

Updates on Projective Texture Mapping

ISO/IEC JTC1 SC24 Meeting

2017. 08. 08

Kwan-Hee Yoo

Chungbuk National University, Korea

Contents



I

Brief Discussion

II

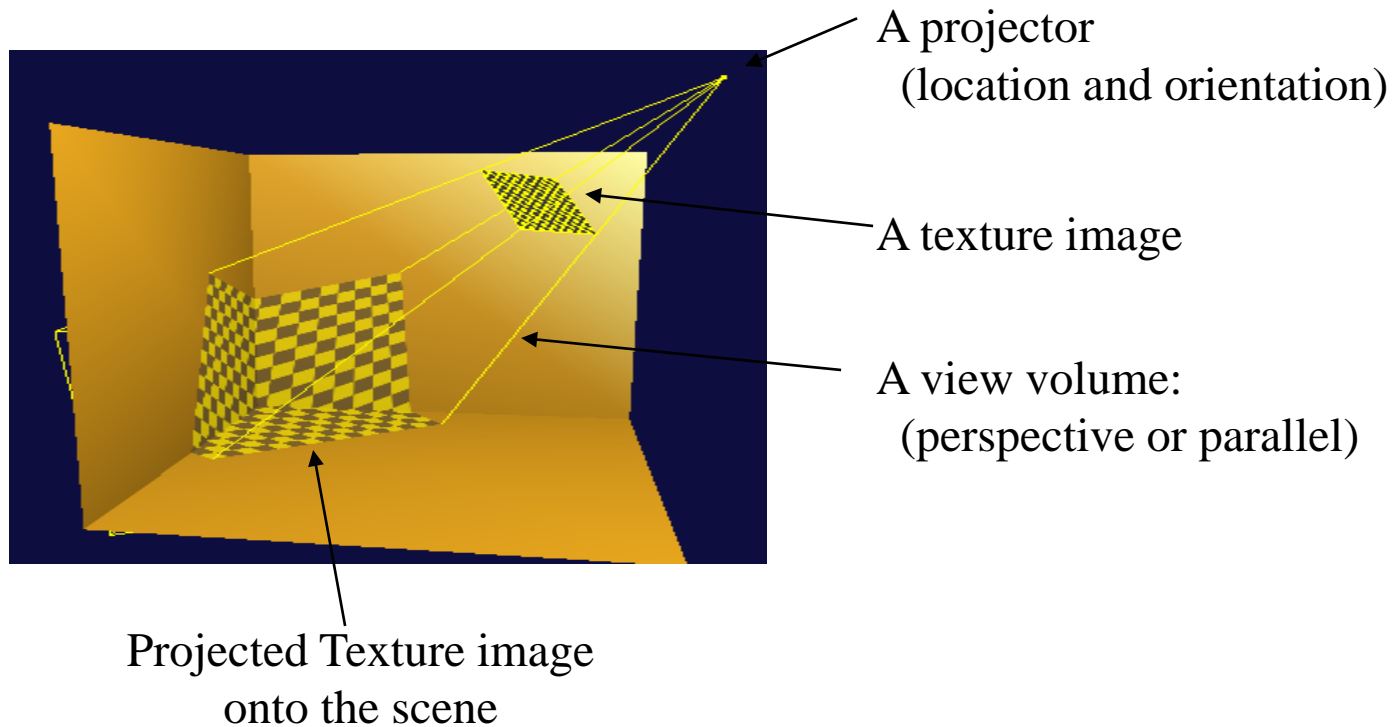
Proposed Nodes

III

Example

I. On Projective Texture Mapping

- A method for texture mapping which allows the texture image to be projected onto the scene as if by a slide projector[Cass Everitt, 1999)



[DEMO](#)

I. History

- The projective texture mapping was proposed as standardization item into X3D at SC24 WC6 Meeting, 2008

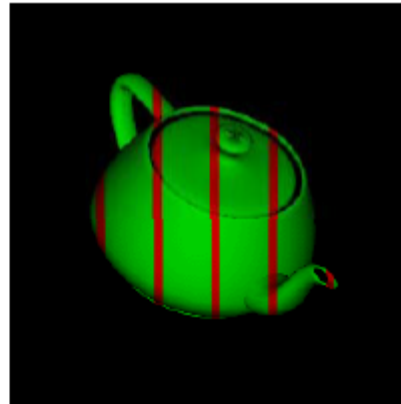
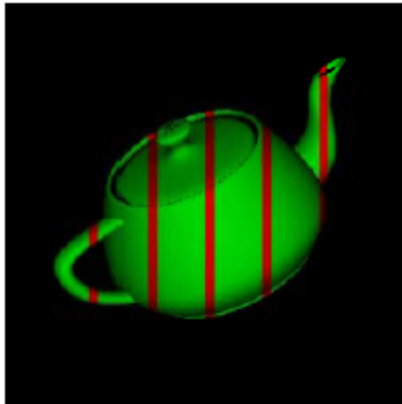
- * **Any nodes for projective texture mapping are not specified into X3D yet**

- For last four years, several sample examples for projective texture mapping have been developed.

- Current, implementation results of projective texture mapping have transferred into X3D browsers such as freeWRL, Xj3D etc.

