

# Elementary Application of Geometry Face Modelling under VRML Complex Object Muster

Duan Xinyu

Media Research Institute, Anyang Normal University, Anyang, Henan, China

Email: dxy@aynu.edu.cn

Gao Guowei

Common Computer Section, Anyang Normal University, Anyang, Henan, China

Email: gaoguowei@126.com

**Abstract**—Geometry face modeling technique is basically a kind of important modeling application when proceeding space complex object construction. First, paper introduced rules for building VRML virtual face model, mainly IndexedFaceSet node's function and its syntax format etc. Here we discussed some of IndexedFaceSet node's elementary fields' meaning, for example coord and coordIndex field, color and colorIndex field, also colorPerVertex field. ccw and solid are two fields distinguished greatly from IndexedLineSet node's. Second, paper designed application examples for illustrating color rendering and object building method for VRML's geometry face Model. Finally, paper pointed out that for any complex natural modeling object, we all can use VRML provided PointSet, IndexedLineSet, IndexedFaceSet, ElevationGrid and Extrusion node, as Shape modeling node's geometry field value, for constructing.

**Index Terms**—IndexedFaceSet, coord, color, ccw, solid

## I. INTRODUCTION

Usually, a programmer can employ Sphere, Cylinder, and Cone etc elementary node in VRML for constructing regular geometry models. Yet it is rarely satisfied and acceptable for human's visual demand, if considering complicate scene existing in natural world. Then we should take this into consideration for some of modeling technique based upon basic geometry point, line and face manner. This paper mainly discusses one kind of VRML complex space object muster. That is the modeling and rendering manner based on geometry face mode.

## II. RULE FOR BUILDING VIRTUAL FACE MODEL

VRML's complex face model is a space sealed, circled and colored polygonal line area. Here, any numbers of polygonal face's coordinate points are sequentially and successively connected, from beginning to end, forming a closed region. When rendering, explorer will sequentially connect every space coordinate point, in accordance with

space coordinate point's index order, and in the same time fill it with lightproof color, so as to construct space geometry flat face. VRML employs IndexedFaceSet node to build scene face model.

In VRML, we employ IndexedFaceSet node to construct space face models. Following is its syntax regulation and format.

```
IndexedFaceSet {
  coord NULL #exposedField SFNode
  coordIndex [ ] #field MFInt32
  texCoord NULL #exposedField SFNode
  texCoordIndex [ ] #field MFInt32
  color NULL #exposedField SFNode
  colorIndex [ ] #field MFInt32
  colorPerVertex TRUE #field SFBool
  normal NULL #exposedField SFNode
  normalIndex [ ] #field MFInt32
  normalPerVertex TRUE #field SFBool
  ccw TRUE #field SFBool
  convex TRUE #field SFBool
  solid TRUE #field SFBool
  creaseAngle 0.0 #field SFFloat
  set_coordIndex #eventIn MFInt32
  set_texCoordIndex #eventIn MFInt32
  set_colorIndex #eventIn MFInt32
  set_normalIndex #eventIn MFInt32}
```

Here we mainly discuss its basic fields' significance and application method. Some of them are similar with IndexedLineSet node [1] [2].

### a) About coord Field

In IndexedFaceSet node, the function of coord field is to set up endpoint coordinates for space face model. Its field value is another Coordinate node.

In Coordinate node, the point field's function is to provide a location coordinate list for a series of three dimension space points. Every space coordinate is constituted by X, Y, Z three component parts. In every location of defined space point coordinate will automatically generate a space point model.

The coord field's default value is NULL, that means not create any face model in default.

### b) About coordIndex Field

Manuscript received January 1, 2010; revised June 1, 2010; accepted July 1, 2010.

Supported by education technology key discipline of AYNU, corresponding author: Xinyu Duan, Email: dxy@aynu.edu.cn.

