Geometric Modeling: Meshes

Theory and Practice

CS 415: Game Development

Professor Eric Shaffer

Polygonal Meshes

In rendering, we typically will generate an image by simulating the reflection of light off surfaces

Rasterization engines most often use polygonal meshes to represent surfaces

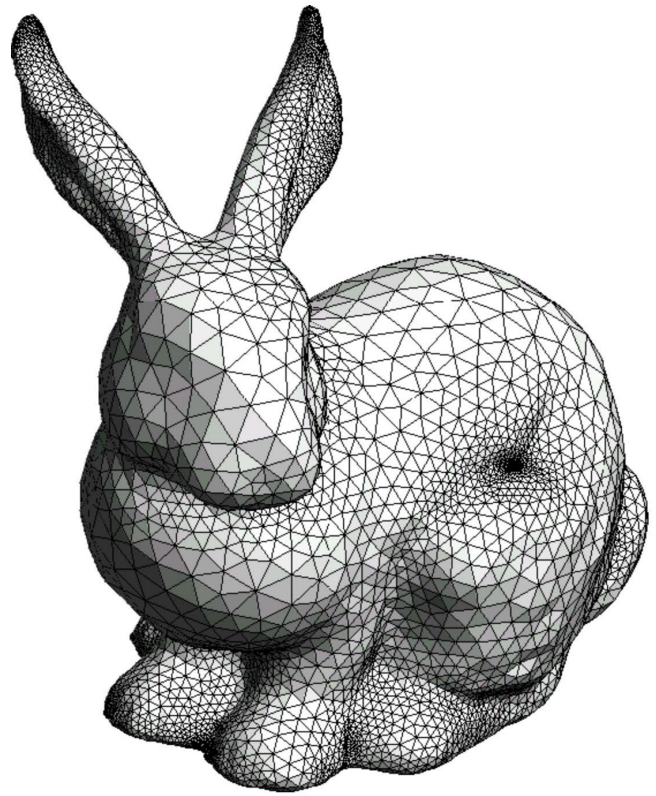
Modern GPUs are designed specifically to rasterize triangles

Why do we say 2D when we are rendering in 3D?

Many advantages to using triangles:

- Simplest 2D primitive…
- Any 2D polygon can be triangulated
- Can easily represent sharp surface features

Any disadvantages you can think of?

