

Implementation of Web-Based Online Image Processing Software

Yet-Chung Chang
Department of Electrical Engineering
Diwan College of Management
Tainan, Taiwan, R.O.C.
Email: ycc@dwu.edu.tw

ABSTRACT

For this research, a piece of web-based online image processing software was designed and implemented. The software allows images to be uploaded to the server and processed following interactive commands from the client which are made through the interface on the web page. The image processing software is designed on the server side as part of the code-behind ASP.NET program in Visual Basic using the .NET framework library. One of the goal in this research was to produce a "Rich Internet Application" (RIA) by using only default programming techniques stipulated by the W3C organization. No attachment needs to be downloaded for the use of this software in client side.

Keywords: online, image processing, web-based, RIA

I. INTRODUCTION

Images and digital photographs are now very popular and are easily obtained through the internet and digital equipment. However, most commercial image processing software is still too expensive for most users and highly resource-consuming. Therefore, an online image processing service seems to be an adequate solution. Unfortunately, the current web-based programming environment is still not suited to image processing for the following reasons:

Firstly, the image files are generally large in size so transporting them frequently between the server and client can be time consuming if the internet bandwidth is limited. Fortunately, transfer speeds are increasing. If the client's internet connection allows the simultaneous opening of several image-rich pages, there should be no long delay when transferring images to online software to be processed. And we can expect that this will continue to improve.

Secondly, most server side technologies developed over the last decade focus more on database applications [1] and less on computational oriented applications such as image processing software. This means that experts developing software on the server side will encounter more difficulties than those working on databases. In this research, the ASP.NET is employed as the server side technology. It is established on the same .NET framework library as other Microsoft general purpose programming languages such as C# and VB. This means that we can design web-based program with any purpose what desktop programs can do, including image processing, in the same integrated design environment. It also means that we need not arrange a CGI mechanism [2] to link the web pages

with an independent image processing program because such a program is actually built in to the pages.

It is also possible to accomplish the major image processing tasks on the client's side. However, web-based programs generally do not have high enough authority to use the resources of the client's computer and a normal web browser is not efficient enough to run a large and somewhat tedious piece of online software. If one insists in promoting the use of such a web based application, an attachment for the user's browser must be designed. This would require that the user accept it and wait for it to download before being able to use the online software. In my opinion, this would be more like installing a desktop program onto the client's computer rather than providing a web-based application.

Another problem is that most image processing software provides user friendly mouse operated functions such as the drag-drop motion to clip a rectangle, readjust the image size or to daub the image. However, it is problematic for a web-based program to provide such user interfaces. HTML is not a standard programming language with complete mouse or key control support. In general, we need additional script languages such as JavaScript embedded in the HTML to design interactive interfaces into web pages, especially mouse and key events. However, performance of pages with functions written with such scripts is generally no match for that of desktop programs.

It is hoped that web based programs, or Rich Internet Applications (RIA) will soon be able to offer an experience as good as desktop programs. [3]. Demand for such programs is high. However, programs written purely using script language are generally not fancy or friendly enough until now. Therefore, many solutions based on attachments to web browsers, including Flash, Silverlight, ActiveX and the Java applet, have appeared. We can see that most of the current online image processing web sites require the lengthy download of such an attached program prior to use[4][5][6][7]. Sometimes, they are simply rejected by the browser for security reasons. In this research, we have purposely looked for solutions purely established on script language to help achieve the RIA goal. A good module for line and shape drawing developed by Zorn [8] was employed as a basis for our mouse operated interactive interface. We found it sufficient for our requirements.

Therefore, we implemented a piece of web-based online image processing software which is located at <http://ycc.dwu.edu.tw/cloud/>. The content and techniques used are described on the following page.

II. SYSTEM CONTENT

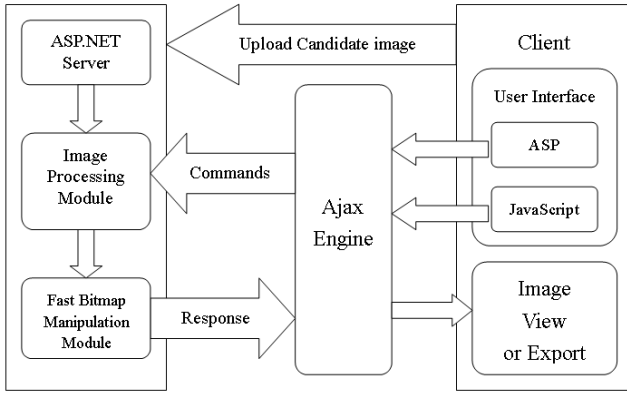


Figure 1. System architecture and data processing flowchart.

