AutoLisp Automatic Creation Program for Hyper-Fun in Auto-CAD

Hiroyuki Nifuku s1070162 Supervised by Prof. Carl. W. Vilbrandt

Abstract

The purpose of this research is to create a Computer Aided Design (CAD) modeling system for HyperFun, a geometric modeling language based on function represetation (F-rep). We extend the AutoCAD Release 12 program by Autodesk to create HyperFun entities in the same manner that Advanced Modeling Extensions (AME) extended constructive solid geometry (CSG) objects in AutoCAD. This research considers the feasibility of a long term project to create a synthetic CAD system based on F-rep.

1 Introduction

1.1 Auto-CAD

CAD(Computer Aided Design) is a computer software system that makes mathematically based models for designing a car, a machine, a house, a bridge, etc. CAD replaces complicated and time-consuming manual drafting methods. CAD using the Cartesian coordinate system models imaginary or actual objects that are visualized by either wire frames or polygonal meshes in two or three dimensions.

AutoCAD is a CAD software made by the Autodesk company. [1] Auto-CAD (Release 12) is a commercial software which became a de facto standard in the CAD industry, but it is now obsolete and no longer available. It uses AME (Advanced Modeling Extensions), which is an extension of the AutoCAD (Release 12) program to make Constructive Solid Geometry (CSG) primitive objects such as: box, sphere, torus, etc.

1.2 HyperFun

HyperFun [2], a free software, is a simple geometric modeling language, based on function representation (F-rep). It is intended for modeling geometric objects described in the form:

$$F(x1, x2, x3, ..., xn) \geq 0$$

This language is applicable to modeling algebraic and skeleton-based "implicit" surfaces, convolution surfaces, distance-based models, voxel objects, constructive solids, and more general F-rep objects. The model in HyperFun is interpreted by the modeling and visualization software tools.

1.3 Auto-Lisp

Lisp (LISt Processor) is a functional programming language developed by John Mc Carthy in the second half of the 1950s. Lisp is good at sign processing, and it has been widely used in the field of formula manipulation or artificial intelligence. Auto-Lisp is used to make short cut commands or complicated entities/objects/primitives in AutoCAD [4].

2 Research and Implementation

F-rep provides a mathematical description for multidimensional space and is a future paradigm for Computer Graphics and CAD. The F-rep based HyperFun language(HF) uses the HF polygonizer, processing HF text files, to visualize HF objects. The text editing is adequate for making simple HF objects for educational purposes. However, for serious research and professional implementation, the HyperFun language needs a robust interactive CAD modeler.

2.1 First approach for HyperFun AutoCAD Modeler

AutoCAD objects are made from three dimensional coordinate data, and HyperFun objects are made from a continuous function using textual input. Thus, it would seem to be very difficult to export AutoCad models to HyperFun which appears to the user to be arbitrary, lacking dimensional or coordinate data, due to the interface in HyperFun. However, HyperFun does have dimensional methods to make primitive objects: box, cylinder, torus, sphere, and cone, and these primitive objects are made with numerical input such as length, radius, etc., which indirectly results in coordinate data.

2.1.1 HyperFun primitives compared to AutoCAD

As mentioned above, HyperFun has some functions to make primitive objects. However, the HyperFun cone and cylinder primitives differ from AutoCAD.

- Cylinder
  The cylinder primitive is a column (or cylinder with ends) in AutoCAD, but the cylinder primitive is just a cylinder in HyperFun and has no ends, the reason being that the cylinder primitive is a simple function

in HyperFun as shown below.

$$f(x, y, z) = R^2 - (x - x0)^2 - (y - y0)^2$$

Therefore, its length is infinite. Refer below.

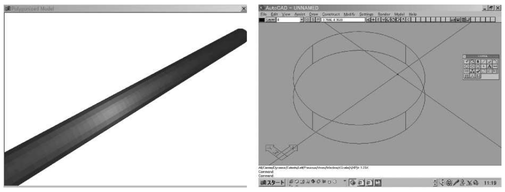

Figure 1: Cylinder in HyperFun and AutoCAD

- Cone
  Cone formula:

  $$f(x, y, z) = (z - z0)^2 - ((x - x0)/R)^2 - ((y - y0)/R)^2$$

  The size of the cone is infinite as seen in the function. It is possible to create ends for the cone, but it makes for programming difficulty.
  Thus, in this research, the cone and cylinder functions are used just as they are in HyperFun. The following are sample graphics of the cone.