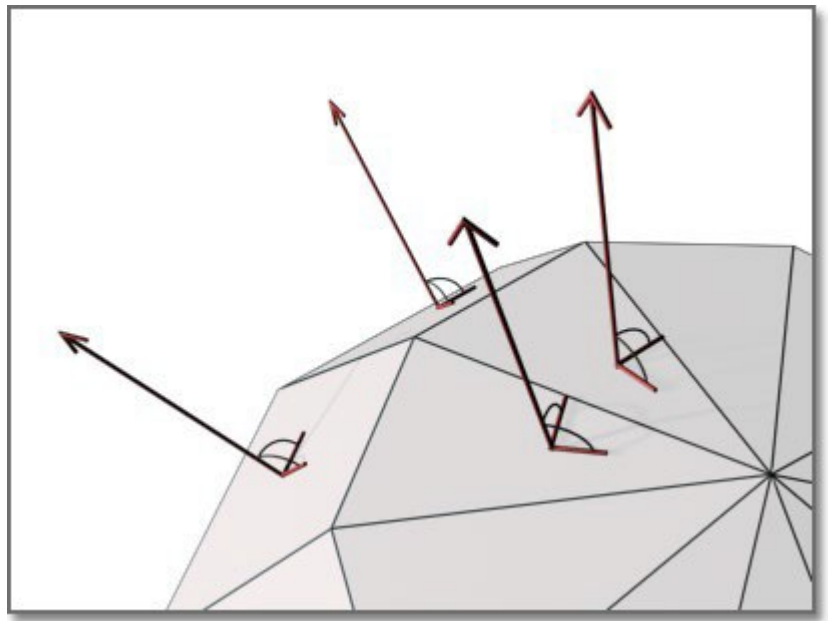


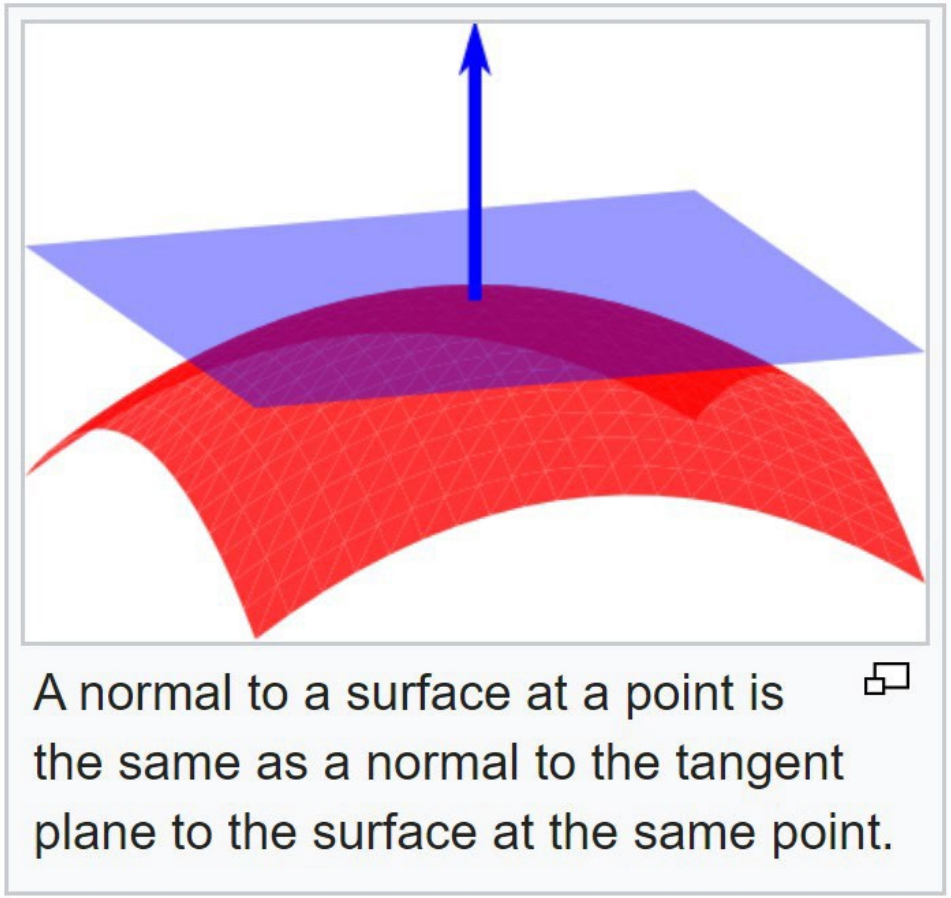
Vocabulary: Surface Normals

A normal is a vector that is perpendicular to object

Each triangle in a surface mesh has outward facing normal

The normal is just the vector perpendicular to the triangle



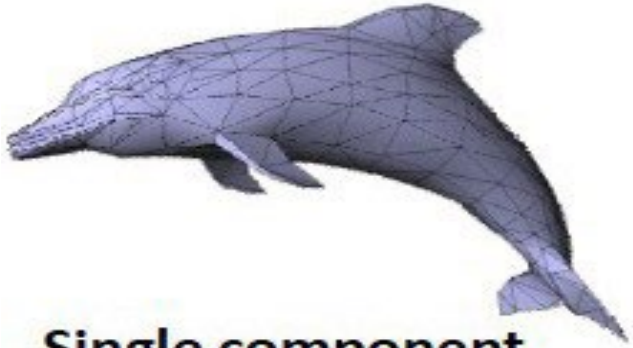


A normal to a surface at a point is the same as a normal to the tangent plane to the surface at the same point.

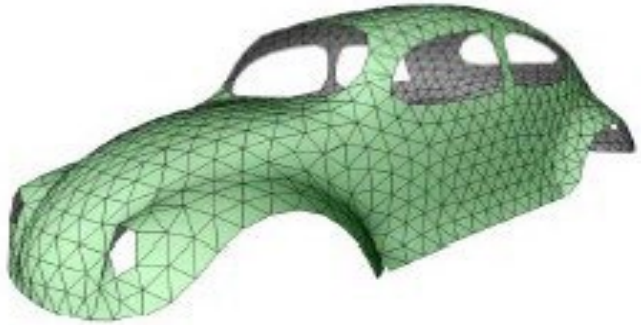
Courtesy Wikipedia

ILLINOIS

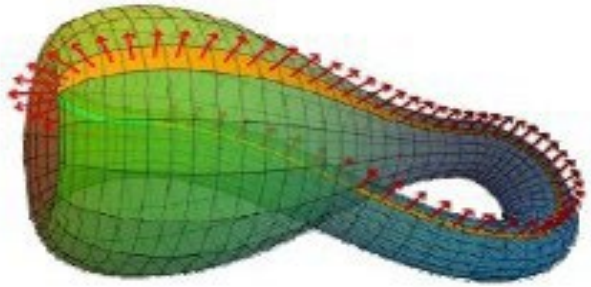
Vocabulary: Surface Mesh Properties



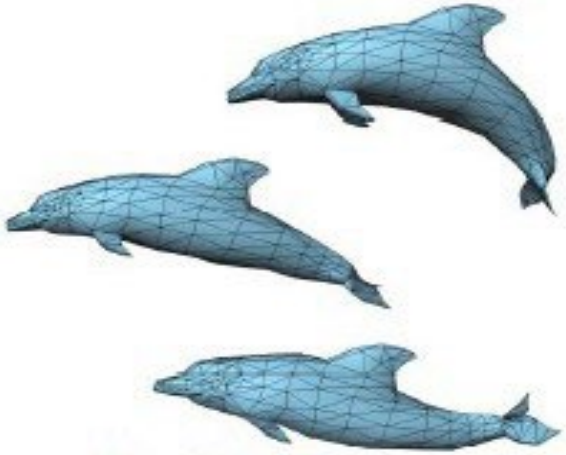
Single component, closed, triangular, orientable manifold



