NASA-IGES TRANSLATOR AND VIEWER

Jin J. Chou and Michael A. Logan
Computer Sciences Corporation
NASA Ames Research Center
Moffett Field, California 94035

SUMMARY

NASA-IGES Translator (NIGEStranslator) is a batch program that translates a general IGES (Initial Graphics Exchange Specification) file to a NASA-IGES-Nurbs-Only (NINO) file. IGES is the most popular geometry exchange standard among Computer Aided Geometric Design (CAD) systems. NINO format is a subset of IGES, implementing the simple and yet the most popular NURBS (Non-Uniform Rational B-Splines) representation. NIGEStranslator converts a complex IGES file to the simpler NINO file to simplify the tasks of CFD grid generation for models in CAD format.

The NASA-IGES Viewer (NIGESview) is an Open-Inventor-based, highly interactive viewer/ editor for NINO files. Geometry in the IGES files can be viewed, copied, transformed, deleted, and inquired. Users can use NIGEStranslator to translate IGES files from CAD systems to NINO files. The geometry then can be examined with NIGESview. Extraneous geometries can be interactively removed, and the cleaned model can be written to an IGES file, ready to be used in grid generation.

INTRODUCTION

In April 1994 the NASA Geometry Data Exchange Specification for Computational Fluid Dynamics (NASA-IGES) was published (ref. 1). The intend of this specification is to facilitate geometry data exchange between CAD and Computational Fluid Dynamics (CFD) grid generation, and to provide high quality data to CFD processes. From the inception of the NASA-IGES standard, the Geometry Data Exchange Subcommittee (ref. 2) realize that it is very difficult to influent CAD system vendors to conform to a standard from a small user community. Hence, the best approach to handle the interface between CFD and CAD systems is to conform to the current CAD convention and yet to simplify the interface. Two steps were taken to achieve to this goal.

First, the IGES file format (ref. 3), the most popular data exchange format among CAD systems, was picked as the base of the CFD geometry standard. However, IGES contains hundreds of data exchange elements, called entities. It would be extremely onerous to CFD processes if they have to handle all the entities. Instead, the Subcommittee concentrated on entities that are essentials to CFD processes. Popular NURBS representation is chosen as the backbone of the standard; a few structural and annotational entities in IGES are also picked to facilitate, but not to complicate, the data exchange. A complete list of entities in the NASA-IGES standard can be found in reference 1. Since NASA-IGES is a subset of IGES, CAD systems have no problems in accepting files in NASA-IGES format. The NASA-IGES standard also specifies an even simpler subset called NASA-IGES-Nurbs-Only format which represents geometries solely with NURBS and trimmed NURBS. This single representation of geometry relieves CFD processes from complicated geometry issues, and yet, grants CFD processes the access of high quality geometry data.

Even though CAD systems can read in NASA-IGES files, they do produce files containing entities beyond NASA-IGES. To reduce the burden on grid generators, a second step was taken by the Subcommittee: a translator that converts a general IGES file to NINO format was developed. Along with the translator, a geometry viewer was also written to allow inspection and manipulation of NINO files. The translator and viewer are detailed in the following sections.

C-10

N

NASA-IGES TRANSLATOR

The translator reads in most geometry entities from an IGES file (Table I). It performs conversion on those entities that require conversion. The conversion map is in Table II.

Translator Behavior Control

An IGES entity (the child) can be "physically depedent" on another entity (the parent). The child entity can exist only when the parent entity exists. An example is that each constituent curve is physically dependent on a Composite Curve entity. The final model space location of a physically dependent entity depends on the transformation matrix of its parent (and grandparents, etc., if any). Normally, the translator discards any entity whose parent is not read in (not in Table I) or whose parent is discarded, even if the entity itself is in Table I. This behavior conforms to the IGES specification. However, an option is provided in the translator to override this behavior. When the option is used, the translator will convert a physically dependent entity without parent. The translator will also change such entities to be physically independent. However, the model space location of the entity in the output may not be correct, since the transformation matrix of its parent (grandparents) may not be available.

