined limit, the script can clean the directory. We put this clean program in a scheduled job (cron job) which runs every night. This will keep the file system clean without adding work load to the image query system's administrator.

4.2.4 File uploading format and size checking

The file uploading system has another security feature: file format and size checking. If a user attempts to upload a non-gif or a non-jpeg file or an over sized file, the system will give an error message as shown in Figure 4.6 – 4.10 to ask the user to submit again.

The built-in PHP function does this checking automatically and all attempts to fool the program are supposed to fail.

4.2.5 Display page range checking

The query results are displayed in a multiple page format. The user can navigate the result pages by clicking the page numbers listed at the bottom of each result page. If a user tries to go beyond the page number by manually changing the page number parameter submitted to the system, a page out of range error will be displayed (see Figure 4.19). This prevents malicious visitors from hijacking the system.

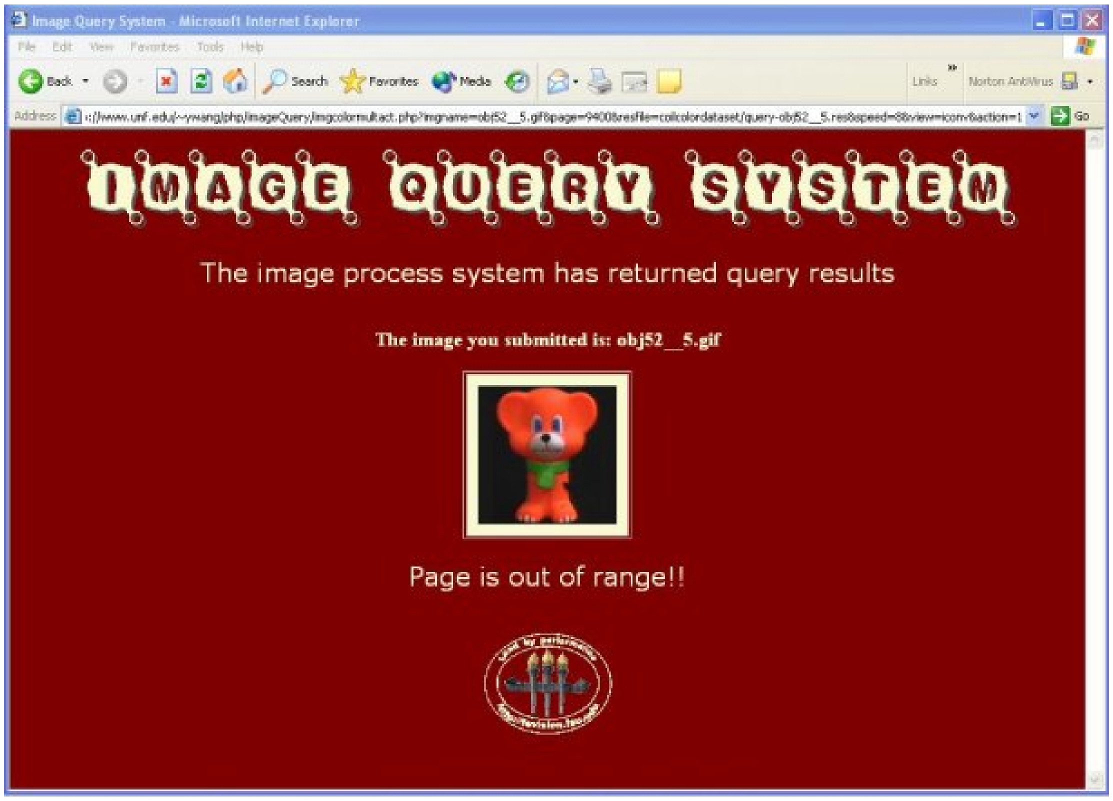


Image Query System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Norton AntiVirus
Address
Go
IMAGE QUERY SYSTEM

::/www.unf.edu/~ywang/php/imageQuery/imgcolormultact.php?imgname=obj52__5.gif&page=9400&resfile=collcolordataset/query-obj52__5.res&speed=8&view=icon&action=1

The image process system has returned query results

The image you submitted is: obj52__5.gif

Page is out of range!!

Figure 4.19 Page out of range error

4.3 System scalability

In our design, we also considered the possibilities of the future system growth. The system is designed to become a module to accommodate different image systems.

Currently, the environment has two distinguished systems: Color and B&W texture. Later if we need to add more image systems, such as a face system and a medical records system, we can simply copy the module and modify some configuration parameters and put a link in the main page. The system will then be up and running.

The system uses a self designed multi-page display system. It can handle display images numbers from small to very big without affecting performance.

All the scripts share common header files and footer files. This makes changing style and header messages much easier.

All configuration information and settings are in the conf directory. The path configuration file defines file path, image path, and program path. The size configuration file defines the file upload size limit and image numbers to display per page. Script reads those configuration files and dynamically assigns them to the variables in the script. Thus the entire system has unified configuration information, which minimizes the possibility of using inconsistent configuration information among scripts.