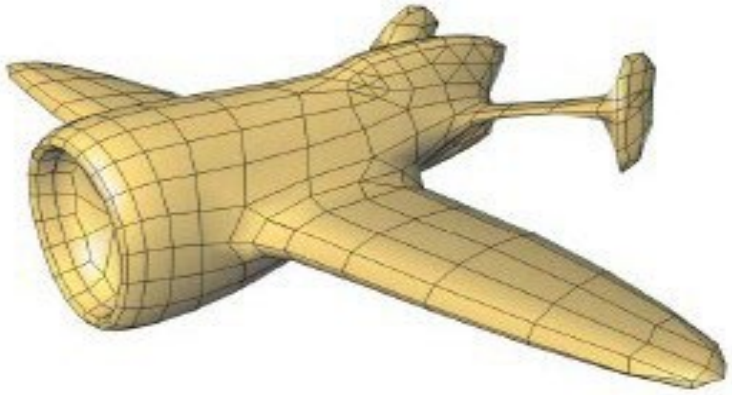
With boundaries



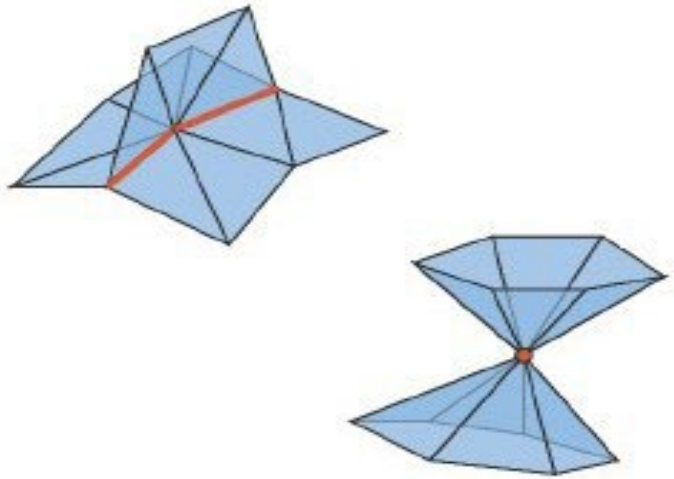
Not orientable



Multiple components

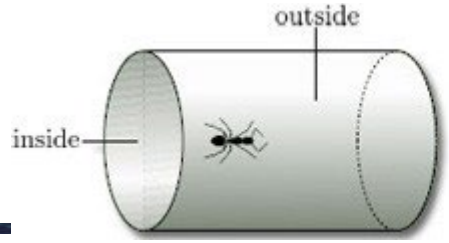


Not only triangles



Non manifold

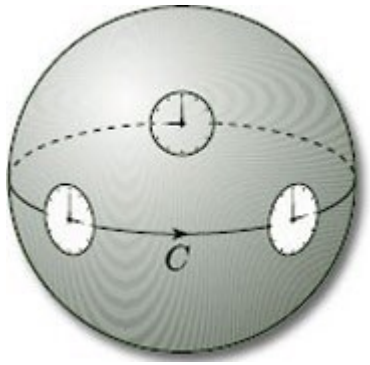
Orientability

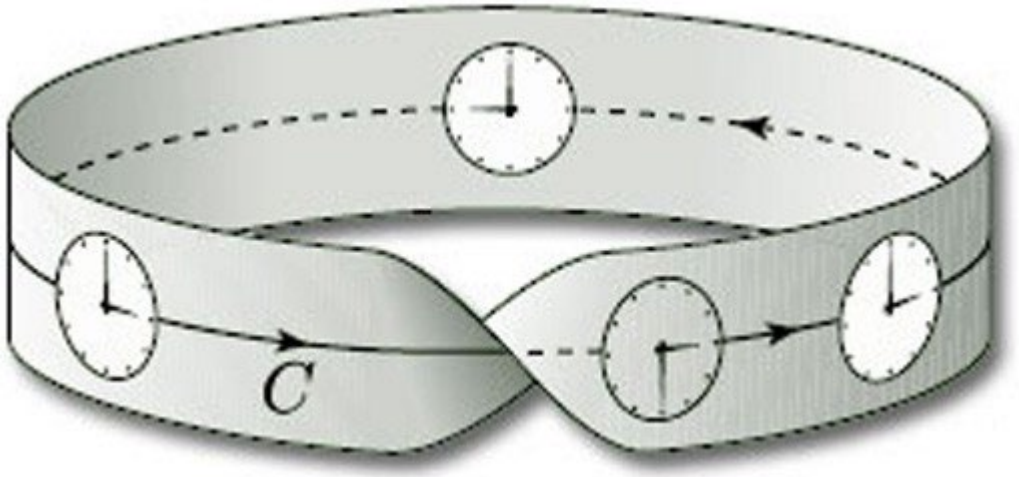


Not that relevant to this course….but it is interesting….

