

Research on the Fast Display of Web-based Nautical Chart System

Yuan-Qiang Zhang

Doctor student of Navigation college, Dalian Maritime
University, Dalian, China
Faculty of Maritime and Transportation, Ningbo
University, Ningbo, China
zhangyuanqiang@nbu.edu.cn

Guo-You Shi, Wei-Feng Li

Navigation college, Dalian Maritime University,
Dalian, China
E-mail: dmulwf@dlmu.edu.cn

Abstract-With ship size and capacity continuing to increase, the safety of ship navigation has become increasingly challenging. Port authorities and shipping companies urgently require quick and easy, real-time, vessel navigational monitoring. Cheap and easy-to-use web based electronic chart application services are increasingly finding favour with companies and port authorities, but many do not conform to the IHO S52 display standard. The purpose of this paper is to propose a solution to these problems . The proposed solution is based on image tile mosaic techniques. Practice has proven that the storage and display technology can work both swiftly and efficiently.

Keywords-electronic nautical chart; B/S mode; tiered storage; asynchronous transmission

I. INTRODUCTION¹

Electronic Nautical Chart Systems provide economic value and may be used in shipping operations, fisheries, accident investigation, pollution control and defense[1]. With the development of Internet technology, and the increased popularity of computers, the traditional client/server (C/S) mode electronic nautical chart system-has not kept pace with the various needs of users. Today a web browser may be used to view real-time ship movements and port dynamic information. Therefore, development of a browser/server (B/S) model of electronic nautical chart system is necessary.

An important problem to be solved in B/S mode electronic chart systems is how to display the chart. The following is a list of academic papers which were studied in this area. [2] did a research into web mapping of vector nautical charts based on HTML5. [3] adapted MapObjects to extend ArcIMS's symbols, and then distributed Electronic Navigational Chart(ENC) data by ArcIMS and ArcSDE. Kam, Seung Chul, Lee, Seong Dae, Park, Hyu Chan Etc.(2006)[4] propose a Web-based system for access to, and services of ENC, which makes use of the Geography Markup First change vector chart into raster chart Language (GML) and the Scalable Vector Graphics (SVG).

The above documents require not only the development of the server-side programs, but also browser compatibility and speed. This paper introduces a new method, This method only needs to develop the server, all of the user's

information processing is to be completed on the server side, the user simply sends a request to the server.

II. THE CHART IMAGE PROCESSING

If you want to achieve B/S mode nautical chart perfectly, you first need to solve the problems of chart cutting and tiered storage.

A. The Basic Principle of Request

Charts described here are in accordance with IHO S57[5] standard transmission and IHO S52[6] but will be displayed similar to raster chart. This scheme according to the S52 standard, provides three kinds of visual pattern: daylight, dusk, night. Also, each visual pattern can use one of three display modes ; base display, standard display, all display.

Different area vector charts are first transformed into different images according to the display mode, and are stored in different folders. Users on the client side send a request to the server according to their different needs, the server will transfer the corresponding chart to the user. This process as shown in figure 1 also shows that the end user is just a passive receiver, all operations are carried out on the server side.

These are translated into raster style charts using chart processing software to cut them into the same size image pixels, called basic chart. The original full chart has been cut into equal size sections and stored in the server in advance. When an end user first visits the site, the chart images can be downloaded to the local computer, to the cache. Users should note some advanced features (such as scale, etc.) are completed by a Javascript program in the browser, the Javascript code can be easily loaded. When the user visits the next site, or refreshes the page, there will be no need to download the map again, only some dynamic information. This will reduce the amount of data flow between server and browser and so improve the access speed. It will also reduce the server load and thus allow more users to access the server at the same time[7].

To explain further, the original download was not the complete chart, only the section required on the first visit. Initially we set the picture size of each section at $256 * 256$ pixels. If the chart is at n level, this level has $X(n) = 2n-1$ columns of chart images, corresponding to the number of rows ; $Y(n) = 2n-1$, thus each level has $N = X(n)*Y(n)$ number of chart "tiles".

