Technical Report: isda08-002
Image Spatial Data Analysis Group
National Center for Supercomputing Applications
1205 W Clark, Urbana, IL 61801

An Overview of 3D Data Content, File Formats and Viewers

Kenton McHenry and Peter Bajcsy
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign, Urbana, IL
{mchenry,pbajcsy}@ncsa.uiuc.edu

October 31, 2008

Abstract

This report presents an overview of 3D data content, 3D file formats and 3D viewers. It attempts to enumerate the past and current file formats used for storing 3D data and several software packages for viewing 3D data. The report also provides more specific details on a subset of file formats, as well as several pointers to existing 3D data sets. This overview serves as a foundation for understanding the information loss introduced by 3D file format conversions with many of the software packages designed for viewing and converting 3D data files.

1 Introduction

3D data represents information in several applications, such as medicine, structural engineering, the automobile industry, and architecture, the military, cultural heritage, and so on [6]. There is a gamut of problems related to 3D data acquisition, representation, storage, retrieval, comparison and rendering due to the lack of standard definitions of 3D data content, data structures in memory and file formats on disk, as well as rendering implementations. We performed an overview of 3D data content, file formats and viewers in order to build a foundation for understanding the information loss introduced by 3D file format conversions with many of the software packages designed for viewing and converting 3D files.

2 3D Data Content

We overview the 3D data content first. The 3D content could be classified into three categories, such as geometry, appearance, and scene information. We describe each of these categories next.

2.1 *Geometry*

The geometry (or shape) of a model is often stored as a set of 3D points (or vertices). The surface of the model is then stored as a series of polygons (or faces) that are constructed by indexing these vertices. The number of vertices the face may index can vary, though triangular faces with three vertices are common. Some formats allow for edges (or lines) containing two vertices.

In graphics applications, these polygonal meshes are convenient and quick to render 3D content. However, sometimes the polygonal meshes are not sufficient. For example, when constructing the body of an airplane, in particular the round hull, a discrete polygonal mesh is not desirable. Though the model may look good at small resolutions, up close the flat faces and sharp corners will be apparent. One trick around this is to set normals at each vertex based on the average of the normals at the faces incident to it. When rendered these vertex normals can then be interpolated to create a smooth looking object (even though it is still a set of

discrete flat faces). Many formats allow for the storage of these vertex normals. Note that these vertex normals do not have to be the average of the incident faces. The normals associated with vertices can come from anywhere and can be used to apply a perception of shape on an object (a process known as bump mapping).

If truly smooth surfaces are required, at any scale, then a convenient option is the use of Non-Uniform Rational B-Spline patches (or NURBS). These parametric surfaces are made up of a relatively small number of weighted control points and a set of parameters known as knots. From knots, a surface can be computed mathematically by smoothly interpolating over the control points. Though slower to render, NURBS allow for much smoother models that are not affected by scale. In addition, these surfaces are just as convenient to manipulate as polygons are since one needs only to apply a transformation to the control points in order to transform the entire surface. Converting from a format that uses parametric patches such as NURBS to one that only supports polygonal meshes can potentially be an issue with regards to preservation. In such a situation the parametric surfaces will have to be tessellated (or triangulated). In other words, the continuous surface represented mathematically by the control points, weights and knots will have to be discretely sampled. Doing this involves a tradeoff as the more accurate the triangulation the more triangles are needed (i.e. more data and larger files). In order to save space on the other hand, the structure of the surface will have to be somewhat sacrificed.

Designing 3D models with only points and faces/patches is not very convenient. A more user friendly method is something along the lines of Constructive Solid Geometry (or CSG). Within the CSG paradigm 3D shapes are built from Boolean operations on simple shapes primitives (such as cubes, cylinders, spheres, etc...). As an example, a washer could be constructed by first creating a large flat cylinder. The hole in the middle of the washer can be inserted by creating a second flat cylinder with a smaller radius and subtracting its volume from the larger one. 3D file formats used within the CAD domain tend to store data that is amenable to this type of solid modeling. Clearly in order to edit your model after saving it you must keep track of each of the primitives and the operations/transformations on them. If you were to just save the resulting triangular mesh you would lose this information and prevent future editing of the model. This is another potential issue with regards to format conversion and preservation. On one hand converting from a format that supports constructive solid modeling to one that does not entails losing information with regards to its construction (which may or not be important in the long run). On the other hand, converting from a format that does not support constructive solid modeling to one that does may not be trivial.

2.2 *Appearance*

In its most common form materials entail applying an image (or texture) to the surface of the model. This is achieved by mapping each three dimensional vertex to a corresponding point within a two dimensional image. During rendering, the points within the faces making up the mesh are interpolated via the assigned texture coordinates at the vertices and the color within the associated image is used at this location. A model that does this must store these texture coordinates within the 3D data file. Most 3D file formats support texture mapping.