Imagine sliding a clock face around a surface.

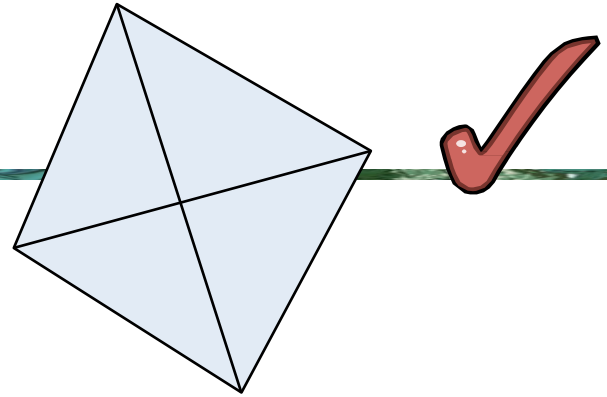
On an orientable surface the clock face will return to starting point and appear the same. (Sphere)
On a non-orientable surface it will be a mirror image of the original clock face (Moebius Strip)
…there is no consistent definition of clockwise on a non-orientable surface.

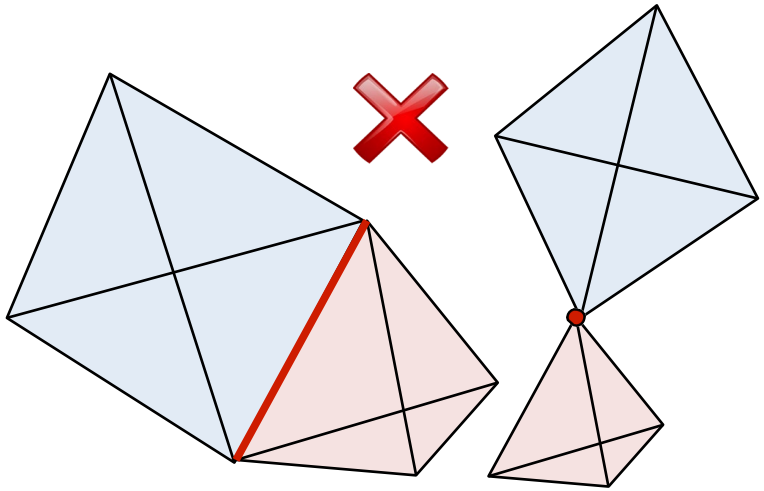




ILLINOIS

Surface Mesh Properties





Manifold:

1. Every edge connects exactly two faces
2. Vertex neighborhood is “disk-like”

Orientable: Consistent normals

Watertight: Orientable + Manifold

Boundary: Some edges bound only one face

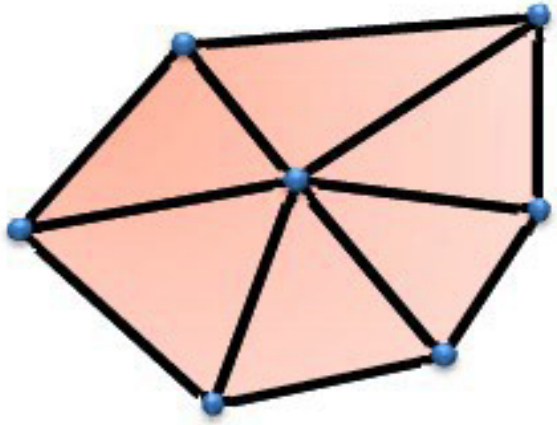
Ordered: Vertices in CCW order when viewed from normal

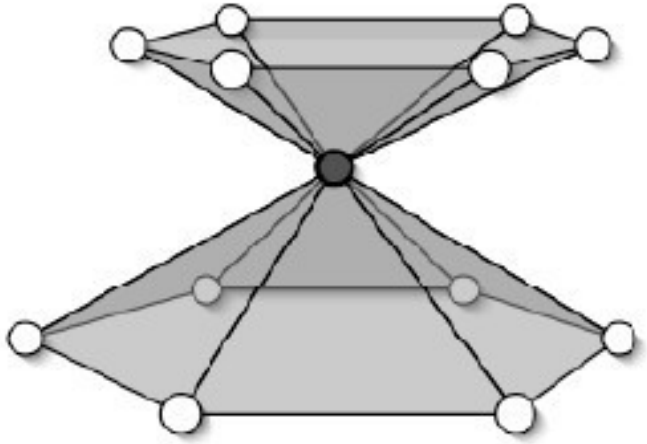
The blue and pink meshes above are tetrahedra, like 4-sided pyramids

