CAD Data Extraction for Simulation: An Application in Camera View Modeling

Chandan Joardar and Leila Notash
Department of Mechanical Engineering
Queen's University
Kingston, Ontario, Canada

Abstract: In the recent years the process of engineering developments has become more and more computer aided. In almost all areas of product development and manufacturing CAD systems are in constant use to support work in terms of productivity, quality and reliability. Due to the diversity of CAD applications, CAD systems are well specified and appropriate to relevant applications and do not support unspecified extension of applications. With the purpose of extending applications of CAD model beyond the limit stipulated otherwise, efforts have been made in this study to develop a utility to extract the geometry data from CAD models.

The software tool developed here extracts the geometry data of 3D CAD models from its ASCII (American standard code for information interchange) format IGES translations. The program has been developed in Java. The extracted geometry data have been used to populate a data structure of a Java class that resembles a solid model. The solid objects of this Java class have been used to develop some visual modeling manipulation operations. A database has been developed to store data of the solid objects corresponding to the CAD model's desired pose in the visual modeling operation. This study was motivated to extract the geometry data of CAD models in view of its extension to a larger research work on a robot assisted visual inspection system.

1 Introduction

The native database of CAD systems maintains precise definitions of features of solid models. If available, this geometry data can be manipulated in a computer to create simulations of various applications such as path planning, material handling, etc. But most CAD software does not support extraction/manipulation of data unless it is for some other specified CAD operations.

To provide the portability of CAD models from one CAD system to another, there exist some translator utilities. Initial graphics exchange specification (IGES) is one such utility that can translate a CAD file into an ASCII (American standard code for information interchange) format file. A software tool has been developed in this study to extract geometry data of CAD models through ASCII format IGES translations. Subsequent to extraction, the data have been processed to correspond to a desired configuration of the model. The extracted geometry data of CAD models have been used in the development of a camera view modeling. Provisions have been made to store the geometry data for multiple models in a database. Through programming interfaces developed in this study, data can be retrieved from the database to reconstruct solid models, which can again be used in subsequent simulations. This study was a part of a larger study to find applications of CAD data in robot manipulations and the study remains limited to only extraction of the geometry data from CAD models.

1.1 Exchange of Graphical Data among CAD Systems

In order to optimize the achievement of goals of all areas of product development and manufacturing, data describing the technical products needs to be accessible to all applications requiring the data. This leads to strong interconnection between all areas/departments of the enterprise involved in the development of a certain product. Hence, exchange of product definition data is a basic requirement for the profitable use of computer–aided technologies in industrial product development.

While attempting to communicate data among various CAD systems, problems are encountered concerning the description of complex curves and surfaces in 3D space. The basic source of these problems is the wide variety of definition formats used for curve and surface data by different CAD/CAM systems. In some cases, theoretically derived mathematical algorithms are available for exact conversion between two different forms of representation. However, in far too many cases no exact conversion is possible [1] and the best that can be achieved is to produce some form of “good” approximation to the original data that is acceptable to the receiving system. These approximations try to preserve, to the extent possible, the geometric intent of the original design. Accordingly, there has been development of translators between different CAx (computer aided) systems to facilitate the exchange of data.

Three different widely used strategies for exchange of geometric data between CAD systems are: direct system-to-system translators, parametric evaluators, and intermediate neutral file format. Because of its relative advantage and simplicity the neutral file format approach continues to be the industrial preference. To meet the growing need of exchange of graphical information there have been several initiatives for developing a perfect neutral file format translator. The prominent neutral file format translators are: initial graphics exchange specification (IGES), DXF, standard d’echange et de transfer (SET), product definition data interface (PDDI), VDA / VDMA-FS, and standard for the exchange of product definitions (STEP).

Among these varieties, the Initial Graphic Exchange Specification (IGES) is a well-developed product that is widely used in industry. IGES translators generate both ASCII format and binary format neutral files. ASCII format scheme structure files are easy to study and analyze. In this study ASCII format IGES translation of CAD models have been used to extract the geometric data.

1.2 Efforts to Find Alternate Applications of CAD Data

To extend the use of CAD modeling in applications not intended for by the CAD software developers, there have been several research initiatives in different parts of the world.

In the late 1980’s, NIST (National Institute of Standards and Technology) developed a prototype of a cable-driven robot (RoboCrane) with graphic programming capability [2]. This enabled programming of controller from the graphical animation of a desired sequence of operations.

In 1988, Dave et al. [3] worked on an automatic generation of paths of a robot from CAD data for developing a water-jet cutting system. The work was developed on 2-D AutoCAD models using an Ingersoll Rand Waterjet cutting system and a five-axis gantry robot. The pertinent information contained in the AutoCAD file was extracted through its ASCII format neutral file translations. The project incorporated path generation from IGES translation and DXF translation of AutoCAD files. The work demonstrated the development of an alternate application of CAD data. However, the work was basically a case of two-dimensional application with some provisions for extension to some three-dimensional use also.