Another way to visually render materials is to physically model the properties of the surface, lights, and environment. This is usually done by assigning each face a number of properties. A surface can have a diffuse component indicating the color of the surface and how much light is reflected. There would be three dimensions for three color components corresponding to color spaces, such as RGB, HSV or YUV, just to mention a few. In addition, a surface can have a specular component indicating the color and intensity of true mirror reflections of the light source and other nearby surfaces. Surfaces can also be transparent or semi-transparent given a transmissive component indicating the color and intensity of light that passes through the surface. Transparent surfaces usually distort light passing through them. This distortion is represented by an index of refraction for the material. As we will talk about more in the next section, light sources have a

position and color assigned to them (usually being approximated as point light sources). However, a light source can be thought of as a surface/material that not only reflects light but also gives off light. This is represented by an emissive component of materials. Lastly, it is often convenient to state a minimal light source shared throughout the scene (i.e. an ambient component).

While combining textures with physically modeled materials can lead to very realistic looking renderings, physically modeled simulations cannot usually be performed in real time (in other words, a user cannot interact with the model in real time). Many file formats support storing material properties. However, an application that loads material properties usually ignores many of these properties when a user is manipulating the object. If the user wishes to see the true appearance of the scene then he/she must select an option to "render" the scene from a menu. After this menu selection, a reverse ray tracing process usually begins which will then simulate light rays bouncing off the surfaces within the scene.

2.3 *Scene*

By the term 'scene' we refer to the layout of a model with regards to the camera, the light source/s, and other 3D models (as a 3D object may be disconnected). To completely define the camera (or view) we must note the camera properties (4 parameters for magnification and principal point), the 3D position of the camera, a 3D vector indicating the direction it is pointing, and another 3D vector indicating the up direction. Assuming a point light source for now, each light source will need to be represented by a 3D vector indicating its position in space and a 3D vector representing its color/intensity. The layout of the 3D model itself can also be stored. This is particularly important if the model is made of up several parts and they must be laid out in a specific way to make up the scene. This requires two things. First, we must be able to designate the separate parts (or groups). Second, we need to store a transformation for each of the parts (i.e. a 3x4 transformation matrix).

A 3D format can often get away without supporting any of these attributes. For example, the parts making up the model can be transformed into their correct positions before saving the file. Thus, the model is always saved with the newly transformed vertices instead of the originals plus transformations. The camera and lighting can be completely ignored assuming that the user has the freedom to set these to whatever he/she desires. A user will likely change the camera position anyway as he/she navigates around the model/scene.

3 Formats

This section presents about 140 file formats. Table 1 reports file format extensions and the name of the format (which is usually an associated software application that opens/saves these types of files). Some of the file formats reported here can also be identified by using the Digital Record Object Identification (DROID) application built on top of the PRONOM technical repository [15-17].

Table 1: 3D file formats.

Extension	Name
3dm	Rhino
3dmf	Quickdraw 3D
3ds	3D Studio
3dxml	3D XML
ac	AC3D
ai	Adobe Illustrator

arc	I-DEAS
ase	ASCII Scene Export
asm	Pro/Engineer, Solid Edge, SolidWorks
atr	Lightscape Material
bdl	OneSpace Designer
blend	Blender
blk	Lightscape Blocks
br4	Bryce
bvh	Motion Capture
c4d	Cinema 4D
cab	TrueSpace
cadds	CADDS
catdrawing, catshape	CATIA V5
catpart, catproduct	CATIA V5
cgr	CATIA Drawing
chr	3Ds Max Characters
dae	AutoDesk Collada
ddf	Data Descriptive File
dem	Digital Elevation Models
df	LightScape Parameter
dlv	CATIA V4
drf	VIZ Reader
dwf	AutoDesk Composer Design Web Format
dwg	Legacy AutoCAD Drawing
dws	AutoCAD Standards
dwt	AutoCAD Drawing Template
dxf	AutoCAD Drawing Exchange Format
eim	Electric Image
eps	Encapsulated Postscript
exp	CATIA V4
fac	Electric Image
fbx	AutoDesk Kaydara FBX
fbl	CADfix Log File
fig	xfig
flt	Flight Studio OpenFlight
fmz	FormZ Project File
gmax	AutoDesk Game Creator
gts	GNU Triangulated Surface
hp, hgl, hpl, hpgl	HP-GL
hrc	SoftImage

