DISPLAYING LARGE AMOUNTS OF IMAGE DATA ON SUN-BLADE2000 WITH MOTIF

Chunquan Cheng [a,] , Jixian Zhang [a], Qin Yan [a], Bin Jiang [b] ,Xinsheng Kang [b]

[a] Dept. of photogrammetry, Chinese academy of surveying and mapping ,BeiTaiPing Road, Beijing, cspring@china.com.cn

[b] Dept. of Recce Reconnaissance, CPE, Jianshe Road, RenQiu, China , jjolly@163.com

KEY WORDS: Image Processing, , Motif, Sun Blade2000, Scrolledwindows, Scrollbar , displaying large amounts of image data

ABSTRACT:

For the limits of computer EMS memory and CPU speed,people ofen show or operate a large amount of image data such as remote sensing in computer by incising the data bulk when reading them from hard disk,It can quicken the speed when opening a file in seeing, but when we zoom or rotate the image showing on screen,it look so slowly for not all data is in memory and data exchanging exist between hard dist and memory. In the past several years, performance of computer hardware has being improved higher and higher ,computer physics and virtual memory ,for SUN Blade2000 as examble,reach to 8 GB maximum, so it is not difficult reading all data from a about 1.5G amounts of data file. But there would have many disadvantages if such large data in memory is not dealt with properly. This article describe some advantages of reading all data into memory when opening the remote sensing image file, and some ways to show, to zoom and to operate the undivided data without taking up additional memory of the computer.In this article we use motif to deal with a user interface for graphical.

About SunBlade2000 and Motif

The Sun Blade 2000 system accommodatas up to two Superscalar, 64-bit, high-performance UltraSPARC III Cu CPUs. It features a high-performance, crossbar-switch system interconnect that provides high bandwidth (up to 4 GBps) for ultra-high-speed processors and graphic subsystems. It also delivers plenty of internal disk and memory and a 64-bit PCI bus for incredibly fast I/O. The Sun Blade 2000 workstation provides both USB and IEEE 1394 interfaces for connectivity to leading-edge third-party peripherals. With high-end 3-D graphics, dual-monitor capabilities, and support for Sun's storage systems, this workstation is truly a powerful, flexible next-generation desktop.

Motif is a set of guidelines that specifies how a user interface for graphical computers should look and feel.It realizes how an application appears on the screen (the look) and how the user interacts with it (the feel). Look and Feel is not something specific to Motif; all windowing toolkits should present a standardized internally-consistent methodology so that the user is comfortable using the controls which the application presents. Specific toolkits, however, have distinct look and feel, although since some toolkits share a common design philosophy there is a cross-over so that users familiar with one platform are not necessarily naked when presented with an alternative.

ScrolledWindows and ScrollBars

The ScrolledWindow provides a convenient interface for displaying large amounts of data when we have limited screen real estate. For most situations, the automatic scrolling mode is all that we really need. In this mode, a ScrolledWindow requires very little care and feeding. By installing callback routines on the ScrollBars, we can even monitor the scrolling actions. However, there are some drawbacks to the automatic scrolling mode: all of the data must be rendered into the work window widget and scrolling occurs in single-pixel increments. If the size of the work window that we need is prohibitively large or if we need to support scrolling in other than single-pixel increments, we must use application-defined scrolling.

There is quite a bit of work involved in supporting real application-defined scrolling because of the different states in the relationship between the size of the work window and the underlying data. We must be able to support not only the underlying data, but also the way it is rendered into the work window, the ScrollBars, and all of the auxiliary variables required for the scrolling calculations. And that work is just to support the scrolling functionality. When we introduce the complexity of a real application, there is a greater chance of a poor design model.

Displaying large amounts of image data

Having the special characteristic of Motif, Motif is one of the best toolkits to design the object that make up a user interface for graphical.with such high-performance Wokstation of SunBlade 2000, It's easy to read all data about 1.5G amounts to memory. But in library of X-Windows and Motif, there is only one function to display image data on screen.It is followed:

```
XPutImage(display, d, gc, image, src_x, src_y, dest_x, dest_y,
          width, height)
    Display *display;
    Drawable d;
    GC gc;
    XImage *image;
    int src_x, src_y;
    int dest_x, dest_y;
    unsigned int width, height
```

From above we can see that we must have all data of image in memory and get its width and heigh before the function is used. When all remote sensing data have entered the emory,with