The function of `coordIndex` field is to illustrate a series of coordination index lists for creating one or multi flat face polygonal line path. The so called coordinate index is referred to that in `coordIndex` field, programmer should use a series of integer value, starting from 0, to sequentially number `Coordinate` node's point field providing point coordinates. And then in every coordinate index, we will apply this number to sequentially designate sort order for `coord` field defined corresponding coordinate point, when constructing VRML flat face's polygonal line model. In `coordIndex` field, the coordinate index value must be greater than or equal to 0, and also be integer value.

Here we should pay attention to that for space face model, when rendering explorer will always successively construct polygonal line sections, from beginning to end in accordance with coordination index sequence, until index list's end or meeting a -1 coordinate index value, and also to connect this polygonal line section from beginning to end for forming a flat face model. Next polygonal line will start building from next coordinate index value behind -1, until index list's end or meeting another -1 coordinate index value, and also to connect this polygonal line section from beginning to end for forming another flat face model. And so forth.

#### *c) About `texCoord` Field*

The function of `texCoord` field's value is to execute special texture render for irregular plane face model created by `IndexedFaceSet` node. Pay attention that `texCoord`'s field value is another `TextureCoordinate` node. The default value of `texCoord` is `NULL`, meaning empty and not implement any special texture render.

The syntax of `TextureCoordinate` node is written below.

```
TextureCoordinate {
    point [ ] #exposedField MFVec2f}
```

Here, VRML apply point field's value to install position area in texture S, T coordinate system, when using texture to render flat face model. The point field's value is consisted of two floating-point numbers list, respectively representing coordination distance in S, T direction for texture picture's four vertexes, all measuring from texture coordination origin. The point field's default value is an empty distance list.

#### *d) About `texCoordIndex` Field*

The function of `texCoordIndex` field's value is to install a series of texture coordinate index list for rendering a piece of or multi plane surfaces. The so called texture coordinate index is referred to that in `texCoordIndex` field, programmer should use a series of integer value, starting from 0, to sequentially number `TextureCoordinate` node's point field providing texture coordinates. And then use this number to correspondingly render series of faces defined by `coordIndex` field's value. In `texCoordIndex` field, its value must be greater than or equal to 0, and also be integer value.

The default value of `texCoordIndex` field is an empty list, meaning sequentially rendering relative face model produced by `coordIndex` field, by applying `TextureCoordinate` node's point field providing texture coordinate list series.

#### *e) About `color` Field*

In `IndexedFaceSet` node, color field's function is to solely color space face model. Its field's value is specifying another `Color` node.

In `Color` node, color field's function is to independently color a series of space face model of `IndexedFaceSet` node built. The color field's value provides a series of color list. Every color is correspondingly to render a flat face or multi vertexes of a face which `IndexedFaceSet` node created. The color field's default value is a null list, meaning not proceeding independent color for space face model.

#### *f) About `colorIndex` Field*

The function of `colorIndex` field is to install a series of color index lists for rendering one piece surface or multi space faces. Here the so called color index is referred to that in `colorIndex` field, programmer should use a series of integer number, starting from 0, to sequentially number `Color` node's color field providing colors. And then we can apply this number to correspondingly render a series space face or theirs vertexes installed by `coordIndex` field's value.

In `colorIndex` field, the coordinate index value must be greater than or equal to 0, and also be integer value. The default value of `colorIndex` field is null list, which means by using color field's color listing sequence of `Color` node to sequentially render relative space coordinate point installed by `Coordinate` node's point field if `colorPerVertex` field's value is `TRUE`, or relative face model installed by `coordIndex` field if `colorPerVertex` field's value is `FALSE`.

#### *g) About `colorPerVertex` Field*

In `IndexedFaceSet` node, its `colorPerVertex` field's function is to set up, for space face model, whether the color rendering is based upon vertex if its value is `TRUE`, or based upon face if its value is `FALSE`.