htr	Motion Analysis HTR file
iam	AutoDesk Inventor
ifc	Industry Foundation Classes
ige, igs, iges	Initial 2D/3D Graphics Exchange Specification
ini	POV-Ray animation script
iob	3D Object TDDDB Format
ipt, iam	AutoDesk Inventor
iv	Open Inventor
jt	JT
k3d	K-3D Native
kmz	Google Earth Model
lay	LightScape Layers
lp	LightScape Presentation
ls	LightScape
lw	LightWave 3D
lwo	LightWave 3D 5.0 Object
lws	LightWave 3D Scene
lxo	Luxology Modo
m3g	JSR-184
ma	Maya Scene ASCII
max	3Ds Max
mb	Maya Scene binary
map	Quake 3
md2	Quake 2 Player Model
md3	Quake 3
mdd	Vertex Key Frame Animation
mel	Maya Embedded Language Script
mf1	I-DEAS
model	CATIA V4
mon	
mot	LightWave 3D Motion
mp	Maya Scene PLE
ms3d	MilkShape 3D
mtx	OpenFX Model
ndo	Nendo
neu	Pro/Engineer
obj	Wavefront
obp	Bryce
off	DEC Object file
p21	

par, psm, pwd	Solid Edge
pdb	PDB Reader v3
pd, pd	CADDS
pdf	Portable Document Format
pkg	I-DEAS, OneSpace Designer
plt	HP-GL
ply	Stanford PLY
pov	POV-Ray Rendering Instructions
pp2	Poser
prc, prd	PRC Adobe 3D Reviewer
prw	Adobe 3D Reviewer
prj	3Ds Studio Mesh
prt	I-DEAS, NX (Unigraphics), Pro/Engineer
ps	Post Script
pwc	Pulse
pz3	Poser
raw	Raw Faces
rib	Renderman
rif	Radiance
rvt, rte, rfa	Rent
rwx	Renderware
s3d	Strata 3D
sab, sat	ACIS
scn	TrueSpace
sda, sdp, sds, sdw, ses	OneSpace Designer
sdac, sdpc, sdsc, sdwc	OneSpace Designer
session	CATIA V4
shp	3D Studio Shape
skp	Google SketchUp
slc, sl, slo	Renderman
sldasm, sldlfp, sldprt	SolidWorks
slp	Pro Engineering
stl	Stereo Lithography
stp, step	Standard for the Exchange for Product Data
svg	Scalable Vector Graphics
trc	Motion Analysis TRC file
u3d	Universal 3D
unv	I-DEAS
vrml	Virtual Reality Modeling Language
vue	AutoDesk 3D Studio Animation

vw	LightScape View
w3d	Shockwave 3D Scene
wings	Wings 3D
wire	Alias Wire
wrl, wrz	VRML 1.0, VRML 2.0, VRML 97
x3d, x3dv	Extensible 3D (VRML 3.0, uses XML, 2004)
x	Direct X
x b, x t, xmt, xmt txt	Parasolid
xas, xpr	Pro/Engineer
xim	
xml	Land XML, VIS Material XML Import
xsi	Soft Image XSI
xv0, xv3	Lattice XVL

4 Viewers/Modelers/Animators/Renderers

This section presents the software found to view, import and export 3D files, as well as to build models, animate 3D models and render 3D content with variable settings. Each sub-section summarizes our findings about manufacturer, version, URL describing the software, cost, file formats it opens, saves, imports and exports.

4.1 *3ds Max*

Manufacturer	AutoDesk
Version	2009 Educational, 11.0
URL	http://www.autodesk.com
Cost	
Opens	chr, drf, max
Saves	chr, max
Imports	3ds, ai, dae, ddf, dem, dwg, dxf, fbx, flt, htr, ige, iges, igs, ipt, iam, lp, ls, obj, prj, shp, stl, trc, vw, wrl, wrz, xml
Exports	3ds, ai, ase, atr, blk, dae, df, dwf, dwg, dxf, fbx, flt, htr, igs, lay, lp, m3g, obj, stl, vw, w3d, wrl

4.2 *Adobe 3D Reviewer*

Manufacturer	Adobe
Version	9, Trial
URL	http://www.adobe.com

Cost	
Opens	3ds, 3dxml, arc, asm, bdl, catdrawing, catpart, catproduct, catshape, cgr, dae, dlv, exp, hgl, hp, hpgl, hpl, iam, ifc, igs, iges, ipt, jt, kmz, mf1, model, neu, obj, pd, par, pdf, pkg, plt, prc, prt, prw, psm, pwd, sab, sat, sda, sdac, sdp, sdpc, sds, sdsc, sdw, sdwc, ses, session, sldasm, sldlfp, sldprt, stl, step, stp, u3d, unv, wrl, vrml, x b, x t, xas, xpr, xmt, xmt txt, xv0, xv3
Saves	prw
Imports	3ds, 3dxml, arc, asm, bdl, catdrawing, catpart, catproduct, catshape, cgr, dae, dlv, exp, hgl, hp, hpgl, hpl, iam, ifc, igs, iges, ipt, jt, kmz, mf1, model, neu, obj, pd, par, pdf, pkg, plt, prc, prt, prw, psm, pwd, sab, sat, sda, sdac, sdp, sdpc, sds, sdsc, sdw, sdwc, ses, session, sldasm, sldlfp, sldprt, stl, step, stp, u3d, unv, wrl, vrml, x b, x t, xas, xpr, xmt, xmt txt, xv0, xv3
Exports	igs, pdf, stl, stp, u3d, wrl, x t