I. Projective Texture theory

- Eye Linear
 - Texture is “fixed” in eye space



I. Projective Texture theory

- Eye Linear

$$\begin{bmatrix} s \\ t \\ r \\ q \end{bmatrix} = \mathbf{T}_e \begin{bmatrix} v_x \\ v_y \\ v_z \\ v_w \end{bmatrix}_{eye}$$

Eye Linear Texgen Transform

– multiply Eye space coordinate and T_e (Texgen matrix)

I. Projective Texture theory

● Eye Linear Texgen Transform

glTexGen automatically applies this when modelview matrix Contains just the eye view transform

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix} = \begin{bmatrix} \text{Eye view (look at) matrix} \\ \text{Modeling matrix} \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \\ w_o \end{bmatrix} \quad \text{Eye space coordinate}$$

$$\begin{bmatrix} s \\ t \\ r \\ q \end{bmatrix} = \underbrace{\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \\ 1 \end{bmatrix} \begin{bmatrix} \text{frustum (projection) matrix} \\ \text{view (look at) matrix} \end{bmatrix} \begin{bmatrix} \text{Inverse eye view (look at) matrix} \end{bmatrix}}_{\text{Te(TexGen matrix)}} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

II. X3DTextureProjectorNode abstract types and **Nodes**

X3DTextureProjectorNode

X3DTextureProjectorNode : X3DChildNode

TextureProjectorPerspective: X3DTextureProjectorNode

TextureProjectorParallel : X3DTextureProjectorNode

Two Implementations: FreeWRL, X3DOM

II. Proposed Nodes

X3DTextureProjectorNode

```
X3DTextureProjectorNode : X3DChildNode{
  SFNode      [in,out]      metadata          NULL
                                     [X3DMetadataObject]
  SFString    [in,out]      description         ""
  SFVec3f     [in,out]      location                0 0 1 (-∞,∞)
  SFVec3f     [in,out]      direction                0 0 1 (-∞,∞)
  SFFloat     [out]         aspectRatio
  SFFloat     [in,out]      nearDistance            1
  SFFloat     [in,out]      farDistance             10
  SFBool      [in,out]      global                  true
  SFBool      [in,out]      on                       true
  SFNode      [in,out]      texture NULL
                                     [X3DTexture2DNode]
}
```

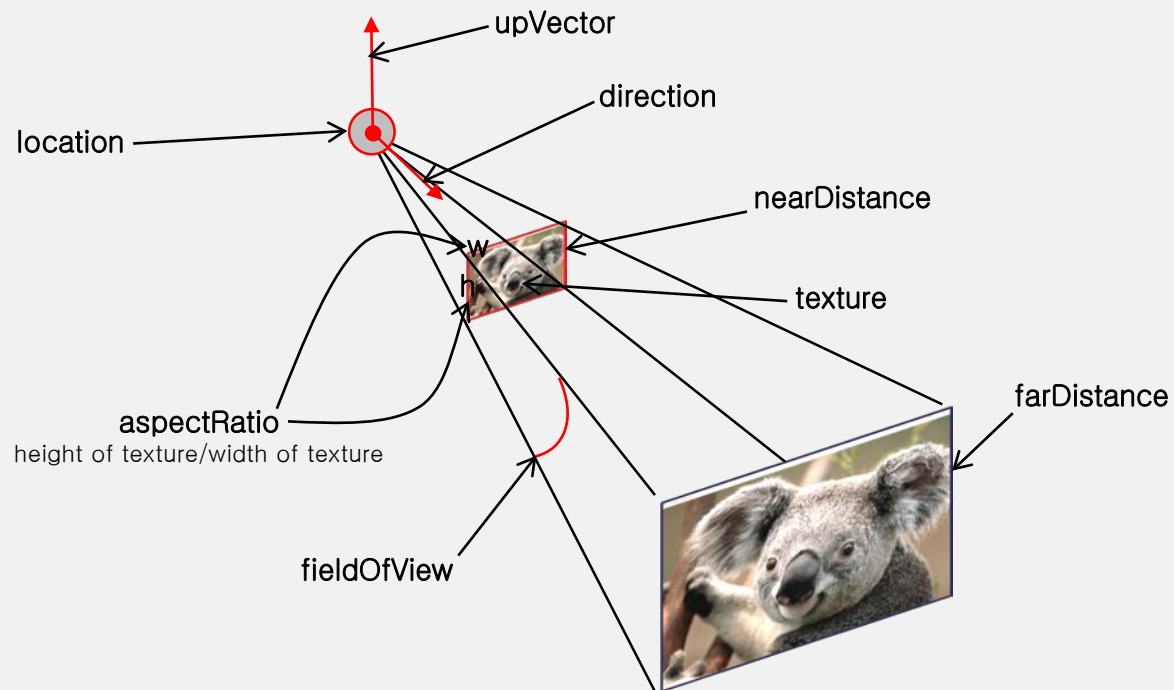
II. X3DTexture abstract types and nodes

TextureProjectorPerspective

```
TextureProjectorPerspective : X3DTextureProjectorNode {  
    SFNode      [in,out] metadata      NULL  
                  [X3DMetadataObject]  
    SFString    [in,out] description   ""  
    SFVec3f     [in,out] location      0 0 1  
    SFvec3f     [in,out] direction     0 0 1  
    SFFloat     [in,out] fieldOfView    $\pi/4$  (0, $\pi$ )  
    SFFloat     [out]   aspectRatio  
    SFVec3f     [in,out] upVector      0 0 1  
    SFFloat     [in,out] nearDistance  1  
    SFFloat     [in,out] farDistance   10  
    SFBool      [in,out] global        true  
    SFBool      [in,out] on            true  
    SFNode      [in,out] texture      NULL  
                  [X3DTexture2DNode]  
}
```

