Wuhan University Journal of Natural Sciences Vol. 5 No. 3 2000, 298～300

Article ID: 1007-1202(2000)03-0298-03

A Generation Method of Mechanical Assembly Drawing in CAD

Guan Zhi-chao, Fang Xiao-chu

Department of Mechanical Engineering, Wuhan Institute of Chemical Technology, Wuhan 430074, China

Abstract: This paper supplies a generation method of mechanical assembly drawing in CAD. By means of the programs of Auto Lisp, the method which based on AutoCAD may complete profiling the parts, forming parts graphic base, calculation of removing hidden line, and establishing assembly drawing.

Key words: Auto Lisp; graphic base; removing hidden lines

CLC number: TP 391.72

1 Overview

A primary objective of using AutoCAD for mechanical design is to generate automatically two-dimensional mechanical drawings. Mechanical drawing consists of parts drawing and assembly drawing. When we create manually mechanical drawings, we need to draw parts one by one in the order of assembly line. After analyzing the positions of every part and its neighbor parts, we then erase the covered portion of parts' profile that is located behind, beneath or right to the view. Similarly, by using CAD, we also need to find out a way to erase the covered profile of parts[1].

There are various shapes of profile, some of which could be extremely complicated. This fact makes line hiding becoming one of the hottest topics in computer graphics, and plenty of line hiding methods have been introduced. The method described in this paper will realize removing hidden lines in AutoCAD with AutoLisp programs.

AutoCAD is a principal application software used for mechanical design and drawing. The popularity of AutoCAD is based on its powerful drawing and graphic editing functions. The whole procedure of mechanical design, from the parts drawing to the assembly drawing which is based on the profile of parts, can be accomplished internally by AutoCAD. This makes program design procedure much easier and faster.

2 Plotting of the Profile of Parts

To obtain parts drawing, it is needed to generate a two-dimensional graphic block for each part, according to its shape profile (only include what needs to be illustrated in the assembly drawing, exclude technical structure). All the two-dimensional graphic blocks are preserved for the future use. This procedure is accomplished by using the drawing commands provided by AutoCAD and Wblock[2]. When this is done, we have the parts drawing for the mechanical design. The drawings of parts in faucet pump are shown in Fig. 1.

3 Program for the Assembling of Parts Profile

3.1 Importing Graphic Blocks of Parts from the Graphic Base

Because it is graphic blocks that we imported, it is necessary to break the entity of graphic block in the program. The program calls explode function and uses a loop to determine whether all the graphic blocks are totally broken. Only those broken graphic blocks can be edited.

Received date: 2000-03-16

Biography: Guan Zhi-chao (1946-), female, Associate professor, research interest: computer graphics and mechanical CAD.

Then the program captures the broken graphic blocks. The end points of parts profile are captured in order, and all the captured end-points are saved into a point set table.

This is the preparation phase of the assembling of parts.

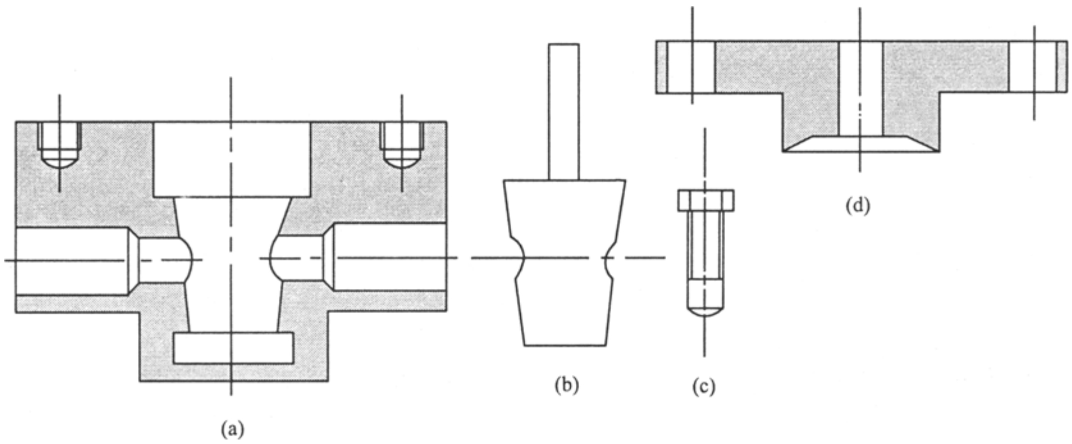

Fig. 1 Parts of faucet

(a): the body of a valve; (b): The faucet; (c): The bonnet; (d): The bolt.

