16

New Techniques in TH-DAIMS 2.0

Xinyou Li, Zesheng Tang, Hanwen Huang
CAD center, Tsinghua University,
Beijing 100084, P.R.China
Tel 01-2571980, Fax 01-2556935
Email dcstzs@tsinghua.edu.cn

Abstract

TH-DAIMS is a drawing renewal and management system which is based on binary image process. New techniques in TH-DAIMS 2.0, which are picking primitives from image data, smoothing edges of lines, mapping memory to hard disk for large drawings, adding Chinese characters to user interface, and making database management system suitable for both image and text data, are described.

Key words

Drawing understanding, Drawing renewal system

1 BASIC FUNCTIONS OF TH-DAIMS 1.x

TH-DAIMS is an engineering drawing renewal system which is essentially based on binary image process technique. It first scans drawings into binary image data by use of optical scanner, such as CONTEX FSS5200. In actual application, for poor contrast blueprints, It first scans drawings in grey image, and then converts the grey image into binary image according to a threshold which is a local lowest grey value selected from grey distributions of the grey image. Drawings can be scanned piece by piece in a smaller scanner and then merged into a single image.

TH-DAIMS provides a tool to clean image, which includes removing spots and filling up holes. It can aid engineers to interactively modify contents of drawings in binary image format without any vectorization by providing them many useful interactive design facilities. Available interactive design tools in TH-DAIMS are line, polyline, arc, circle, rectangle, polygon, arrow, ellipse, text, and Chinese character.

The image data of drawings and some attributes of the drawings can hierarchically be stored into FOXBASE compatible databse by the types of projects and can be accessed through image database management system. Vector data from other 2D CAD systems can also be accepted For example, DXF, the data exchange file of AUTOCAD 11.0, can be read

Q. Sun et al. (eds.), *Computer Applications in Production Engineering*

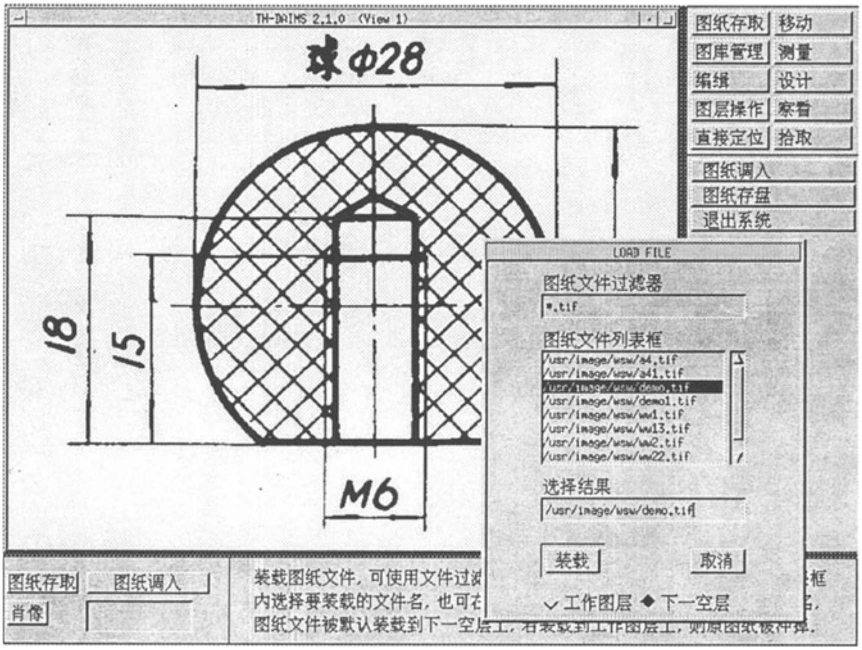


Figure 1 TH-DAIMS user interface with kanji menus and messages.

and interpreted into image data. DXF, the data exchange file of AUTOCAD 11.0, can be read and interpreted into image data. New drawings can be easily and rapidly printed by image data printer, such as HP DesignJet 600.

TH-DAIMS has a well defined graphic user interface which consists of one message area, two level menus, two view regions, four level drawings and temporal dialogues, as shown in Figure 1.