Normally the output of the translator contains the following entities: 124, 126, 128, 141, 142, 143, 212 (Form 0), 402 (Forms 1, 7, 14, 15), 406 (Form 15). A Subfigure Instance (Entity 408) is instantiated and converted to a group entity (Entity 402) containing a copy of all the geometries (converted) in the subfigure. This behavior eliminates instances from the output file and conforms to NINO format. The advantage of this format is that CFD processes which handle the NINO files do not have to deal with instances, which can be quite complex. The disadvantage of instantiating instances is that the output file may become quite large. The current translator contains an option that allows the user to suppress the instantiation of instances. When this option is used, the Subfigure Definition (Entity 308) and Subfigure Instance (Entity 408) are written to the output file.

Conversion Approximation

Most of the conversions listed in Table II are mathematically exact (ref. 4). When exact conversions do not exist, approximations are used. Tolerance control is provided to the user through a tolerance variable in the translator resource file. Three kinds of approximation may take place. The first kind is for offset curves and offset surfaces. In general offset curves and surfaces can not be represented in the forms of its progenitors (ref. 5). For example, the offsets of bspline curves/surfaces are generally not bspline curves/surfaces. To convert offset entities to bsplines, approximation to the offset is required.

The second kind of approximation is for ruled surfaces when the rail curves of the ruled surfaces can not be converted to bsplines parametrically identically (ref. 6). An example is that of a ruled surface constructed between a circular arc and a line. Since the parametrization of arcs in IGES is defined trigonometrically, it is not rational. Hence, it is not possible to represent the parametrization with bsplines. IGES also allows ruled surfaces be defined by arc length parametrizations of the rail curves. It is well known that nearly all the arc length parametrizations are not rational.

The third kind of approximation occurs on trimmed or bounded surfaces when the parametrization of the underlying surfaces cannot be converted to bsplines exactly. This may happen when the underlying surfaces are ruled surfaces, surfaces of revolution, tabulated cylinders, and offset surfaces. When the parametrization of the surfaces cannot be converted exactly to bsplines, the parameter space curves of the trimmed/bounded surfaces need to be regenerated. Approximations are used for this regeneration.

NASA-IGES VIEWER

Overview

NIGESview is an application that permits the viewing and manipulation of objects read from a NINO file. The program reads in the NASA-IGES-Nurbs-Only subset of IGES and converts the resulting geometry into SGI's Open Inventor format which is then displayed. The input file should conform to the IGES Version 5.2 Specification (ref. 3). It is intended to be used in conjunction with the NIGEStranslator (Figure 1). NIGESview2.0 was developed using Open Inventor-2.0 and compiled on an SGI Indigo^2/ Extreme running IRIX-5.2 with 64Mbytes of RAM and 132Mbytes of swap space.

Open Inventor is a C++ API that rests on top of OpenGL and implements a display-list oriented graphical structure of your scene. This is stored as a "scene graph" and contains all of the items relevant to the geometry as well as the geometry itself. Many of the tedious aspects of OpenGL are removed so that the developer may concentrate on the application rather than the idiosyncrasies and event-handling nature of a low-level graphics API. The Open Inventor libraries should be available on vendor platforms other than SGI some time in 1995.

One of the goals of NIGESview is to eliminate the need for an expensive CAD package just to verify the correctness of NINO files. The other is to provide very simple editing of files to delete entities or merge entities from multiple files. In addition, few CAD packages have the ability to view entities 143 and 141 (Bounded surfaces), which NIGESview can display and manipulate.

The graphical interface permits a variety of editing features to manipulate the geometry in NINO files. The modified geometry can then subsequently be written out to either Inventor or NINO files. For example, if an IGES file contains a wing-body aircraft configuration, but also contains engine struts and cowls, these can be selected and deleted. Then the remaining geometry may be written to another NINO file.

During parsing of NINO files, entities in the input file that are not part of the NINO specification are thrown away. Entities that are not used for geometry viewing (e.g., properties, groups, etc.) will not be displayed, but are kept in memory so that they may be written to a NINO file, if the user chooses so. Physically dependent entities are ignored until the parent of the entity is parsed, then it is processed. If the entity has no parent, then the entity is lost, which is acceptable since the file is actually invalid. Note that the NIGEStranslator would not write out such an invalid file.

