High Performance Internet image Server Design and Implementation

Genyuan Zhang

The Zhejiang University of Media and Communications, Hangzhou, P.R. China

zgenyuan@163.com

Keywords: internet image server, image patch analysis, image patch predicate.

Abstract. How to provide cost effective, high performance image patch query and image patch analysis implementation is one important issue for internet image server design. In this paper, we present one novel approach to internet image server design and implementation. Like most modern internet image server implementation, this system stores pre-rendered image tiles for different scales on the web server, which are then laced together on the client browser. The key issues and solutions of image patch query, image patch predicate and image patch analysis are proposed and illustrated in this paper. Furthermore, the key details of the algorithm design are presented in this paper. The prototype adopting the above methods is presented and the related benchmark result details reveal the effectiveness and efficiency of the proposal.

Introduction

Image information service has gained wider and more advanced application in recent years. Web has become the dominant platform for facilitating the access, processing of information. Retrieval and display image information according to some query criteria (including image patch predicate) is the essence of image server. But getting this functionality from modern internet image server product is CPU intensive and memory demanding, if not impossible. How to provide cost effective, high performance image patch query and image patch analysis implementation is one important issue for internet image server design. In this paper, we present one novel approach to internet image server design and implementation. The key issues and solutions of image patch query, image patch predicate and image patch analysis are proposed and illustrated in this paper. The prototype framework adopting the above methods is presented and the related benchmark result details reveal the effectiveness and efficiency of the proposal.

System Architecture Design

On the client side, Flex entered the rich internet application (RIA) scene, leveraging its ubiquitous cross-platform flash player by creating a programmatic way to make a flash application. This programmatic approach uses Flex's core languages: the XML templating language (MXML) and its scripting language (ActionScript)[2]. Flex integrates well with J2EE using an additional server-side layer known as LiveCycle Data Services or BlazeDS deployed on application server. This additional layer facilitates Flex application to invoke and communicate directly with Java classes on the server so that they can be called and accessed from the Flex application. The request pre-processed and forwarded by the servlet is processed by the corresponding java modules(image patch identification, image patch query and so on) which are backed up by C++ implementation components. C++ is chosen as the implementation language for its platform independence and high performance.

Like most modern internet image server implementation, this system stores pre-rendered image tiles(and corresponding index files whose mechanism will be discussed in later section)for different scales on the web server, which are then laced together on the client browser. The image images can be made up of single layer or combined layers to make a base image or background image. Using an image cache allows more complex symbology and cartographic quality. Pre-rendering is also useful to improve server side performance – at the expense of some disk space. One interesting and

profound finding is that: the cached image tiles are not only the visualization of vector data but also the approximation of the exact geometry.

It is not hard to figure out thatraster approximation is much more similar to image patch boundary and shape of features than MBR. At the same time, raster approximation can assist to judge image patch relationship. For example, if one grid identified overlaps with the same grid. The grids overlapped are raster approximation of features intersection. When to vectorise approximation, it can obtain image patch operation such as intersect union erase and so on which become the base of image patch analysis on web. The details of this algorithm will be explained.

To get the tiled images, we utilize the Anti-Grain Geometry (AGG). AGG [3] is an open source graphic library written in industrially standard C++. Basically, AGG can be viewed as platform independent, high performance and lightweight rendering engine that produces pixel images in memory from some vector data. It can render arbitrary polygons or lines with anti-aliasing and sub-pixel accuracy. Traditional method for computer graphics is only conscious of the geometry coordinates and the feature information will not be kept in the rendered image [4]. If we keep the feature ID in one index file (we call this file “shadow image”) which records the feature ID for every pixel of the rendered image, we will establish the imaging relationship between pixel of the image and the feature attribute. We can implement this idea by drawing the polygon using the feature ID as filling color. Some modification to the AGG rendering engine is needed to generate the shadow image. For every cell of rendering buffer, feature ID is kept besides the cover value which records the cell area (in percent) covered by the polygon. The shadow image assumes the role of index file when implementing the special method in Web Imaging Service(WIS) protocol and image patch filters in Web Feature Service(WFS) protocol. If the user wants to know the feature attribute when he/she clicks on one image on browser side, the server can figure out the feature ID from the shadow image by taking the pixel position of click operation as index. This query can be accomplished in constant time on the basis of shadow image.

Instead of search paradigm of R-tree(descending from the root), the searching process used in this new method is based on grid system. Notice also that the generated index file allows the search algorithm to eliminate irrelevant regions of the indexed space, and examine only the grid cells of search region which are specified by the users. The idea is to first allocate width_tile*height_tile*3 bytes of memory for the current index image (here, width_tile and height_tile are the width and height of the index image). For point query, the search point’s coordinates need to be converted to the row and the column on the index image(specified grid cell in memory through scan line). Then resolving the color at this position to geographic feature ID. The color components can be got as follows:

For region query, after determining the region row and column according to users’ inquire demand, the specified lines should be scanned in loop in memory image. At the same time, several different values of RGB can be calculated. Thereby, the desirable IDs could be easily got transformed from these color values.