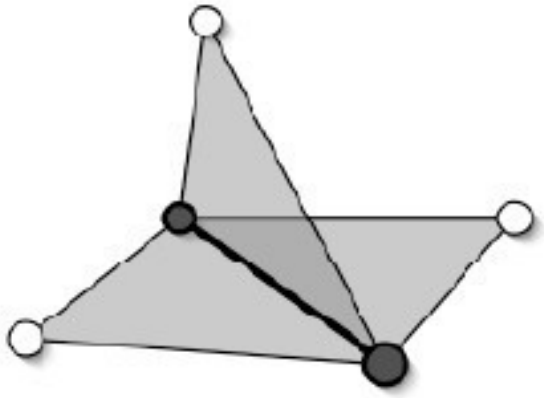
ILLINOIS

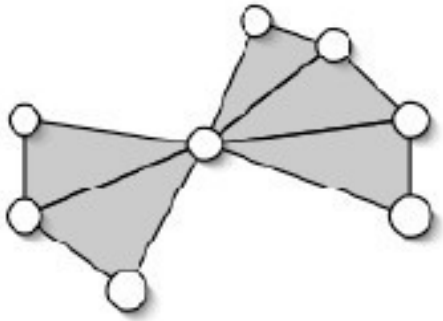
2-Manifold Mesh Examples

Disk-shaped neighborhoods





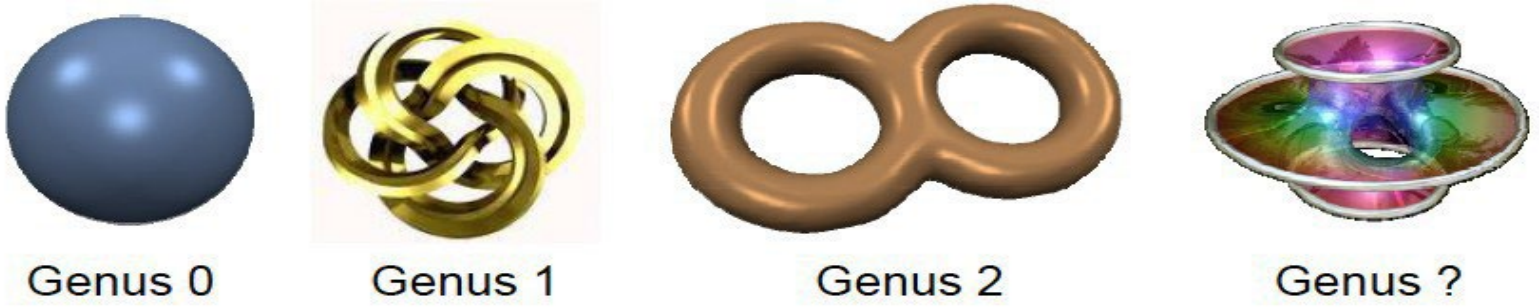


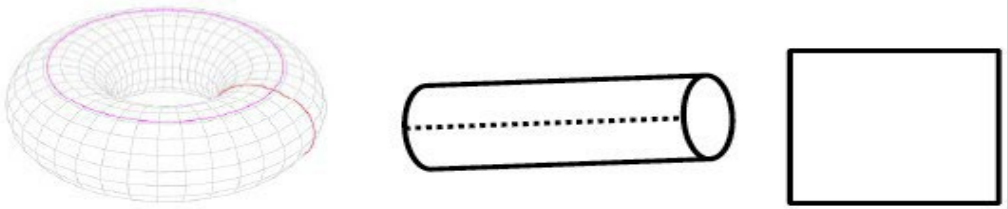


non-manifolds

Genus


Genus of orientable surfaces
genus 0
genus 1
genus 2
genus 3


Genus 0
Genus 1
Genus 2
Genus ?



Euler Characteristic

For a closed (no boundary), manifold, connected surface mesh:

$$V-E+F= 2(1-G)$$

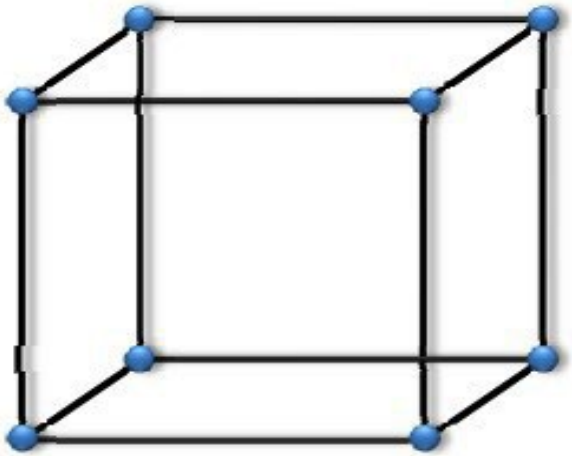
- V = number of vertices
- E = number of edges
- F = number of faces
- G = genus (number of holes in the surface)

Leonhard Euler
1707 – 1783

A 2-manifold is a surface (locally like a plane)