In 1992, Madurai and Lin [4] worked to develop a computer-aided process planning system linking computer aided design (CAD) with computer aided manufacturing (CAM). To extract the features required for manufacturing, IGES translations of CAD models were used. Although, the extracted features were not very accurate, they were sufficient to generate the process plans. The system was limited to use in rotational parts and could not recognize flat features.

In 1994, Liu et al. [5] worked on generating 3D parts from IGES translation of 2D drawings. The work was motivated to provide better understanding of part shape and to support CAM operations downstream.

To facilitate the application of CAD in computer-aided process planning, in 1996, Chu et al. [6] worked on automatic extraction of features of CAD models from their IGES translation. An IGES file of the raw material was taken as an input to the system. The system classified the relevant features in the IGES file. The entities in the IGES file were used to extract the geometric and topological information of the parts. Algorithms were developed to merge relevant features based on geometric and topological information. Process operations were determined on the basis of the extracted features.

In 1999, Mackulak et al. [7] worked on reduction in model creation cycle time in an automated material handling system (AMHS). They extracted the relevant data from selected entities and attributes in the IGES file. After extraction, the data were converted into a format (.asy type) to make it acceptable to automated material handling system (AMHS) simulation software.

All these earlier studies were conducted with some specific type applications not in line with that in the current study and appear to be not applicable directly for the purpose of extraction of geometry data. More over, through time, there have been fundamental changes both in CAD systems as well as in IGES standards. Expanding the capabilities of CAD systems from 2D to 3D has improved the aesthetics (shape, color, shading, etc.) but has been accompanied by complexities. Accordingly, the current study was initiated to extract the geometry data of CAD models and make it available for other applications.

2 Extraction of Geometry Data from CAD Models

Efforts have been made in this study to develop a simulation of application based on the geometry data of CAD models. Since most of the standard CAD software do not support extraction or use of native data in applications not intended for, it has been decided to use some standard translation of CAD files that gives access to the geometry data of the CAD models. In that regard ASCII format IGES translation of CAD files has been quite useful. The consistent structure of these files makes it suitable to be treated as a reliable data source. The characteristics of IGES files were useful in this study, and the development of a software tool to extract the necessary data from IGES files are explained as follows:

2.1 Useful Characteristics of IGES files

Initial graphics exchange specification (IGES) is concerned with the data required to describe and communicate the essential engineering characteristics of physical objects as manufactured products [8, 9, 10]. Such products are described in terms of their physical shape, dimensions, and information, which further describe or explain the product. The development of this specification was triggered by the requirements for a common data communication format for product definition that is understood by the CAD/CAM environment.

The fundamental unit of data in IGES file is the entity. Entities are categorized as geometric and non-geometric. Geometric entities represent the definition of the physical shape and include points, curves, surfaces, solids, and relations, which are collections of similarly structured entities. Non-geometric entities typically serve to enrich the model by providing a viewing perspective in which a planar drawing may be composed by providing annotation and dimensioning appropriate to the drawing.

A file may include any number of entities of any type as required to represent the product definition. Each entity occurrence consists of a directory entry and a parameter data entry. The directory entry provides an index and includes descriptive attributes about the data. The parameter data provides the specific entity definition. The directory data and parameter data for all entities in the file are organized into separate sections, with pointers providing bi-directional links between the directory entry and parameter data for each entity. The characteristics of IGES files that were utilized in this study are described in the following sub-sections.

2.1.2 IGES File Structure

As mentioned earlier, an IGES file can have three different forms - ASCII, Compressed ASCII and Binary. Compressed ASCII and Binary forms are the means to create shorter IGES files. They store the same data as the ASCII format but in a compressed format that is not human readable.

ASCII format IGES files are structured with 80 character records, representing a line, wherein the data is stored in free format [8, 9, 10]. The file is divided into sections. The section identification character occupies column 73 of each line. Columns 74 through 80 are specified for the section sequence number of each line.

Sections of a file

An IGES file consists of five or six sections: Flag (in the case of the binary or compressed ASCII form), Start, Global, Directory Entry, Parameter Data, and Terminate. Each of the five or six sequentially numbered sections in an IGES file has a role in describing the CAD data. The arrangement and intend of data in each sections are described as follows:

Flag (B or C): This is an optional section used only when remainder of the file is in compressed ASCII or binary form.

Start Section (S): The start section is to record the user's exclusive engineering information such as part number, description, etc. All records in this section have the letter "S" in column 73 and sequence number in columns 74 through 80.

Global Section (G): All records in the Global section contain the letter "G" in column 73 and a sequence number in columns 74 through 80. This section contains 26 pieces of information pertaining to the entire file.

Directory Entry Section (DE): The Directory Entry section has one Directory Entry record for each entity in the file. The Directory Entry record for each entity is fixed in size and contains 20 fields of eight characters each, in two consecutive 80-character lines. Values are right justified in each field.

Parameter Data Section (P): The Parameter Data section of the file contains the parameter data associated with each entity. Parameter data are placed in free format with:

- The first field always containing the entity type number. Therefore, the entity type number and a parameter delimiter (default is comma) precede parameter one of each entity.