4.4 Design challenges

Three major challenges I have faced throughout the building of the project are the following:

1. How to seamlessly call the query_image program (C program) from PHP, pass inputs to the query program, and get results from it.

   The PHP has a safe mode for handling security related issues. If a script needs to do system calls, like system() or exec(), the safe mode only allows script to call programs allocated in a pre-specified directory. So in order to run the system, we need to get permission from the system administrator of the Department of Computer Science and ask them to put the convert and query_image into a designated directory that is specified in the php.ini file.
   Passing inputs to the query_image is as easy as passing arguments to a program in the command line. However, to get results from the program

took me a while and much effort to realize. First, I attempted to have the value by catching the query_image program's return value. But that failed. Next, I tried to let the query_image write results into a file and the dynamic image query system read the file and display the results. When all the systems were finished and loaded into web server for testing, no results were displayed. By doing extensive study, I found that for security reason a called program cannot write any files to the system even though the program is set to world executable and the program and directory is owned by me. Our PHP programs call convert and query_image and both of the two C programs attempts write files to the system.

I did many experiments and found the script can catch the stdout output of the called C program. Dr. Liu modified the query_image program to output results via stdout. This critical issue finally got resolved.

2. How to combine displaying file uploading form, checking error condition, processing file uploading, calling image query program, and displaying multiple results into one PHP processing program.

   In the file uploading system, the image query system needs to display a file uploading form to the user to allow them to browse their file system. After the user clicks the "send image" button, the image query system needs to check if the user uploaded a file or not at all, if the file size is within the limit, and if the file is a gif or jpeg image. After a series of checks, the uploaded image needs to be saved into a designated directory. The query_image program is then called to process the uploaded image. Results are displayed either in icon view choice or in full view choice. When the user navigates the results, the system retrieves results within the page range and displays them.

   Initially I tried to use multiple scripts to accomplish this goal. However, there are technical difficulties that made this approach almost impossible.

For example, if the “send image” button calls a different PHP script, then how can we display an error message and the re-upload form in a single page? What script should the re-upload “send image” button call? What if the second time the uploaded file is still in error state? Do we call a third script to handle this? If so, we will end up with a chain of scripts.

After studying the problem, I used a technology called self-processing to handle this complicated situation. Self-processing means we use one script to handle all of the tasks. This, however, brings another technical challenge that we need to resolve. Since all actions are processed by one script, we need to make sure those actions do not mix up and confuse the script. We defined three actions: action 0, action 1, and action 2. Upon calling the file uploading script first time, the action is set to 0 so the script displays the uploading form. After the user clicks the “send image” button, the action becomes 1. The script does error checking and displays the uploading form if necessary or calls the query_image and passes the image to it and displays the results in multi-page format. When user navigates the results by clicking the page numbers in the result page, action 2 is assigned and the script displays the corresponding results and does not call the query_image again to get the results.

3. Form submission variations.

This dynamic image query system handles and displays lots of images and forms. Besides using traditional submit button, we used three other formed “buttons”. When the user chooses icon view, we want to let them click the icon to show the corresponding full sized image. If we put a submit button next to each icon, it would ruin the page layout and takes up additional display spaces. We used the icon image itself as the submit button to give us neat looking pages and display more images. We also designed Flash buttons to enrich our webpage’s dynamic effects. The last form button we used is the text “button”. The text page number listed in the results display page invokes a form action and passes parameters to the called scripts.

4.5 System Installation

Since the PHP is a scripting language, no compilation is needed. One simply copies those scripts to a designated directory. There is an INSTALL file that tells one how to modify configurations and settings.

We need to ask system to copy the query_image and convert program into a system defined directory and set them world executable.

4.6 Project work loads

Thirteen PHP scripts, five configuration files, two common files, one admin file, 17 pictures and flash buttons have been written and created for this project. Total PHP codes are over 3200 lines.

5. CONCLUSION

The dynamic online image query system provides a convenient way to interact with the image query system in real time. Visitors can visit the system at any time without pre-notification to the query system researchers. The whole system is designed to show the high performance of the image query system. The system has reached the expectations of the initial designs in both the presentation and performance. There are also areas for which we can improve the system:

- Use a database instead of files to take advantages of the PHP’s database manipulation abilities.
- Provide an access controlled GUI interface for the administrator to set configurations, such as file path and image size.
- Design non-flash alternative interfaces for users who do not have a flash player installed.

REFERENCES

[1] Xiuwen Liu and DeLiang Wang (2002) A spectral histogram model for texton modeling and texture discrimination. *Vision Research,* vol. 42, pp. 2617-263.

[2] Xiuwen Liu and DeLiang Wang (2001) Appearance-Based Recognition Using Perceptual Components. *International Joint Conference on Neural Networks,* vol. 3, pp. 1943-1948.

[3] Rasmus Lerdorf and Kevin Tatroe (2002) Programming PHP. Creating dynamic web pages. *O'Reilly & Associates, Inc.*

[4] PHP manuals online http://www.zend.net/.

[5] Hugh Williams and David Lane (2002) Web Database Applications with PHP & MySQL. *O'Reilly & Associates, Inc.*