A. Basic Architecture

Figure 1 shows the system's architecture and data processing flowchart. Firstly, the candidate image is uploaded to the server and post immediately back to the client's page. Then, the user can edit the image. The commands are intercepted by an Ajax engine, rather than being sent directly back to the server. This blocks out most unnecessary page post back actions. It both saves internet bandwidth and prevents the page from flickering. The full definition, benefits and basic mechanism of Ajax can be seen in reference [9]. In this research, the Ajax engine was mainly provided by the ASP.NET 3.5 library, with only some additional client side JavaScript functions written by the author of this paper.

The commands sent back to the server are handled by the image processing module which performs any necessary mathematical computations. At the same time, a module called "Fast Bitmap Manipulation" helps to extract information from the image and create a new image as the output. Whenever a step of image processing is completed, the result is returned to the client immediately via the Ajax passage and the system awaits the next command. The interactive command and processing cycles via the Ajax engine can be repeated as many times as the user requires. The experience for the user is very similar to using a desktop program.

B. About the ASP.NET user interface

The purpose of the user interface in this project is to form a command for each image processing attempt. In most cases, it requires only a command with few parameters. For example, when we want to make the image brighter, a text command like "bright" and a parameter to denote "how much brighter?" is necessary. For these requests we simply employ the tool box objects provided with the ASP.NET interface such as the command button and some selection controls to make our interface function. To save the limited space on the page, the interface is displayed using a cascading multi-view architecture and most buttons are small icons with a tooltip. An example of rotating image 10 degrees clockwise is shown in Figure 2.

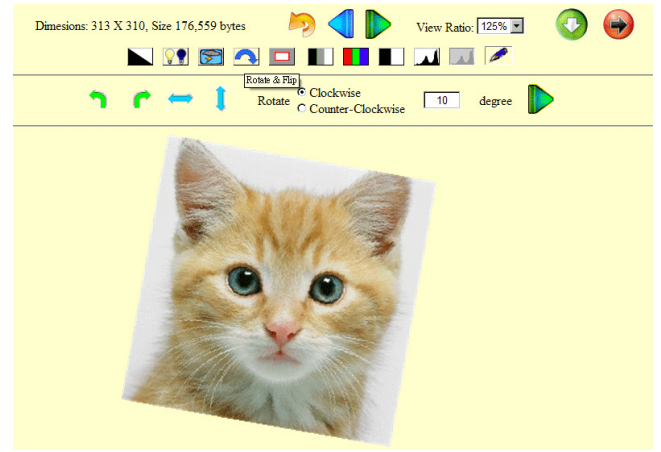


Figure 2. Main processing page, example of rotating an image 10 degrees.

The first row of the tool strip provides basic image information, roll back and forth buttons, view scale, image download and exit buttons. The second row provides the main categories of image processing tools including rotate and flip, clipping, contrast and others. When the user clicks buttons on the second row, the third row of different buttons corresponding to the user's selected button appears. More detailed parameters and settings can be defined in the third row which also contains OK button which confirms the actions.

C. About the JavaScript user interface

Although all the parameters required for image processing can be input via text box or selection controls, allowing the user to visually define the changes by mouse movement directly on the target image, such as clipping or brush painting, is much more user friendly. However, mouse movement detection and response control are not well-supported by normal HTML pages. Even though script languages such as JavaScript can handle mouse events, they still offer no support for drawing functions.

We discovered that Zorn [8] has designed a freeware JavaScript module which provides simple commands to draw lines and shapes on normal web pages. By combining the mouse event detection ability in JavaScript and this drawing module, we were able to design the mouse driven painting and selection functions. An example of some drawings and an ellipse shaped clipping is shown on Figure 3.

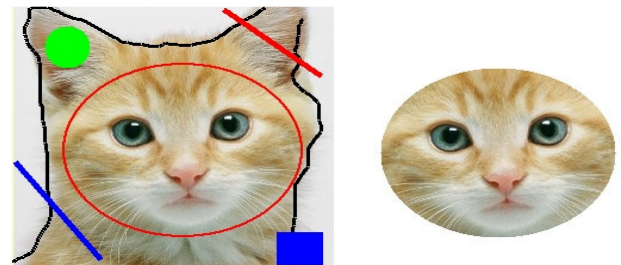


Figure 3. Example of a mouse-driven drawing and clipping interface.