4.3 *Acrobat Pro Extended (3D Portion Only!)*

Manufacturer	Adobe
Version	9, Trial
URL	http://www.adobe.com
Cost	
Opens	pdf
Saves	pdf, prw
Imports	3ds, 3dxml, arc, asm, bdl, cadds, catpart, catproduct, cgr, dae, dlv, exp, ifc, iges, igs, ipt, iam, jt, mfl, model, neu, obj, par, pd, pd, pkg, prc, prd, prt, psm, pwd, sab, sat, sda, sdac, sdp, sdpc, sds, sdsc, sdw, sdwc, ses, session, sldasm, sldprt, step, stl, stp, u3d, unv, vrml, wrl, x b, x t, xax, xpr, xv0, xv3
Exports	igs, iges, stl, stp, step, vrml, wrl, x t

4.4 *AutoCAD*

Manufacturer	AutoDesk
Version	2009 Educational, C.56.0 (UNICODE)
URL	http://www.autodesk.com
Cost	
Opens	dwg, dws, dxf, dwt
Saves	dwg, dws, dxf, dwt
Imports	
Exports	

4.5 *Blender*

Manufacturer	BlenderFoundation
Version	2.46
URL	http://www.blender.org
Cost	Free
Opens	blend
Saves	blend
Imports	3ds, ac, ai, bvh, dxf, dae, eps, flt, off, lwo, md2, ms3d, obj, ply, ps, raw, slp, svg, x
Exports	3ds, ac, dae, fbx, fig, flt, iv, lwo, m3g, map, md2, mdd, mot, obj, off, ply, raw, wrl, x, x3d, xsi

4.6 *CINEMA 4D*

Manufacturer	Maxon
Version	R11.004, Demo
URL	http://www.maxon.net
Cost	
Opens	3dm, 3dmf, 3ds, ai, c4d, dem, dxf, iob, lwo, lws, mon, obj, ps, wrl
Saves	
Imports	
Exports	

4.7 *Flux Player*

Manufacturer	
Version	2.1
URL	http://mediamachines.wordpress.com
Cost	Free
Opens	wrl, x3d, x3dv
Saves	
Imports	
Exports	

4.8 *IDA-STEP*

Manufacturer	LKSoft

Version	Basic, 4.0
URL	http://www.ida-step.net
Cost	
Opens	p21, step, stp, xim
Saves	xim
Imports	
Exports	

4.9 *K-3D*

Manufacturer	Tim Shead
Version	0.6.7.0
URL	http://www.k-3d.org
Cost	Free
Opens	k3d
Saves	k3d
Imports	gts, k3d, obj, off, mtx, raw, rib
Exports	gts, k3d, raw

4.10 *Lightwave 3D Layout*

Manufacturer	Newtek
Version	9, Temporary Licensce
URL	http://www.newtek.com
Cost	
Opens	lws
Saves	lws
Imports	
Exports	wrl, w3d, lws

4.11 *Lightwave 3D Modeler*

Manufacturer	Newtek
Version	9, Temporary Licencse
URL	http://www.newtek.com
Cost	
Opens	lwo
Saves	lwo

Imports	eps, pdb
Exports	3ds, dxf, eps, lwo, obj

4.12 *Maya*

Manufacturer	AutoDesk
Version	8.5, Personal Learning Edition
URL	http://www.autodesk.com
Cost	
Opens	ai, dae, dxf, eps, fbx, fbl, ma, mb, mel, mp
Saves	mp
Imports	ai, dae, dxf, eps, fbl, fbx, ma, mb, mel, mp
Exports	fbl, mp

4.13 *POV-Ray*

Manufacturer	Persistance of Vision
Version	3.6
URL	http://www.povray.org
Cost	Free
Opens	pov, inc, ini, txt
Saves	pov, inc, ini, txt
Imports	
Exports	

4.14 *SwirlX3D Viewer*

Manufacturer	Pinecoast Software
Version	1.2.10
URL	http://www.pinecoast.com
Cost	Free
Opens	wrl, x3d, x3dv
Saves	
Imports	
Exports	

4.15 *Wings 3D*

Manufacturer	Izware
Version	0.99.02
URL	http://www.wings3d.com
Cost	Free
Opens	wings
Saves	wings
Imports	3ds, ai, fbx, lwo, lxo, ndo, obj, stl
Exports	3ds, fbx, eps, lwo, lxo, ndo, obj, pov, rwx, stl, wrl, x

4.16 *Xj3D*

Manufacturer	Web3D Consortium
Version	1.0
URL	http://www.xj3d.org
Cost	Free
Opens	wrl, x3d, x3dv
Saves	
Imports	
Exports	

5 Format Details

In this section we take a closer look at a few of the listed formats.