If `colorPerVertex` field's value is `TRUE`, and in the same time `colorIndex` field's value is null, under this condition the color field's first color list of `Color` node's will to color the first coordinate vertex built by `Coordinate` node's point field. According to this regulation, the color field's second color list of `Color` node's will to color the second coordinate vertex built by `Coordinate` node's point field. Now, the color number in color field of `Color` node is at least equal to the vertex coordinate number installed by point field of `Coordinate` node. If the color number in color field of `Color` node is less than the vertex coordinate number installed by point field of `Coordinate` node, that we should use `coordIndex` field's coordinate index list, which is relative to `colorIndex` field, to install vertex color index list. This is aiming to reuse color field in `Color` node. Please pay attention to color rendering mode based upon vertex, the surface part will be presented special color gradual change and transition among vertexes.

If `colorPerVertex` field's value is `FALSE`, and in the same time `colorIndex` field's value is null, under this condition the color field's first color list of `Color` node's will to color the first indexed face in `coordIndex` field's coordinate index list. According to this regulation, the

color field's second color list of Color node's will to color the second indexed face in coordIndex field's coordinate index list, and so on. Now, the color number in color field of Color node is at least equal to the surface indexed number installed by coordIndex field. If the color number in color field of Color node is less than the surface indexed number installed by coordIndex field, that we should use coordIndex field's surface indexed list, which is relative to colorIndex field, to install surface color index list. This is aiming to reuse color field in Color node. Pay attention to color rendering based on face, the surface part will be solely colored, none any special effects presented.

The default value of colorPerVertex field is TRUE, that means single color for each vertex in default.

#### *h) About normal Field*

The usage of normal field's value is to set another Normal node. This Normal node is taken to install unit normal vector for surfaces or vertexes in designated face union. The unit normal vector can be used to change light and shade render effect for scene luminous system. Its default value is NULL, meaning empty and not install any unit normal vector.

Normal node's syntax format is written below.

```
Normal {
    vector [ ] #exposedField MFVec3f}
```

In Normal node, the vector field is used to install unit normal vector for series of space face models created by IndexedFaceSet node. Every unit normal vector is consisted of three float numbers, separately expressing normal vector's X, Y, Z component. A series of unit normal vector lists is provided by vector field. Each unit normal vector will be applied to correspondingly render one surface or every vertexes of a surface constructed by IndexedFaceSet node. Its default value is an empty list, meaning automatically produce unit normal vector in explorer's default mode.

#### *i) About normalIndex Field*

The function of normalIndex field's value is to install a series of unit normal vector's index list for the purpose of dark and light control. The so called unit normal vector index is referred to that in normalIndex field, programmer should use a series of integer value, starting from 0, to sequentially number Normal node's vector field providing unit normal vector. And then use this number to correspondingly control series of faces or vertexes defined by coordIndex field's value. In normalIndex field, its value must be greater than or equal to 0, and also be integer value.

The default value of normalIndex field is an empty list. There are two levels of meaning here. One means sequentially control light and shade level of space coordinate points provided by Coordinate node's point field, by applying Normal node's vector field providing unit normal vector list series, if normalPerVertex field's value is TRUE. The other means sequentially control light and shade level of relative face model installed by coordIndex field, by applying Normal node's vector field providing unit normal vector list series, if normalPerVertex field's value is FALSE.

#### *j) About normalPerVertex Field*

The field value of normalPerVertex is to set up whether the unit normal vector control is based on vertex if normalPerVertex's value is TRUE, or based on plane surface if normalPerVertex's value is FALSE.

If normalPerVertex's value is TRUE and simultaneously normalIndex field's value is empty, under this condition the vector field's first unit normal vector list of Normal node's will to install unit normal vector for the first coordinate vertex built by Coordinate node's point field. According to this regulation, the vector field's second unit normal vector list of Normal node's will to install unit normal vector for the second coordinate vertex built by Coordinate node's point field. Now, the unit normal vector number in vector field of Normal node is at least equal to the vertex number installed by point field of Coordinate node. If the unit normal vector number in vector field of Normal node is less than the vertex number installed by point field of Coordinate node, that we should use coordIndex field's coordinate index list, which is relative to normalIndex field, to install vertex's unit normal vector index list. This is aiming to reuse vector field's value in Normal node.