Euler Characteristic for Closed 2-Manifold Polygonal Meshes

$$V + F - E = \chi$$ **Euler characteristic**

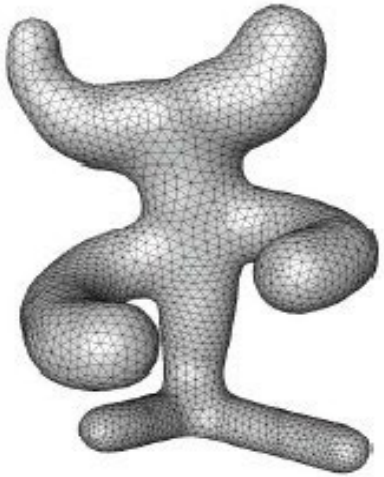


$V = 8$

$E = 12$

$F = 6$

$\chi = 8 + 6 - 12 = 2$



$V = 3890$

$E = 11664$

$F = 7776$

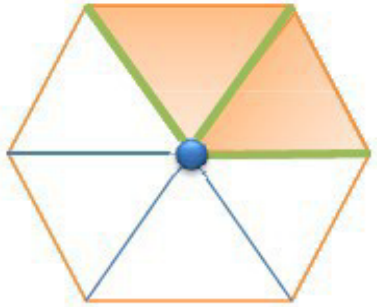
$\chi = 2$

..and if they are triangle meshes

- *Triangle* mesh statistics

  $E \approx 3V$

  $F \approx 2V$



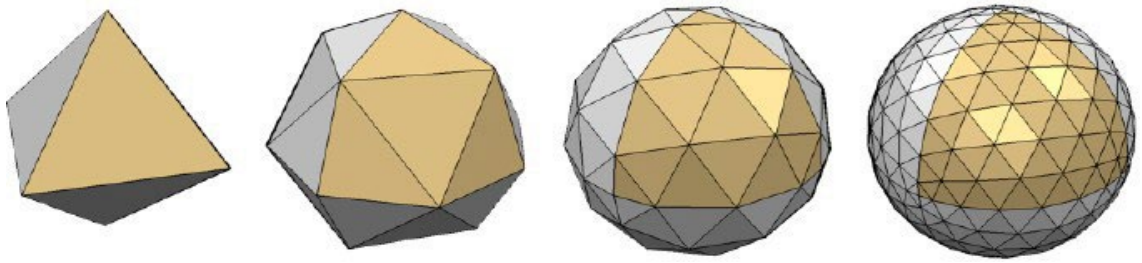
- Avg. valence ≈ 6

  Show using Euler Formula

Aside from being totally interesting on their own, these formulas are useful for computing memory usage

ILLINOIS

Mesh Data Structures



Need to store

- Geometry
- Connectivity

Can be used as file formats or internal formats

Considerations

- Space
- Efficient operations

Mesh processing has different requirements than rendering

- Example: Deforming a mesh when simulating physics…

ILLINOIS

Mesh Data Structure: Face Set (STL)

- face:
  - 3 positions

Triangles		
x_{11} y_{11} z_{11}	x_{12} y_{12} z_{12}	x_{13} y_{13} z_{13}
x_{21} y_{21} z_{21}	x_{22} y_{22} z_{22}	x_{23} y_{23} z_{23}
...	...	...
x_{F1} y_{F1} z_{F1}	x_{F2} y_{F2} z_{F2}	x_{F3} y_{F3} z_{F3}

36 B/f = 72 B/v

no connectivity!

Designed in 1987 for stereolithography (3D printing technology)

No explicit information about which vertices are shared by which triangles
Consider how efficiently we could:

- Deform mesh by moving vertices
- Lookup vertex neighbor information

We will be focusing on triangle meshes…but all of these structures generalize to other polygons…storage requirements will change.

ILLINOIS

Mesh Data Structure: Indexed Face Set (OBJ)

- vertex:
  - position
- face:
  - vertex indices

Vertices
x_1 y_1 z_1
...
x_V y_V z_V

Triangles
v_{11} v_{12} v_{13}
...
...
...
...
v_{F1} v_{F2} v_{F3}

12 B/v + 12 B/f = 36 B/v

no neighborhood info

The OBJ file format is a popular storage format for meshes

Developed by Wavefront Technologies…now part of AutoDesk

Text files with .obj extension

```
# List of geometric vertices
v 0.123 0.234 0.345
v …
v …
# List of triangles
f 1 3 4
f 2 4 5
…
```

ILLINOIS

Indexed Face Set

One block of data are the vertices

- Each vertex is a set of 3 coordinates
- Often referred to as the geometry of the mesh

Vertices
x_1 y_1 z_1
...
x_V y_V z_V

Another block of data is the set of triangles

- Each triangle is set of 3 integers vertex IDs
- The vertex IDs are indices into the vertex block

Triangles
v_{11} v_{12} v_{13}
...
...
...
...
v_{F1} v_{F2} v_{F3}

What are some advantages of this representation?

Disadvantages?

ILLINOIS

Questions

- How easy is it for artists to work directly with surface meshes?
- How important is it for artists to work easily with meshes?
- What would be an alternative surface representation?

- What are the advantages compared to meshes?

- What are the disadvantages compared to meshes?