5.1 *3ds*

The 3ds file format is the primary format of AutoDesk's 3ds Max software. It is a binary format consisting of chunks that hold various pieces of information. Chunks contain an identification indicating what information is stored there and the offset to the next chunk [3]. In this way software that doesn't support certain rendering properties can simply ignore them.

The 3ds file format supports: geometry in the form of vertices/faces and parametric surfaces, textures, physical material properties, transformations, camera information, and lights.

5.2 *obj*

The obj file format is a text based, open file format developed by Wavefront Technologies (now Alias|Wavefront) [3]. The format has been adopted by other 3D graphics applications vendors and can be imported/ exported by a number of them.

The obj file format consists of a number of lines each containing a key and various values. The key on each line indicates the type information to follow. Because of this the obj file format doesn't require a header. Below is a list of some of the keys that can be used:

Table 2: OBJ file keys

Key	Description
#	Comment
v	Vertex
l	Line
f	Face
vt	Texture Coordinate
vn	Normal
g	Group
. . .	. . .

.

A simple example containing one face would look like:

```
v 0.0 0.0 0.0
v 0.0 1.0 0.0
v 1.0 0.0 0.0
f 1 2 3
```

The obj file format supports: geometry in form of vertices/edges/faces and parametric surfaces, vertex normals, textures, material properties, and groups.

5.3 *igs*

The Initial Graphics Exchange Specification (or IGES) format, published by the National Bureau of Standards in 1980 (NBSIR 80-1978), is a popular neutral format for digital the exchange of CAD information. The iges format is designed to store both 2D and 3D data.

The file itself is composed of 80-character ASCII records (which is derived from the punch card era). Text strings are represented in the Hollerith format with the number of characters in the string coming first, followed by an ”H”, then the string. There are five sections to an iges file. A ‘start’ section which begins the file. A ‘global’ section which is a type of header containing information about the file such as data type resolution, units, scale, and author information. A corresponding ‘terminate’ section ends the file indicating the number of records in each of the other four sections. Each geometric item that the file holds is split among the last two sections. One half is stored in what is called ‘the data entry’ section. It holds attributes associated with the particular item, for example, the color and style. The other half is stored in what is called ‘the parameter data section. It holds the actual values for the geometric item. For example, if the item is a point, then it will hold the x, y and z coordinates. Both the data entry section and parameter data section also identify themselves as being a particular type. For instance, points are identified by the number 116, lines by the number 110, and arcs by 100.

The iges format supports: vertices, lines, polylines, arcs, curves, parametric surfaces, CSG and B-Rep Objects.

5.4 *ply*

The polygon file format (or stanford triangle format), was designed for the purpose of being both a flexible and portable 3D file format [3]. The ply format has both an ASCII and a binary version. The binary version

includes information to make it machine independent, specifying the types used for each value, number of bytes per type, and whether it's big or little endian. In addition the format allows for user defined types allowing it to be extensible to the needs of future 3D data. Because of its simplicity and flexibility the ply format is very popular in the academic and research world.

A ply file begins with text header similar to the following:

```
ply format ascii 1.0
comment author: John Smith
element vertex 3
property float x
property float y
property float z
element face 1
property list uchar int vertex_index
end_header
```

After the first line identifying the file as a ply the header contains a number of keys, types and values specifying the layout of the file. The "format" keyword identifies this file as text based or binary. Following this are a number of "element" keywords, in this example defining the layout of vertices first, then faces. Specifically the line "element vertex 3" states that the file contains 3 vertices. Following an "element" keyword are a number of "property" keywords. These properties define what exactly a vertex is. In this case it indicates that a vertex has 3 parts, an x, y, and z component, and each of them is represented by a floating point value. Similarly, we define a face. In the above example a face is defined to be of a variable number of vertices. The number of vertices is defined as the first element of a list, or a line in our text file, by an 8 bit uchar. Next, we have the vertex indices represented by 32 bit integers. The end of the header is identified by the keyword "end header". The actual data follows the header laid out in the format specified. For example a body going allowing with the above header might look like:

```
0.0 0.0 0.0
0.0 1.0 0.0
1.0 0.0 0.0
3 0 1 2
```

The ply file format supports: geometry in the form of vertices/edges/faces, vertex colors, textures and materials. However, due to user defined types the ply format could potentially hold much more. Nevertheless, since user defined types are not standardized, a program reading the file may not know what to do with these properties.