II. Proposed Nodes

TextureProjectorPerspective



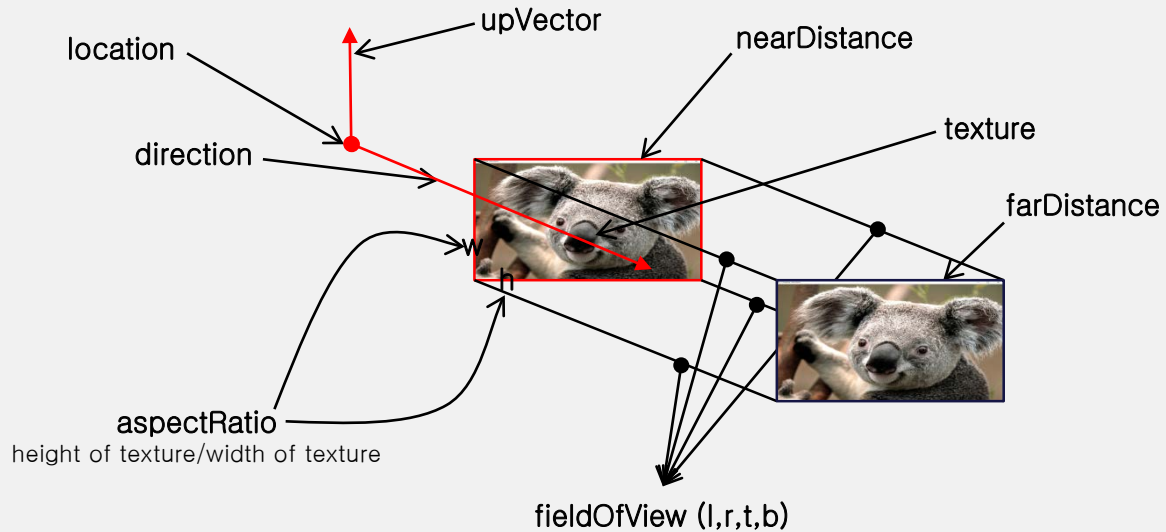
II. Proposed Nodes

TextureProjectorParallel

```
TextureProjectorParallel : X3DTextureProjectorNode{
    SFNode      [in,out] metadata      NULL
                [X3DMetadataObject]
    SFString    [in,out] description  ""
    SFVec3f     [in,out] location      0 0 1
    SFVec3f     [in,out] direction     0 0 1
    SFFloat     [out]   aspectRatio
    MFFloat     [in,out] fieldOfView   -1 -1 1 1 (-∞,∞)
    SFFloat     [in,out] nearDistance  1
    SFFloat     [in,out] farDistance   10
    SFBool      [in,out] global        true
    SFBool      [in,out] on            true
    SFNode      [in,out] texture      NULL
                [X3DTexture2DNode]
}
```

II. Proposed Nodes

TextureProjectorParallel



II. Implementation

Two Implementations

FreeWRL

X3DOM

Implementation PTM by using FreeWRL

- Register Three Nodes for PTM
- Generate parsing program automatically (C++)
- Parsing PTM files
- Shading Program for rendering PTM files



FreeWRL

```
X3DTextureProjectorNode => new VRML::NodeType("X3DTextureProjectorNode", {
    addChildren => [MFNode, undef, inputOnly, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31
| SPEC_X3D32 | SPEC_X3D33)",
    removeChildren => [MFNode, undef, inputOnly, "(SPEC_VRML | SPEC_X3D30 |
SPEC_X3D31 | SPEC_X3D32 | SPEC_X3D33)",
    children => [MFNode, [], inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)",
    metadata => [SFNode, NULL, inputOutput, "(SPEC_X3D30 | SPEC_X3D31 | SPEC_X3D32
| SPEC_X3D33)",
location=> [SFVec3f, [0, 0, 1], inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 | SPEC_X3D32 |
SPEC_X3D33)",
    direction => [SFVec3f, [0, 0, 1], inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31
| SPEC_X3D32 | SPEC_X3D33)",
    aspectRatio => [SFFloat, 1, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)",
    nearDistance => [SFFloat, 1, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)",
    farDistance => [SFFloat, 10, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)",
global => [SFBool, FALSE, inputOutput, "(SPEC_X3D31 | SPEC_X3D32 | SPEC_X3D33)",
    on => [SFBool, FALSE, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)",
    texture=>[SFNode,NULL,inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)",

}, "X3DGroupingNode"),
```


FreeWRL

```
TextureProjectorPerspective => new VRML::NodeType("TextureProjectorPerspective",{
    metadata => [SFNode, NULL, inputOutput, "(SPEC_X3D30 | SPEC_X3D31 | SPEC_X3D32
| SPEC_X3D33)],
    description => [SFString, "", inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    location=> [SFVec3f, [0, 0, 1], inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    direction => [SFVec3f, [0, 0, 1], inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31
| SPEC_X3D32 | SPEC_X3D33)],
    upVector => [SFVec3f, [0, 1, 0], inputOutput, "(SPEC_VRML | SPEC_X3D30 |
SPEC_X3D31 | SPEC_X3D32 | SPEC_X3D33)],
    fieldOfView => [SFFloat, 45, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    aspectRatio => [SFFloat, 1, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    nearDistance => [SFFloat, 1, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    farDistance => [SFFloat, 10, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    global => [SFBool, FALSE, inputOutput, "(SPEC_X3D31 | SPEC_X3D32 | SPEC_X3D33)],
    on => [SFBool, FALSE, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    texture=>[SFNode,NULL,inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
}, "X3DGroupingNode"),
```