- The free format part of a parameter line ends in column 64.

- Column 65 is a blank.

- Columns 66 through 72 on all parameter lines contain the sequence number of the first line in the directory entry of the entity for which parameter data is being presented.

- Column 73 of all lines in the parameter section shall contain the letter "P".

- Columns 74 through 80 shall contain the sequence number inside the parameter data section.

Terminate Section (T): There is only one line in the Terminate section of the file. It is divided into ten fields of eight columns each. The Terminate section is the last sequenced line of the file and identified by "T" in column 73. Fields in the Terminate section record the number of lines contained in each sections of the file.

2.2 CAD Model Data in IGES File

A CAD model is a solid object. A solid is composed of a set of bounded surfaces. A bounded surface is defined by a set of closed curves. A closed curve is composed of a set of curve segments. A curve segment is described by a set of points. A point in the three dimensional space is a set of triplet of real numbers representing its coordinates.

In the IGES file information about the CAD model is maintained in the form of a set of entities [9, 10] arranged sequentially. A unique number identifies each entity type. Each entity type has a specific purpose in maintaining the CAD model information in the IGES file. The values pertaining to an entity that appear in the parameter and directory entry sections uniquely specify its scope in defining the CAD model. The directory entry section of the file maintains an index of all the entities that appear in the file. Starting at line one (1) in the directory entry section, two consecutive lines pertain to an entity.

On sequential progress, starting at line one, in the directory entry section, the surfaces pertaining to a CAD model are sequentially represented by Bounded Surface Entities (Type 143). Data associated with a Bounded Surface Entity leads to the associated Boundary Entities (Type 141). A boundary of a surface is defined by a closed curve. Each Boundary Entity leads to a set of associated curve segments (like Entity Type 126). A set of associated curve segments together forms a closed curve. A curve entity leads to a set of associated points that together define the curve segments. Collection of all the surfaces pointed by the Bounded Surface Entities enables defining the complete CAD model.

The software tool developed in this study extracts the geometry data of 3D solid model from its IGES translation. Object oriented programming (OOP) approach has been used to develop the software tool. The programs of this tool have implemented the above-mentioned sequential association of IGES entities for extracting the geometry data points of CAD models from its IGES translations.

3 Extraction of CAD Data and Development of Application

In this study programs have been developed in Java using object oriented programming approach. The developed programs support three different operations: extraction of geometry data of CAD models from its IGES translations, manipulation of the geometry data of CAD models in visual modeling, and a database application for storage of the geometry data. Works on these three areas are described as follows:

3.1 Software Tool to Extract Geometry Data from an IGES File

The software tool developed in this study to extract the CAD geometry data from its IGES translation was built on the structure of IGES files. In the program IGES entities are described by a combination of two classes – one for the directory entry data (DirectoryData) and the other for the parameter data (PData or its sub classes). The basic unit of geometric information contained in the geometric entities, which is of interest in this study, is the co-ordinate information of a point - a group of three decimal numbers, and that have been defined in the class Point to capture this information. During the course of the program, objects corresponding to particular IGES entity types are created and used to extract, and accumulate the point information in collections to represent: Curve segment – collection of points, Closed curve – collection of curve segments, Surface – collection of closed curves, and Solid – collection of surfaces.

The types of entity objects to be created to capture the point information depends on the entity types used in the IGES file. The software tool evaluates the entity type at run time and creates the proper object at the run time. Types of entities used in the IGES files for a piece of geometric information in the CAD drawing may vary from one IGES file to another. However, if there exists the class with the interface defined in the package for the types of entities, then the program should be capable of extracting the associated geometric information.

The design of the program reflecting the extraction of geometry points have been shown in Figure 1 as a logical view of the object diagram. This design has been implemented by using Java programming language. The direction of arrow in the diagram shows the direction of seeking information from. The sequence of events to extract the points and to create the "solid" object, may be stated as follows:

- The operation starts with taking the name of the IGES file as the input.

- The IGES file is converted into a file data vector ("fileDataVector").

- A reference to this file data vector is passed to the class Solid and the object "solid" is created.

- Solid class makes an object "terminate" of the Terminate class with the "fileDataVector". Then "terminate" gives the information about the number of lines each section of the IGES file has. Solid class evaluates all the elements in the directory data section and looks for entity Type 143 (bounded surface entity). Finding an occurrence, directory data object of class DirectoryData is created as “dData1”, and parameter data object of class P143Type is created as “pData143”. The objects “dData1”, “terminate” and “fileDataVector” are passed to the class Surface to create an object “surface”.

- Surface class evaluates the number of closed curves the surface has. For a closed curve, a directory data object “dData2” and a P141Type object “pData141” are created. The objects “fileDataVector”, “terminate” and “pData141” are passed to the class ClosedCurve and an object “closedCurve” is created.

- The class ClosedCurve evaluates the number of curve segments the closed curve has. For a curve segment a directory data object “dData3” is created. The objects “fileDataVector”, “terminate”, “dData3” and the associated orientation flag are passed to the class P126Type to create an object “pData126”.