5.5 *stp*

The Standard for the Exchange for Product Data, ISO 10303, was developed as a successor to the iges format. The step format is a plain text format that deals with named objects rather than just raw geometric information [3]. For example a point can be set as:

```
#1000 = CARTESIAN_POINT('', (0.0, 0.0, 1.0));
```

From now on this point will be identified by its corresponding number, #1000. For example if we were to now add a circle we would only reference that point and provide the radius:

```
#2000 = CIRCLE( '', #1000, 1.5);
```

Any future reference to this circle would use its number, #2000.

Like the iges file format the step format relies on solid modeling, which is convenient for CAD developers.

5.6 *u3d*

The Universal 3D format [4] was developed by the 3D Industry Forum which consisted of companies such as Intel, Boeing, Adobe and HP. Their goal was a universal standard for 3D data of all kinds that would facilitate exchange with a focus on promoting 3D graphics development in manufacturing, construction and various other industries. The format was approved in 2005 by the European Computer Manufacturer Association (Ecma-363).

The format is largely backed by Intel who began work on it after leave the Web3D Consortium after disagreements over x3d [5]. Like x3d this format is intended to be the 3D standard. It has since been adopted by Adobe to embed 3D graphics within PDF documents.

5.7 *wrl*

The wrl format has two versions, VRML 1.0 and VRML97 [1, 2]. Both versions are open, standardized formats by the Web3D Consortium, and were designed to integrate 3D virtual worlds into the World Wide Web (thus portability and performance were important considerations). VRML files are text based and laid out in a structured human readable manner so as to promote authorship of 3D data. VRML files are widely used to transport 3D models between graphics applications.

The wrl format supports: geometry in the form of vertices/faces and parametric surfaces, textursa, cameras, lights, and transformations.

5.8 *x3d*

The Extensible 3D Format [5] is an extension to VRML that was approved by the International Organization for Standardization in 2005 (ISO/IEC 19776). Like VRML the data files are text based to promote human authorship. In this way it is more than just file a format but a language/environment for construct 3D scenes.

6 Data Sets

This section presents a few collections of 3D data sets that are available for downloading over the Internet. Each sub-section contains URL of the data set and the file formats available at the URL.

6.1 *MINOS Experiment*

URL	http://www-numi.fnal.gov/minwork/scint
Formats	dxf, igs, plt, prt
Files	161

6.2 *NASA Wind Tunnel*

URL	http://www.windtunnels.arc.nasa.gov/{11ft1,9x7ft1}.html, http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3
Formats	catpart, dwg, dxf, igs, model
Files	29

6.3 *NIST*

URL	http://cic.nist.gov/vrml
Formats	ifc, kmz, skp, stp, wrl, x3d
Files	340

6.4 *NOvA Experiment*

URL	http://nova-docdb.fnal.gov/cgi-bin/ShowDocument?docid={2380,2517,2699,2874}
Formats	asm, dxf, igs, prt, stp, x t
Files	45

7 Format Conversions

7.1 Errors Due to Format Conversions

According to [12] (section '3.1.2 Data Exchange Problems'), there are several common problems with data exchanges related to format conversions. The errors during format conversions would be due to software/hardware incompatibility and due to product data quality. Some examples include:

- missing, collapsed, or inverted faces;
- models that do not form closed solids (surfaces and edges do not connect);
- models with incorrect feature orientation;
- lines that do not meet at corners;
- lines that cross at corners;
- curves or lines drawn as many short line segments;
- multiple occurrences of the same feature at the same location;
- lines or surfaces coincident with other lines or surfaces;
- surfaces that do not meet at lines;
- some or all of the geometry not translated;
- geometry, dimensions, and notes not correctly separated into different layers;
- planar features drawn out of plane;
- geometry of features not drawn to scale.

The study in [12] concluded that some of these problems could be overcome by developing a neutral format, such as the STEP format. The STEP format has been developed into the ISO 10303standard and the specifications of format modules can be found in [10], [11], [13] and [14].

7.2 Consideration about Format Conversion Evaluation

Preservation of 3D data involves basic understanding of 3D data characteristics, 3D file formats and viewing software. One of our objectives is to understand the information loss introduced by 3D file format conversions with many of the software packages designed for viewing and converting 3D data files. In order to quantify the information loss, a possible approach is to rank the characteristics of 3D data sets and design metrics for scoring 3D file conversions. This approach would depend on the application defining why 3D models would be preserved. For example, if the 3D model contained a 3D simulation of a crime, then the scene information would be ranked higher than the appearance and geometry of the individual objects. On the other side, if the 3D model was being preserved for the future users of the model in order to replace a part of the object being modeled, then the ranking of 3D data characteristics would follow the order of geometry, appearance and scene. It is also conceivable to build 3D models of wild fire where the appearance of flames would have higher preservation priority than the geometry and scene since the appearance conveys the information about what burned.