Patch Predicate

The image patch filters in WFS specification provide one convenient way of retrieving features from server side by specifying some image patch predicate.. Image patch extension to relational database management system (oracle image patch for instance) or geometry library (GEOS and Terralib for instance) often takes the approach of computational geometry, which is CPU intensive and memory demanding, to the implementation of image patch predicate and image overlay. The work of [5] motivates us to work on one viable approach to image patch predicate and image overlay. In fact, in modern internet image server, the image tiles is pre-rendered or dynamically rendered on initial client request. In such scenarios, the cached image tiles are not only the visualization of vector data but also the approximation of the exact geometry. Such images can provide more precise approximation than minimum bounding rectangle (MBR) and can be utilized as the basis of image patch predicate and image overlay implementation. It is not hard to figure out that taking the approach to image patch predicate or image overlay requires two steps: (1) the vector data utilizing the method

in above section; (2) overlay operations and image patch relation judgment based on raster representation. The four color raster signature (4CRS) [5] uses four colors representing an intersection type between the feature object and the grid cell: empty (the cell is not intersected by the polygon)、weak(the cell contains an intersection of 50% or less with the polygon)、strong(the cell contains an intersection of more than 50% and less than 100% with the polygon) and full(the cell contains an intersection of 100% with the polygon). Only strong × strong is determined situation and weak × weak, strong × weak and weak × strong are uncertain situations needing further calculation. In the calculation of the approximate area and confidence interval, it uses mathematical expectation and probability formula to estimate which may not be suitable for the real data. The rendering engine proposed in above section can record coverage area of border grid cells accurately based on sub-pixel accuracy, so it can determine whether two polygons overlap or not through judging coverage area of the corresponding grid cells. For example, given that coverage area of one cell in the first layer is 49% and that of the second layer is 52% in the same grid position, it can determine that two layers overlap in this cell because the coverage area sum is more than 100%. But it is uncertain situation if using 4CRS because it can not handle weak × strong condition (not to mention weak × weak situation). At side effects, the rendering engine can preserve feature attribute information in the cell structure which offers more useful hints for image overlay (polygon IDs etc.).

Given that there are two input feature sets-A and B, judge whether the two layers overlap or not. If it is true (regarding the overlap possibility as filtering condition), return the qualified polygon IDs. The pseudo code is listed as follows.

For image overlay analysis, most requirements (the result layer's feature count, the area and ID of every feature of result layer and so on) can be met through the method mentioned.But if the user wants to get the geometry of every feature, we need vectorization of raster representation of the result layer. The design and implementation details can be found in [6] which gives thorough discussion on the key steps and implementation tricks.

Benchmark and Experiment Results

Runtime environment is notebook computer T43 from IBM with Intel Pentium M 760(2.0GHZ,) and 2GB memory (DDR2). Hard disk speed is 5400 r.p.m. Operating system is windows xp(service pack 3). The two input feature sets (D1 is sparse dataset and D2 is dense dataset) are provided in ESRI shapefile format and the information are displayed in Table I. In the experiment, the first dataset D are random polygons. When do operators (overlap and contain) one million times, the average time of one transaction compared to the corresponding module of GEOS (Geometry Engine - Open Source)[7] is listed in Fig.5. The experiment and data analysis shows: for sparse dataset, efficiency of image patch predicate operations is improved not so much, but for dense dataset, time cost is cut down by about 60 percent. The result of point query performance (in Table below) shows that: for point query, the proposed approach presents better performance than the R-tree method whichever data set chosen. When searching for one object which contains the given points, the algorithm will find the row and the column on the index image according to its geographic coordinates and resolve the color of this grid to real ID. The computational complexity is constant. By contrast, the R-tree approach will search data from its root which degrades the query performance for redundant search paths.

Conclusion and Future Works

This paper presents one approach to internet image server design and implementation. The key issues and solutions of image patch query, image patch predicate and image patch analysis are proposed and illustrated in this paper. The prototype framework adopting the above methods is presented and the related benchmark result details reveal the effectiveness and efficiency of the proposal. As future works, we will conduct more complete tests and evaluate the use of new algorithm in the performing of image patch predicate between sets of polygons and so on. Still, a

more detailed study must be carried on to determine dynamically what is the best number to be used as limit to the maximum number of cells.

References

[1] P. A. Longley, M. F. Goodchild, D. J. Maguire and D. W. Rhind, Geographical Information Systems: Principles, Techniques, Management, and Applications, 2nd ed., New York: Wiley, 2005,pp.97-99.

[2] T. Ahmed, J. Hirschi and F. Abid. Flex3 in Action. Manning.2009.

[3] http://www.antigrain.com/doc/index.html

[4] J. D. Foley, A. V. Dam, S. K. Feiner and J. F. Hughes, Computer Graphics: Principles and Practice in C, 2nd ed., Addison-Wesley Professional, 1996, pp.67-72.

[5] L. G. Azevedo, G. Zimbrão and J. M. Souza, “Approximate Query Processing in Image patch Databases Using Raster Signatures,” VIII Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 19-22, 2006, INPE, p.3-17.

[6] Hui Dong, Zhenlin Cheng and Jinyun Fang, “One rasterization approach algorithm for high performance image overlay,” The 17 International Conference on Geoinformatics 2009, 12-14 Aug. 2009.

[7] http://wiki.woodpecker.org.cn/moin/lilin/geos-introduce