OpenCascade Primitives BRep - Box

eryar@163.com

Abstract. BRep is short for Boundary Representation. Boundary Representation gives a complete description of an object by associating topological and geometric information for solid modeling. In this case, objects are described by their boundaries. There are two types of information in BRep: Topological information and Geometric information. This paper is concerned with the Box BRep in OpenCascade, and also show how to use Tcl script to dump box BRep info.

Key words. OpenCascade, BRep, Boundary Representation, Box, Winged-Edge Structure

1. Introduction

边界表示法（BRep: Boundary Representation）通过拓朴（Topology）和几何（Geometry）给出了一个物体完整的描述，即用边界来表示物体。

在 OpenCascade 中的拓朴实体有以下几种：

- Vertex: a zero-dimensional shape corresponding to a point in geometry;
- Edge: a shape corresponding to a curve, and bound by a vertex at each extremity;
- Wire: a sequence of edges connected by their vertices;
- Face: part of a plane(in 2D geometry) or a surface(in 3D geometry) bounded by a closed wire;
- Shell: a collection of faces connected by some of the edges their wire boundaries;
- Solid: part of 3D space bound by a shell;
- Compound solid: a collection of solids.

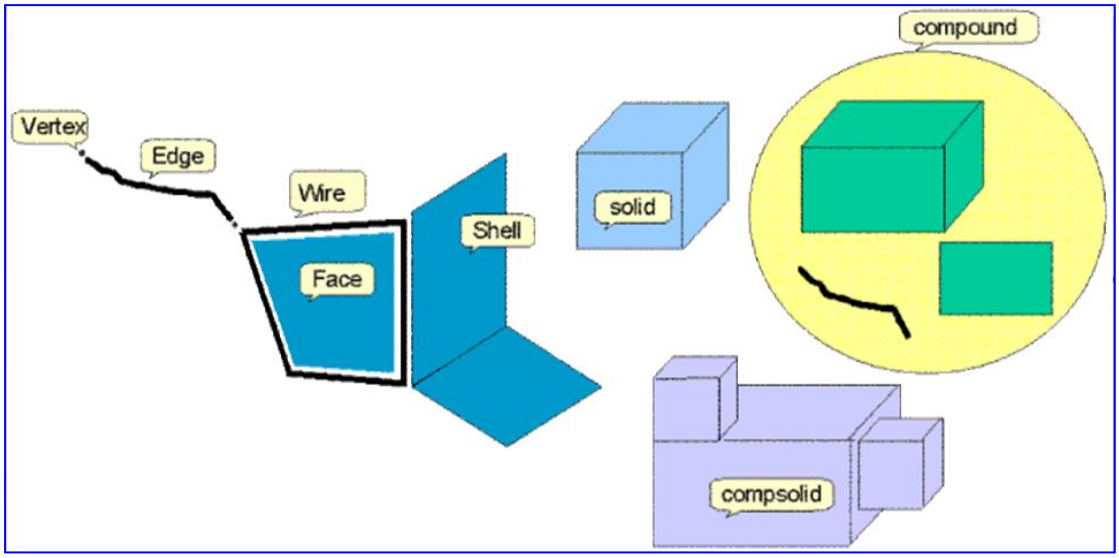


Figure 1.1 Topological Entities in OpenCascade

OpenCascade 中的拓朴实体如上图所示，其中 Compound 可以包含很多 Solid；Solid 由 Shell 包围而成；Shell 由相连的 Face 组成；Wire 由相连的 Edge 组成；Edge 对应一条曲线，且曲线的端点处由 Vertex 组成；Vertex 对应空间中一个点。OpenCascade 的拓朴结构的类层次关系如下图所示：

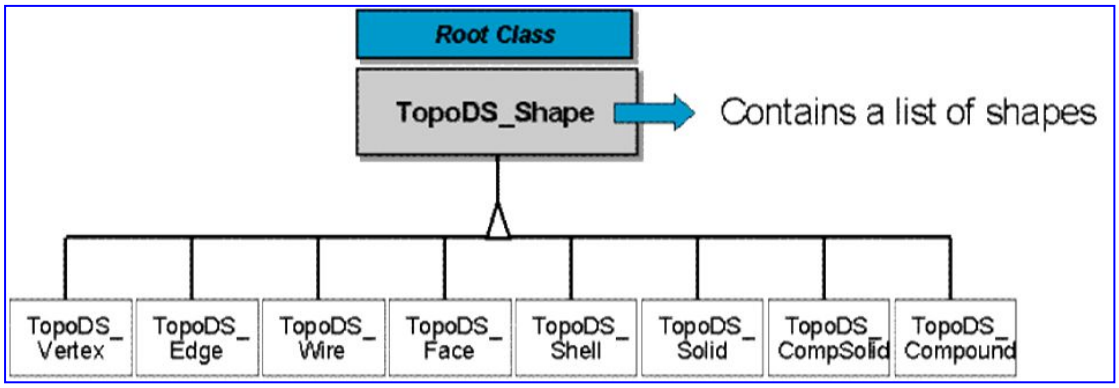


Figure 1.2 Topological Shape Hierarchy of OpenCascade

OpenCascade 的 BRep 表示中几何曲线曲面是参数化表示的，即曲线上的点与一个参数 u 有关，曲面上的点与两个参数 u,v 来有关。因为是参数化表示的，所以曲线曲面是有向的（naturally orientated）。方向（Orientation）是重要的，对面而言方向是面上每个点处的法向。