FreeWRL

```
TextureProjectorParallel => new VRML::NodeType("TextureProjectorParallel",{
    metadata => [SFNode, NULL, inputOutput, "(SPEC_X3D30 | SPEC_X3D31 | SPEC_X3D32
| SPEC_X3D33)],
    description => [SFString, "", inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    location=> [SFVec3f, [0, 0, 1], inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    direction => [SFVec3f, [0, 0, 1], inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31
| SPEC_X3D32 | SPEC_X3D33)],
    upVector => [SFVec3f, [0, 1, 0], inputOutput, "(SPEC_VRML | SPEC_X3D30 |
SPEC_X3D31 | SPEC_X3D32 | SPEC_X3D33)],
    fieldOfView => [SFColorRGBA, [-1, 1, -1, 1], inputOutput, "(SPEC_VRML | SPEC_X3D30 |
SPEC_X3D31 | SPEC_X3D32 | SPEC_X3D33)],
    aspectRatio => [SFFloat, 1, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    nearDistance => [SFFloat, 1, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    farDistance => [SFFloat, 10, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    global => [SFBool, FALSE, inputOutput, "(SPEC_X3D31 | SPEC_X3D32 | SPEC_X3D33)],
    on => [SFBool, FALSE, inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
    texture=>[SFNode,NULL,inputOutput, "(SPEC_VRML | SPEC_X3D30 | SPEC_X3D31 |
SPEC_X3D32 | SPEC_X3D33)],
}, "X3DGroupingNode"),
```

Example

- TextureProjectorPerspective – Example(Apple)

```
<X3D profile="Interactive" version="3.3"
<Scene>
```

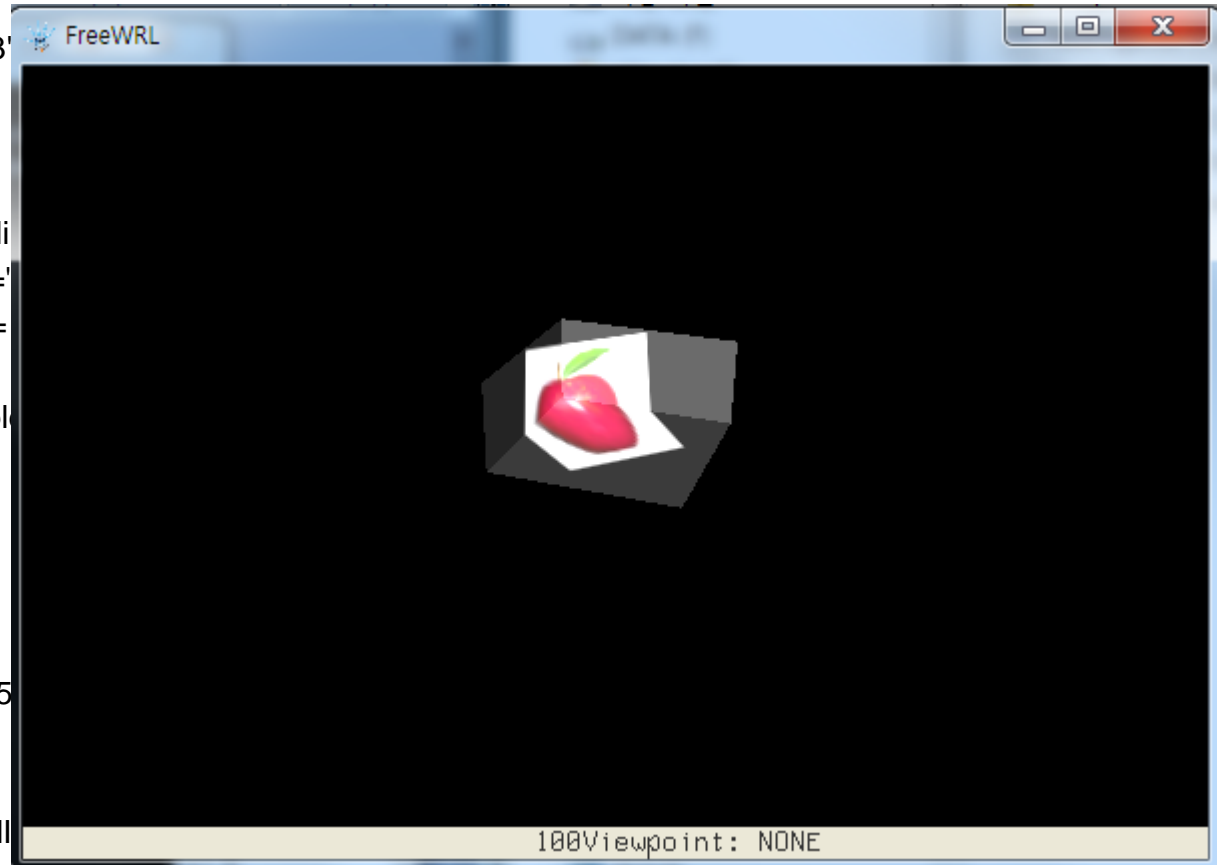
```
<TextureProjectorPerspective
  description='pt1' location='3 3 3' di
  fieldOfView= '0.26' nearDistance='
  upVector='0 1 0' global= 'true' on=
```

```
  <ImageTexture url='C:/image/appl
</TextureProjectorPerspective>
```

```
<Shape>
  <Appearance>
    <Material diffuseColor='0.5 0.5 0.5
  </Appearance>
```

```
  <IndexedFaceSet solid='false' coordl
    <Coordinate point="1 0 1, -1 0 1, -1 0 -1, 1 0 -1, 1 1 -1, -1 1 -1, -1 1 1 "/>
  </IndexedFaceSet>
```

```
</Shape>
</Scene>
</X3D>
```



