Introduction to X3D

- Paradigms, Primitives, Coordinates -

Felix G. Hamza-Lup, Ph.D
Associate Professor / Director NEWS Lab
Computer Science and Information Technology
Armstrong State University
Savannah, Georgia, USA
Outline

• Understanding the Camera Model
  • Clipping, Projection (View Frustum), Rasterization

• The Polygon
  • Normal

• X3D
  • What is X3D?
  • Standards
  • Application domains

• Scene Graph Terminology

• Geometric Primitives

• Cartesian Coordinate
Application Programmer’s Interface

- API functionality should match the conceptual model
- Synthetic Camera Model used for APIs like OpenGL, Direct 3D, Java3D, VRML, X3D etc.

Functionality needed in the API to specify:
- Objects
- Viewers
- Light sources
- Material properties
Application Programmer’s Interface

Camera specification in APIs:

- **position** – usually center of lens

- **orientation** – camera coordinate system in center of lens
camera can rotate around those three axis

- **focal length** of lens determines the size of the image on the film actually viewing angle

- **film plane** - camera has a height and a width
Synthetic Camera Model (1)

Computer-generated image based on an optical system – **Synthetic Camera Model**

Viewer behind the camera can move the back of the camera – change the distance $d$ i.e. additional flexibility

Objects and viewer specifications are independent – different functions within a graphics library
Imaging with the Synthetic Camera Model

• film plane position in a camera
• projected scene to the projection plane
Synthetic Camera Model (3) - **Clipping**

Not all objects can be seen - limit due to viewing angle

Solution: *Clipping rectangle or clipping window* placed in front of the camera

(b) shows the case when the clipping rectangle is shifted aside – only part of the scene is projected
What is **Projection**?

Any operation that reduces dimension (e.g., 3D to 2D)

1. Perspective Projection
   - Focal point = the eye
   - Parallel lines may converge at infinity

2. Orthographic Projection
   - focal point at infinity
   - rays are parallel and orthogonal to the image plane
Comparison

Orthographic
- Everything seems equal
- No Vanish-Point
- Parallel lines never touch

Perspective
- Closest things seems bigger
- Has Vanish-Point
- Parallel lines touch at infinity
Projection - Transformation from 3D (world) to 2D (eye/screen)

• From XYZ (world coordinates) to UV (screen coordinates)

• Frustum Culling
Frustum Culling
View Frustum in X3D

Oblique view

X3D ViewFrustum - Example
Clipping, Projection & **Rasterization**

- **Clipping** is used to remove those parts of the world that cannot be seen.

- Objects representation is “kept” in 3D as long as possible. After transformation and clipping must be **projected** to 2D somehow.

- Projected objects or their parts must be displayed – and therefore **rasterized**.

All those steps are performed on your graphics cards in hardware nowadays.
All together: Graphics Pipeline Architectures

There are 4 major steps in the geometric pipeline:

- Transformations – like scaling, rotations, translation, mirroring, sheering etc.
- Clipping – removal of those parts that are out of the viewing field
- Projection – world (xyz) to screen (uv) coordinates
- Rasterization – from points to pixels (analog to digital)
Modeling vs Rendering
Polygonal Models

Dr. Felix Hamza-Lup - Fulbright Lecture, Bangkok University
2015
Modeling vs. Rendering – Understanding the Paradigms

• In many applications the modeling is separated from production of an image – rendering (CAD systems, animations etc.)

• In this case the modeling might be different from the renderer

• The connection between both parts can be simple or highly complex using distributed environments
Application Programmer’s Interface (API)

• Objects are defined by points or vertices, line segments, polygons etc. to represent complex objects

• API primitives are displayed rapidly on the hardware

• Usual API primitives:
  • points
  • line segments
  • **polygons**
  • text
An Example API

OpenGL defines primitives through list of vertices – triangular polygon is drawn by:

```c
glBegin(GL_POLYGON);
    glVertex3f(0.0, 0.0, 0.0);
    glVertex3f(0.0, 1.0, 0.0);
    glVertex3f(0.0, 0.0, 1.0);
    glVertex3f(0.0, 0.0, 1.0);
    glVertex3f(0.0, 0.0, 1.0);
    glEnd();
```
What is a Polygon?
Outline

• Understanding the Camera Model
  • Clipping, Projection (View Frustum), Rasterization

• The Polygon
  • Normal

• X3D
  • What is X3D?
  • Standards
  • Application domains

• Scene Graph Terminology

• Geometric Primitives

• Cartesian Coordinate
What Is X3D?