As each entity is parsed, it is converted to a group of Open Inventor "nodes," which represent graphical/attribute objects that we want to place in our scene graph. NIGESview uses the Open Inventor concept of a "kit" to represent an IGES geometry. The kit contains Open Inventor equivalents of IGES concepts such as subscript names (for info only), line widths and fonts, color, transforms, and of course, control points and NURBS data. So a kit is an entire encapsulation of an IGES geometry and its directory section information (Figure 2). See references 1 and 3 for details of IGES geometry attributes.

Two problems were encountered when we used Inventor for NURBS display. First, Inventor (also OpenGL) is extremely slow when the NURBS surface(s) is large. Second, Inventor can not render bsplines of order greater than 8. We circumvented these problems partially by tessellating unbounded NURBS surfaces ourselves. When unbounded NURBS surfaces are encountered, NIGESview calls routines to evaluate and tessellate the surface into a quadrilateral mesh. The resolution of the evaluation is determined using heuristics at parsing time, but the user may change this interactively after the file is loaded - to increase or decrease resolution. Due to the excellent performance of our tessellation routines and the high rendering speed of quadrilaterals by Inventor, NIGESview can handle large unbounded NURBS surfaces reasonably well.

NIGESview supports two kinds of "resolution" or complexity of geometries. The one for unbounded surfaces (discussed above) is called "Unbounded Complexity". Everything else in the scene (curves, bounded surfaces, text) is driven by "General Complexity," which is controlled by one value. A dialog box is provided for users to change either or both complexities at any time. In addition, a popup menu in the drawing area provides even more resolution flexibility with drawing styles such as Bounding Box, Low-res, Points, and Wireframe.

Editing

After a file is loaded and converted to Open Inventor, there exists two copies of the data in memory: one for Inventor, the other for IGES. When users modify geometries with NIGESview editors, both the Inventor and IGES data are modified. For example, when a user deletes an entity, not only is the Inventor data deleted, but also the IGES data in the model; if the NINO file is written out, the entity will not be in the output. For cases where the user translates, rotates, or scales the geometry, the result of writing an IGES file is that an IGES Transformation Matrix entity (124) is created or added to the modified entity. All IGES protocols regarding order of matrix multiplication, and so forth, are obeyed (Figure 3).

Editing Physically dependent entities (bounded surfaces) requires more caution. There are many IGES entities associated with one bounded surface - the trimming curves (parameter and model space NURBS curves), the coordinates, and the underlying surface. When copying/pasting/deleting a surface, NIGESview provides an interface dialog to specify how to process physically dependent surfaces and their "relatives".

Also available to the user is the ability to inquire IGES information that comes from the input file. The Start Section, for example, can be edited like standard text, and it will be reformatted and written out to the output IGES file when a user requests it. Data such as the list of read entities and the global section can not be directly modified, but can be viewed in a separate dialog window.

Issues

One of the difficulties of any package is dealing with large files having many large entities (or worse - lots of trimmed surfaces). NIGESview is no exception, but if the file can be parsed by lower level libraries, then the user can improve the usability of NIGESview with the complexity options discussed earlier. NIGESview can be sensitive to large files because there are always two copies in memory at any time. IGES data is not structured in a way that allows OpenGL or Open Inventor to merely reference the data - they must copy the data.

There are features provided by Open Inventor which are virtually free of (engineering) cost. These features are included in NIGESview for high quality image rendering. A brief list includes the following: editing of 6 axial lights, surface material editing (color, transparency, lighting reflective properties), background color, depth cueing, and choice of box or line highlighting of picked objects. These types of attributes are NOT saved to NINO files (IGES provides no support for these).

NIGESview limitations include features such as transforming groups of entities at once, editing raw control points of NURBS curves and surfaces, automated evaluation/tessellation of bounded surfaces and curves, saving color modifications of entities, and UNDO operations while editing.

DISTRIBUTION

The NIGES software including NIGEStranslator, NIGESviewer, and a set of IGES test files is available from NASA Ames Research Center. The software runs on SGI IRIX 5.2 machines. All software distribution includes source code. In case recompilation is necessary, SGI C++3.1 is required; in addition, OpenInventor 2.0 development environment is required for NIGESViewer. In the near future the software will be forwarded to COSMIC for distribution.