If normalPerVertex's value is FALSE and in the same time normalIndex field's value is empty, under this condition the vector field's first unit normal vector list of Normal node's will to install unit normal vector for the first face index installed by coordIndex field's coordinate index list. According to this regulation, the vector field's second unit normal vector list of Normal node's will to install unit normal vector for the second face index installed by coordIndex field's coordinate index list. And so on. Now, the unit normal vector number in vector field of Normal node is at least equal to the face index number installed by coordIndex field's face index number. If the unit normal vector number in vector field of Normal node is less than the face index list number installed by coordIndex field, that we should use coordIndex field's face index list, which is relative to normalIndex field, to install face's unit normal vector index list. Its aim is to reuse vector field's value in Normal node.

The default value of normalPerVertex is TRUE, that means installation of unit normal vector for each vertex in default.

#### *k) About ccw Field*

This ccw field value's function is, in optical line simulation equation and relative to customized or default unit normal vector, to install the setting mode of geometry object vertex coordinate system.

If this field is TRUE, system will adopt right hand rule to produce unit normal vector. That means the right hand thumb's direction is the unit normal vector direction, in the same time other fingers' direction is the direction for each vertex to be sequentially arranged and connected to form this flat face.

If ccw's value is FALSE, system will adopt left hand rule to produce unit normal vector. That means the left hand thumb's direction is the unit normal vector direction, in the same time other fingers' direction is the direction for each vertex to be sequentially arranged and connected

to form this flat face. Please refer to Fig.1 for reference. The left one representing right hand rule, on the contrary the right standing for left hand normal vector direction. Its default value is TRUE.

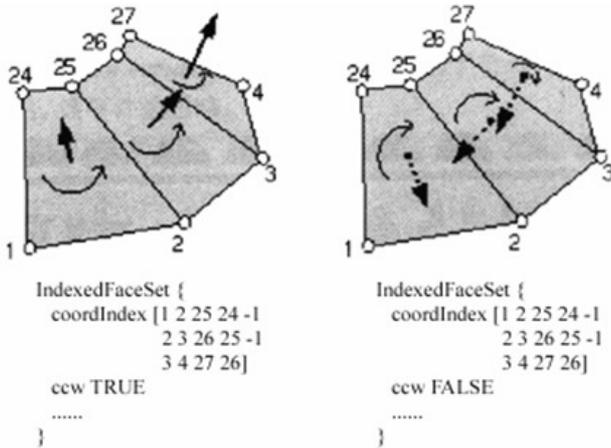


Figure 1. Function of ccw field

If ccw field's value is TRUE, referring to every unit normal vector direction when vertexes arranged by sequence, when observing built model in the opposite direction of unit normal vector, the vertexes is arrayed in anti-clockwise direction. If ccw field's value is FALSE, referring to every unit normal vector direction when vertexes arranged by sequence, when observing built model in the opposite direction of unit normal vector, the vertexes is arrayed in clockwise direction.

l) About convex Field

The field value of convex is to decide whether the surface of built face model is a convex surface or a concave surface. If the value is TRUE, then the surface is a convex one. In this situation, explorer will no longer execute partition operation when rendering. It aims to optimize arithmetic process. If the value is FALSE, the surface is a concave one. Under this situation, when rendering explorer will automatically partition this concave face into a series of smaller convex one and then implementing operation process.

m) About solid Field

Function of solid field value is to set up whether to construct a solid entity or not for the faces of indexed face muster. If its value is TRUE, the faces of indexed face muster will be connected and sealed up to form a solid entity. Under this condition, we can only see face model in the opposite direction of unit normal vector. That is to say the explorer will only render model's outside surface, not rendering inner side surface.

If its value is FALSE, the faces of indexed face muster will not be connected and sealed up to form a solid entity. Now we can see face model, not only from the opposite direction of unit normal vector, but also from the direction of unit normal vector. That means the explorer will, in the same time, not only render model's outside surface, but also render its inner side surface. The solid field's default value is TRUE. Its significance is to realize

explorer optimization, when the built model is a sealed up geometry entity,

Finally, the set\_coordIndex eventIn of IndexedFaceSet node is used to install new coordinate index list. The set\_colorIndex eventIn of IndexedFaceSet node is used to install new color index list.

n) About creaseAngle Field

A schematic diagram about creaseAngle field's function is referred below.