¹ This paper is supported by the Fundamental Research Funds for the Central Universities with number of 3132015009

B. Cutting the Chart

You should define the scope of the chart before cutting. The origin of the coordinates is located at the center, (0, 0). As shown in figure 2 on the left, n=2, scope of pixels (-256,256) to (256,-256), consists of 4 tiles. When we zoom in to level, n=3, it will have 16 tiles as shown in figure 2 on the right. The number of images is four times that of level 2, and the scope is now (-512, 512) to (512, -512) pixels. When we cut the chart we leave no gaps between the tiles.

C. Tiered Storage of Charts: Selecting the Chart You Want.

Usually, the larger the scale, the more information that is shown. Users choose a different scale according to demand. Web-Based nautical chart systems should also provide the choice of different scale, so we need to cut different scale chart. For each region each chart level must be stored independently.

The scale of a level 1 chart is the smallest, the general layout for the area, chart size is 256 x 256 pixels, composed of one picture / tile. The second scale is composed of four images, the size of each tile is still 256 x 256 pixels, a total of (256 + 256) x (256 + 256) of pixels, the third grade scale cuts each picture into four pieces. The fourth level will consist of 64 pieces of 256 x 256 pixel tiles, the total size will now be 2048 x 2048 pixels. The higher the level, the larger the scale of a chart, each higher level having more detailed information.

When users change the scale, the browser Javascript program will automatically judge the level of the current request and send it to the server. In order to decrease the difficulty of development on the server side, we choose openlayers as the server, it will send corresponding chart according to the request [8]. On the server side, the chart of different scale are stored in different folders (as shown in figure 3). When the user selects a different scale of chart, server will call the corresponding folder chart for display.

These chart images are in the server hard disk, in the folder named numerically n-1 respectively. There're a total of five folder in figure 3, so there are five zoom levels to choose from. Each folder has rows and columns of 2n - 1 pictures / tiles respectively.

Because we have a lot of pictures in each level scale, if there is no reasonable way of storage this can bring difficulties to selecting your chart. So the idea of this article is: the chart will be stored in a separate folder for each level, and we establish subfolders (Chart Column Number) in the primary folders (Chart Level). Subfolders are named according to the column number, all images stored in these subfolders are numbered in accordance with row numbers starting from 0, as shown in figure 3. If the client needs to request the fourth grade scale column 1 row 4 picture, the following request can be sent: "http://... /3/0/ 3. PNG" link to the server.