ACKNOWLEDGEMENT

The authors would like to thank NASA for the support of this project. Special thanks are attributed to Matthew Blake, Dr. Bill Van Dalsem, Francis Enomoto, and Cathy Maksymiuk at NASA Ames Research Center.

REFERENCES

1. M. Blake, et al, NASA Geometry Data Exchange Specification for Computational Fluid Dynamics (NASA IGES), NASA RP 1338, April 1994.

2. M. Blake, et al, The NASA-IGES Geometry Data Exchange Standard, *NASA Workshop on Software Systems for Surface Modeling and Grid Generation*, NASA CP 3143, April 1992.

3. Initial Graphics Exchange Specification (IGES) Version 5.2, US PRO/IPO-100.

4. L. Piegl, W. Tiller, Curve and Surface Constructions Using Rational B-Splines, *Computer-Aided Design*, Vol 19, No 9, 1987, pp 485-498.

5. B. Pham, Offset Curves and Surfaces:A Brief Survey, *Computer Aided Design*, Vol 24, No 4, 1992, pp 269-279.

6. J. Chou, Higher Order Bezier Circles, *Computer-Aided Design*, Vol 27, No 3, 1995, pp 250-259.

Table I: Entities in the IGES input file read in by NIGEStranslator.

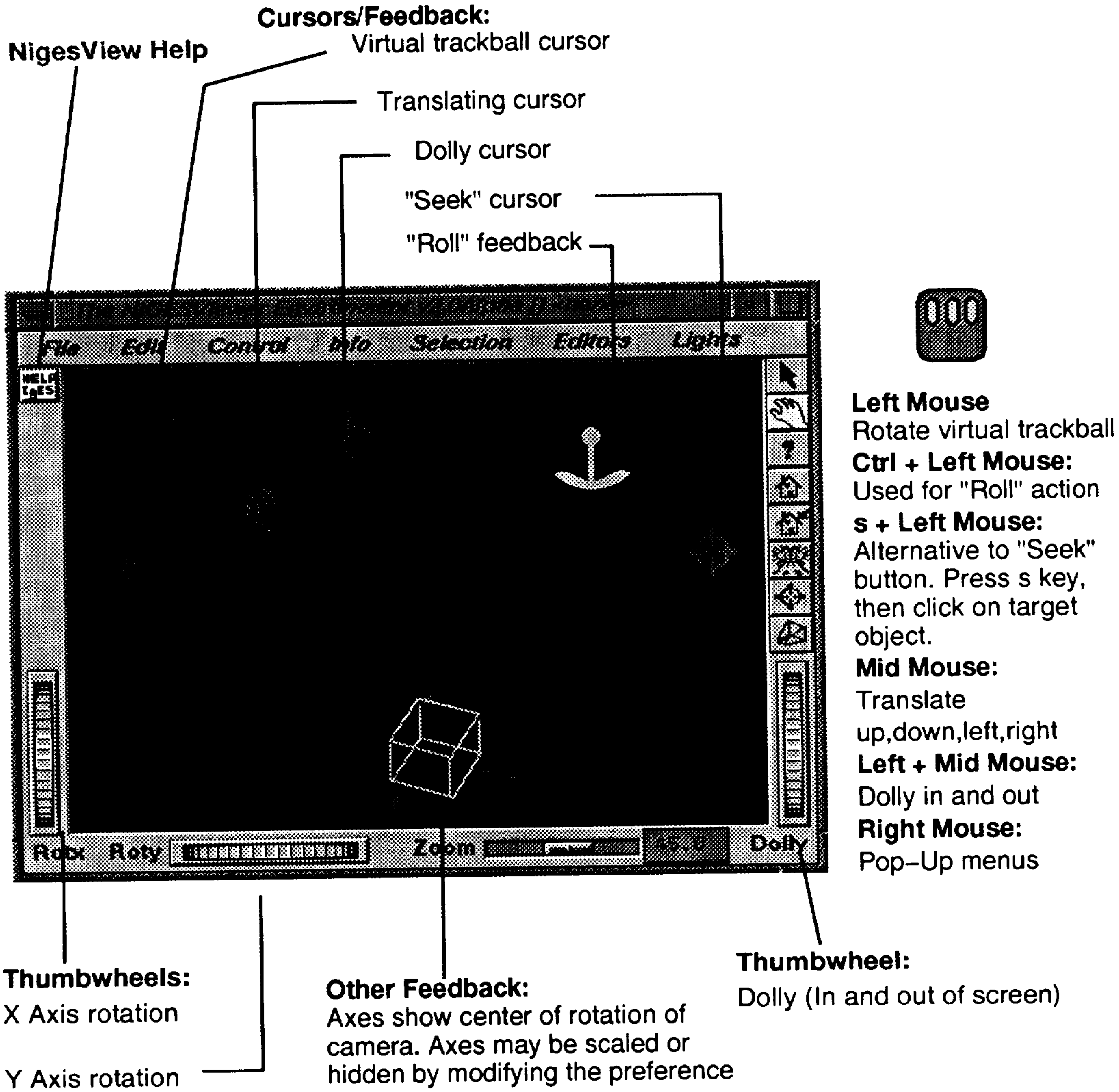
IGES Entity #	Entity Name
Entity 100	Circular Arc
Entity 102	Composite Curve
Entity 104	Conic Arc
Entity 106 (Forms 1,2,3,11,12,13,63)	Copious Data
Entity 108 (Form 1)	Bounded Plane
Entity 110	Line
Entity 112	Parametric Spline Curve
Entity 114	Parametric Spline Surface
Entity 116	Point
Entity 118	Ruled Surface
Entity 120	Surface of Revolution
Entity 124	Transformation Matrix
Entity 126	Rational B-Spline Curve
Entity 128	Rational B-Spline Surface
Entity 130	Offset Curve
Entity 140	Offset Surface
Entity 141	Boundary
Entity 142	Curve on Parametric Surface
Entity 143	Bounded Surface
Entity 144	Trimmed Parametric Surface
Entity 212 (Form 0)	General Note
Entity 314	Color Definition
Entity 402 (Forms 1, 7, 14, 15)	Associativity Instance,
Entity 406 (Form 1)	Property, Definition Levels
Entity 406 (Form 15)	Property, Name
Entity 308	Subfigure Definition
Entity 408	Singular Subfigure Instance