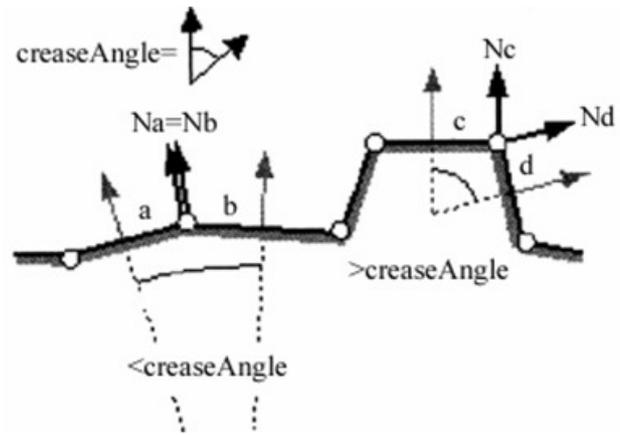


Figure 2. The function of creaseAngle field

The creaseAngle field's value is to set up a fold angle's threshold value that is indicated by radian measure. If the included angle between two adjacent surfaces' unit normal vector is small than designated threshold value, when rendering the contact edge of two surfaces will be vague handled. That is the so called smoothness draft. If the included angle between two adjacent surfaces' unit normal vector is great than designated threshold value, the contact edge of two surfaces will keep its original looks, no more smoothness draft process. This field's value must be greater or equal to 0.0. Its default value is 0.0.

Finally the set\_texcoordIndex eventIn of IndexedFaceSet node is used to install new texture coordinate index list. The set\_normalIndex eventIn of IndexedFaceSet node is used to install new unit normal vector index list.

III. MODELLING AND RENDERING OF GEOMETRY FACE

A. Pentagon Plane Modelling

```

#VRML V2.0 utf8
Shape {
  geometry IndexedFaceSet {
    coord Coordinate {
      point [ 0.0 1.0 0.0
            -0.951 0.309 0.0
            -0.588 -0.809 0.0
            0.588 -0.809 0.0
            0.951 0.309 0.0]
    coordIndex [0 1 2 3 4]
    solid FALSE}}
    
```

In this example, we use point field to provide a vertex position list for five calculated space vertex coordinates for forming a pentagon plane object. Then employ

coordIndex field to install a polygonal line path coordinate index list of this pentagon flat face. When rendering, VRML explorer will automatically connect these polygonal lines, from end to end, to form a space face model, also rendered it. Its scene rendering effect is shown in Fig.3.



Figure 3. A pentagon plane object built by IndexedFaceSet node

For this scene, user can employ  button rotating pentagon model to its reverse side. Both sides are seeable. Try to change solid field's value from FALSE to TRUE. Can you see both the two sides? This means solid field's function.

*B. A Solid Regular Tetrahedron - Color Rendering Based on Vertex*

```
#VRML V2.0 utf8
Shape {
  geometry IndexedFaceSet {
    coord Coordinate {
      point [ 0.0 3.266 0.0
             -2.0 0.0 1.155
             2.0 0.0 1.155
             0.0 0.0 -2.309]
    coordIndex [ 0 1 2 -1
                0 2 3 -1
                0 3 1 -1
                3 2 1 -1]
    colorPerVertex TRUE
    color Color {
      color [ 1 0 0
             0 1 0
             0 0 1
             1 1 0]
    }}}}
```

In this example, we use point field to provide a vertex position list for four calculated space vertex coordinates for forming a regular tetrahedron. Then employ coordIndex field to install four piece of regular triangle polygonal line path coordinate index list. It has a -1 ending mark for each flat plane's coordinate index. When rendering, VRML explorer will automatically first connect each polygonal lines, from end to end, to form a space regular triangle plane, then group these four planes into a space regular tetrahedron model. Due to the TRUE value of colorPerVertex field, the surface parts among each vertex are rendered color gradual change and transition effect.