It should be noted that the shapes or lines are only temporary objects on the client side page after the mouse operations. They are merely overlays on the image which do not become part of the image until the user clicks the confirm button. The background processes to complete an action are: 1) The key mouse coordinates are saved in an HTML object such as a hidden field; 2) The user clicks a button to submit the information and command to the server; 3) The server side program executes the drawing or clipping on the image and posts back the result to the client.

D. The Image Processing Module

The image processing module in this project is written in Visual Basic as part of the server side code-behind ASP.NET program. Since the ASP.NET program is founded on the .NET framework library which is fundamentally the same as the general purpose C# or Visual Basic language, most of the functions for the general purpose program could also be used in the ASP.NET code-behind program. Therefore, the sub-library of "System.Drawing" in .NET could be employed for our image manipulation. The "module" is actually not as independent as the name implies. There are, in fact, many subroutines respectively corresponding to the client side command button inside the web page.

In our design, there is no need for an overly complicated CGI mechanism to be introduced to the server to "connect" the operations between the page server and the image processing program, because they are actually in the same program file. It certainly eases the programmer's task if he or she only needs to design one program for all of the server side functions including the web page response and data processing related to the software. The simplicity of the architecture also implies higher reliability and stability of the server side system.

E. The Fast Bitmap Manipulation Module

Image processing itself is already a relatively time-consuming task and using a web-based program requires even more time due to internet data transportation. Therefore, one theme in this project is to find ways to reduce the time consumed during processes. As many programmers know, the typical methods in the .NET framework to get or set pixels from or into a bitmap are slow. On MSDN, a solution to increase the speed of processes has been suggested. This works by locking the position of the bitmap object in the memory of the server, then directly reading or saving the color information as a normal numerical array in the memory [10]. Generally, this could be up to ten times faster than the typical method. However, the suggested method may be a little tricky for most programmers, particularly the complicated computations of the memory pointer position for each pixel. We therefore summarized it as a module with several public functions [11]. Its procedures are simple as: 1) The bitmap is locked; 2) The module uses the X and Y co-ordinates of each pixel to set or get color data; 3) The Bitmap is unlocked. We used this module in our application to accelerate processing. This proved very successful.

F. Some efforts for professional usage

Most existing web sites for image processing are designed for simple tasks performed by non-advanced users. In our project, we tried to introduce more quantitative information for advanced users. For example, the histogram of brightness distribution is provided as optional information. This is important for advanced usage such as the decision of threshold to create a binary image - a critical process for most image recognition or machine vision. In our software, if the user chooses to see the histogram, every time a new image appears, the histogram and the corresponding numerical values are displayed beside the major image as shown in Figure 4.

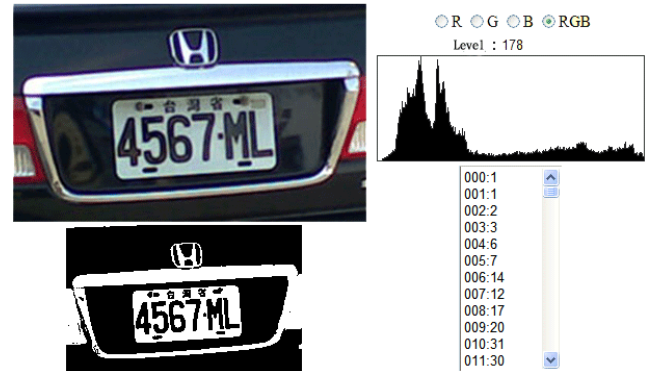


Figure 4. Example a plate image with its histogram information and a result of binary image ready for recognition.

G. The Image File Denomination System

This web-based program may be used by many different users simultaneously but each client has his own instance of the program. This means different users have independent variables and memory occupations. These do not influence each other. However, if data files are uploaded, the files coming from different users are saved into the same file system on the server's computer. If different data files which happen to have the same name are uploaded, there is a risk of file overlapping. In our case, users upload image files and these are processed to create more temporary files. It is inevitable that files coming from different users which are at different processing stages will be crowd in the server. Therefore, a reliable image file denomination system is required.

Our mechanism uses an automatically-increasing public variable belonging to the web site to denote the sequential number of each user. This number later forms the prefix of the user's filename, for example "0001xxxx.jpg". When the user conducts any image processing, a new image with a different file name is created. Each individual user's files are named according to a private sequential number. Therefore, a series of files belonging to a user with public sequential number of "0001" might be "00010001.jpg", "00010002.jpg", "00010003.jpg" and so forth. An example is shown in Figure 5.

0	0	0	1	0	0	1	1	.	Jpg
User ID				Image ID					Format

Figure 5. Example of temporary file denomination rule.