Table II: NIGEStranslator Conversion Map

From Entity Name	Type (,Form)	To Entity Name	Type (,Form)
Circular Arc	Type 100	Rational B-Spline Curve	Type 126, Form 2, Degree 2, PROP1 1, PROP3 0, PROP4 0
Composite Curve	Type 102	Rational B-Spline Curve	Type 126, Form 0
Conic Arc	Type 104	Rational B-Spline Curve	Type 126, Form 3, 4, or 5 as appropriate, Degree 2, PROP1 1, PROP3 1 for parabola, PROP3 0 for others, PROP4 0
Copious Data	Type 106, Form 1 or 11	Rational B-Spline Curve	Type 126, Form 0, Degree 1, PROP1 1, PROP3 1, PROP4 0
Copious Data	Type 106, Form 2, 3, 12 or 13	Rational B-Spline Curve	Type 126, Form 0, Degree 1, PROP3 1, PROP4 0; Note: The information about the vectors associated with the points will be lost.
Copious Data	Type 106, Form 63	Rational B-Spline Curve	Type 126, Form 0, Degree 1,PROP1 1, PROP2 1, PROP3 1, PROP4 0
Plane	Type 108, Form 1	Bounded Surface/ Boundary	Type 143, Form 0/ Type 141, Form 0
Line	Type 110	Rational B-Spline Curve	Type 126, Form 1, Degree 1,PROP1 1, PROP3 1, PROP4 0
Parametric Spline Curve	Type 112	Rational B-Spline Curve	Type 126, Form 0
Parametric Spline Surface	Type 114	Rational B-Spline Surface	Type 128, Form 0
Ruled Surface	Type 118	Rational B-Spline Surface	Type 128, Form 8
Surface of Revolution	Type 120	Rational B-Spline Surface	Type 128, Form 6