- The class P126Type extracts the relevant lines of parameter data, processes it, collects the points and accumulates the points in a Vector type collection “bsCurveV”. Vector is a class in the Java class library.

Completion of the above process accomplishes the creation of the object “solid”.

3.2 Visual Modeling

Techniques of orthographic projections, which are very common for visualizing geometric models, have been used in this study. The principles of computer graphics have been used to draw pictures of solid objects with realism [11].

A simplified approach using orthographic projection and wireframe diagrams have been adopted in this study to draw the object on the computer screen. The geometry data representing the object has been transformed based on the chosen viewing direction. It is then projected orthographically on a plane perpendicular to the viewing direction. This data have then been used to draw the wireframe diagram on the screen.

In this study a solid object has been described internally in terms of a set of three-dimensional coordinate points and the object’s corresponding wireframe diagram has been drawn on the computer terminal by joining consecutive points by lines that represent the edges of the solid. The software tool developed here enables generation of graphical presentation of solid objects and provides for some manipulation with the views. This enables synthesis of the geometry data on the basis of pictorial description of the model. The correspondence between the geometry data and the graphical presentation, as well as the transformation of the geometry data from one configuration of the object to another has been achieved by using the principles of geometric transformations.

3.2.1 Geometric Transformations

Rigid-body motion is a simple motion where in the relative distances between object particles remain constant. Accordingly, the object does not deform during this motion. Translation, scaling, reflection, rotation, and any combination of them are some of the common geometric transformations that can be considered on a solid model. These transformations have been applied directly to the parametric representations of objects such as points, curves, surfaces and solids. Matrix notations provide an easy way of developing and implementing geometric transformations into graphic packages [11, 12].

Transformations of points represent the basic operation of geometric transformation, as points are the basic elements of object representation. For example, a line is represented by its two end points, and a curve, surface, or solid is represented by a collection of points. Given a point **p** belonging to a geometric model that undergoes a rigid-body motion, the corresponding point **p*** for the new position of the model is found as $\mathrm{p}^* = \mathrm{f}(\mathrm{p},\ \textit{transformation parameters})$. Both **p*** and **p** are 3×1 vectors.

To make it convenient to implement, both **p*** and **p** are expressed in their 4×1 homogeneous forms $\widehat{\mathbf{p}}^*$ and $\widehat{\mathbf{p}}$, and they are related by the relation $\widehat{\mathbf{p}}^* = [\mathbf{T}]\ \widehat{\mathbf{p}}$, where [**T**] is the 4×4 homogeneous transformation matrix used to concatenate transformations involving translation. The homogeneous transformation matrix applies to all rigid-body transformations (translation, rotation, reflection, and scaling) and is applicable to both two– and three–dimensional graphics applications.

Repeated application of the homogeneous transformation to key points in a geometric model database or to a particular entity enables the transformation of the model or the entity. For example, to transform a straight line, its two end points are transformed and then connected to produce the transformed line. Similarly, to transform a curve, points on the curve are transformed, and then connected to give the transformed curve. The transformed entities have been displayed by drawing the line segments connecting the transformed key points.

In the development of the graphical applications of this study, homogeneous representation of points has been implemented in Java programs. Provisions of applying translation, rotation and scaling have been developed to apply on the solid objects constructed out of the extracted geometry data of a CAD model. A camera view of the solid is displayed (in terms of a wireframe diagram) on the screen based on the position of eye and the position of look as selected by the user of the software tool (developed here). On the selection for a specific modeling operation, appropriate functions are called inside the program and the desired action is performed resulting to drawing the object on the screen. All the points of a solid object are systematically subjected by the desired transformation matrix to achieve the overall transformation of the solid.

3.3 Database Application

After the extraction of geometry data of a CAD model from its IGES translation, the extracted data remain within the Java objects and will be lost (if not saved in a permanent storage) on closure of the Java program. Accordingly, a relational database management system (RDBMS) has been developed using DB2 software of IBM. This database maintains the geometry data of CAD models extracted from the IGES files. This creates a provision for permanent storage of the extracted geometry data from the Java objects. Data pertaining to many models can be stored simultaneously in this database. The database stores data of a solid model for a particular configuration in the visual modeling environment. The database enables subsequent recreation of the model as well as further visual manipulation of the model. This has been accomplished with the associated programs that implement the database queries and updates. These programs provide user-friendly graphical user interfaces (GUIs) utilizing forms, menus and tables.

In this study, a solid model has been abstracted in terms of a set of surfaces. The database has a provision of storage/retrieval of a surface normal per surface. Each of the surfaces also has one or more closed curves. Each of the closed curves is composed of a set of points and each of the points is a set of triplet of decimal numbers. A set of triplet of a point has the form as (x, y, z). The surface normal also has the same form as a point (i.e., a triplet of doubles x, y, z).