Figure 4. A regular tetrahedron model with based upon vertex color rendering

Now we try to modify coordIndex field's value to below.

```
coordIndex [ 0 1 2 -1
            0 2 3 -1
            0 3 1 -1
            1 2 3 -1]
```

In the scene, we rotate the built model by using  button. Pay attention to observe this regular tetrahedron's bottom. Here we will understand ccw and solid fields' meaning and function. Because solid's default value is TRUE, all faces in face union are constructed a solid entity. In this moment, observe can only see the face model in the opposite direction of unit normal vector. Explorer only renders model's outside surface, not rendering its inner side surface.

Try to change solid's value from its default TRUE to FALSE, further learn ccw and solid fields' significance. When solid's value is FALSE, all faces in face union are no longer constructed a solid entity. In this time, observe will see the rendered model not only in the opposite direction, but also in the direction of unit normal vector. That means explorer will simultaneously render both the outside surface and inner side surface.

In above color rendering based on vertex, different faces around a same vertex angle are rendered the same

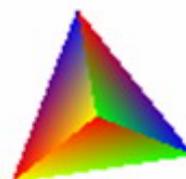


Figure 5. A regular tetrahedron model, with different color effect for different surfaces around a same vertex

color. Some times we need to execute more special color render, to render different plane surfaces around a same vertex with different color set up. Now we need apply colorIndex field, in matching along with coordIndex field, to specifically install different surface's relative vertex with different color render. See following application.

```

#VRML V2.0 utf8
Shape {
  geometry IndexedFaceSet {
    coord Coordinate {
      point [ 0.0 3.266 0.0
             -2.0 0.0 1.155
             2.0 0.0 1.155
             0.0 0.0 -2.309]}
    coordIndex [ 0 1 2 -1
                0 2 3 -1
                0 3 1 -1
                3 2 1 -1]

    colorPerVertex TRUE
    color Color {
      color [ 1 0 0
             0 1 0
             0 0 1
             1 1 0]}

    colorIndex [
      0 1 2 -1
      1 3 2 -1
      2 0 3 -1
      3 1 0 -1]}
  }
}

```

This is a color rendering example, based upon vertex. Although the color number in color field of Color node meets the needs of coloring render based on vertex, we yet are able to apply colorIndex field, in matching along with coordIndex field, to specifically install different surface's relative vertex with different color render. Its effect is shown in Fig.5.

#### C. A Regular Tetrahedron - Color Rendering Based on Face

```

#VRML V2.0 utf8
Shape {
  geometry IndexedFaceSet {
    coord Coordinate {
      point [ 0.0 3.266 0.0
             -2.0 0.0 1.155
             2.0 0.0 1.155
             0.0 0.0 -2.309]}
    coordIndex [ 0 1 2 -1
                0 2 3 -1
                0 3 1 -1
                3 2 1 -1]

    colorPerVertex FALSE
    color Color {
      color [ 1 0 0
             0 1 0
             0 0 1
             1 1 0]}
  }
}

```

In following effect, we simply change colorPerVertex field's value from TRUE to FALSE. This determines to execute color render based upon face. Here, each plane surface is colored with single color, none color gradual change and transition effect. The building scene can be referred to following Fig.6 [3].

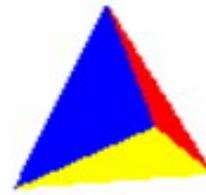


Figure 6. A regular tetrahedron with based upon plane surface color rendering

#### D. Advance Application-Ancient Fan and Special Texture Render

Following is an ancient Chinese fan model and special texture render effect.

```

#VRML V2.0 utf8
Shape {
  appearance Appearance {
    texture ImageTexture {
      url "cp.jpg"}
    geometry IndexedFaceSet {
      coord Coordinate {
        point [.....]}
      coordIndex [
        0 1 2 3 4 5 6 7 8 9
        10 11 12 13 14 15 16 17]
      solid FALSE}}
}

```

This model's most significant feature is the application of picture texture in texture field of Appearance node. Here we directly apply texture image to render a fan model created by IndexedFaceSet node. Of course each vertex's space location of the fan should be calculated first. The built ancient fan object is referred by following Fig.7. By using control panel's  button turning or rotating this model at one's will, on two sides of the model we can see fan's effect rendered by picture texture.