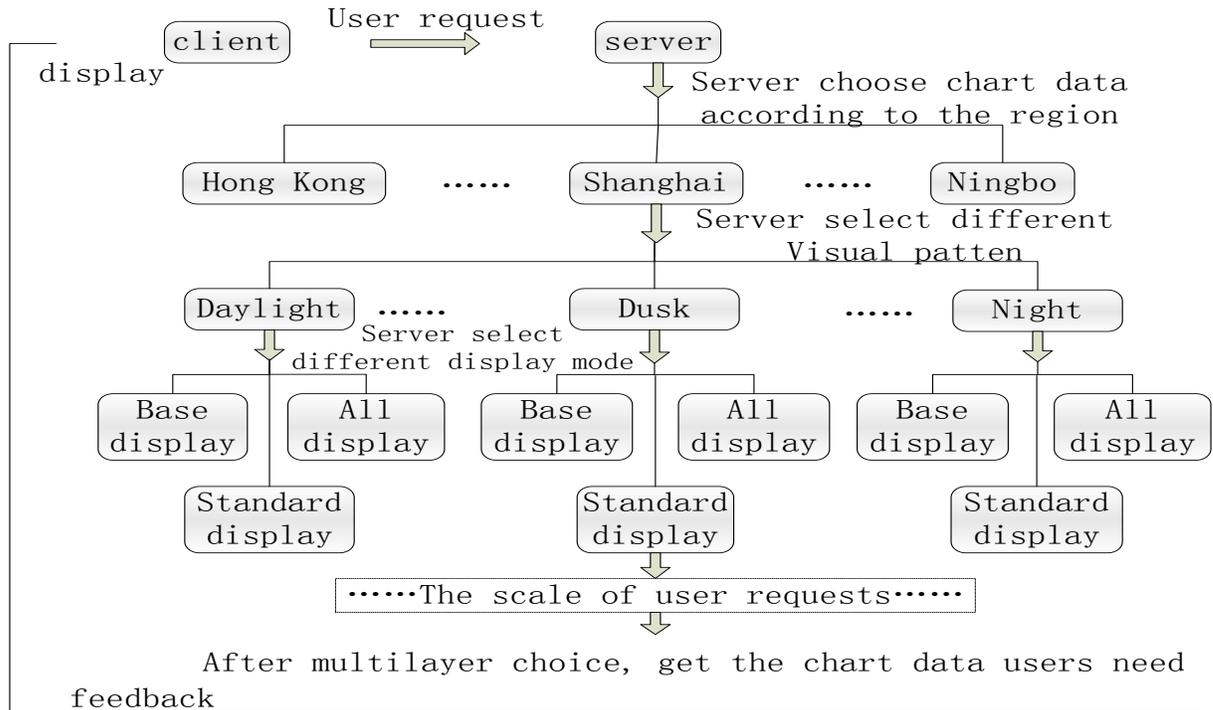


Figure 1. The process of chart request.

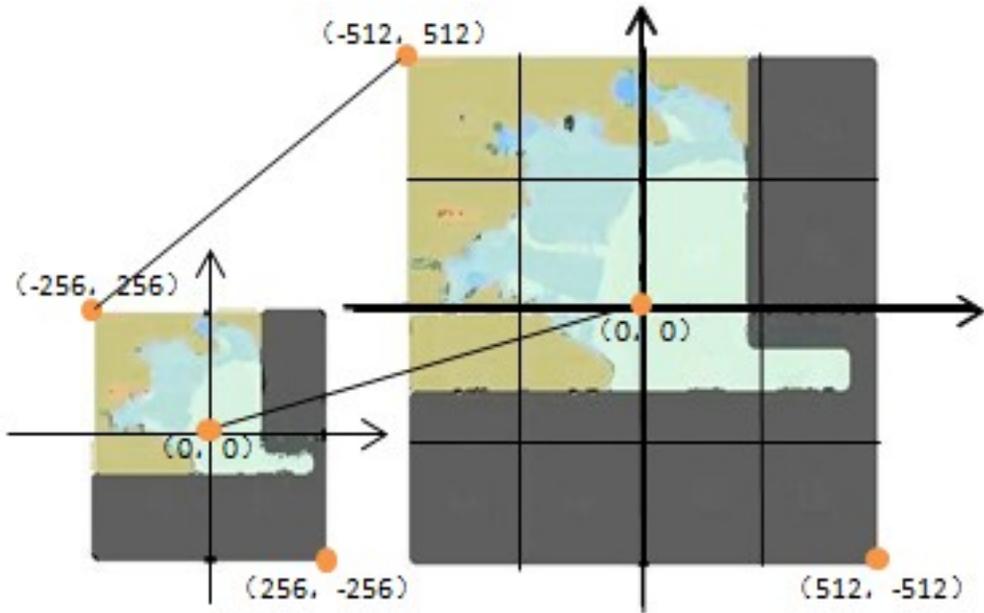


Figure 2. Cutting chart.