The database has been implemented by creating five relational tables:

Solid: Solid Name, Solid Number

Surface: Surface Number, Solid Number

Normal: Normal Number, X, Y, Z, Surface Number

Closed Curve: Closed Curve Number, Surface Number

Point: Point Number, X, Y, Z, Closed Curve Number

The database has been developed corresponding to the entities and their relationships represented by the entity relationship (ER) diagram in Figure 2. These relations enable linking among the tables to retrieve data using SQL statements.

To enable smooth use of all the programs a user-friendly interactive graphical user interface (GUI) has been developed in this study. Use of this GUI interface is being discussed in the next section.

4 Processing of Data in Application

To improve usefulness of the geometry data, arrangements have been made to process the data subsequent to its extraction from an IGES file. An interactive graphical user interface (GUI) has been developed in this study to make the user's interactions easy with the underlying programs. Integration of all the provisions developed in this study has been achieved through this GUI interface. In this interface, the user's intended use options are described in terms of graphics, menus, buttons, combo boxes, lists, etc. On appropriate selection of options followed by confirmation by clicking appropriate buttons, the underlying program performs the desired action. A view of the GUI displaying a hexagonal solid model is shown in Figure 3.

The program begins with the creation of an empty frame with four menu items: File, View, Movement, and Database.

Menu item "File" gives the option to open an IGES file or to close the program.

Menu item "Database" gives the options to interact with the database.

At the beginning only a solid model can be displayed either from an IGES file or from the database, or an existing model can be deleted from the database.

The GUI frame consists of three display panels, one text area and a control panel. Descriptions of these are as follows:

(a) "3D Model" display panel: The camera view of the solid model is displayed in this panel.

(b) “Surface View” panel: The orthogonal view of the selected surface is displayed in this panel.

(c) “Model for DataBase” panel: This panel displays the model at a configuration at which it is ready to be saved in the database.

(d) “Display Text” area: Coordinate points of the solid model and the surface normals are displayed in this text area.

(e) “Controls” panel: This panel contains a number of controls that the user can interact to manipulate the modeling of the solid object. The controls are:

Scale: To control the displayed size of the solid model,

Surface: To select the surface being displayed in the “Surface View” panel as well as in the “Display Text” area,

“Normal” text field: To display the surface normal values,

Radio buttons: To decide if only a surface data to be displayed or all the data of the model to be displayed in the “Display Text” area, and

ModifyNormal list: The surface normal value is not always available from the IGES file and sometimes it shows the opposite of the correct sense. Accordingly, arrangements have been made, under this dropdown list, to calculate a surface normal from three points on a surface. Sometimes, the mathematical calculation may not ensure correct sense of the normal. The 3D Model View and Surface View displayed side-by-side can enable the user selecting the correct sense of the normal. The surface view marks the first point on the curve as “0” and the second point as “1”. With the order of these two points, the user can ascertain if the calculated normal is pointing inside or outside the model. The direction of the normal can be reversed by clicking an appropriate option.

To calculate the surface normal for a surface, the program picks up first three points of the first closed curve of the surface. However, a check has been incorporated to see that the three consecutive points are not collinear. If the first three points become collinear then the program drops the third point and goes to the fourth point. This process continues until a set of three non-collinear points are found out.

Use of the menu item options are described as follows:

Open File: “Open File” option gives a file open dialog to open the appropriate .igs file. Opening a file is followed by extraction of the geometry data from the selected file and painting of the displays on the display panels.

Camera View Dialog: “Camera View Dialog” option enables creation of a camera view of the solid by giving the eye-position and look-position.

DBSolid Movement: “DBSolid Movement” option gives the choice to perform rotation, translation and scaling on the solid model.

Save Solid in DB: “Save Solid in DB” option enables saving the geometry data in the database under a given name.

Get Solid from DB: “Get Solid from DB” option enables retrieval of the geometry data from the database and painting the display panels with the retrieved data.

Delete Solid from DB: “Delete Solid from DB” option enables deletion of all data from the database for a selected CAD model.

Figure 3 is displaying the views of a hexagonal solid model in the GUI environment. The “3D Model” panel shows the camera view of the model. The “Surface View” is showing the eighth surface of the model, as it is the surface that has been selected in the “Controls” panel. The “Surface View” shows the marking of two points “0” and “1”, indicating the order of the points defining the surface. The view in the panel “Model for DataBase” shows the configuration for which the data is ready to be saved in the database – the model has given a translation to have its centroid coincide with the origin.

5 Discussion and Conclusion

The software tool developed in this study could extract geometry data of CAD models through its IGES translations. To extract and use the geometry data of CAD models, the work of this study may be listed as: analysis of IGES files to reach the geometry data of CAD models, formulation of extraction of geometry data from IGES files, building a data structure to resemble a solid model, development of software tool to extract geometry data of CAD models from its IGES translations, development of visual modeling operations, development of a database for storage and retrieval of geometry data of CAD models, and development of a user-friendly graphical user interface (GUI) integrating all operations of the software tool. The software tool thus developed here could find an alternate application of CAD modeling beyond the stipulated limit.