As the popularity of the website increases, the amount of files dumped on the server may become an issue. As a solution, a mechanism for quick clearness had to be designed. Unfortunately, web-based programs have no definite “client left” event to allow the program to delete the temporary files belonging to the user who has just exited the site. Therefore, like many other web-based programs, we provided an exit button for the user. When the button is clicked, all files belonging to that user are deleted immediately. This not only lowers the burden on the server but also keeps helps to ensure the client’s privacy. However, as we cannot expect all users to click this button, we designed a subroutine to check and delete images older than 2 hours. This is automatically activated each time a newcomer enters the site.

III. DISCUSSION AND CONCLUSION

As the internet evolves, not only is more and more data available online but also more and more jobs can be performed through a remote host. In the near future, it may well be that computers contain very few programs as all data processing including document editing, image processing and versatile computations will be performed on a wide variety of web sites, just as restaurants or shopping stores on the street provide many different services. Thus, a local computer will become a terminal with access to web sites providing software or services. The new term “cloud computing” [12] is being increasingly used to describe this developing phenomenon.

Undoubtedly, “cloud computing” is an excellent way to save resources. One might agree that most software in our computers is seldom or never used. If image processing can become a convenient web service, then most people need not buy such software which is cumbersome and costly. The user need only pay a small fee, if anything, for the use of software online. To accomplish this, the main technical challenge for the programmer is to design a web-based program with similar or better performance than a desktop application. The difficulties can be subdivided as follows:

1) The software must work together with the web server. Not only does the logic of the software itself need to be considered, the logic of web server operation should be taken into account. The data is processed on a different computer from that containing the interface and they are generally located at great distance from each other. The methods of communication between server and client must become an integral part of the software design.

2) The client side of the software is restricted by the capabilities of the web browser or the rules stipulated by the W3C organization for HTML, CSS, JavaScript and others. A web page is merely a document. Establishing a high efficiency

program with an elegant user interface in an HTML page is extremely challenging.

Essentially, web servers need to expand their range of services from normal pages and databases to other computational oriented programs. In this project, we employed the .NET framework based Visual Basic and ASP.NET objects as server side designing tools. Since they have same function library and developing environment, we found it simple to accomplish both the jobs for the page server and the image processing.

On the client side, RIA is being developed in two different ways. The first is that more complicated and versatile JavaScript program modules and CSS skins and themes are being created. This will lead to the improvement of user interfaces which run under the current framework of internet documents defined by W3C. The other way is that developers are experimenting with attaching new objects, programs or run-time environments to the client’s browser or computer. Such efforts include the Flash Player, Silverlight and Java Virtual Machine.

So far, we favor the former approach. The latter can certainly improve the online interfaces to the extent that desktop programs can as they are basically the same as a downloaded desktop application only they work together with the browser and not independently. This is the reason why most current online image processing web sites can provide elegant user interfaces but generally require a long download time before the work can be started. In this project, techniques which are not necessarily safe for ordinary browsers have purposely been avoided.

Our web-based image processing software is located at: <http://ycc.dwu.edu.tw/cloud/>. The page also contains a summary of the core techniques employed and concepts of the design.

REFERENCES

- [1] A. Greene, "Managing subject guides with SQL Server and ASP.Net," Library Hi Tech, vol.26, pp.213-231, 2008.
- [2] P. Edwards, "DIY Intranets with CFML/XML/HTML/CGI/ASP/JSP," VINE, vol.30, no.2, pp.53-60, 2000.
- [3] P. Cavaleri, "The use of AJAX in searching a bibliographic database: A case study of the Italian Biblioteche oggi database," Program: electronic library and information systems, vol.42, no.3, pp.275-285, 2008
- [4] LunaPic, LunaPic online photo editor, <http://www.blibs.com/editor/>
- [5] Zygomatic , Online photo editor, <http://www.freeonlinephotoeditor.com/>
- [6] Picnik, Picnik, <http://www.picnik.com/>
- [7] DrPic, Free web picture editor and image host, <http://drpic.com/>
- [8] W. Zorn, DHTML: Draw Line, Ellipse, Oval, Circle, Polyline, Polygon, Triangle with JavaScript, http://www.walterzorn.com/jsgraphics/jsgraphics_e.htm
- [9] J. J. Garrett, Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [10] MSDN, Bitmap.LockBits method, <http://msdn.microsoft.com/zh-tw/library/5ey6h79d.aspx>
- [11] Yet Chung Chang, Fast bitmap manipulation module, unpublished, <http://ycc.dwu.edu.tw/lmgProc/FastPixelModule.htm>
- [12] Wikipedia , Cloud computing - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Cloud_computing