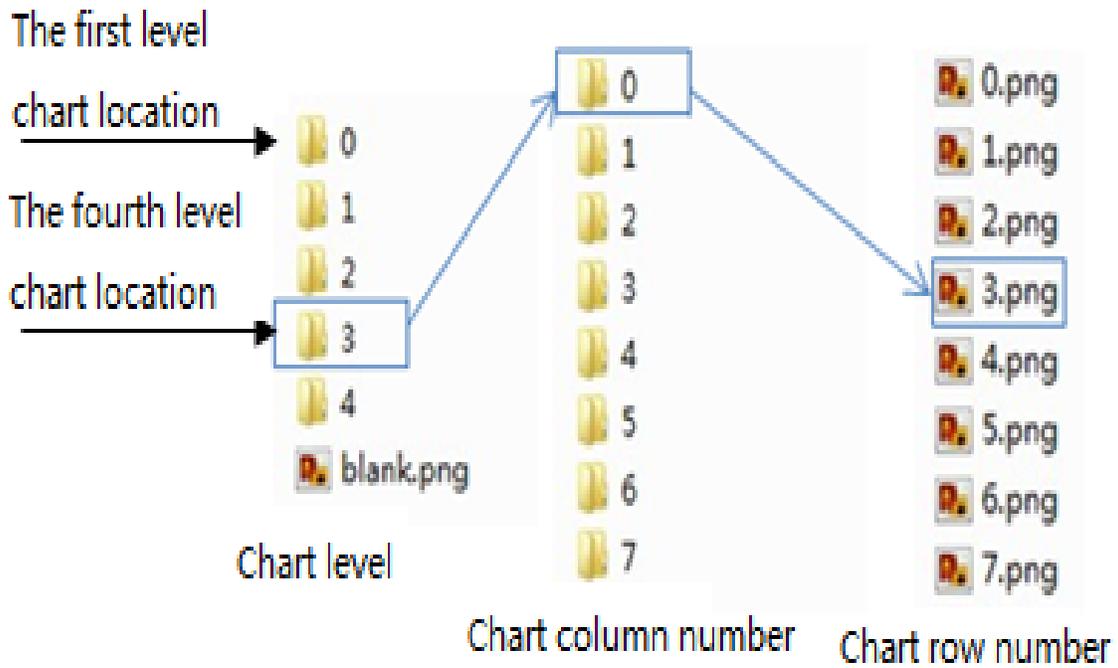


Figure 3. Storage chart images.

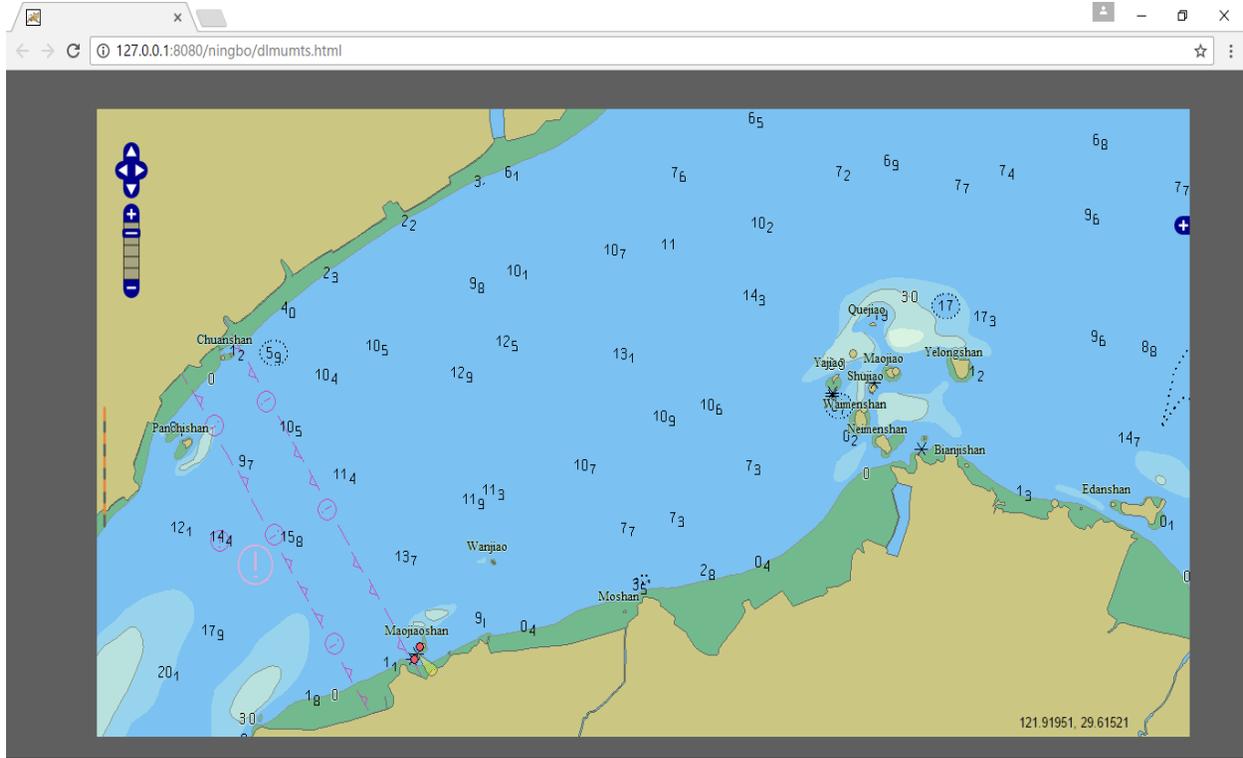


Figure 4. Chart display effect.