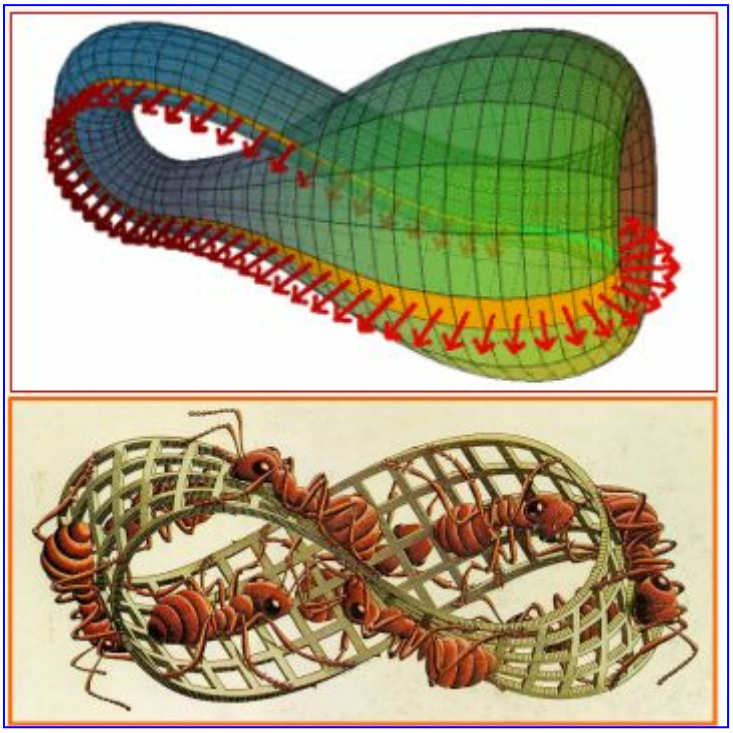


Figure 1.3 Orientation of Faces

使用参数表示的曲线曲面还需要注意曲线曲面的有界性 Bounded，奇异性 Singularity（曲面上的奇点 Singular Point），曲面上的曲线（PCurve: Curve on Surface）等概念。

本文通过使用 Tcl 脚本将 OpenCascade 中的基本形状长方体 Box 的边界表示 BRep 数据导出到文件，继而方便分析 Box 在 OpenCascade 中的边界表示。

2. Dump Box BRep Info by Tcl

为了得到长方体的边界表示数据，先要用类 BRepPrimAPI_MakeBox 来生成一个 TopoDS_Shape，再使用 BRepTools::Dump()可以将这个 TopoDS_Shape 的信息以便于理解的方式输出。如果用 C++来编程，只有这两句，但是为了编译链接成功，需要包含相关的头文件及引用相关的库，需要编译工具来编译链接，还是比较麻烦的。本文使用 Tcl 脚本来输出，体验一下在 OpenCascade 中使用 Tcl 脚本编程的便利。

在 Draw Test Harness 中输入以下三行 Tcl 命令，即可以将 Box 的 BRep 信息输出到屏幕：

Figure 2.1 Dump Box BRep info in Draw Test Harness

考虑到输出内容过多，在命令窗口查看不便，有些信息被覆盖，于是将这些信息导出到文件，Tcl 代码如下所示：

```
#
#    Copyright (c) 2014 eryar All Rights Reserved.
#
#           File : box.tcl
#         Author : eryar@163.com
#           Date : 2014-03-17 19:00
#        Version : 1.0v
#
#    Description : Dump OpenCascade primitive box topology info to file.
#
#      Key Words : OpenCascade, BRep, Tcl, Box
#
#


# Load Modeling algorithms.
pload MODELING

# Make the box.
box theBox 1.0 2.0 3.0
set boxInfo [dump theBox]

# Save box BRep info to file.
if [catch {open d:/box.txt w+} theFile] {
    puts "Cannot open d:/box.txt for writing: $theFile"
} else {
    puts $theFile $boxInfo
    flush $theFile
    close $theFile
}

puts "Dump Info Finished!"
```