Table II: NIGEStranslator Conversion Map

From Entity Name	Type (,Form)	To Entity Name	Type (,Form)
Tabulated Cylinder	Type 122	Rational B-Spline Surface	Type 128, Form 7
Offset Curve	Type 130	Rational B-Spline Curve	Type 126, Form 0
Offset Surface	Type 140	Rational B-Spline Surface	Type 128, Form 0
Curve On Parametric Surface	Type 142 (Not part of 144)	Curve On Parametric Surface	Type 142, with all curves and surfaces converted to B-Splines
Curve On Parametric Surface	Type 142 (Part of 144)	Boundary	Type 141, with all curves and surfaces converted to B-Splines
Trimmed Surface	Type 144	Bounded Surface	Type 143, with all curves and surfaces converted to B-Splines
Definition Levels	Type 406, Form 1	The entity with this property is placed in the first level identified by this Definition Levels entity.	none
Singular Subfigure Instance	Type 408	A group (Associativity Instance Entity) of the geometry using original entities. These entities are then converted as specified in these Conversion Maps. This conversion takes place if the "-s" flag is not used	Type 406



Cursors/Feedback:
Virtual trackball cursor
Translating cursor
Dolly cursor
"Seek" cursor
"Roll" feedback

NigesView Help

Left Mouse
Rotate virtual trackball
Ctrl + Left Mouse:
Used for "Roll" action
s + Left Mouse:
Alternative to "Seek" button. Press s key, then click on target object.
Mid Mouse:
Translate up,down,left,right
Left + Mid Mouse:
Dolly in and out
Right Mouse:
Pop-Up menus