The software tool developed may be termed as a limited scope “post-processor”. It is limited in scope, as it could not extract data pertaining to all aspects of CAD modeling. This tool extracts the geometry data from IGES files. The performance of the tool has been tested on files generated by version 5.3 of IGES translator of Mechanical Desktop 3 and AutoCAD 14.

The extracted geometry data conforms perfectly to CAD model dimensions for straight edges of planar surfaces. Future work will include extraction of points on curved edges and development of applications of the extracted data. For a curve, the extracted points remain outside of the curve. In the IGES file the points are associated with weight factors, and these values can be used to draw spline curves.

The storage of the geometry data in the database for multiple CAD models allows independent existence of the data for its easy use in subsequent applications.

The developed software could be used with the optical tracking system to control the motion of robot for part inspection. The robot could carry an optical scanner to register the outer surface of a work piece and the digitized data collected by the probe could be compared with the expected values. The software could also be used for the modeling and simulation of robot-assisted manufacturing operations such as welding or painting. The software tool of this study gives both the coordinates information of vertices as well as the surface normal information of a CAD model, where the edges of a CAD model are represented by a series of points. The surface normal information can be used to decide the orientation of the tool. With the coordinate information of points and the surface normal information, it would be possible to decide about a series of locations and orientations, which can then be used to simulate a welding operation by a robot on a CAD model. With the development of some additional programs, it is possible to simulate movement of a CAD model along a trajectory path.

To conclude, this work is part of the study on the application of geometry data of CAD models in a robot assisted visual inspection system. The structure of IGES files has been analyzed and a mechanism has been developed to extract data from an IGES file. The storage of the extracted data in the database for multiple CAD models is expected to provide a program independent existence of the data for its easy use in any application. It is up to the system in the downstream to develop useful applications of the data in the IGES file. The extracted data and the data structure (an object of the class Solid), developed in this study, were successfully used in the visual modeling operations. On successful formulation of application, IGES translations of CAD models may be used as a reliable data source to find extensions of CAD modeling.

Acknowledgement

The authors would like to acknowledge the help of Ms. Susanne Thüringer in the improvement of the GUI.

References

[1] Goult, R. J., Sherar, P. A. "Improving the Performance of Neutral File Data Transfers", research reports ESPRIT, Project 322.CAD Interfaces (CAD*I), Vol. 6, Springer-Verlag, 1987.

[2] Dagalakis, N. "Remote Graphic Programming and Monitoring Tools of the NIST RoboCrane Controller", Deneb International Simulation Conference, Troy, 1997.

[3] Dave, R. N., Vij, S. and Padhalkar, U. "Generation of Robot Path Programs from CAD Database for a Water-Jet Cutting System", Proceedings of Robots 12 and Vision'88, Conference, Vol.1, Detroit, 1988, pp. 109-117.

[4] Madurai, S. S., and Lin, L., "Rule-Based Automatic Part Feature Extraction and Recognition from CAD Data", Computers and Industrial Engineering, Vol. 22, No. 1, 1992, pp. 49-62.

[5] Liu, C. H., Perng, D. B., and Chen, Z., "Automatic Form Feature Recognition and 3D Part Reconstruction from 2D CAD Data", Computers and Industrial Engineering, Vol. 26, No. 4, 1994, pp. 689-705.

[6] Chu, L. S., Miguel, G., and Gwo, C. J., "Development of an Automatic Part Feature Extraction and Classification System Taking CAD Data as Input", Computers in Industry, Vol. 29, No. 3, 1996, pp. 137-150.

[7] Mackulak, G. T., Zhao, W., and Paprotny, I., "Reducing Model Creation Cycle Time by Automated Conversion of a CAD AMHS Layout Design", Proceedings of 1999 Winter Simulation Conference, URL: http://www.cs.dartmouth.edu/~igorpapa/paprotny_5.pdf.

[8] Stokes, H. "Solid Modeling and the Initial Graphics Exchange Specification (IGES)" published in "Handbook of Solid Modeling", McGraw-Hill, 1995, Editor in chief: LaCourse D. E., 1995.

[9] IGES/PDES Organization, "Initial Graphics Exchange Specification, IGES 5.3", U.S. Product Data Association, 1996

[10] IGES/PDES Organization, "Initial Graphics Exchange Specification, (IGES), Version 4.0", Society of Automobile Engineers, Inc., 1988.

[11] Hill, F. S., JR. "Computer Graphics Using OpenGL", 2nd edition, Prentice Hall, 2001.