将上述内容保存到文件 box.tcl，并在 Draw Test Harness 中输入如下命令：

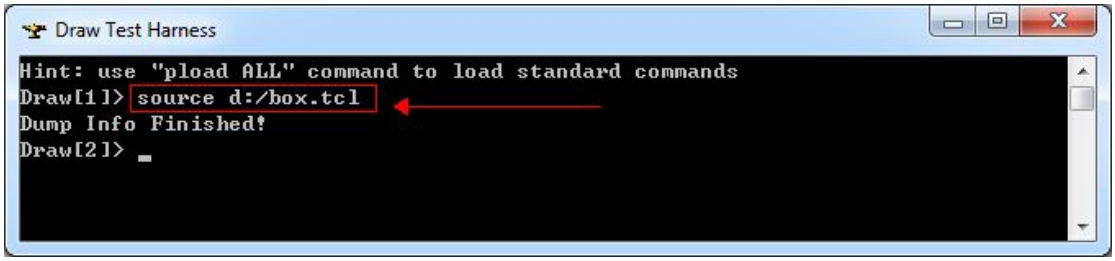


Figure 2.2 Run a Tcl File in Draw Test Harness

执行成功后会在 D 盘中生成一个 box.txt 的文件，文件部分内容如下所示：

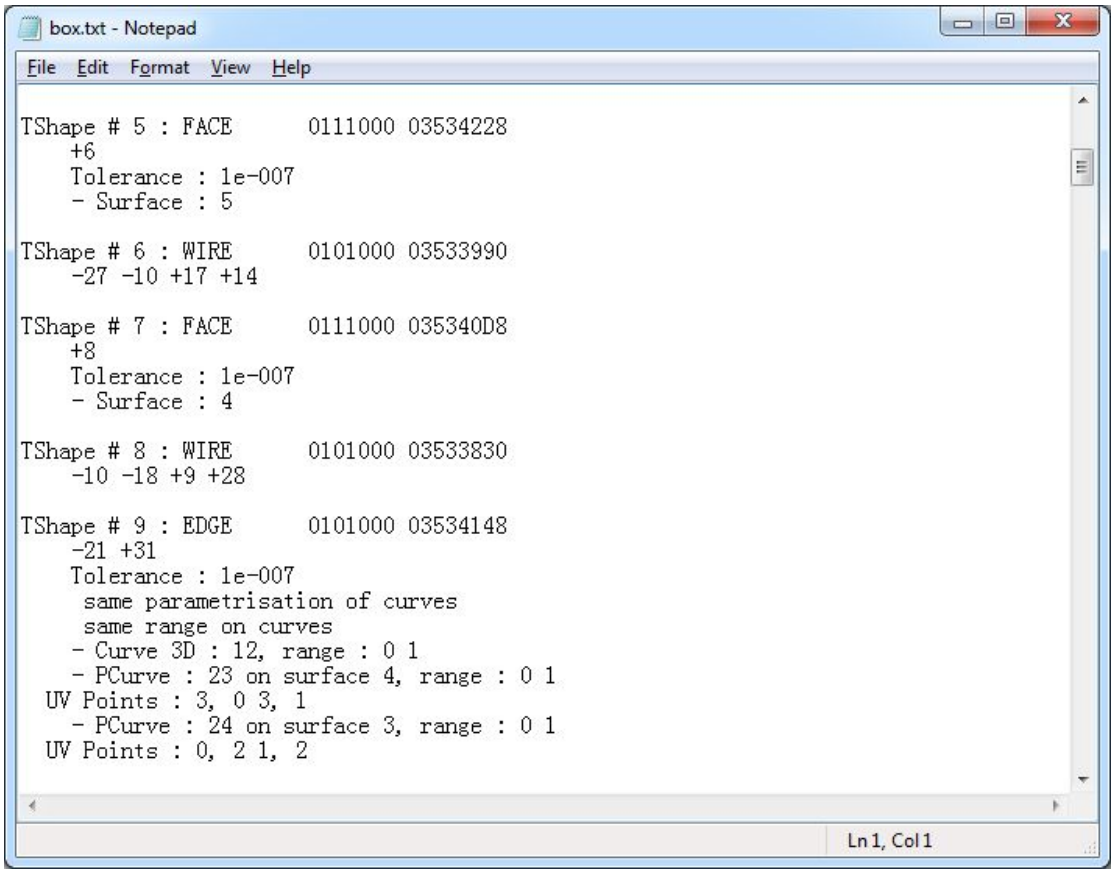


Figure 2.3 Box BRep Info Generated by Tcl Script

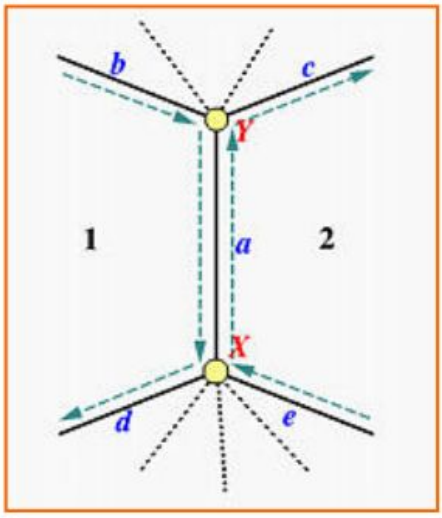
为了方便读者测试及本文中分析的 Box 数据的准确性，box.tcl 和 box.txt 都可通过文章后面的链接下载。