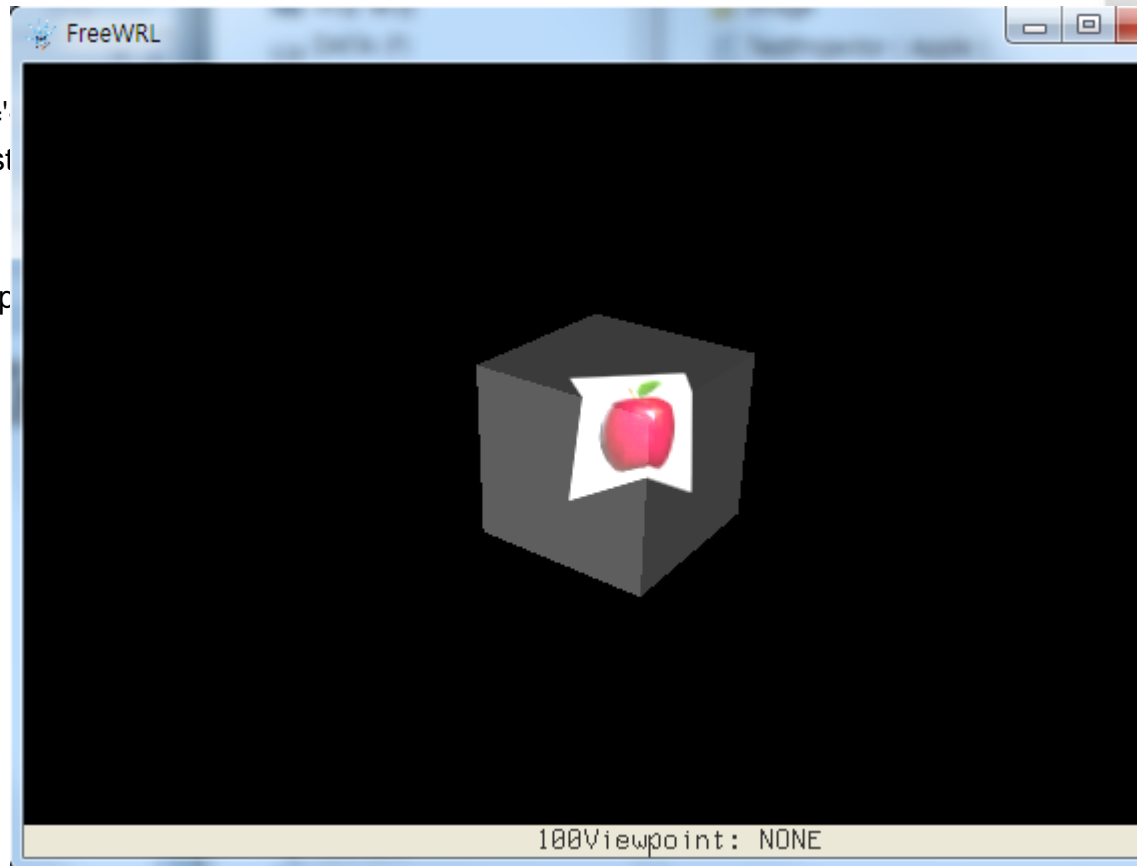
Example

- TextureProjectorPerspective – Example(Box)

```
<X3D profile="Interactive" version="3.3">
<Scene>
<TextureProjectorPerspective
  description='pt1' location='3 3 3' direction='
  fieldOfView= '0.26' nearDistance='1' farDist
  upVector='0 1 0' global= 'true' on= 'true'>

  <ImageTexture url='C:/image/apple.jpg' rep
</TextureProjectorPerspective>

<Shape>
  <Appearance>
    <Material diffuseColor='0.5 0.5 0.5'>
  </Appearance>
  <Box/>
</Shape>
</Scene>
</X3D>
```