• X3D:
  • Stands for eXtended 3D
  • Royalty-free, international standard for 3D graphics on the Web
  • Uses a variety of tags and attributes (some resemblance with HTML)

• Use text editors or IDEs to create X3D document
  • We use NotePad++
  • With syntax highlighter
Motivation (1)

• Over 35 years of steady growth and innovation have made 3D graphics an exciting field

• Key professional organization is SIGGRAPH for computer graphics and interactive techniques
  • technical experts
  • artists

• Nevertheless, few people actually build 3D models themselves
  • Usually requires advanced programming skills
  • Costly proprietary tools and approaches compete
Motivation (2)

- Rather than creating another expensive technical niche, X3D is designed for Web interoperability
  - Support capabilities common to most (or all) tools
  - Provide import/export publishing compatibility for many other formats
  - Align 3D with the Architecture of the World Wide Web

- This approach works well for simple 3D models, scaling up to large-scale virtual environments

- Ultimate X3D success means that 3D graphics becomes a “first-class citizen” for Web Multimedia
Why is X3D important?

• There are many types of 3D graphics engines and plugins available. Best known:
  • Computer graphics games
  • Animated movies

• Well-kept secret: these are rarely interoperable

Example:
• no 2 experts can run the other’s demo
• “Silly” question: hey, let's mix 2 games together!
• ... why should adding models together be so hard?

• Proprietary software actively prevents such mergers

• Interoperability over Web can change all that
Why is X3D important?

• Web standards let different companies do what they do well, then interoperate together

• Today there are many small islands of functionality

• Tomorrow might bring a much bigger playing field for 3D graphics to work with

• A shared Web is good for everyone
  • Business, public, government, universities
  • Best practices emerge
  • More information, more connectivity, more progress
  • “A rising tide lifts all boats”
X3D and other Standards

X3D Specification itself is componentized and extensible

X3D File Encodings
.x3d XML Encoding DTD, Schema
.ISO 19776-1

.x3dv Classic VRML Encoding
.ISO 19776-2

.x3db Binary Encoding
.ISO 19776-3

H-Anim
.ISO 19774

DOM Document Object Model
.Recommendations W3C

Scene Access Interface (SAI) scripting API for Java
.ISO 19777-2

Scene Access Interface (SAI) scripting API
.ISO 19777-1

XML Encryption, Authentication
.Recommendations W3C

Programming Language Bindings
Application Domains (1)

- Entertainment Industry
  - Online games (MMRPG), Avatar interaction
  - Virtual Museums, Virtual galleries

- Medical Industry
  - Medical Modeling, Surgical training, Patient education
  - Initial Technical Focus
    - Representation of human anatomy in X3D
    - Association of 2D images (from multiple modalities) with 3D skeletal structure with registration
    - Extension of X3D to accommodate image textures in context of 3D anatomy model
Application Domains (2)

- Architecture and Design Industry
  - City planning, Landscaping

- Manufacturing Industry
  - Production planning automated manufacturing
  - Real-time system simulation of plants, machinery, peripherals and components

- Advertisement Industry
  - see Google Earth + 3D restaurants + cities etc.
Many Case Studies/Examples
– Commercial & Prototypes –

www.web3d.org/casestudies
X3D Technical Overview
X3D Browsers

• X3D browsers/plug-ins
  • parse (read) X3D scene models and
  • render (draw)
  • simulation capabilities for animation and user interaction

• Often implemented as plugins to web browsers:
  • Internet Explorer, Mozilla Firefox, Opera, Safari