3. Box BRep in OpenCascade

翼边结构（Winged-Edge Structure）及其边的表格方式（Edge Table）可以清晰地表达出形状的拓朴关系，表中内容有：

- 边的名称，即边的编号 Edge Name；
- 边的起止顶点 Start vertex and end vertex；
- 相连的两个面 Left face and right face；
- 遍历左面时访问边的顺序 The predecessor and successor edges when traversing its left face；
- 遍历右面时访问边的顺序 The predecessor and successor edges when traversing its right face；

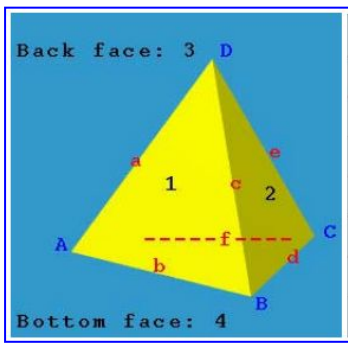
如下图所示为 Edge Table 表示的翼边结构：



Edge	Vertices		Faces		Left Traverse		Right Traverse	
Name	Start	End	Left	Right	Pred	Succ	Pred	Succ
a	X	Y	1	2	b	d	e	c

Figure 3.1 Edge Table of Winged-Edge Structure

从上图可知，给出一个边 a 后，从表中可以清晰看出与边 a 相连的顶点及面的信息。如下图所示为一个 Edge Table 的示例：



Edge	Vertices		Faces		Left Traverse		Right Traverse	
Name	Start	End	Left	Right	Pred	Succ	Pred	Succ
a	A	D	3	1	e	f	b	c
b	A	B	1	4	c	a	f	d
c	B	D	1	2	a	b	d	e
d	B	C	2	4	e	c	b	f
e	C	D	2	3	c	d	f	a
f	A	C	4	3	d	b	a	e

Figure 3.2 Complete Edge Table for Pyramid

由上可知，给出翼边结构表示中的一个边，可以很方便得到与之相连的顶点和面的信息。

但是 OpenCascade 中没有使用翼边结构来表示形状，可以从图 1.2 所示的类结构得出。因为每个形状只包含子形状的数据，不包含其父形状的数据，所以在 OpenCascade 中不能反向访问其父形状（In OpenCascade, there is no back pointer from sub-shapes to ancestor shapes.），所以若想得到与给定顶点或边相连的面或环等信息，必须使用这个工具：

```
TopExp::MapShapesAndAncestors()
```