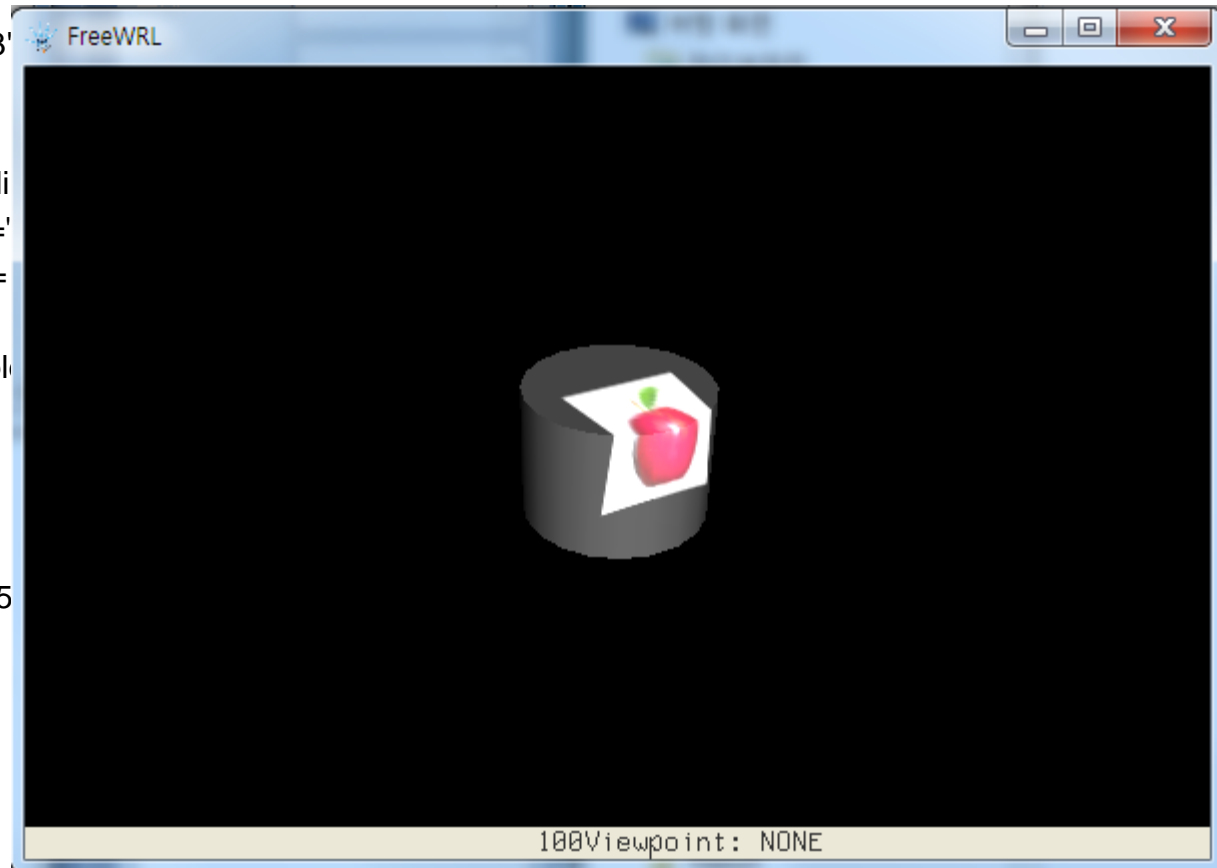
Example

- TextureProjectorPerspective – Example(Cylinder)

```
<X3D profile="Interactive" version="3.3"
<Scene>
<TextureProjectorPerspective
  description='pt1' location='3 3 3' di
  fieldOfView= '0.26' nearDistance='
  upVector='0 1 0' global= 'true' on=

  <ImageTexture url='C:/image/appl
</TextureProjectorPerspective>

<Shape>
  <Appearance>
    <Material diffuseColor='0.5 0.5 0.5
  </Appearance>
  <Cylinder radius='1' height='1.5'/>
</Shape>
</Scene>
</X3D>
```



Example

- TextureProjectorPerspective – Example(Face + Sphere)

```
<X3D profile="Interactive" version="3.3">
  <Scene>
    <TextureProjectorPerspective
      description='pt1' location='3 3 3' direction='
      fieldOfView= ' 0.26' nearDistance='1' farDis
      upVector='0 1 0' global='true' on= 'true'>

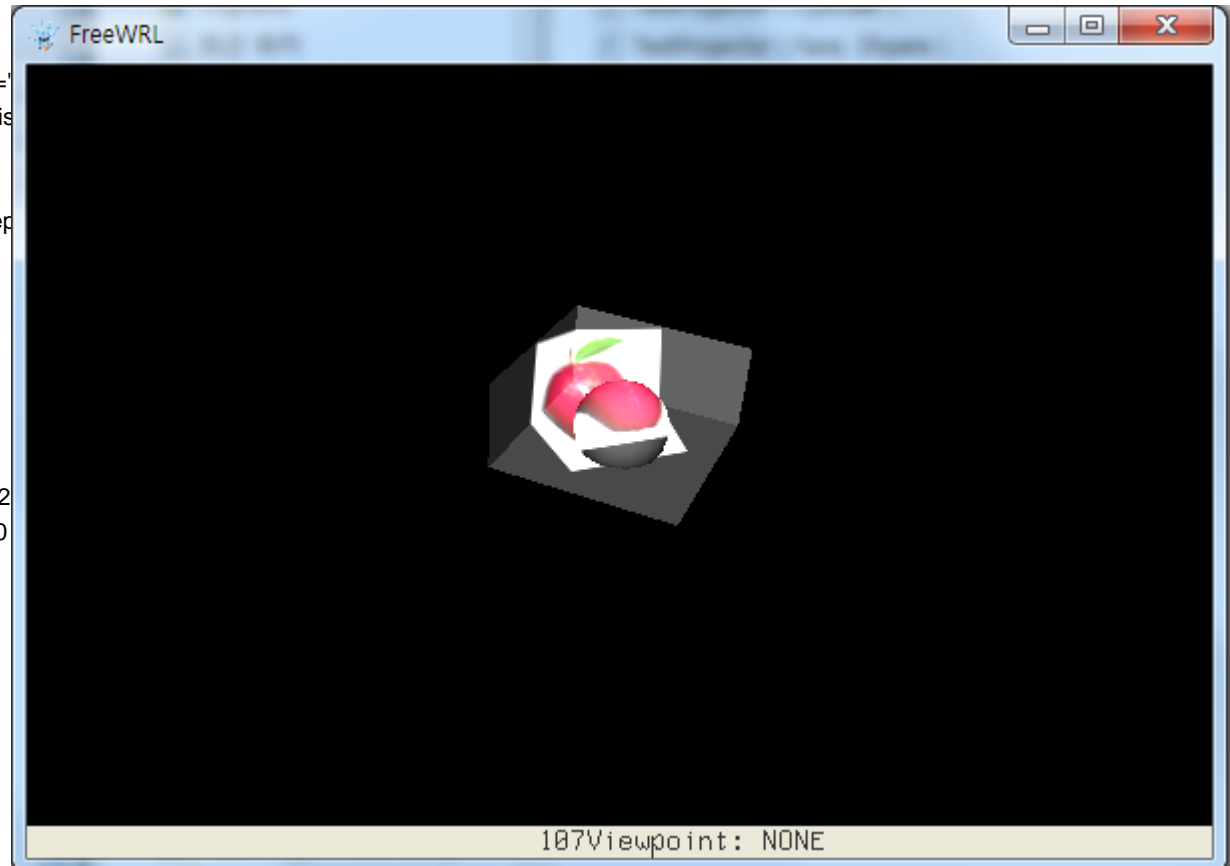
      <ImageTexture url='C:/image/apple.jpg' rep
    </TextureProjectorPerspective>

    <Shape>
      <Appearance>
        <Material diffuseColor='0.5 0.5 0.5'>
      </Appearance>

      <IndexedFaceSet solid='false' coordIndex="3 2
        <Coordinate point="1 0 1, -1 0 1, -1 0 -1, 1 0
      </IndexedFaceSet>
    </Shape>

    <Transform translation='0,0.25,0'>
      <Shape>
        <Appearance>
          <Material diffuseColor='0.5 0.5 0.5'>
        </Appearance>

        <Sphere radius = '0.5'>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```