• X3D Players can also run as stand alone applications
X3D plugins

• There are many X3D plugins for Web browsers

  • Bitmanagement Contact http://www.bitmanagement.de
  • InstantReality http://instantreality.org
  • Octaga http://www.octaga.com
  • Xj3D http://www.xj3d.org

• Most also operate as a standalone application
  • Either commercial source code or open source
  • Same X3D graphics content runs on each one
Software Architecture Overview

X3D Browser

- X3D scenes, X3D streams
- Event passing with external HTML Web pages or applications

Parsers
- X3D XML encoding
- Classic VRML encoding
- Binary encoding

New node and prototype construction
- X3D nodes, node types
- Prototype and External Prototype
- Scene graph manager

Scene Authoring Interface (SAI)
- Application programmer interfaces

Scripting engines
- EcmaScript
- Java
- Others

Scene Graph Renderable Nodes

Event Graph Animation Nodes

Dr. Felix Hamza-Lup - Fulbright Lecture, Bangkok University 2015
Software Support

• Lots of free plugins, tools and resources provided
  http://www.bitmanagement.com/

• X3D Resources at
  http://www.web3d.org/x3d/content/examples/X3dResources.html

• Steps:
  • Install an X3D plugin into your default Web browser
  • Set up to author X3D scenes using plain-text editor, or else by using an X3D-aware authoring tool
    • X3D-Edit provided free for any use

Dr. Felix Hamza-Lup - Fulbright Lecture, Bangkok University 2015
Examples

• Numerous (thousands) of X3D examples are available online
  http://x3dgraphics.com/examples/X3dResources.html#Examples

• Can browse all examples in X3D for Web Authors
  http://x3dgraphics.com/examples summary
  http://x3dgraphics.com/examples/X3dForWebAuthors archive
  http://x3dgraphics.com/X3dExamplesX3dForWebAuthors.zip

• Recommended approach:
  • Browse examples online
  • Download and edit on local system
X3D-Edit Authoring Tool

• Available free for any use
  • https://savage.nps.edu/X3D-Edit
  • Written using Java, XML and X3D
  • Windows, MacOSX, Linux, Solaris operating systems

• Standalone application with automatic updates available once installed.
X3D-Edit Features

• X3D-Edit features include:
  • direct editing of X3D scenes using the XML (.x3d) encoding,
  • embedded visualization of scenes using the Xj3D viewer
  • XML validation against X3D DTDs
  • and Schemas,
  • drag-and-drop palette for X3D nodes,
  • popup panels for node editing, and
  • extensive help resources
  • classic VRML and X3D compressed binary encoding support
  • encryption and digital-signature authentication using XML security standards.
Bit Contact Authoring Tool
"Hello World" Example

```xml
<X3D xsd:noNamespaceSchemaLocation="http://www.web3d.org/specifications/x3d-3.3.xsd"
xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance" version="3.3" profile="Immersive">
  <head>
    <meta name="title" content="HelloWorld.x3d"/>
  </head>

  <Scene>
    <Viewpoint position="0 0 20"/>
    <Shape>
      <Text maxExtent="16" string="Hello World">
        <FontStyle justify="middle"/>
      </Text>
      <Appearance>
        <Material/>
      </Appearance>
    </Shape>
  </Scene>
</X3D>
```
Outline

• Understanding the Camera Model
  • Clipping, Projection (View Frustum), Rasterization

• The Polygon
  • Normal

• X3D
  • What is X3D?
  • Standards
  • Application domains

• Scene Graph Terminology

• Geometric Primitives

• Cartesian Coordinate
X3D Scene Graph?