III. CHART DISPLAY

If you want to achieve the rapid and correct display of the chart, you still need to solve the two problems, first is chart positioning and the second is quick response.

A. Chart Positioning

The rows and columns that identify the chart image start from zero, The lower left corner of the chart section is defined as (0,0), increasing column number to the right, increasing row number upwards.

When a user drags or enlarges a chart the browser will determine the blank areas in the window and will find the corresponding chart tiles according to the position of the blank space. The browser will then put the chart splicing into the blank Spaces. The request time is very short. It can realize smooth transition and display the zoomed in image swiftly.

The browser's decision process is as follows: Because the amplification of a chart is a step by step operation, so we define the center of the chart as the origin coordinate, (0,0), as shown in figure 4. When a user requests level 4 chart data, this will have a total of 64 tiles, pixel size is 2048 * 2048. The lower right coordinate values are (-1024,1024), if the current computer screen is 1280 * 1280, all the chart cannot be displayed, so we need to drag a hidden area to the display. On a computer screen the point coordinates are set to the (x, y), the requested scale level of a chart is set as A, chart column set as B, chart row set as C, the browser will request chart according to the following code sent to the server:

$$A=this.map.getZoom() \tag{1}$$

$$B=Math.Floor(\frac{\Delta X}{256})= Math.Floor(\frac{x-(-\frac{256 \times 2^A}{2})}{256}) \tag{2}$$

$$C= Math.Floor(\frac{\Delta Y}{256})= Math.Floor(\frac{y-(-\frac{256 \times 2^A}{2})}{256}) \tag{3}$$

$$x = 1024 - \Delta x ; y = 1024 - \Delta y \tag{4}$$

“getZoom” means get the current zoom level, here you can get the value of A is 4. “Math.Floor” means take the lower limit of an integer, so we can calculate the value of B is 4, the value of C is 2. The computer can get the chart tile path used to fill the blank area, link address: [http://..... + 4 + "/" + 4 + "/" + 2 + "." + "png"](http://..... + 4 +), the chart will be downloaded and the blank area displayed. If you are not zooming in, then the browser does not require chart level data, the browser just needs to judge the values of B and C.

B. Chart Quick Request

In section 3.1 we have already obtained the required chart, and have sent the request information to the server. Now we need to solve the problem of the rapid transmission and display of the chart. For this we need to use the Ajax

asynchronous transmission mechanism[9]. To explain this further ; the asynchronous transmission mechanism is not a way to improve the transfer rate of the chart. The velocity of the transmission of the chart data mainly depends on the bandwidth of the current user network, the user's browser version, computer configuration, etc. If the user network is unstable, or bandwidth is too narrow, it will cause slow downloading, quite often the user's web page download will stop, the user cannot operate, he can only wait for the transfer to be completed, when the browser continues the operation after refreshing the page automatically. This makes the user's visit and operation discontinuous and enlarging or dragging the chart needs additional time. Ajax's asynchronous transmission technology can effectively solve the problem, even if the chart data is not fully transmitted, the user can still operate basic functions, i.e. EBL, VRM, cursor, zooming, etc.

To realize the technology we need to add the Ajax engine between browser and server, it runs in the background uninterrupted, processing user requests sent to the server[10], it is entirely "on-demand request data", to avoid some unnecessary data.

Ajax request basic process is:

- Create the XMLHttpRequest and send the XMLHttpRequest ;
- Specify the response function: tell the HTTP request object which JavaScript function to deal with this response. You can set the onreadystatechange property of objects to use JavaScript function name. `httpRequest.onreadystatechange = nameOfTheFunction;`
- Sends an HTTP request, invokes the XMLHttpRequest open and send two functions `http_request.open ('GET', 'http://www.xxx....', true);`
`http_request.send (null);`

Use Open () to create a new request. The first parameter is the HTTP request method ; GET, POST or HEAD. The second parameter requests the URL of the page, the third parameter sets whether the request is asynchronous or synchronous, the default is true, the JavaScript function will continue without waiting for the server response. Use the send() function to send data.