3.2 Removing Hidden Line of Parts Profile

If parts are assembled from inside to outside, every time a part is put in, we need to hide the covered portion of the profile of assembled parts. In order to do so, first we need to find out the point of intersection of two profiles, then to get a set of entities for the covered portion of profile, which is called inter-ent. For each entity en in the set inter-ent, we look for its start point and end point, as well as all the point of intersection of entity en and profile point set. Then we begin to clip, and erase the portion which is totally covered and parts profile no intersect[2]. The program adopts the following techniques:

To get intersectant entity set inter-ent, we use 'F' selecting unit manipulation option to select, where 'F' is abridged from 'Fence' (open polygon). The format is (ssget "F" pt), here pt is a table of points within a definable range. In our program, pt is polylist. The function is:

```
(setq inter-ent (ssget "f" polylist))
```

To get the start point and end point of intersect entities, we use a cyclical function which goes through each entity in the set to detect the entity data tables and returns to 'psel', the list of end points of intersection lines. The format of 'psel' is:

```
( (ps1 pd1) (ps2 pd2)…)
(repeat n
    (setq pse nil)
    (setq ed1 (entget (nth I en1)))
    (setq pst (cdr (assoc 10 ed1)))
    (setq pse (cons pst pse))
    (setq ped (cdr (assoc '11 ed1)))
    (setq pse (cons ped pse))
    (setq pse1 (cons pse pse1))
    (setq I (+ 1 I))
)
```

To get points of intersection set, a two-level cycle is used to request points of intersection of every intersectant entity and every entity of the profile. Function inters returns the points of intersection set pin1. This part of the program is:

```
((while (< I n)
(setq pse (nth I pse1))
(setq j 0)
(while (< j (- m 1))
(setq pin (inters (nth 0 pse) (nth 1 pse) (nth j
polylist) (nth (+ j 1) polylist))
)
if (/= pin nil) (setq pin1 (cons pin pin1)));;to
judge intersectant point. If there is,save it.
        (setq j(+ 1 j))
)
(setq I (+ 1 I))
)
```

To get hidden entity set erase-ent, we use "wp" entity selection option, where "wp" is abridged from "window polygon". The format is

(ssget "wp" pt), here pt is a point set of definable polygon dialog box. In the program
(setq erase-ent (ssget "wp" polylist))
(command "erase" erase-ent "")

Polylist is pt. And the program of this part is:

```
(setq pcen (getpoint "\n Center point:"))
(setq n (length pinl))
(setq I 0)
        (while (< I n)
                (command "block" (nth I pinl)
pcen)
              (setq I (+ 1 I))
        )
(setq erase-ent (ssget "wp" polylist))
(command"erase" erase-ent "")
)
```

In Fig. 2, we use the method above to erase the portion of profile which is covered[2,3].

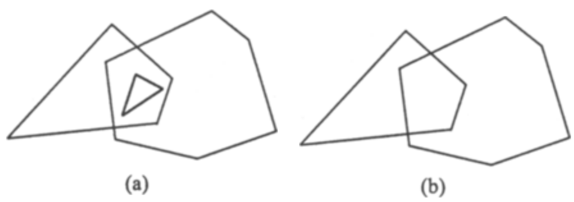

Fig. 2 Climinafion of covered contour line

4 Conclusion

With the procedure above, we can accomplish the whole assembly drawing by putting the parts according to the order from inside to outside, from bottom to top, and from right to left.

Program flow chart:

start graphic processing program
↓
initialization of graphics
↓
get point set of profiles
↓
find points of intersection set
↓
prune and hiding
↓
redraw parts profile
↓
end

The programming and algorithms adopted in this method are simple. It takes advantage of AutoLisp functions and entity function to increase the speed of drawing. Assemble drawing is shown in Fig. 3. Program interactively samples points, which is flexible, general and suitable for assembly drawing of parts with polygon shaped profile. For the case of circle or arc shaped profile, it is necessary to edit them to be polyline. We will adopt interactive method of point selection to preserve the normal processing, then request intersection point, and the principles of this method still works.

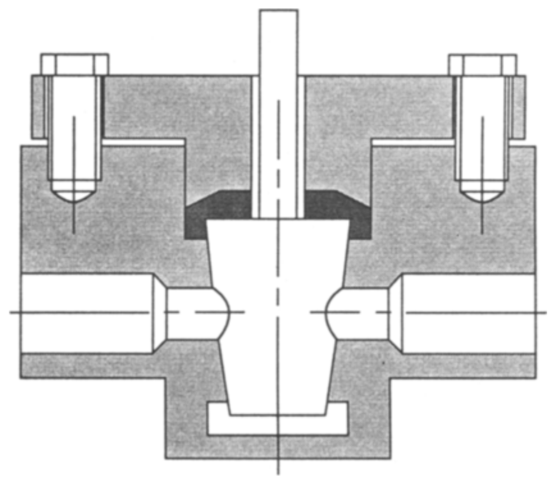

Fig. 3 Assemble drawing

References:

[1] Hearn D, Baker M P. *Computer Graphics*. Newjersey: Prentic-Hall, Englewood Cliffs, 1986.
[2] Liang Chun-xue. *Practicality Tutorial for Auto Lisp*. Beijing: People's Posts & Telecommunications Publishing House, 1998.
[3] Weiler Kevin. Hidden Surface Removal Using Polygon Area Sorting: [Masters Thesis]. Program of Computer Graphics, Cornell University, January 1978.