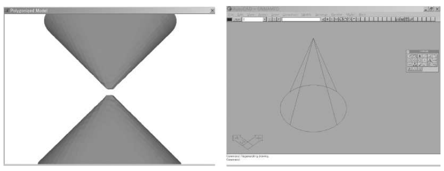

Figure 2: Cone in HyperFun and AutoCAD

2.1.2 Algorithm

This project uses AutoLisp [3] programming to make HyperFun primitive objects in AutoCAD. First, the user inputs the command “ex. hfbox”(below, all HyperFun AutoLisp commands start with the letters hf), and the origin point and the other corner point (same as in AutoCAD). Then, an object is made, and the entity data is placed into a file called “data.dat”. After the user makes all the objects, the user inputs the command “make”, then AutoLisp makes a HyperFun program using “data.dat”. A flow chart is shown Fig.3.

2.1.3 Consideration

This approach has some problems. One of them was the need to create a temporary file (data.dat) and then the creation of a final output file, which was a waste of I/O

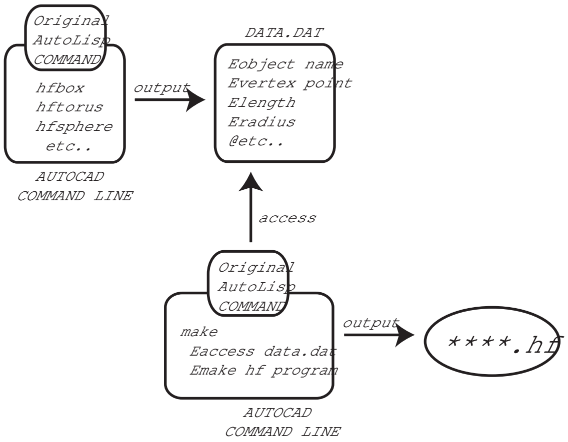

Figure 3: The contents of AutoLisp program (first approach)

capacity, slow, complicated and took a great deal of programming. It also was a problem in that objects made by the original hf command can not be moved or deleted. Therefore, a new approach was researched.

2.2 Second approach

2.2.1 Modifications to previous approach

In the previous approach, the program was made without using AutoCAD original commands. A point of improvement in our new approach is making objects using the original AME commands and using the extended data facilities provided by AutoCAD to attach extended HyperFun data to AutoCAD entities. This means that when an AutoCAD entity is modified or deleted, so is the extended HyperFun data.

2.2.2 Algorithm

This new program is separated into two parts. One is a “get” part, and an another is a “make” part. Details are explained below.

⊙ GET

“GET” command is used to access the AutoCAD entity data in a DXF (Data eXchange Format) structure. DXF is a file format made by AutoDesk for AutoCAD. However, it is become a common format in almost all 3D graphics software because it includes three dimensional data.
The version of AutoCAD used in this research is release 12, and its DXF is divided into four sections: header section, tables section, block section, and entity section. The version after R13 has added a class section and object section. In this research,

an entity section is used to get the data of the entity, that is displayed on a screen. An AutoCAD entity has entity data and XDATA (eXtended entity data). The XDATA feature of the AutoCAD software was used in the implementation of AME primitive objects by AutoCAD. In this project, we used XDATA to implement HyperFun primitives. Refer to Fig.4 for the contents of the entity data section.

Group code	Value Type
-4 ' 999	entity data
1000 ' 1071	extended entity data (XDATA)

using code

-3	Extended entity data (XDATA) sentinel
10	Primary point(start point of Line, center of Circle, etc.)
1010	A point in XDATA
1070	16 bit integerin XDATA

other usable code (future works)

40 ' 48	Floating point value (scale factor etc)
50 ' 58	Angles
1040	Floating point value in XDATA.
1042	Scale factor in XDATA.

Figure 4: Group Code list [3]

In AutoCAD Release 12, we do not fully understand some of the group code for AME. XDATA made from AME has two or more group codes using 1070, for example.

AME's XDATA is the following:

(-3 ("AME_SOL" ("1002."{")
(1070.1)(1071.158946e+006)(1070.1536)(1070.1)

We need to create our own extended HF data structure, because HF has extended requirements beyond that of AME.

⊙ MAKE

"MAKE" command creates a HyperFun program from the data made by the "GET" command. The HyperFun program consists of four parts: a declaration part, a substitution of an initial value part, a function part, and an output part. A declaration part is made from group code 1070, a substitution of an initial value part is made from group code 10, a function part is made from 1010, and an output part is made from group code 10.

AutoLisp "data arrays" are lists, and the "GET" program makes four lists. When processing the data to one section, it needs to transform real numbers to strings by the Real TO String "RTOS" AutoLISP function so that the number variables can be written to a file as a string.

The following program exports an HF data file directly. The flow is shown Fig.5.