Figure 7. An ancient Chinese fan model

Rewrite above program as following.

```

#VRML V2.0 utf8
Shape {
  appearance Appearance {
    texture ImageTexture {

```

```

        url "cp.jpg" }}
    geometry IndexedFaceSet {
        coord Coordinate {
            point [.....]}
        coordIndex [
            0 1 2 3 4 5 6 7 8 9
            10 11 12 13 14 15 16 17 ]
        texCoord TextureCoordinate {
            point [ 0.3 0.3
                0.6 0.3
                0.6 0.6
                0.3 0.6]}
        texCoordIndex [
            0 1 0 1 0 1 0 1 0 3 2 3 2 3 2 3
2 3]
        solid FALSE}}
    
```

For this application, we not only executed picture texture rendering technique, but also applied texCoord field in IndexedFaceSet node. Bu using texCoord field, we can choose necessary picture texture's two dimensional area for texture application, particularly applied TextureCoordinate node's point field. Then we use texCoordIndex field, in matching with coordIndex field, to implement exceptional texture rendering. Its special effect is shown below.



Figure 8. Special image texture render

*E. Advance Application-Unit Normal Vector Application*

```

#VRML V2.0 utf8
Shape {
    appearance Appearance {
        material Material {
            diffuseColor 1.0 0.5 0.0}}
    geometry IndexedFaceSet {
        coord Coordinate {
            point [ -2.0 -2.0 0.0
                2.0 -2.0 0.0
                2.0 2.0 0.0
                -2.0 2.0 0.0]}
        coordIndex [0 1 2 3]
        normal Normal {
            vector [ 1.0 0.0 0.1
                0.1 0.0 1.0]}
        normalIndex [0 1 0 1]
        solid FALSE}}
    
```

In this application, we apply IndexedFaceSet node built a square plane face model, specially its unit normal

vector is the positive direction of Z axis. In above code, we apply normalIndex field to index vector field set unit normal vector in Normal node, in corresponding with coordIndex field's coordinate index. In this way we changed four vertexes' unit normal vector direction of this square plane face, for the reason that normalPerVertex field's default value is TRUE. So we acquire a kind of quite extraordinary rendering effect. See Fig.9.

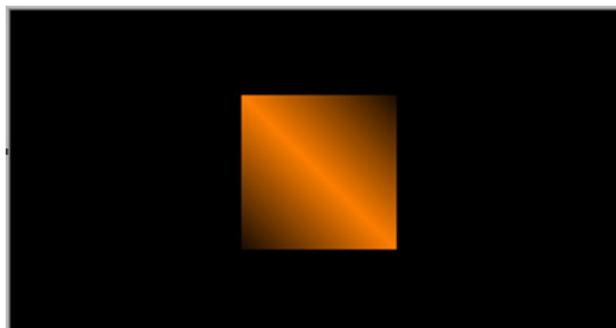


Figure 9. Light and shade affection for vertex's normal vector

Try to modify normalIndex field's value as following.  
 normalIndex [ 0 0 1 1]

We will get different scene light ray variation effect. Then change normalPerVertex field's value as FALSE and also change unit normal vector direction through vector field of Normal node, one can have in-depth knowledge about plane face normal vector.