- The server response. When dealing with the information returned from server to browser, the first thing to check is the status of request, this is because it is only when the complete server response has been received, that the function can perform the response. XMLHttpRequest provides the readyState attribute to determine the server response, the attributes include five possible values, only when the readyState is 4, the function can perform the, return to the page.
- The browser receives the information returned.
- the browser deals with the information from the server.

IV. EXPERIMENTAL RESULTS

The test method is to install the server and browser under the general net bandwidth in two faraway place, the

browser visit the server in the morning, noon and night respectively, test for a week, get chart display complete time, Test whether accord with S57 display time requirements. The test results are as follows.

Install the server in Dalian City, Liaoning Province, China, the browser address in Ningbo city, Zhejiang province, China, the distance between them is about 1000 km. the bandwidth of the server and browser are 4M. The chart display time within 4 seconds for the first time, when repeat the operation, time within 2 seconds, display effect is shown in figure 4. With reference to S52 requirements, electronic chart display time should be less than or equal to 5 seconds. According to the experimental results, the design scheme can satisfy the requirement of S52 under the general net bandwidth. Repeat operation completion time is far lower than the requirements of S52, design scheme is feasible.

V. CONCLUSION

Through the rapid chart release technology based on the chart tiles the electronic charts meet IMO S57 data transfer standard and IMO S52 chart display standard. The charts are cut and stored in tiered format according to the scale. The browser/client display is based on Openlayers which allows chart tiles so be selected quickly. This scheme is completely built on open source software and code, it does not rely on any third-party commercial software. The system combination achieves efficiency, it is also convenient to modify and to extend functionality if required. The realization of the system mainly divided into client and server, a lot of work on the server side implementation to alleviate the burden of the client.

ACKNOWLEDGMENT

This research was sponsored by Ningbo University Funds (Project No. xk113d214); Ministry of Education of Humanities and Social Science Youth Fund Projects(14YJCZH081).

REFERENCES

- [1] Vadus, Joseph R(1995). Electronic nautical charting: role in sustainable development of marine transport operations. Oceans Conference Record (IEEE), 3, 1955-1967.
- [2] Pan Mingyang, Pang Bobo, Li Chao Etc.(2014). Research on web mapping of vector nautical charts based on HTML5. Proceedings-2013 International Conference on Information Science and Cloud Computing Companio. 1, 611-617.
- [3] Zhu Feixiang, Zhang Yingjun, Zhu Berlin(2007). Web system for electronic nautical charts service based on ArcIMS. Proceedings - International Computer Software and Applications Conference. 2(1), 657-658.
- [4] Kam, Seung Chul, Lee, Seong Dae, Park, Hyu Chan Etc.(2006). Web-based system for electronic navigational charts using GML and SVG. Proceedings - 5th IEEE/ACIS Int. Conf. on Comput. and Info. Sci. 485-490.
- [5] IHO Transfer Standard for Digital Hydrographic Data. Edition 3.1.International hydrographic organization.
- [6] Specifications for Chart Content and Display Aspects of ECDIS. Edition 6. International hydrographic organization.

- [7] Domanski Adam, Domanska Joanna, Chmiel Sebastian(2014). JavaScript Frameworks and Ajax Applications. *Communications in Computer and Information Science*. 431(25), 57-68.
- [8] Huang Ping-Yu, Jan Jihn-Fa(2011). Comparison of Google maps API and openlayers for WebGIS development. *32nd Asian Conference on Remote Sensing 2011*. 4(4), 2379-2382.
- [9] Ying Ming, Miller James(2013). Refactoring legacy AJAX applications to improve the efficiency of the data exchange component. *Journal of Systems and Software*. 86(1), 72-88.
- [10] Zaidman Andy, Matthijssen Nick, Storey Margaret-Anne(2013). Understanding Ajax applications by connecting client and server-side execution traces. *Empirical Software Engineering*. 18(2), 181-218.