如下图所示使用上述工具得到与指定边相连的所有面的信息：

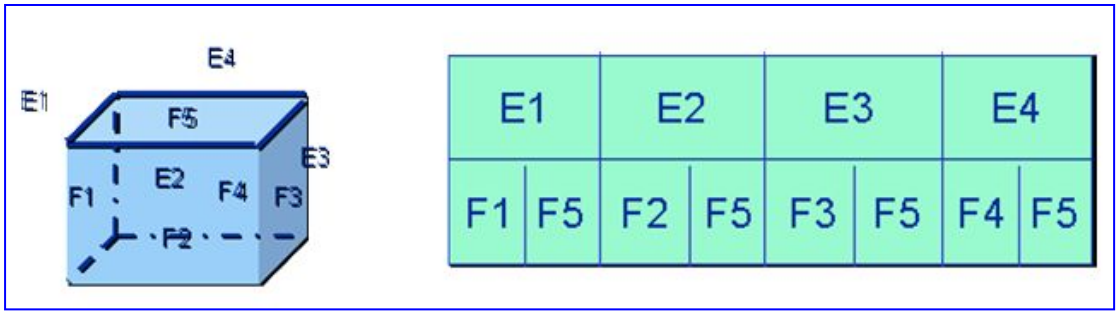


Figure 3.3 Get Ancestor Shapes in OpenCascade

根据前面使用 Tcl 脚本生成的 Box，尺寸分别为 1.0，2.0，3.0：

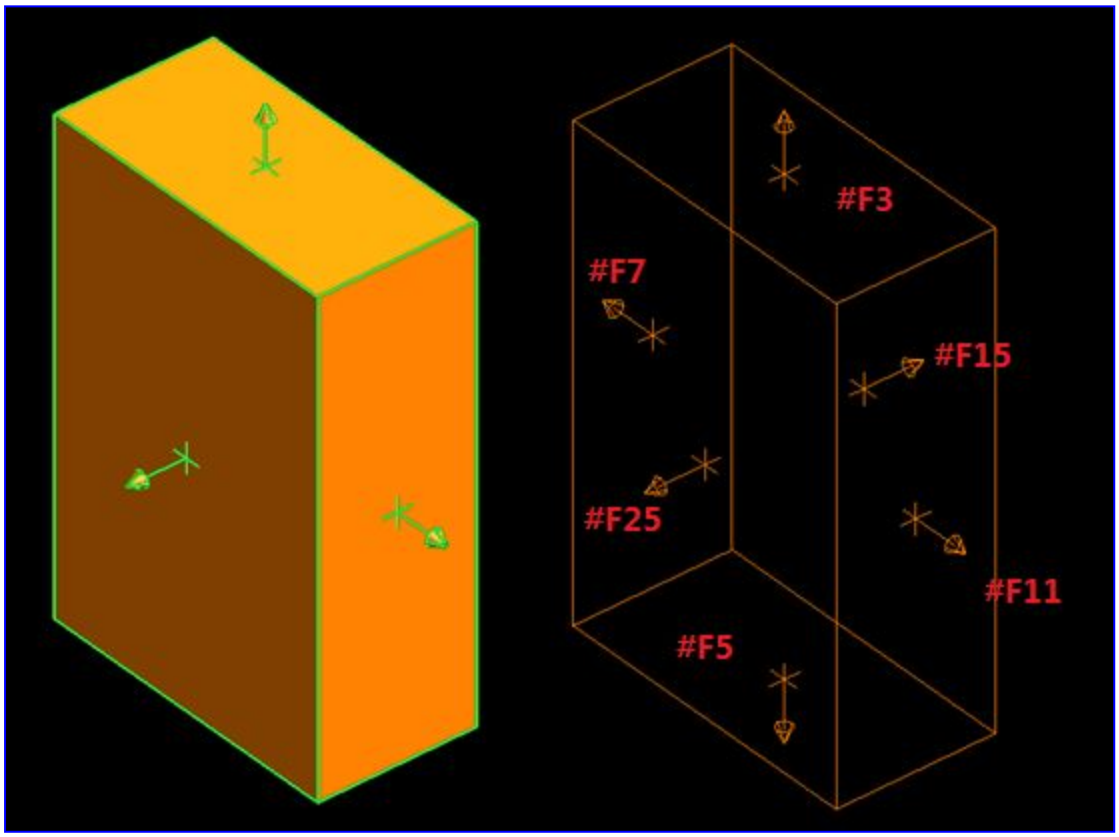


Figure 3.4 The Box generated by Tcl Script

为了正确显示出 Box，每个面的法向必须与上图中的箭头方向一致。下面通过 box.txt 中的拓朴信息的编号，从 vertex 开始来组装成一个 Solid 的形状。对应顶点的编号及其在空间中的坐标如下图所示：