- Scene graphs: model-centric approach to 3D that hierarchically defines geometry shape, appearance, position and orientation, etc.
  - Directed acyclic graph (DAG), meaning a tree with a root node and no loops
  - Declarative listing of parameters of interest

- Unlike most imperative programming approaches draw this triangle, that triangle, recompute, etc. the scene graph is traversed => one frame.
Scene graph terminology

• Scene graph data file
  • contains model description, may refer to data files

• Scene graph viewer
  • Reads and renders scene-graph models
  • Implemented as application or web browser plugin

• Scene graph editor
  • Special text editor for scene graph development

• Executable application
  • Specific 3D model capable of running on a specific operating system
Scene graph rendering

• The browser traverses the scene graph, updating any values within nodes and building an image
  • New image then replaces previous screen image, process known as **double buffering**
  • Rapid repetitions are very important
  • Frame rate faster than 7-10 Hz (cycles per second) provides appearance of smooth motion

• **Rendering** is defined as this drawing process

• **Off-line rendering** is performing such operations to image or movie files, rather than display
Other scene graph architectures

- OpenInventor (OI), predecessor of VRML

- Virtual Reality Modeling Language (VRML), direct predecessor of X3D
  - [http://www.web3d.org/x3d/specifications](http://www.web3d.org/x3d/specifications)

- Java3D quite similar to X3D scene graph
  - [https://java3d.dev.java.net](https://java3d.dev.java.net)

- OpenSceneGraph (OSG)
  - [http://www.openscenegraph.org](http://www.openscenegraph.org)

- OpenSG
  - [http://www.opensg.org](http://www.opensg.org)
Geometric Primitives in X3D
Geometric Primitives
Common field: \textit{solid}

- In 3D graphics, all triangles have 2 sides
  - Graphics term: backface culling only draws front sides

- The solid field defines whether a geometry node has an inside or not, with a default value of true
  - solid='true' means do not render (draw) the inside
  - solid='false' means render both inside and outside

- This approach reduces the number of polygons needing to be drawn, thus improving performance

- Confusing if user gets lost inside invisible geometry
  - Hint: set solid='false' to draw both sides
Shape Parent with Geometry Child

```
<Shape>
  <Box size='1 2 3'/>
  <Appearance>
    <Material/>
  </Appearance>
</Shape>
```

Shape must be parent node, can only hold one geometry node

```
<Shape>
  <Sphere radius='1'/>
  <Appearance>
    <Material/>
  </Appearance>
</Shape>
```

Appearance and Material nodes define colors, transparency, etc.

Primitives have simple dimensions
Typical volume ~1 m radius

All units are in meters
Box node

• Six-sided rectangular parallelepiped
  • meaning: not necessarily a cube, but it can be
  • three non-zero non-negative size dimensions for x y z

• Centered at local origin, size field has X3D data type SFVec3f
  • SF Vec = Single-field vector
  • array length of 0 or 1 only
  • 3f = 3 floating-point values
Cone node

• Circular bottom
  • Radius non-zero non-negative
  • height above bottom

• Centered at local origin

• Can hide different parts
  • side='false'
  • bottom='false'

• Set side='false' (for bottom only) defines a flat circle
Cylinder node

• Right-angle cylinder with top and bottom caps

• Non-zero non-negative height above bottom

• Circular radius

• Centered at local origin

• Can hide different parts
  • side='false'
  • top='false'
  • bottom='false'
Sphere node

• Circular radius

• Centered at local origin
  • phi and theta are implicit
  • not defined by author
Text node

• Produce readable flat, 2D text strings in X3D world
  string field is MFString array of “quoted strings”
  • Each “quoted string” appears on a separate line length
    field is MFFloat array of lengths for each line
  • Can shrink or stretch size of each line if needed
    maxExtent is maximum length for all substring lines

• Note characters have no 3D depth
  • Flat when viewed from alongside
  • Not viewable from behind unless solid='false’

• Hint: use Billboard to face user
X3D Cartesian Coordinate System

Dr. Felix Hamza-Lup - Fulbright Lecture, Bangkok University
2015
X3D Measurement Units

• Linear measurements in meters
  1 meter = 39.3 inches

• Angular measurements in radians
  2\pi = 360 degrees

• Time is measured in seconds
  Starting January 1, 1970 (Unix time)

• Colors
  RGB model floating points ranging [0..1] – normalized
  In contracts with HTML [0..255]
X3D Coordinate System

Right hand rules!

First three fingers of right hand must align with the X Y Z axes, in that order

Right hand rule also provides direction of positive rotation about an axis