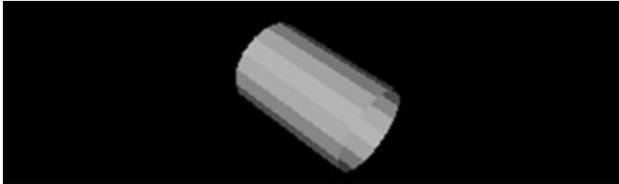
*F. Advance Application-Fold Angle Application*

```

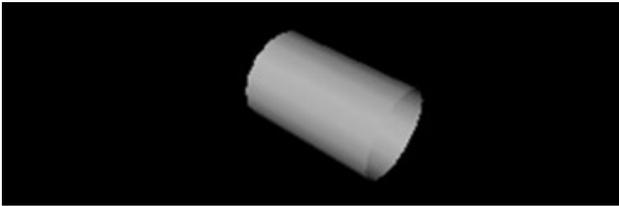
#VRML V2.0 utf8
Shape {
    geometry IndexedFaceSet {
        coord Coordinate {
            point [ -1.5 0.383 0.924
                -1.5 0.707 0.707
                -1.5 0.924 0.383
                -1.5 1.0 0.0
                -1.5 0.924 -0.383
                -1.5 0.707 -0.707
                -1.5 0.383 -0.924
                -1.5 0.0 -1.0
                .....]}
        coordIndex [
            0 16 17 1 -1
            1 17 18 2 -1
            2 18 19 3 -1
            3 19 20 4 -1
            4 20 21 5 -1
            5 21 22 6 -1
            6 22 23 7 -1
            .....]
        solid FALSE}}
    
```

In this application, we apply IndexedFaceSet node to build a roller model for industry usage. The modeling and rendering effect is shown in Fig.10. In Fig.10 (a), we can clearly distinguish every contact edge between any two

component surfaces. Under this basis, we append `creaseAngle` field to above program code, and also set up its field value as 0.4. Once more rendering, the effect is shown in Fig.10 (b). For the reason that the included angle between any two adjacent surfaces' unit normal vector is small than this 0.4 threshold value, when



(a) distinguishable contact edge



(b) vague handled rendering effect for indistinguishable contact edge

Figure 10. Fold angle application

rendering the contact edge of two surfaces will be smoothness drawn. Finally we obtain vague and smoothness process effect for this industry roller model.

#### IV. CONCLUSION

In generally saying, VRML provides some elementary nodes for building regular scene. But they are quite insufficient when constructing complicated object in nature. So we have other kind of modeling technique by adopting the most basic geometry point, line and face manner, as well as terrain elevation manner and industry extrusion manner.

For any complex natural modeling object and scene, we can use VRML provided `PointSet`, `IndexedLineSet`, `IndexedFaceSet`, `ElevationGrid` and `Extrusion` node, as `Shape` modeling node's geometry field value, to construct.

#### REFERENCES

- [1] ISO/IEC 14772-1:1997, VRML97 International Standard [EB/OL], The VRML Consortium, <http://www.vrml.org>.
- [2] Huang Tiejun, Liu Jian, VRML International Standard and Application guide [M], Beijing: Electronic Engineering Publishing Company, 1999
- [3] Duan Xinyu, Virtual Reality Foundation and VRML Program [M], Beijing: Higher Education Press, 2004



**Duan Xinyu**, born in Quanzhou city, Fujian province, China, in 1963. Graduated from Physics Department of Zhengzhou University in Henan province, China, and earned his bachelor's degree in Radio Technology in 1985. During 1990-1993, studied in Bio-engineering Department, Huazhong University of Science and Technology, Hubei province, China, and earned his master's degree in 1993, majoring in Biomedical Engineering.

Now, he works as a dean at Public Computer Teaching Section of Anyang Normal University, Anyang, Henan province, China. He is now an academic and technical senior scholar of Anyang Normal University in education technology domain. His research interests are multimedia technology and virtual reality technology application on education field. In these areas, he has published 4 books, one of them is *Virtual Reality Foundation and VRML Program* (Beijing, China: Higher Education Press, 2004). Up to now, sixties of research papers have been issued; tens of them be indexed by EI or ISTP. Two of them: *Development and Application of Virtual Avatar Interactive Technology*, Proceedings of ICACTE 2008, pp.286-289, 2008 and *Three Dimension Human Body Format and Its Virtual Avatar Animation Application*, Proceedings of IITA 2008, pp.1016-1019, 2008.

Professor Duan is now the IITA senior member and the member of International Association of Computer Science and Information Technology (IACSIT).