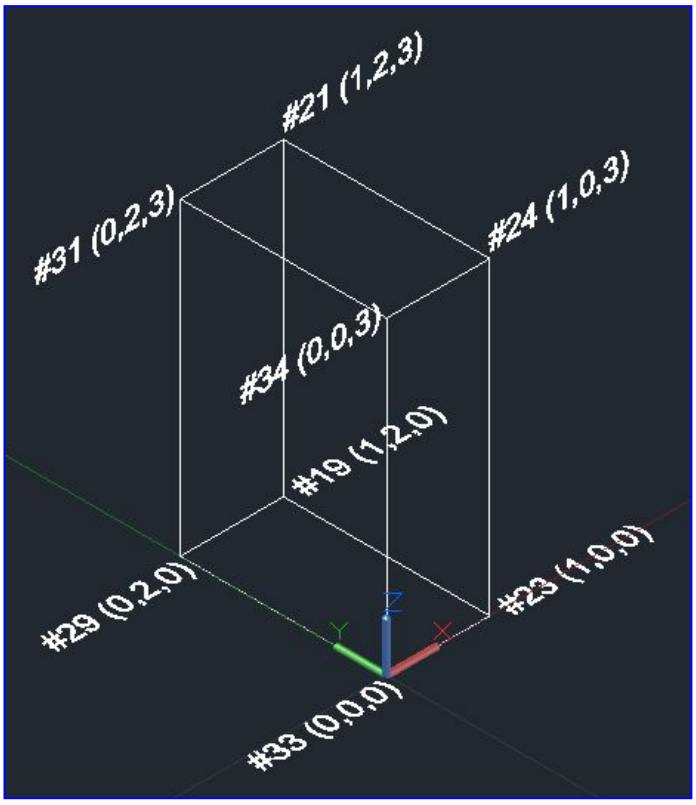


Figure 3.5 Vertex of the Box BRep

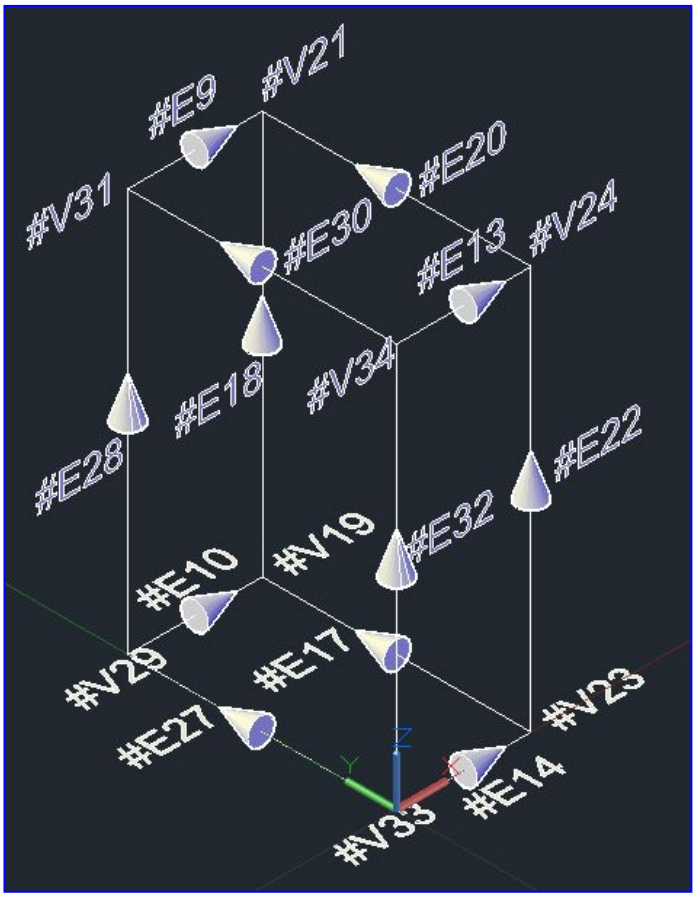


Figure 3.6 Edges of the Box BRep

上图中根据边中的顶点的方向，标示出边的方向：从标号为+正的顶点到标号为－负的顶点。

```
TShape # 4 : WIRE        0101000 03533A70
    -30 -9 +20 +13

TShape # 6 : WIRE        0101000 03533990
    -27 -10 +17 +14

TShape # 8 : WIRE        0101000 03533830
    -10 -18 +9 +28

TShape # 12 : WIRE        0101000 035336D0
    -14 -22 +13 +32

TShape # 16 : WIRE        0101000 03533450
    -22 -20 +18 +17

TShape # 26 : WIRE        0101000 03208F28
    -32 -30 +28 +27
```

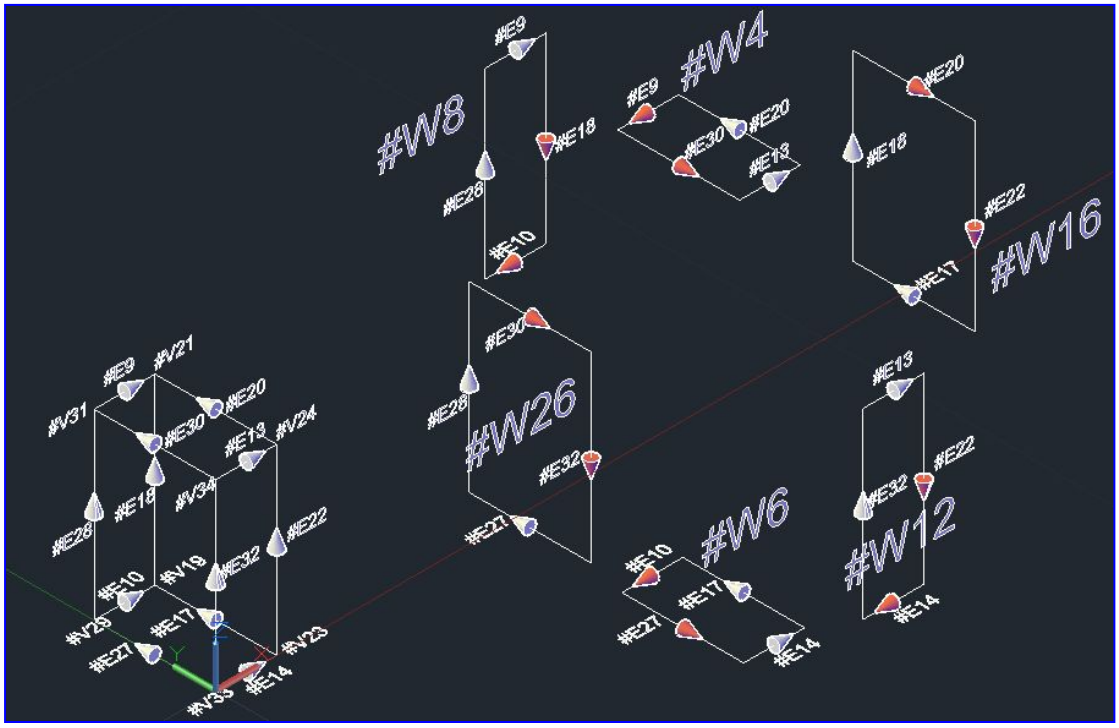


Figure 3.7 Wire #4 of Box BRep

由图可知，由于 Wire #4 是由边 E30，E9，E20，E13 组成，由于 30 和 9 号边前有负号，所以需要反向，方向反向的边用红色箭头标示。其它 Wire 的处理与此类似。通过上面左右两幅图的对比，可以清楚知道哪些 Edge 在组成 Wire 时反向了。