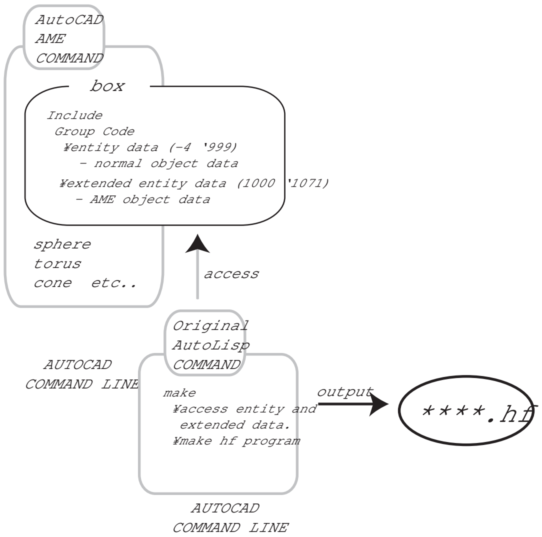

Figure 5: The contents of AutoLisp program (second approach)

3 Comparison

- axis

  In the first approach, primitive HF objects were made by this author's program so that XYZ axis data was easily available, because the program handled all the user's input.
  In the second approach, the AutoCad AME program was used to create a CSG model and the user input is not stored so that XYZ axis data is not easily available.

- move

  The first approach does not allow the object to be modified once the object has been made, because the program writes the object data into an external data file, "data.dat".
  The second approach allows modification of the HyperFun/AME object, because the eXended HyperFun DATA has been added to the AME object.

- speed

The first approach is slow requiring a lot of data collection and processing to create temporary files and consequently can not modify the object but must remake the object in order to edit it.
The second approach is faster than the first because it memorizes to a variable and does not use external file I/O. Furthermore, it uses original AutoCAD commands, and when AutoCAD AME data is modified, the XDATA is modified at the same time so editing is flexible and fast.

4 Result

Several methods were developed to check the data being exported correctly and confirm the placement of the objects. The result is shown in Fig.6.

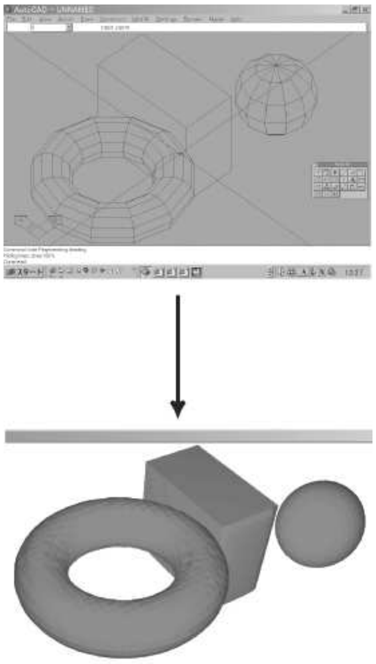

Figure 6: Result of AutoCAD and HyperFun

5 Conclusion

This research shows that it is feasible to extend AutoCAD with the third party AutoLisp program to be a HyperFun function representation modeler. We have been able to make HyperFun objects using the two approaches above. However, there is much left to be done, because the programming is incomplete, and the full implementation of cylinder and cone is complicated.

5.1 Future works

In the future, a robust HyperFun AutoCAD modeler can be completed by adding improvements, such as:

- Rotation angle

  Understanding of the meaning of group code 50, and inclusion to the program.

- Scale

  Understanding or the meaning of group code 40 to 48, and inclusion to the program.

- Subtraction

  Recognizing the trim, and using HyperFun's minus function.

- Cross-sectional creation

  Example, cut the cylinder $cut < bottompoint$ and $toppoint < cut$, and make two ends.

6 References

References

[1] "Autodesk - The Design Resource Leader," Autodesk, Inc., http://www.autodesk.com (current Jan. 2003).

[2] http://www.hyperfun.org

[3] *AutoCAD Release 12 Advanced Tools*, Autodesk, Inc., 1993.

[4] Akemi Ibe, Youji Ebata, Haruna, Mura, *AutoCAD/AutoCAD LT Reference Guide*, Shoeisya, Inc., 2001.

[5] "Mura's Home Page," http://www.mura.sh (current Dec. 2002).

[6] M Hiroi, "xyzzy Lisp Programming," http://www.geocities.co.jp/SiliconValley-Oakland/1680/xyzzylisp.html (current Jan. 2003).

Acknowledgment

The author would like to express appreciation for helpful information on how to get XDATA provided by Mr. Mura's homepage [5] and on Lisp programming provided by Mr. Hiroi's homepage [6].