```

<X3D profile="Interactive" version="3.3">
<Scene>
<TextureProjectorParallel
  description='pt1' location='3 5 3' direction='0 1 1'
  fieldOfView='0 1 -0 1' nearDistance='1' farDistance='10'
  global= 'true' on= 'true'>

```

```

  <ImageTexture url='C:/image/apple.jpg' repeatS

```

```

</TextureProjectorParallel>

```

```

<Shape>
  <Appearance>
    <Material diffuseColor='0.5 0.5 0.5'/>
  </Appearance>
  <IndexedFaceSet solid='false' coordIndex="3 2 1 0
    <Coordinate point="1 0 1, -1 0 1, -1 0 -1, 1 0 -1, 1
  </IndexedFaceSet>

```

```

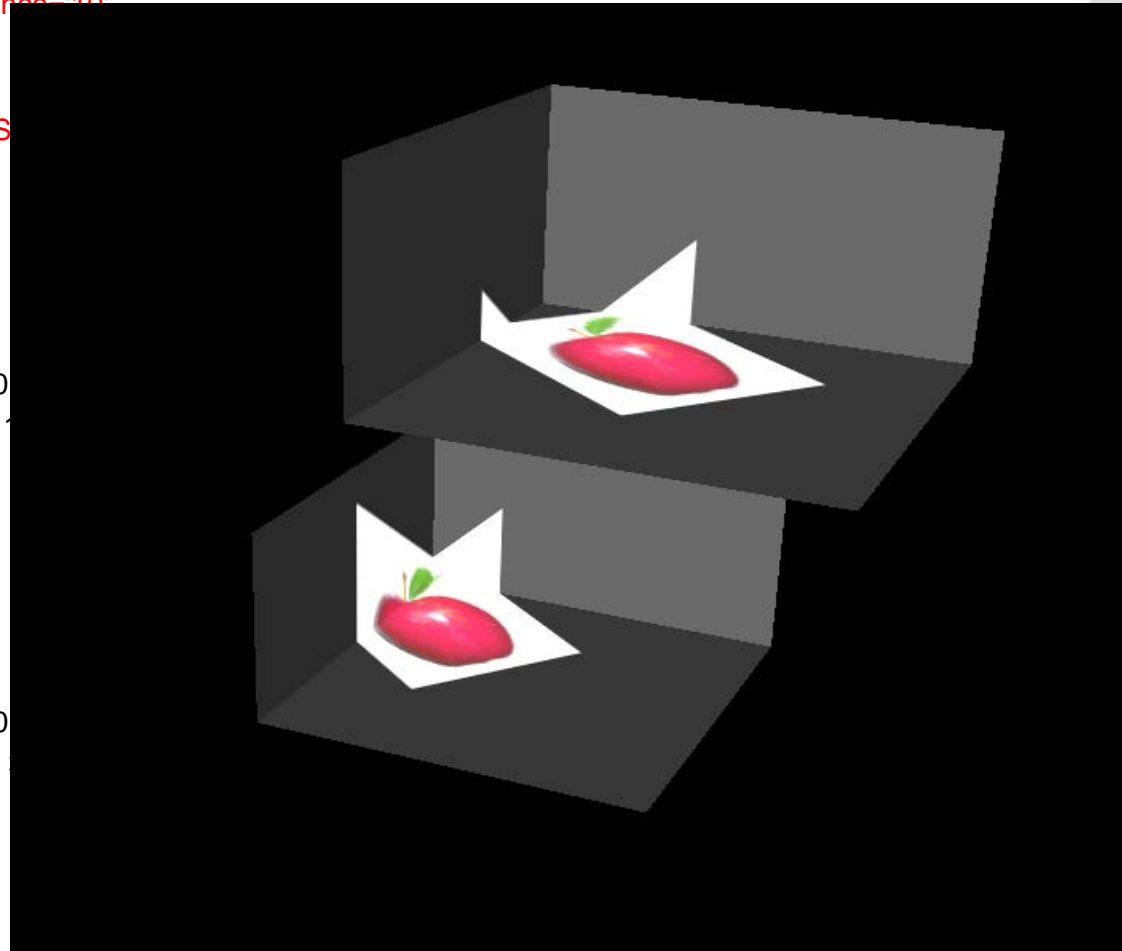
</Shape>
<Shape>
  <Appearance>
    <Material diffuseColor='0.5 0.5 0.5'/>
  </Appearance>
  <IndexedFaceSet solid='false' coordIndex="3 2 1 0
    <Coordinate point="2 2 2, 0 2 2, -0 2 -0, 2 2 0, 2
  </IndexedFaceSet>

```