We provide an illustration of the ranking approach driven by the part replacement application in Table 3. It is many time the case that conversion utilities would ignore material appearance characteristics when converting between 3D file formats assuming the order of priorities to be geometry, appearance and scene. After ranking categories of characteristics, one could rank individual characteristics and map the presence or absence of characteristics before and after 3D file format conversions. During this process, one has to be aware of not only the information loss but also the feasibility of conversions. In the case where the conversion is possible one must usually consider a tradeoff between storage requirements and information loss.

Many 3D formats also contain information that allows for convenient editing of 3D models. This was discussed in the context of Constructive Solid Geometry (CSG) frequently used for building CAD models from a set of primitives. The loss of the constructive sequence and/or the primitives during file format conversions impacts further editing capabilities. While not relevant to preserving the appearance of the model, this loss can hinder those who may wish to edit the model in the future.

Table 3: Parameters supported by a number of popular 3D file formats and their classification according to the part replacement application criteria for preservation. Geometry (red) would be of high importance; Materials/appearance (green) would be of medium importance; Scene/environment (blue) would be of low importance. The following abbreviations are used in the table for the characteristics of 3D formats: C = Color, M = Material, I = Image texture, B = Bump map, L = Light sources, V = Viewport/Cameras, T = Transformations, G = groups, A = Animation. The rows of the table contain only a subset of the file formats mentioned in this report. They serve as an example of how each file format specification has to be analyzed to meet the preservation objective. The superscripts in the column labeled as 'Format' refer to the reference list and the documentation studied to place the checkmarks. The empty spots refer either to missing placeholders for the parameters or to missing (or impossible to find) information in the format specifications about the existence of the specific parameters.

Format	Geometry				Appearance				Scene				Animation
	Faceted	Parametric	CSG	B-Rep	C	M	I	B	L	V	T	G	
3ds [8]	√	√			√	√	√	√	√	√	√		
igs [9]	√	√	√	√	√						√	√	
lwo [9]	√	√			√	√	√	√					
obj [7]	√	√			√	√	√	√				√	
ply [3]	√				√	√	√	√					
stp [10-14]	√	√	√	√	√							√	
wrl [1, 8]	√	√			√	√	√	√	√	√	√	√	√
u3d [4]	√				√		√	√	√	√	√	√	√
x3d [5]	√	√			√	√	√	√	√	√	√	√	√

8 Summary

In this technical report, we have listed many of the 3D file formats currently available (more than 140 file formats), as well as examined the types of data/parameters that many of the formats support and outlined the conversion errors and considerations about conversion evaluations.

We classified the 3D data content into three categories, such as geometry, appearance and scene. With regards to geometry, 3D formats can store a sampled version of the object through vertices, edges, faces, and surface patches or it can store the exact solid through CSG or B-Rep. With regards to appearance 3D formats can store: vertex colors, vertex texture coordinates, and physical material properties. Lastly, scene information can be stored as: lights, cameras, groups of geometry, and transformations on these groups.

Finally, we provide an estimate of the number of models per file format in Table 4 according to www.turbosquid.com as of 8-25-2008. The visualization of the model distribution is presented in Figure 1.

Table 4: Number of models according to file formats available at www.turbosquid.com as of 8-25-2008.

Format	Number of Models
max	71,506
3ds	47,633
obj	25,306
lwo, lw, lws	18,333
dxf	10,895
c4d	10,629
ma, mb	9,082
fbx	5,978
hrc, xsi	5,069

blend	2,977
x	2,108
dwg	1,922
wrl, wrz	1,724
cab, scn	1,481
gmax	1,369
3dm	1,205
w3d	1,063
flt	855
br4, obp	829
pz3, pp2	712
vue	596
ige, igs, iges	527
rib, slc, sl, slo	435
stl	252
skp	228
lxo	149
fmz	95
wire	91
eim, fac	60
dae	56
s3d	53
3dmf	49
m3g	12
md3	7
rvt, rte, rfa	7
pwc	1