2 TECHNICAL OVERVIEW

The motivation to develop TH-DAIMS is that in Beijing heavy electrical machinery plant, houndreds of drawings have to be drawn when a new product is designed. But most of the drawings are much similar to the one of some old products. Only a little modification is needed. The designers have to spend a lot of time on drawing the drawings again. They need a system which can read the drawings to computer, provide them some interactive design tools for making a little modification, and finally print out new drawings.

Vectorization (Nagasamy, 1990 and Parker, 1988) is one way to solve the drawing renewal problem. Vectorization system converts binary image into vector automatically or manually. Automatical vectorization is to extract vector information from image by use of characteristic extraction, AI recognition, and thinning techniques. Manual vectorization is first to display the image on screen or put up the drawing on digitizer as background, and then interactively draw vector primitives, for example, line, circle and text, according to what displayed in the screen or digitizer. The resulted vectors are then sent to a 2D CAD system in hand to do some modification or further design, and new drawings can be got by printing the vector data.

Although there are not a few vectorization system in sale, the systems can not reach the level of actual usage, because there still remain some complicated problems, such as curve recognition, Chinese character recognition, dimension recognition, to be solved in automatical vectorization system, and the manual vectorization process is so time_consuming and terrible bother that one does not like to use it.

TH-DAIMS is another way to solve the drawing renewal problem. TH-DAIMS is a binary image system. The most simple idea in TH-DAIMS is that if we can provide user an interactive design tool in image data, we do not need to do any vectorization for drawing renewal. But it is not a simple thing. It at least needs to solve the following problems:

(1) Picking graphic primitives from image data
Picking primitives such as line, arc, text is one of four basic interactive taskes (Foley, 1990). The picked primitives can be manipulated by user interactively. For example, copy or move a line segment from one place to another, delete a text string, specially, smooth edges of lines. In graphic system, picking primitives is easily implemented if the inputed primitives are well organized in a data structure. The system searches the primitive in the data structure which is the nearest primitive to a given point on the screen. In image data system, there is no primitive information. Given a point on screen, the system should scan the image data from the point, and extract the primitive. Picking primitives from image data is new idea and new work.

(2) Allocating memory for large size drawings
Large memory space is required by image data system for manipulating image data. For an A0 size drawing with 300dpi, about 16 MByte memories are essential. If a user wants to load four A0 size drawings at a time, it will be impossible in a memory limited PC486. The system should allocate memory space available for large size drawings by mapping memory into disk.

(3) Smoothing edges of lines
Smoothing edges of lines is a distinct task in image data system. The edges of lines in original image data are not smooth enogh to pass the qualitative verification of engineering drawings.

TH-DAIMS (was named IMCAD) is one of the 8th national 5 years projects and has been finished in last summer (Li, 1993). From that time, we began to put TH-DAIMS 1.x in market, and it was well received by users in many application fields. Till now, TH-DAIMS 1.x got more than 20 users to use it.

At the same time, we developed a new version of the system, TH-DAIMS 2.0. TH-DAIMS 2.0 will be better than TH-DAIMS 1.x in the following aspects:

(1) Picking line segments from image data;
(2) Smoothing edge of lines;
(3) Allocating memory for large drawings;
(4) Adding Chinese characters to user interface;

(5) Making database management system suitable for both image and text data.

In the rest of this paper, we will describe some techniques used by TH-DAIMS 2.0 in detail.

3 PICKING LINE FROM IMAGE DATA

Although Hough transformation (Illingworth, 1988) can be used to extract line from image data, we would like to use local scanline algorithm to pick lines from image data. Given a point (x_s, y_s) on screen, the algorithm will scan image data line by line from the point until a line segment is found.

3.1 Find the parameters which define a line

A line is defined by
$(k,\ b,\ mean\_line\_width,\ start\_point,\ end\_point)$
where $y = kx + b$.

We scan image data from point (x_s, y_s) in X direction. If the inclination of the line is less than 45°, we will exchange X and Y in image data.