```

</Shape>
</Scene>
</X3D>

```



Implementation of PTM by using X3DOM

- Register Three Nodes for PTM
- Connection of defined PTM nodes and Scene Graph (Java Script)
- Parsing PTM files
- Shading Program for rendering PTM files

(Until Oct. 2017)



X3DTextureProjectorNode for X3DOM

```
x3dom.registerNodeType (  
  "X3DTextureProjectorNode",  
  "X3DChildNode",  
  defineClass (x3dom.nodeTypes.X3DChildNode,  
    function (ctx) {  
      x3dom.nodeTypes.X3DVNode.superClass.call (this, ctx);  
      this.addField_SFNode (ctx, 'metadata', null);  
      this.addField_SFString (ctx, 'description', '');  
      this.addField_SFVec3f (ctx, 'location', '0 0 1');  
      this.addField_SFVec3f (ctx, 'direction', '0 0 1');  
      this.addField_SFFloat (ctx, 'aspectRatio', '0');  
      this.addField_SFFloat (ctx, 'nearDistance', '1');  
      this.addField_SFFloat (ctx, 'farDistance', '10');  
      this.addField_SFBool (ctx, 'global', true);  
      this.addField_SFBool (ctx, 'on', true);  
      this.addField_SFNode (ctx, 'texture', null);  
    }  
  )  
);
```



TextureProjectorPerspective for X3DOM

```
x3dom.registerNodeType (  
  "TextureProjectorPerspective",  
  "X3DTextureProjectorNode",  
  defineClass(x3dom.nodeTypes.X3DTextureProjectorNode,  
    function(ctx) {  
      x3dom.nodeTypes.TextureProjectorPerspective.superClass.call(this, ctx);  
      this.addField_SFNode(ctx, 'metadata', null);  
      this.addField_SFString(ctx, 'description', '');  
      this.addField_SFVec3f(ctx, 'location', '0 0 1');  
      this.addField_SFVec3f(ctx, 'direction', '0 0 1');  
      this.addField_SFFloat(ctx, 'fieldofview', 3.14/4);  
      this.addField_SFFloat(ctx, 'aspectRatio');  
      this.addField_SFVec3f(ctx, 'upVector', '0 0 1');  
      this.addField_SFFloat(ctx, 'nearDistance', 1);  
      this.addField_SFFloat(ctx, 'farDistance', 10);  
      this.addField_SFBool(ctx, 'global', true);  
      this.addField_SFBool(ctx, 'on', true);  
      this.addField_SFNode(ctx, 'texture', null);  
    }  
  )  
);
```



TextureProjectorParallel for X3DOM

```
x3dom.registerNodeType (  
  "TextureProjectorParallel",  
  "X3DTextureProjectorNode",  
  defineClass(x3dom.nodeTypes.X3DTextureProjectorNode,  
    function (ctx) {  
      x3dom.nodeTypes.TextureProjectorParallel.superClass.call(this, ctx);  
      this.addField_SFNode (ctx, 'metadata', null);  
      this.addField_SFString (ctx, 'description', '');  
      this.addField_SFVec3f (ctx, 'location', '0 0 1');  
      this.addField_SFVec3f (ctx, 'direction', '0 0 1');  
      this.addField_SFFloat (ctx, 'fieldofview');  
      this.addField_SFFloat (ctx, 'aspectRatio');  
      this.addField_SFFloat (ctx, 'nearDistance', 1);  
      this.addField_SFFloat (ctx, 'farDistance', 10);  
      this.addField_SFBool (ctx, 'global', true);  
      this.addField_SFBool (ctx, 'on', true);  
      this.addField_SFNode (ctx, 'texture', null);  
    }  
  )|  
);
```



Implementation PTM by using X3DOM

```
x3dom.registerNodeType (  
  "TextureProjectorParallel",  
  "X3DTextureProjectorNode",  
  defineClass(x3dom.nodeTypes.X3DTextureProjectorNode,  
    function (ctx) {  
      x3dom.nodeTypes.TextureProjectorParallel.superClass.call(this, ctx);  
      this.addField_SFNode (ctx, 'metadata', null);  
      this.addField_SFString (ctx, 'description', '');  
      this.addField_SFVec3f (ctx, 'location', '0 0 1');  
      this.addField_SFVec3f (ctx, 'direction', '0 0 1');  
      this.addField_SFFloat (ctx, 'fieldofview');  
      this.addField_SFFloat (ctx, 'aspectRatio');  
      this.addField_SFFloat (ctx, 'nearDistance', 1);  
      this.addField_SFFloat (ctx, 'farDistance', 10);  
      this.addField_SFBool (ctx, 'global', true);  
      this.addField_SFBool (ctx, 'on', true);  
      this.addField_SFNode (ctx, 'texture', null);  
    }  
  )|  
);
```



Thank you

Q&A

khyoo@chungbuk.ac.kr