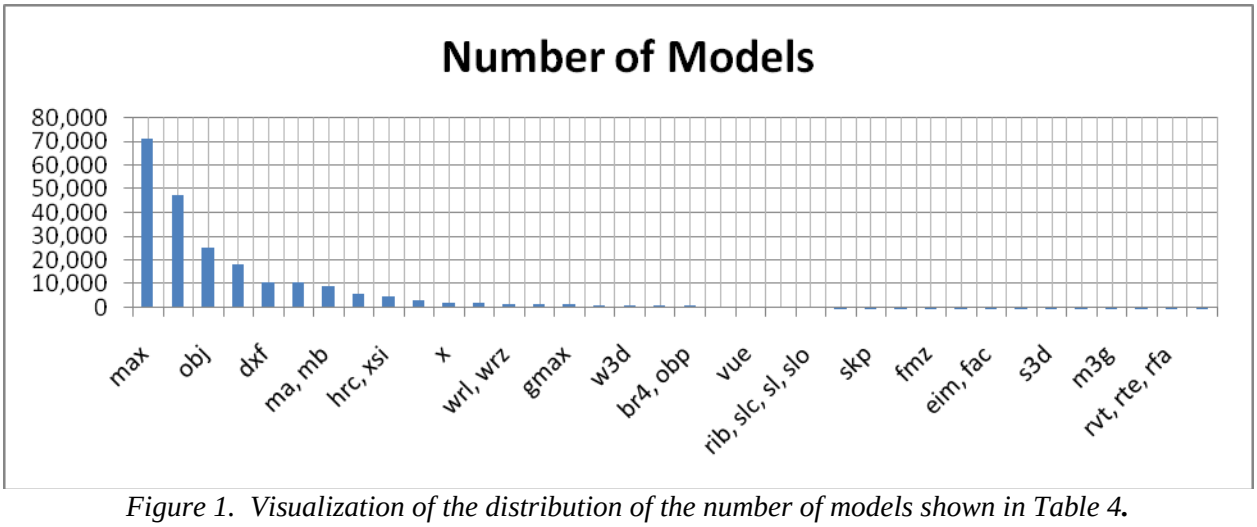

Figure 1. Visualization of the distribution of the number of models shown in Table 4.

9 Acknowledgment

The funding for this research came from the National Archive and Records Administration (NARA) as a supplement to NSF PACI cooperative agreement CA #SCI-9619019. The views and conclusions contained in this document are those of the authors, Kenton McHenry and Peter Bajcsy, and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the National Archive and Records Administration, or the U.S. government.

10 References

[1] *The Virtual Reality Modeling Language (VRML) – Part 1: Function Specification and UTF-8 Encoding*, ISO/IEC 14772-1, 1997.

[2] G. Bell, A. Parisi, and M. Pesce. *The Virtual Reality Modeling Language, Version 1.0 Specification, 1995*.

[3] P. Bourke. *Data Formats*, URL: http://local.wasp.uwa.edu.au/~pbourke/dataformats

[4] *Universal 3D File Format*, ECMA 363, 2006.

[5] *Extensible 3D (X3D) Specification*, ISO/IEC 19776-1, 2006.

[6] A. Del Bimbo and P. Pala, *Content Based retrieval of 3D objects*, ACM Transactions on Multimedia Computing, 2006

[7] *Wavefront OBJ: Summary*, URL: http://www.fileformat.info/format/wavefrontobj

[8] *3D File Formats*, URL: http://www.ibrtses.com/opengl/fileformats3d.html

[9] *3D Graphics Files*, URL: http://www.wotsit.org
[10] *The Canadian STEP Centre's "Introducing STEP" Publication*, URL: http://www.ic.gc.ca/epic/site/ad-ad.nsf/en/ad03581e.html#navigation

[11] *PDES Inc*. STEP overview and software tools, URL: http://pdesinc.org/step_overview.html; http://pdesinc.org/free_STEP.html

[12] Brunnermeier, S.B., & Martin, S.A. (March 1999), Interoperability Cost Analysis of the U.S. Automotive Supply Chain, Research Triangle Institute, Technical Report RTI Project Number 7007-03; URL: http://www.rti.org/pubs/US_Automotive.pdf

[13] *STEP standard*, ISO 10303 TC184/SC4, URL: http://www.tc184-sc4.org/SC4_Open/Projects/documentation.cfm?CFID=395674&CFTOKEN=32799048&FID=1104

[14] *STEP Application Handbook*, ISO 10303, Version 3, SCRA, 2006; URL: http://www.uspro.org/documents/STEP_application_hdbk_63006_BF.pdf

[15] PRONOM project, the National Archives of the UK. URL: http://www.nationalarchives.gov.uk/pronom/

[16] DROID (Digital Record Object Identification), a software tool developed by The National Archives to perform automated batch identification of file formats, URL: http://droid.sourceforge.net/wiki/index.php/Introduction.

[17] A. Brown, 'Automatic File Format Identification using PRONOM and DROID,' Digital preservation technical paper, The National Archives of the UK, March 7th, 2006, http://www.nationalarchives.gov.uk/aboutapps/fileformat/pdf/automatic_format_identification.pdf