For each scanline, we will have a line width L_w and a middle point (x_m, y_m) as shown in Figure 2. The next scanline will be started from $(x_m, y_m + 1)$, and the scanline width will be three times of L_w. The line equation will be given by

$$\frac{y - y_{m1}}{y_{m2} - y_{m1}} = \frac{x - x_{m1}}{x_{m2} - x_{m1}}$$

where (x_{m1}, y_{m1}) is the middle point of the first scanline, and (x_{m2}, y_{m2}) is the middle point of current scanline. The line parameters will be calculated as followings:

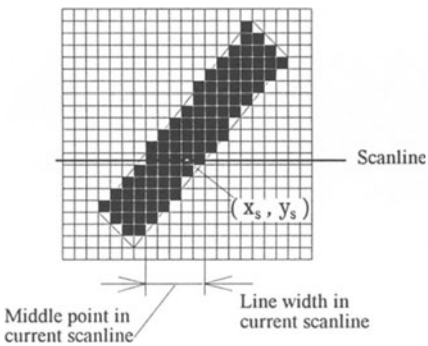


Figure 2 Scanline in an image.

$$k = \frac{y_{m2} - y_{m1}}{x_{m2} - x_{m1}}, \quad b = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}$$

The mean *line_width*, k and b of the line will be computed from the mean sum of L_w, k and b for each scanlines except the one whose slope is far difference from the mean slope.

The start point and end point of the line will be detected if there is no line width in some successive scanlines.

3.2 Find and recover all cross points

The purpose of picking lines is to manipulate them. After a move or deletion, the cross points of the picked line with other lines will be destroyed. Figure 3 shows this case. To find all cross points and to recover these cross points after interactive manipulation are necessary.

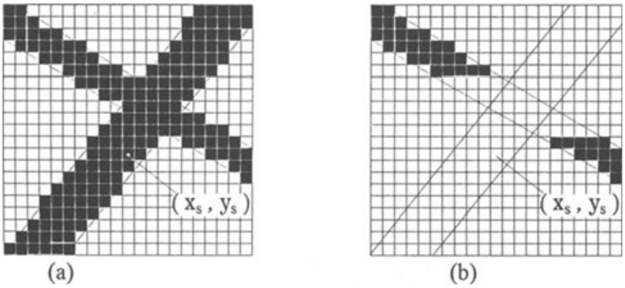


Figure 3 (a) Two lines intersected (b) Cross point destroyed.

When two segments in one scanline within the scanline width are detected, there may be a cross point. See Figure 4 where two lines are intersected. We can also use scanline method presented in section 3.1 to find the parameters which define the intersection line.

If the cross point is destroyed after interactive manipulation, we will redraw the intersection line to recover the cross point.

An example is given in Figure 6.

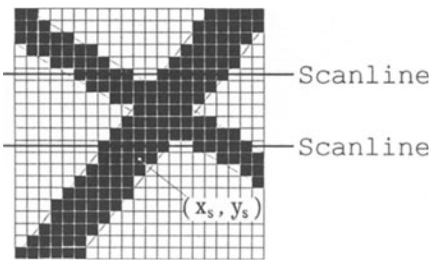
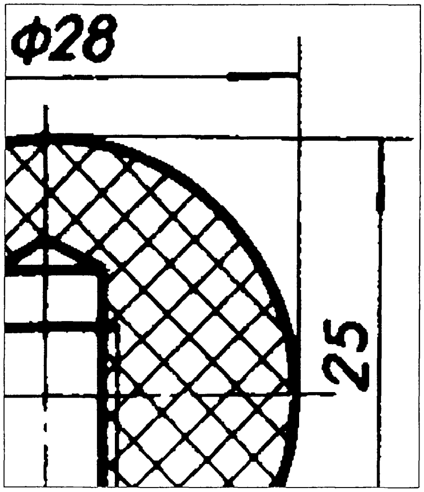
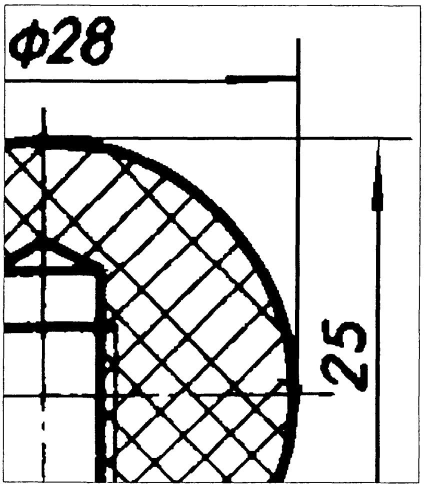


Figure 4 Cross point detection.