AutoDesk FBX File Format

- Popular 3D data interchange format utilized between 3D editors and game engines
- Created as file format for Kaydara's Filmbox motion capture tool
  - Name derived from FilmBox
- In 2005 Kaydara released a public SDK for the closed and proprietary format
- Filmbox was renamed MotionBuilder and was later acquired by Autodesk
- Alternative 3D model and scene file formats are:
  - Wavefront OBJ 3D format
  - Khronos Group glTF 3D format

ILLINOIS

FBX File Format

The FBX SDK is part of Autodesk FBX technology, a family of tools which allow 3D content developers to import and export 3D data. Autodesk FBX contributes to the flexibility of organizations involved in the media and entertainment industry by allowing them to mix and match a variety of 2D and 3D digital content creation applications.

FBX File Format for 3D Scenes

FBX files (*.fbx*) are normally saved in a binary (or native) format, but they can also be saved in ASCII format. Binary FBX files and ASCII FBX files both use the *.fbx* filename extension.

FBX files store data about cameras, lights, meshes, NURBS, and the other *elements* of a 3D scene. Applications such as Autodesk 3ds Max, Autodesk Maya, and Autodesk MotionBuilder can import all or part of the scene data from an FBX file. They can also export their scene data to an FBX file. All these operations make use of the FBX SDK.

Note **The FBX file format is not documented**. Applications should use the FBX SDK to export and import scene data to and from FBX files (and other file formats supported by the FBX SDK).

http://docs.autodesk.com/FBX/2014/ENU/FBX-SDK-Documentation

ILLINOIS

FBX Scene Elements

Gobo (lighting) - Wikipedia

A **gobo** is an object placed inside or in front of a light source to control the shape of the emitted light and its shadow. For studio photography purposes, ...

The FBX SDK lets you access, create, or modify the following elements of a scene (`FbxScene`):

- Mesh - `FbxMesh`
- Level of detail (LOD) groups - `FbxLodGroup`
- Cameras (including stereo cameras for 3D) - `FbxCamera`
- Lights and gobos - `FbxLight`, `FbxGobo`
- NURBS - `FbxNurbs`, `FbxNurbsCurve`, `FbxNurbsSurface`, `FbxTrimNurbsSurface`
- Texture mapping over a geometry - `FbxTexture`
- Material mapping over a geometry - `FbxSurfaceMaterial`
- Constraints - `FbxConstraint`
- Vertex cache animation on the control points of a geometry - `FbxDeformer`
- Scene settings that provide Up-Axis (X/Y/Z) and scene scaling (units) - `FbxGlobalSettings`, `FbxAxisSystem`
- Transformation data including position, rotation, scale, parent - `FbxNode`
- Markers - `FbxMarker`
- Skeleton segments (root, limb, and limb node) - `FbxSkeleton`
- Animation curves - `FbxAnimCurve`
- Rest and bind poses for a list of nodes (bones and geometry) - `FbxPose`

FBX Example

```
; FBX 7.1.0 project file
; Copyright (C) 1997-2010 Autodesk Inc. and/or its licensors.
; All rights reserved.
; ----------------------------------------------------
FBXHeaderExtension:  {
       ; header information: global file information.
    FBXHeaderVersion: 1003
    FBXVersion: 7100
    CreationTimeStamp:  {
        Version: 1000
        Year: 2010
        Month: 1
        Day: 19
        Hour: 16
        Minute: 30
        Second: 28
        Millisecond: 884
    }
    Creator: "FBX SDK/FBX Plugins version 2011.2"
    SceneInfo: "SceneInfo::GlobalInfo", "UserData" {
 ...
}
```

FBX Example

```
GlobalSettings:  {
    Version: 1000
    Properties70:  {
        P: "UpAxis", "int", "Integer", "",1
        P: "UpAxisSign", "int", "Integer", "",1
        P: "FrontAxis", "int", "Integer", "",2
        P: "FrontAxisSign", "int", "Integer", "",1
        P: "CoordAxis", "int", "Integer", "",0
        P: "CoordAxisSign", "int", "Integer", "",1
        P: "OriginalUpAxis", "int", "Integer", "",-1
        P: "OriginalUpAxisSign", "int", "Integer", "",1
        P: "UnitScaleFactor", "double", "Number", "",1
        P: "OriginalUnitScaleFactor", "double", "Number", "",1
        P: "AmbientColor", "ColorRGB", "Color", "",0,0,0
        P: "DefaultCamera", "KString", "", "", "Producer Perspective"
        P: "TimeMode", "enum", "", "",6
        P: "TimeSpanStart", "KTime", "Time", "",0
        P: "TimeSpanStop", "KTime", "Time", "",46186158000
    }
}
```

FBX Example

```
  ObjectType: "Model" {
      Count: 86
      PropertyTemplate: "FbxNode" {
          Properties70:  {
              P: "QuaternionInterpolate", "bool", "", "",0
              P: "RotationOffset", "Vector3D", "Vector", "",0,0,0
              P: "RotationPivot", "Vector3D", "Vector", "",0,0,0
              P: "ScalingOffset", "Vector3D", "Vector", "",0,0,0
              P: "ScalingPivot", "Vector3D", "Vector", "",0,0,0
..}
  ObjectType: "Material" {
      Count: 1
      PropertyTemplate: "FbxSurfacePhong" {
          Properties70:  {
              P: "ShadingModel", "KString", "", "", "Phong"
              P: "MultiLayer", "bool", "", "",0
              P: "EmissiveColor", "ColorRGB", "Color", "",0,0,0
              P: "EmissiveFactor", "double", "Number", "",1
              P: "AmbientColor", "ColorRGB", "Color", "",0.2,0.2,0.2
..}
```