box.txt - Notepad

File Edit Format View Help

```
TShape # 2 : SHELL      0101100 03208EE8
    -25 +15 -11 +7 -5 +3

TShape # 3 : FACE       0111000 03534260
    +4
    Tolerance : 1e-007
    - Surface : 3

TShape # 5 : FACE       0111000 03534228
    +6
    Tolerance : 1e-007
    - Surface : 5

TShape # 7 : FACE       0111000 035340D8
    +8
    Tolerance : 1e-007
    - Surface : 4

TShape # 11 : FACE       0111000 03534030
    +12
    Tolerance : 1e-007
    - Surface : 2

TShape # 15 : FACE       0111000 03219B58
    +16
    Tolerance : 1e-007
    - Surface : 6

TShape # 25 : FACE       0111000 0320CD40
    +26
    Tolerance : 1e-007
    - Surface : 1
```

Figure 3.8 Faces and Shells of Box BRep

由上图可知，每个 Face 由一个 Wire 组成。注意到由 Face 组成 Shell 时，Face 前面有方向性。如 Face 5#前面有负号，意思是面的法向与 Wire 的方向相反，其他类似。

最后的 Solid 由 Shell 组成。这样 Box 的拓朴数据就形成了。其中 Face, Edge, Vertex 中包含了几何数据。Face 的参数表示的几何数据位于 surfaces 部分，如下图所示：