[12] Zeid, I. "CAD/CAM Theory and Practice", McGraw-Hill, Inc., 1991

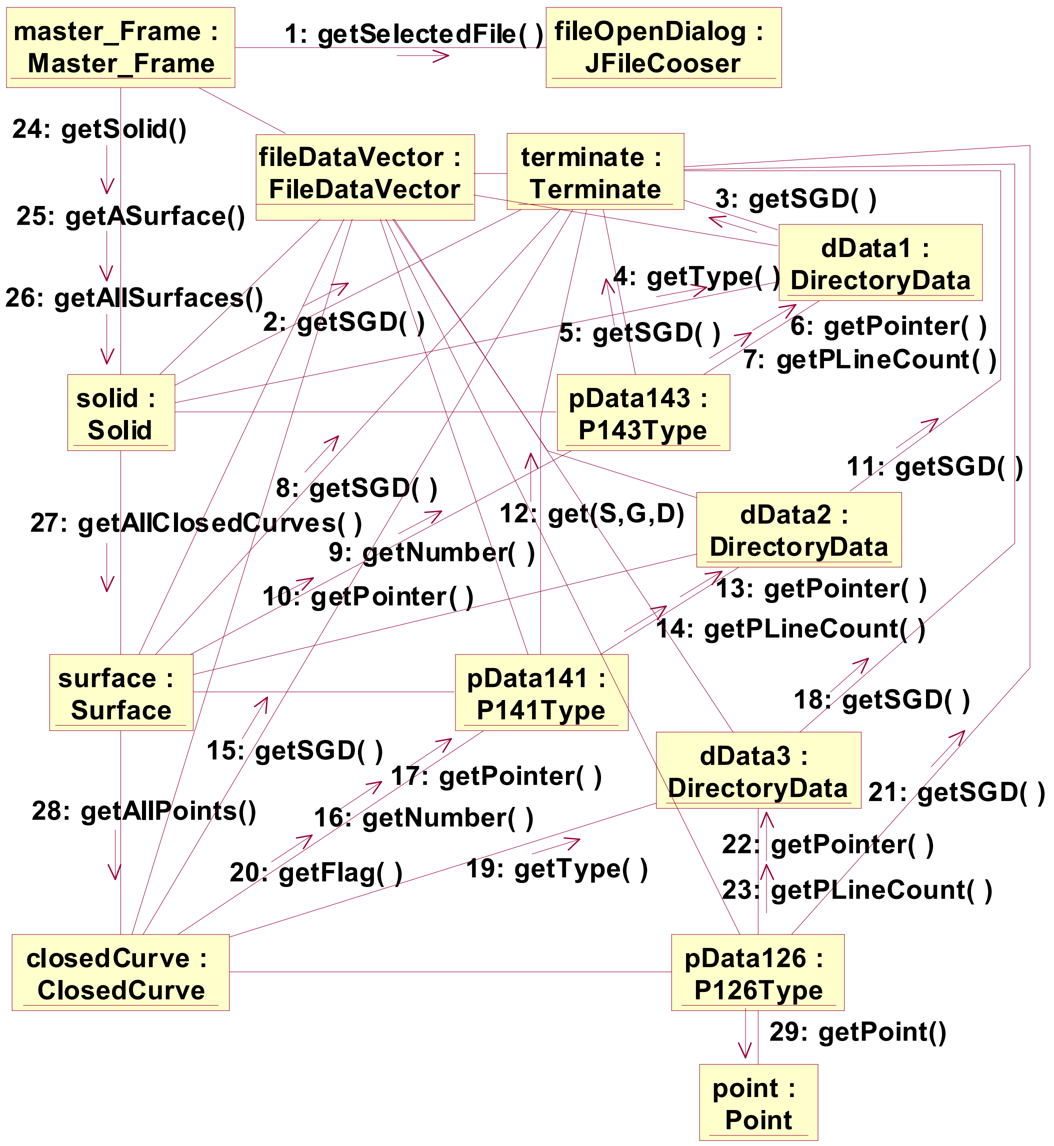


Figure 1: Logical view of object diagram.