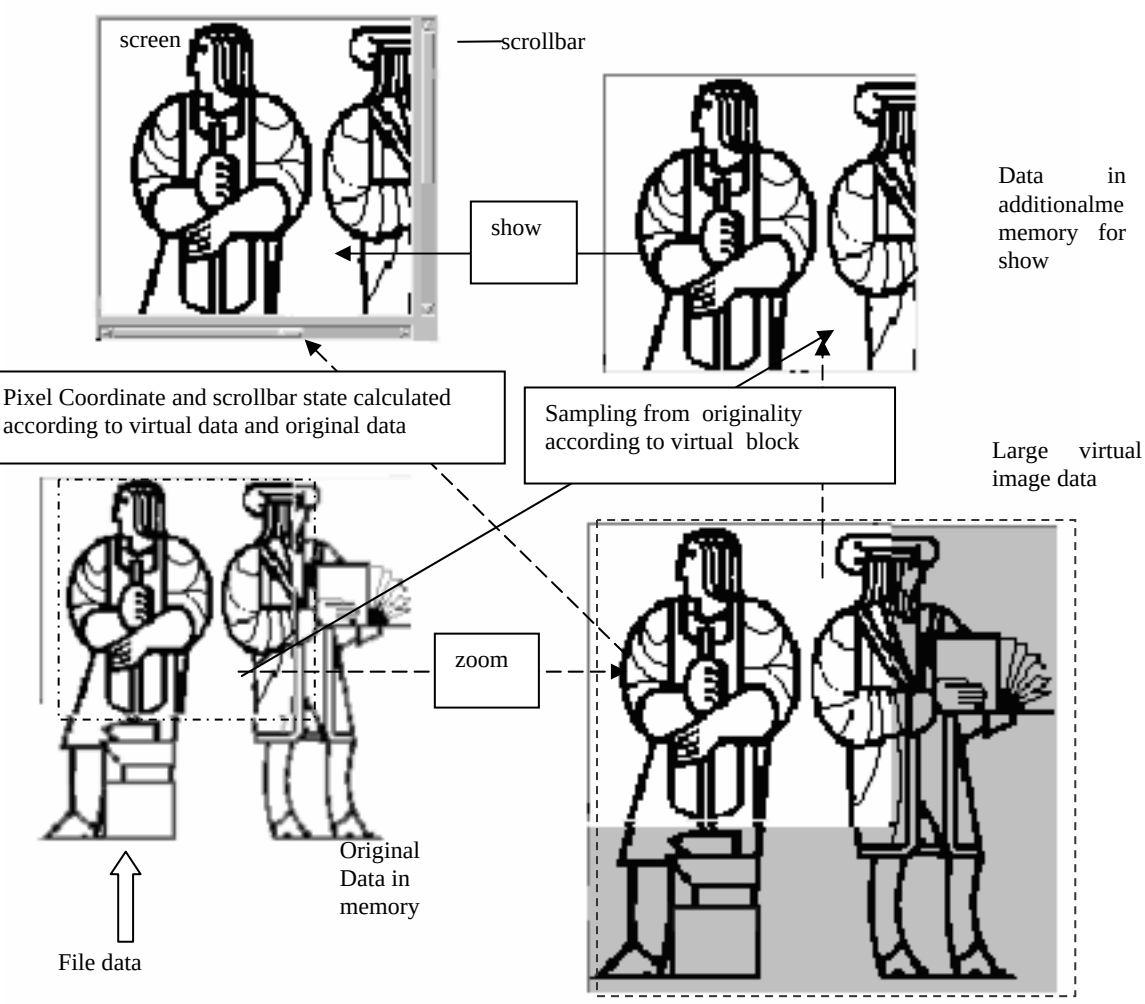
this function we can show the image without zoom with automatic scrolling mode.the states of scrolledwindow is settled by itself. But we must get all resampled data using auto scrolling mode or using above function when we zoom a image on screen .It is out of the question when we zoom in a image for the follewing reason: first we can't alloc so large EMS memory to hold the resampled data,second we can't wait for the computer to cost so long time to resample ,and finally the auto scrolled mode don't s bear so great data. So we can only use application-defined scrolling to show image,and the ScrollBars and all of the auxiliary variables required for the calculations of scrollbar and coordinate showing on statebars.

To avoide the waste of time and memory and make the scrollbars and states messages in same step with operating of image showing,We must write code not only to control imgae cruising and zooming but set the states of scrollbars,show on statebar the coordinate of image pixel.The following base function is write for zooming and cruising of image.

void StrechImg(display, win, gc, image, src_x, src_y, dest_x dest_y, dest_width, dest_height, s)
Display *display,
Drawable win,
GC gc,
XImage *image,
int src_x,
int src_y,
int dest_x
int dest_y,
int dest_width,
int dest_height,
double s

In this function, input parameters include original image data and its row and line amounts,the start place of screen on which image want to show,the ratio(double s) to zoom and so on. With it we can roam over or zoom a image freely with little additional memory room.The way we realize the function is as following: First set the states(size and position in work windows) of scrollbar in keeping with the handling on work windows by mouse or keyboard, then get the area of work window and alloc the same size of memoy, resample the pixel values want to show on screen based on zoom ratio and position of scrollbars , put the calculated values to the alloced memory, and put all resampled value on screen with XputImage() function, last calculate the coordinate value of pixel which is the nearest to mouse moving on windows according to zoom ratio and relative location of showing image to all image before showing the pixel coordinate value on statesbar .

The following picture show their relative. We can see that we only get showing data from original data by resampling and calculation to display on screen directly. This means that we can place a large view area inside a smaller one and then view portions of the view area. So it is easy to carry out roam and zoom and set position and size of scrollbar when we show large amounts of remote sensing image data.



large amounts of image processing

For all original remote sensing data has existed in memory, we can see the result immediately if appoint part of image want to be deal with such as we want to see the part result of image strengthen and recover selected by mouse on screen,we can follow above ways mentioned.

Conclusion

By test,It takes the computer about one minute and a half to read all 1.5G image data to memory from hard disk.when we drag the showing image or zoom or even zoom out all image to fit to screen we don't have the feeling of waiting for the result.In this course of operation,we don't set up image pyramid, so this technique is suit for a great deal of operation during image is showed on screen,It is quite a good choice for it don't cost a lot of time during working with big block image data.and the practicality of this technique is creasing more and more while the performance of computer hardware is getting better.

References

MOTIF PROGRAMMING MANUAL (VOLUME 6A)
Chapter: 10 ScrolledWindows and ScrollBars
http://www.ist.co.uk/motif/books/vol6A/ch-10.fm.html

Sun Blade 2000 Workstation–Overvie
http://www.sun.com/desktop/workstation/sunblade2000

Chapter: 4 The Main Window
http://www.ist.co.uk/motif/books/vol6A/ch-4.fm.html