(a) Original drawing image



(b) After a line deleted and some lines smoothed

Figure 5 Example of line deletion and smooth.

4 SMOOTHING EDGES OF LINES

We use two methods to smooth edge of lines. One is first to pick lines and then to redraw the lines. Another is to tracing edge of lines. The first method is described in section 3.1. Here we will address the second one.

In tracing edge algorithm, we scan total image two times started from the first horizontal line and the first vertical line, respectively, to record the runlength codes of the image. From horizontal runlength code, we can trace edges of the lines whose angle from vertical line is less than 45°. From vertical runlength code, we can trace edges of the lines whose angle from horizontal line is less than 45°.

By tracing edge, we can have the parameters which define the edge by the same method as mentioned in section 3.1. Using the parameters, we can smooth the edge.

5 ALLOCATING MEMORY FOR LARGE DRAWINGS

The method we used to allocate memory for large drawings is dynamic memory-disk swapping. Here, What we mean by the large drawing is that the size of drawing is so large that it can not be loaded into memory even through all virtual memory are used.

We divide an image data space into $N \times M$ blocks and store them in a mapping file which has an index table. Each element of the index table has a pointer to a head of a block and a structure to record the location of the block in image space. We also have a memory block queue to record the blocks which are loaded into memory.

Block swapping between mapping file and memory block queue is carried out when an image manipulation is applied to a block which is just not in memory block queue. The mechanism of dynamic swapping is shown as Figure 6 where block swapping first reads image manipulation commands from user interface, then decides whether the block swapping is needed or not, and swaps blocks between mapping file and memory block queue when needed. Finally, block swapping calls X_lib API to carry out image manipulation on memory block queue. The strategy of block swapping is that the blocks which are the farthest to current block are swapped to mapping file when block queue is overflow.

Speed for dynamic swapping is critical problem. Table 2 with Table 1 is a test report of dynamic swapping and non_swapping system.

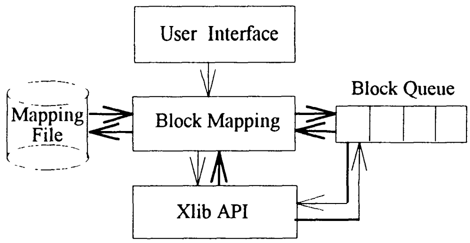


Figure 6 Mechenism of dynamic swapping.

Table 1 Test conditions

Computer	Compaq 386/33m	AST 486/33m
Memory size	8MB	10MB
CUP speed	36	72.1
Disk speed	740KB/s	601KB/s

Table 2 Test results

Manipulation	Seconds for swap/nonswap in PC386	Seconds for swap/nonswap in PC486
Load drawing (A2, 300DPI)	90/60	44/20
Store drawing (A2, 300DPI)	90/70	23/17
Size extend	8/20	2/1
Draw line (70cm, 35°)	25/35	7/2
Draw circle (R=18cm)	40/40	15/2
Move (When swap needed)	2/1	<1/<1

6 ADDING CHINESE CHARACTERS TO USER INTERFACE

Two methods can be used to add Chinese characters to user interface. One is to change UNIX and MOTIF kernel so that an application can have a Chinese interface without or with a little modification. Another is to change all text functions provided in MOTIF toolkit to the functions which can display Chinese characters. We adopted the second method because by the first one the system properties are deeply down while by the second these have not been obviously influenced.

Adding Chinese characters to user interface of an application includes three tasks:

(1) Create Chinese character font(CCF)

CCF can be got from any CC-DOS or CC-Windows but it can not be directly used by X_Window. We first translate CCF into BDF format which is a readable ASCII file and then use *bdftosnf* command to convert it to SNF format.

(2) Display Chinese characters

We use 16-bit text function to display Chinese characters. First we create a multi_font string using *XmStringCreate* function. And then we send the multi_font strings to widgets such as label, title, or other resources.