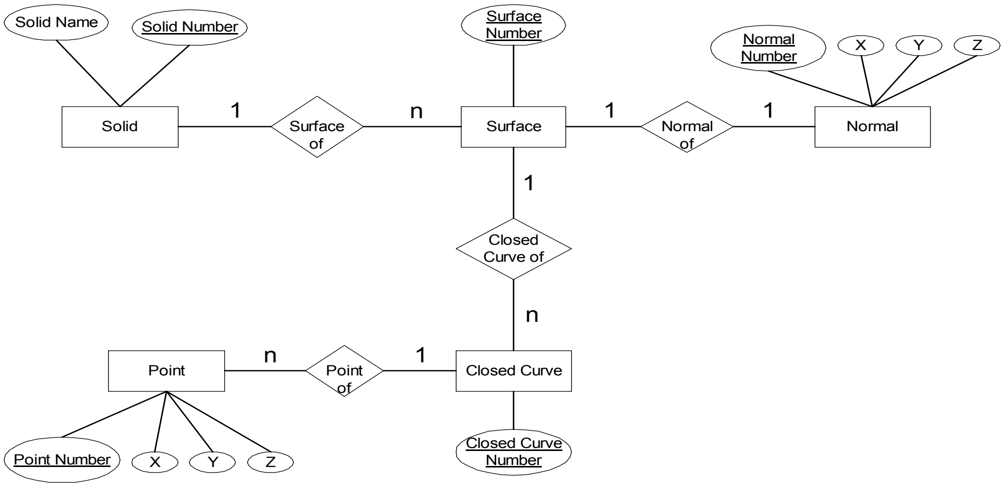
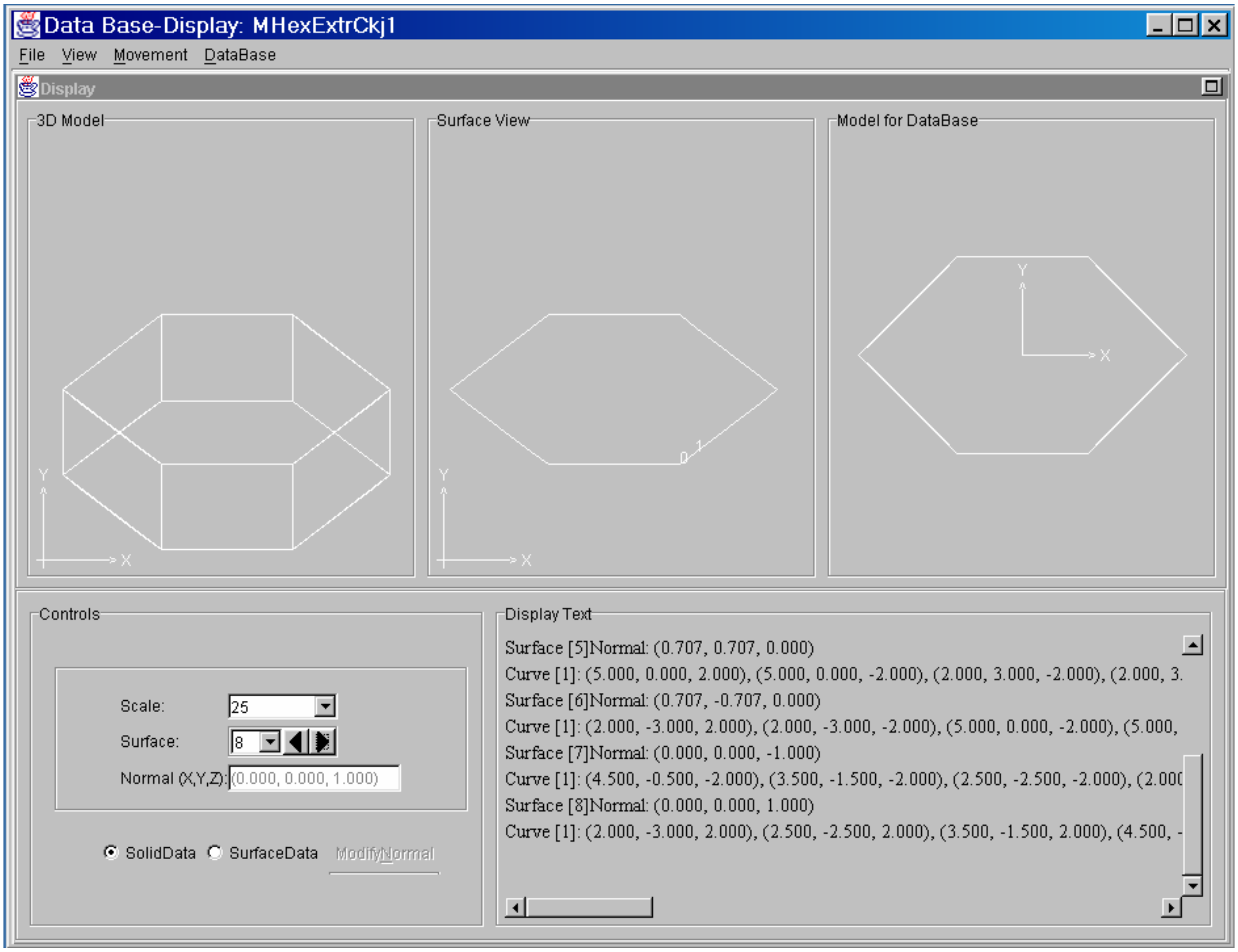


Figure 2: Entity-relationship schema diagram for the database.



Scale: 25

Surface: 8

Normal (X,Y,Z): (0.000, 0.000, 1.000)

● SolidData ○ SurfaceData

```
Surface [5]Normal: (0.707, 0.707, 0.000)
Curve [1]: (5.000, 0.000, 2.000), (5.000, 0.000, -2.000), (2.000, 3.000, -2.000), (2.000, 3.
Surface [6]Normal: (0.707, -0.707, 0.000)
Curve [1]: (2.000, -3.000, 2.000), (2.000, -3.000, -2.000), (5.000, 0.000, -2.000), (5.000,
Surface [7]Normal: (0.000, 0.000, -1.000)
Curve [1]: (4.500, -0.500, -2.000), (3.500, -1.500, -2.000), (2.500, -2.500, -2.000), (2.000
Surface [8]Normal: (0.000, 0.000, 1.000)
Curve [1]: (2.000, -3.000, 2.000), (2.500, -2.500, 2.000), (3.500, -1.500, 2.000), (4.500, -
```

Figure 3: A view of the GUI displaying a hexagonal solid model.