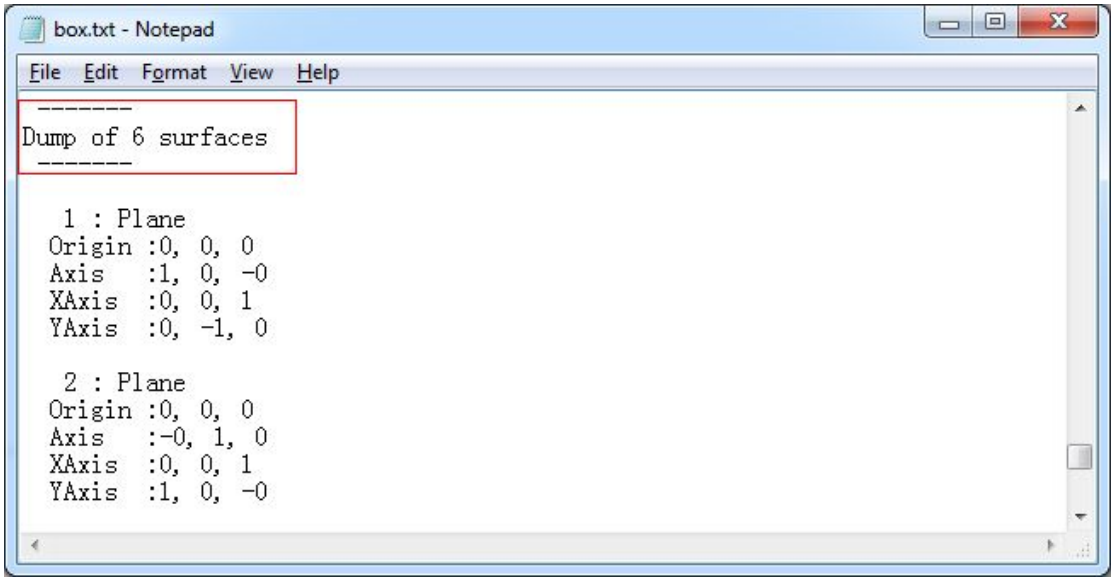


Figure 3.9 Parametric Surfaces of the Box

Edge 中包含了多种曲线数据，在 Box 的 Edge 中就包含两种曲线数据，一种是三维曲线；一种是曲面上的曲线。如下图所示：

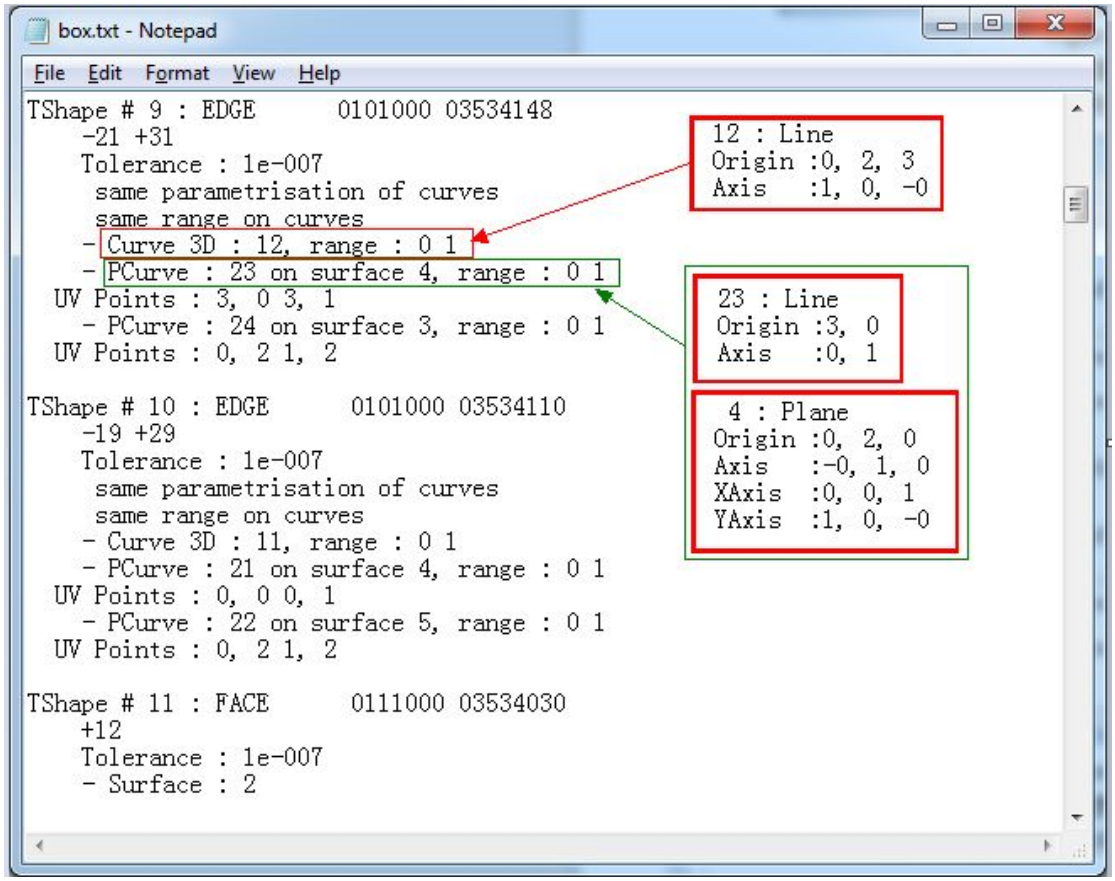


Figure 3.10 Curve info of the Edge in Box

其中 Curve3D 对应的就是参数化的三维曲线，这个很好理解。PCurve 为曲面上的曲线。下面以 Edge#9 为例，来说明边中的几何数据。三维曲线#12 是原点为（0，2，3）方向为 X 方向的直线，在参数区间[0，1]的端点处，直线对应的两个点为（0，2，3）和（1，2，3），分别对应#31Vertex 和#21Vertex。所以 Vertex 前面的－负号表示 Edge 中曲线终点，+正号表示 Edge 中曲线的起点。

PCurve 为#4 号平面上的#23 号参数曲线，由下图可知#4 号曲面对应的 Face 为#7，对应的 Wire 为#8：

```
TShape # 7 : FACE       0111000 035340D8
    +8
    Tolerance : 1e-007
    - Surface : 4

TShape # 8 : WIRE       0101000 03533830
    -10 -18 +9 +28
```

Figure 3.11 Surface of Face

$$S(u,v)=P+u\cdot D_u+v\cdot D_v,\ (u,v)\in(-\infty,\infty)\times(-\infty,\infty).$$

根据#23 号二维参数曲线计算得出曲面上的 U,V 分别为（3, 0）和（3, 1），将得到 U，V 代入#4 曲面中得出对应的曲面上的点，计算过程如下所示：

$$\begin{aligned}
&S(3,0)=(0,2,0)+3\cdot(0,0,1)+0\cdot(1,0,0)\\
&S(3,0)=(0,2,3)\\
&S(3,1)=(0,2,0)+3\cdot(0,0,1)+1\cdot(1,0,0)\\
&S(3,1)=(1,2,3)
\end{aligned}$$

从中可以看出，PCurve 表示的曲线与三维曲线#12 相同，不过其中还保存了曲面的信息。

4. Conclusion

本文通过使用 Tcl 在 OpenCascade 中生成 Box 的边界表示信息，通过对数据的分析，得出 Box 在 OpenCascade 中的边界表示方式。并通过与翼边结构的对比，说明 OpenCascade 中的拓朴结构不是翼边结构。通过分析边界表示中的几何数据，来理解边界表示中几何数据的应用。

若对文中内容有任何意见、建议都可以与我取得联系，邮箱：eryar@163.com，欢迎讨论、交流、指导。

5. References

1. OpenCascade, Test Harness User's Guide 2013
2. OpenCascade, BRep Format Description White Paper, 2013
3. John K. Ousterhout, Tcl and Tk Toolkit, 1993