(3) Receive Chinese character input

Three tasks should be finished. The firest is to grap keyboard to catch user's input, provide user an echo and translate the input into Chinese character standard code. The second is to display Chinese characters in input area. The third is to control cursor movement while user adds or deletes characters or moves cursor.

7 IMAGE DATABASE MANAGEMENT SYSTEM

IDBMS is compatible to FOXBASE in file structure. In order to management both image and

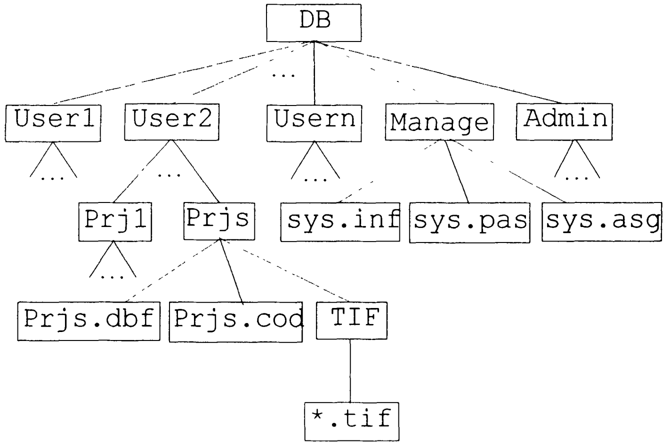


Figure 7 Logical structure of IDB.

text information in a same database, we designed a logical database structure as shown in Figure 7.

A database system has at least a directory called manage in which there are three manage files:

sys.inf information about database system;
sys.pas password for each user;
sys.asg assignment of access right to DBF, DBF record, and TIF file.

A database system may have one or more users. A special user is admin who has a right to access all three manage files. A user may have one or more projects. In each project, there is a DBF file which records text information, a COD file which records image file status, and a TIF directory in which all image files used by the project are stored.

DBF is a relationship database whose structure can be defined by user. DBF is same as the DBF of FOXBASE except that TIF field type and DB field type. TIF field is a pointer to TIF file and DB field is a pointer to another DBF file so that a hierarchical database structure can be constructed.

When user wants to add a drawing to one of his projects, what he needs to do are first to store image data of the drawing into TIF directory and then to add a record about the drawing into DBF file. IDBMS automatically adds a pointer from the record to the TIF file. By the pointer, system can easily find the image data of the drawing from the DBF record when retrieval.

A user can only access his own projects. He can also assign others an access right to one or more his own DBF files, records of a DBF, or TIF files. He can not access DBF file, record of DBF, or TIF file of others if he has no the right. A user can add his own crypogram to his TIF files so that others can not understand the means of his drawings recorded in the files even through the others have a way to get the TIF files of his drawings.

A plant or institute can create one or more such image database system in network environment with optical disks. IDBMS can access these database systems through ethernet and can keep concurrence by locking access right to a DBF or TIF file.

8 CONCLUSION

We have presented some techniques of TH-DAIMS2.0. Used these techniques, TH-DAIMS 2.0 becomes more powerful.

There still remain some problems in these methods to be solved. For example, in picking primitives, we only described picking line. Other primitives, such as arc or circle, arrow, symbol, are not addressed. In smoothing edge of lines, how to speed up the process is what we concentrated on. We expect these problems will be solved in the near future.

9 REFERENCES

Foley, J. D. (1990) Computer Graphics Principles and Practice. Addison Vesley Publishing Company.

Illingworth, J. and Kittler, J. (1988) A survey of the Hough transform. *Computer Vision, Graphics & Image processing,* **44**, 87-116.

Li, X.Y. (1993) IMCAD — A Drawing Reading and Interactive Design System Based on Binary Image Process, in *New Advances in CAD&CG* (ed. Z.S.Tang), 809-814, International Academic Publishers.

Nagasamy, V. (1990) Engineering Drawing Processing and Vectorization System. *Computer Vision, Graphics & Image Processing.*

Parker, J. R. (1988) Extraction Vectors From Raster Images. *Computer and Graphics*, **12(1)**, 75-79.