FBX Example

```
    Model: 21883936, "Model::Humanoid:Hips", "LimbNode" {
        Version: 232
        Properties70:  {
            P: "ScalingMin", "Vector3D", "Vector", "",1,1,1
            P: "NegativePercentShapeSupport", "bool", "", "",0
            P: "DefaultAttributeIndex", "int", "Integer", "",0
            P: "Lcl Translation", "Lcl Translation", "", "A+",-271.281097412109,-762.185852050781,528.336242675781
            P: "Lcl Rotation", "Lcl Rotation", "", "A+",-1.35128843784332,2.6148145198822,0.42334708571434
            P: "Lcl Scaling", "Lcl Scaling", "", "A+",1,0.99999988079071,1
...
```

FBX Advantages

Wide Feature Support

The FBX format is the gold standard in its support for the variety of data in its format. FBX supports 3D models, scene hierarchy, materials lighting, animations, bones, skinning, and blend shapes. FBX, as an older format, also supports data that isn't widely used today, such as NURBS surfaces and curves. When you choose FBX as your interchange format, you know that when it comes to geometry-related properties, it can likely represent the data you need.

https://www.threekit.com/blog/when-should-you-use-fbx-3d-file-format

ILLINOIS

FBX Advantages

Fast and Efficient

The FBX file format, because it utilizes a binary format, is both fast and efficient. This is because when one stores data as binary it is faster to write and read it, whereas a text-based format must convert binary data to and from human readable numbers. The FBX file format is also efficient in terms of space because binary representations of numbers take up less space than human-readable numbers. Binary files can often be complex if you need to read and write to them directly, but the FBX SDK hides this completely from the user of the FBX SDK.

https://www.threekit.com/blog/when-should-you-use-fbx-3d-file-format

ILLINOIS

FBX Advantages

Good Compatibility

In part because the SDK ensures that it can read all previous versions of the FBX file format, it ensures that most modern versions of tools can correctly read the FBX files produced by other tools. Even if the FBX file format changes, the SDK can ensure that it can read both the old format and the new format via different code paths that are transparent to the user of the SDK. As such, FBX is not riddled with compatibility problems that have plagued similar complex formats such as COLLADA (.dae). If FBX properly supports a feature, it can transfer that data between applications without much concern.

Accordingly, FBXs can be used to transfer complex 3D scene data easily between Clara.io, 3DS Max, Maya, Unity 3D and Unreal Engine.

https://www.threekit.com/blog/when-should-you-use-fbx-3d-file-format

ILLINOIS

FBX Disadvantages

Closed Format

The biggest downside of the FBX format is that it is a closed format. The only official way to access the data in FBX files is to use the official SDKs. If you want to load an FBX on a system that is not officially supported by the FBX SDK (such as in a web browser or an open source application) you cannot do so. (Although recently, after a couple years of effort, Blender has successfully reverse engineered most of the current version of the FBX format.)

Because it is a closed format, it is also not possible for anyone to evolve the format except for its owners Autodesk. Its owners have not recently made many changes to the FBX file format, and as such, it has somewhat stagnated.

https://www.threekit.com/blog/when-should-you-use-fbx-3d-file-format

ILLINOIS

FBX Disadvantages

Legacy Lighting

The lighting model in FBX also uses 1990s-era conventions. Modern games and tools like ThreeKit use a physically based lighting model where all lighting parameters are grounded in physical quantities such as lumens, luminosity and physically based falloff. Modern game engines want to transfer global illumination information such as light probes and image-based lighting, but FBX does not support this.. FBX uses a more historic and arbitrary lighting model, which makes it difficult to communicate between tools that use the more modern and physically-based approaches. As with materials, most lighting information transferred using the FBX file format has to be fixed or simply redone once it is imported into its target application.

https://www.threekit.com/blog/when-should-you-use-fbx-3d-file-format

ILLINOIS

FBX Disadvantages

Legacy Materials

The FBX file format uses a 1990s-era lighting model premised on the Blinn-Phong spec-glossy model, along with a single color or texture per material property. While this was sufficient for a minimalist material representation, it does not reflect the capabilities of modern 3D editors or game engines that utilize complex shader networks in concert with the Physically-Based Rendering material model centered around roughness-metalness. Nor does the simplistic FBX materials support sub-surface scattering, proper layered materials nor anisotropic effects. Because of the severe limits of FBX's material model, most materials on 3D models have to be redone after they have been transferred between tools via the FBX format.

https://www.threekit.com/blog/when-should-you-use-fbx-3d-file-format

ILLINOIS