Thumbwheels:
X Axis rotation

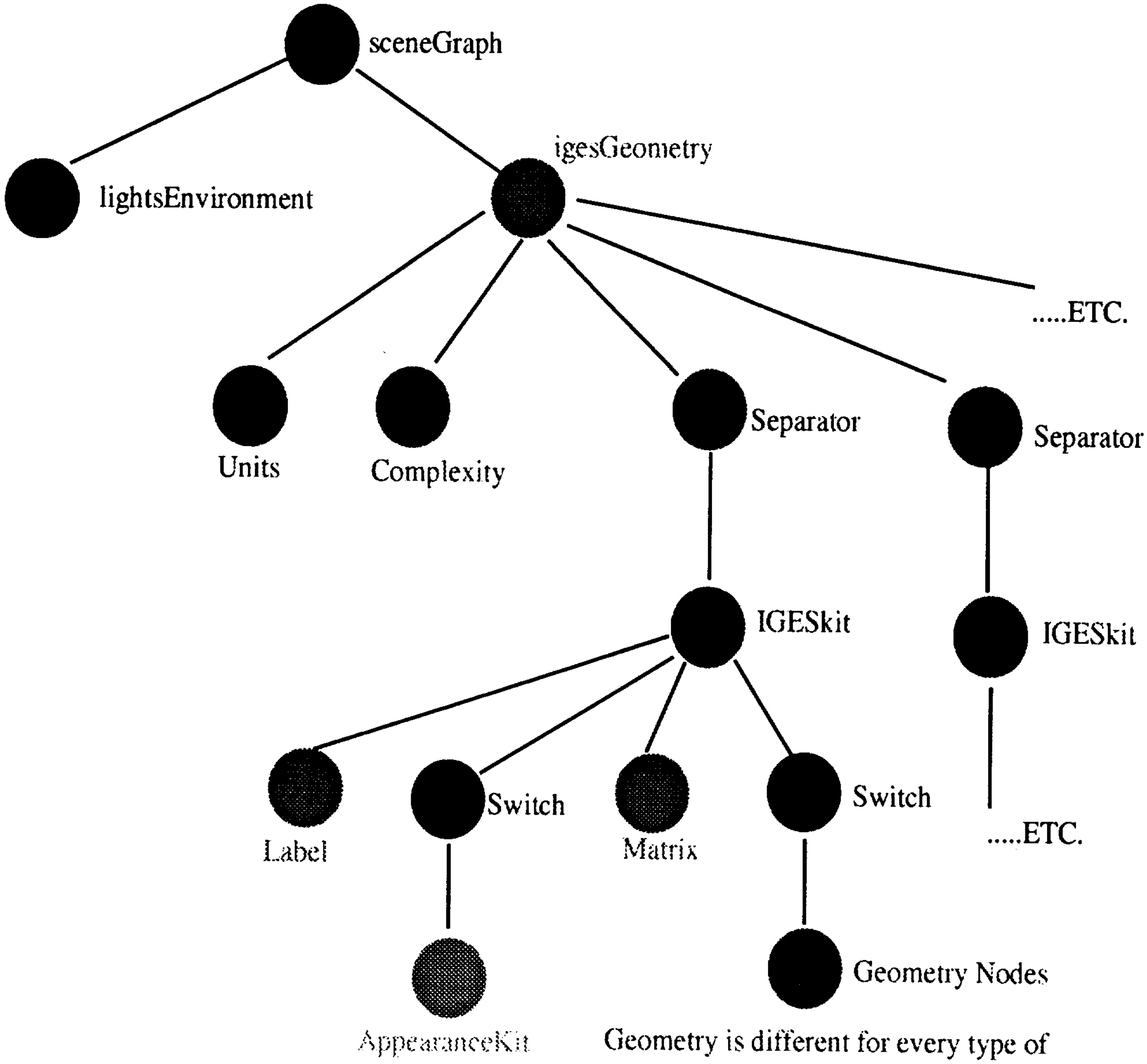
Y Axis rotation

Other Feedback:
Axes show center of rotation of camera. Axes may be scaled or hidden by modifying the preference

Thumbwheel:
Dolly (In and out of screen)

Figure 1.

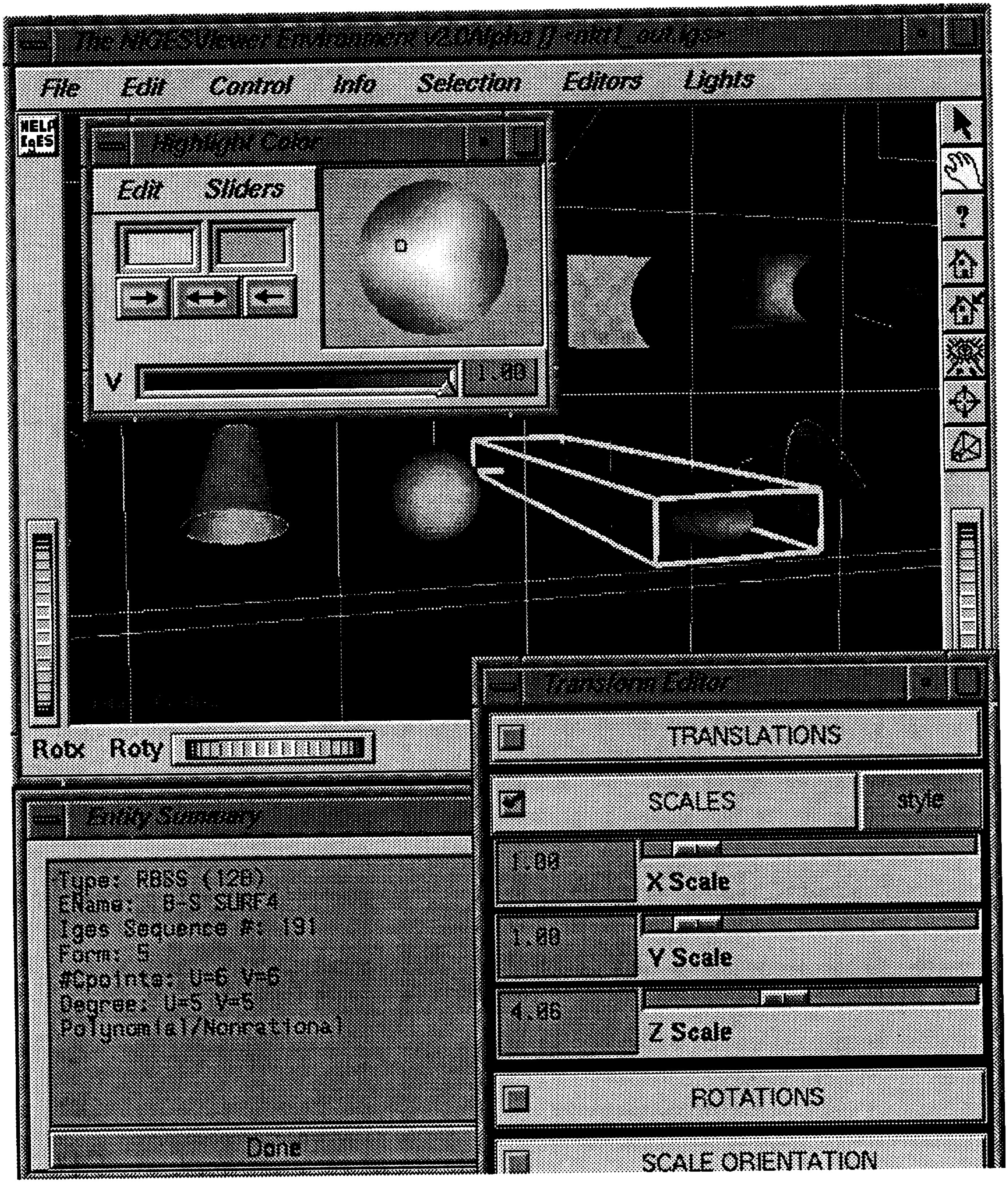
This viewer uses a virtual trackball to rotate the view. The point of rotation is by default the center of the scene bounding box, but can be placed anywhere in the scene. This viewer also allows you to translate in the scre͘en plane, as well as dolly in and out (forward/backward movement).



Geometry is different for every type of IGES entity -> IV kit conversion. Nurbs have Coordinates, maybe Profiles, Nurb Nodes. General Notes have 3D–Text nodes.

Figure 2:

Inventor uses a directed acyclic graph to represent data. To render a scene, Inventor traverses the graph depth–first, left to right. The graph is easy to manipulate and only needs created once, which for NIGESview, is when the IGES file is parsed.



```
Type: RBSS (128)
EName:  B-S SURF4
Iges Sequence #: 191
Form: 5
#Cpoints: U=6 V=6
Degree: U=5 V=5
Polynomial/Nonrational
```

☐ TRANSLATIONS

☑ SCALES

☐ ROTATIONS

☐ SCALE ORIENTATION

Figure 3: example of a surface being edited by NIGESview Inventor options. Note that the Scaling change **will** be saved to an IGES geometry file.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1995

3. REPORT TYPE AND DATES COVERED
Conference Publication

4. TITLE AND SUBTITLE
Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamic (CFD) Solutions

5. FUNDING NUMBERS
WU-505-62-52

6. AUTHOR(S)
Yung K. Choo, compiler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E-9458

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, D.C. 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CP-3291

11. SUPPLEMENTARY NOTES
Proceedings of the workshop on Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamic (CFD) Solutions held at NASA Lewis Research Center, May 9–11, 1995. Responsible person, Yung K. Choo, organization code 2610, (216) 433-5868.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 02

This publication is available from the NASA Center for Aerospace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The NASA Steering Committee for Surface Modeling and Grid Generation (SMAGG) sponsored a workshop on surface modeling, grid generation, and related issues in Computational Fluid Dynamics (CFD) solutions at Lewis Research Center, Cleveland, Ohio, May 9–11, 1995. The workshop provided a forum to identify industry needs, strengths and weaknesses of the five grid technologies (patched structured, overset structured, Cartesian, unstructured, and hybrid), and to exchange thoughts about where each technology will be in 2 to 5 years. The workshop also provided opportunities for engineers and scientists to present new methods, approaches, and applications in SMAGG for CFD. This Conference Publication (CP) consists of papers on industry overview, NASA overview, five grid technologies, new methods/approaches/applications, and software systems.

14. SUBJECT TERMS
Surface modeling; Grid generation; Computational fluid dynamics

15. NUMBER OF PAGES
865

16. PRICE CODE
A99

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT