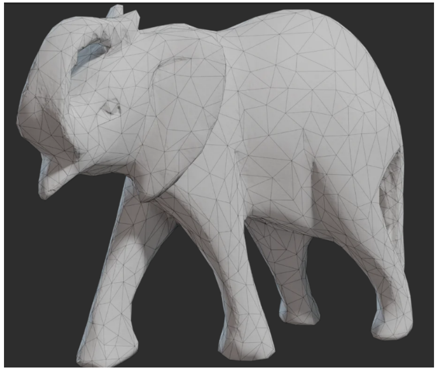
Foundations of 3D Scene Modeling

3D scene in big picture

Cornerstones of image generation:

- **3D scene modeling**
  - Representing 3D models, lights and cameras
  - Animation and interaction
- Rendering
  - Creating 2D images from 3D scenes
- Raster image
  - Representing 2D images
  - Post-processing



https://polyhaven.com/

3D scene modeling for image synthesis

- Image synthesis: rendering a 3D scene
- Analogy: taking real world photograph/video
- Elements:
  - observer/**camera** placed somewhere in space,
  - this space should contain **objects** with various shapes and materials
  - and finally, **light source** must be present to shed light onto this space.

Models from: https://polyhaven.com/

3D scene modeling

A way of acquiring/creating, representing and manipulating:

- Objects
- Lights
- Camera

3D scene modeling questions

How do we represent scene elements in a computer?

- Intuitive for user?
- Efficient for rendering?

How we create/acquire/manipulate scene elements?

- How to use scene elements to create real-world phenomena and objects?
- How do we manipulate 3D models?

3D modeling foundations

3D scene is approximated using 3D primitives:

- 3D space
- Points
- Vectors
- Line segments
- Rays
- Lines
- Planes
- Polygons

3D objects

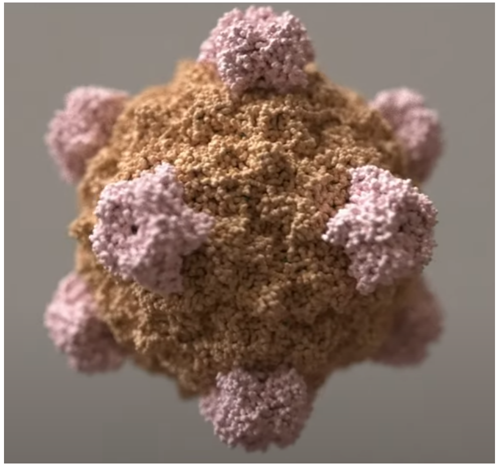
3D models: intuition

Depending on application, we would like our 3D scene to contain different elements

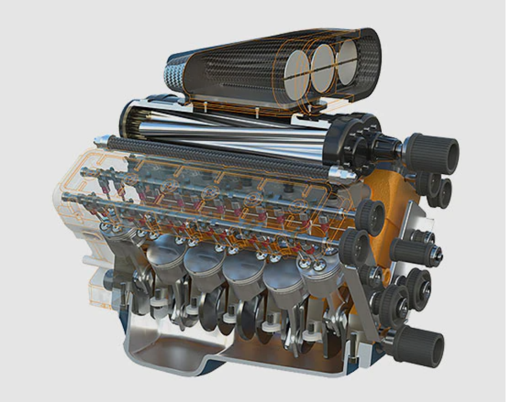
OTHER

How do we even start with this?

Arhitecture, Revit:
https://www.autodesk.com/products/revit



Brady Johnston, Visaulising viruses:
https://www.youtube.com/watch?v=adhTmwYwOiA&ab_channel=Blender



Autodesk Fusion 360:
https://www.autodesk.com/products/fusion-360/overview

3D models: shape and material

Looking at real world phenomena, we can say that in order to create objects in a 3D scene, we need a way of representing:

- Shape
- Size, relative position in the space and to other shapes
- Appearance

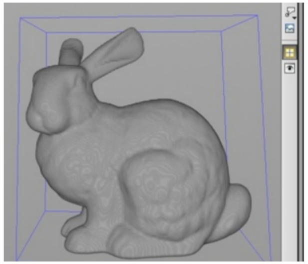
Therefore, we can conclude that representing 3D models requires:

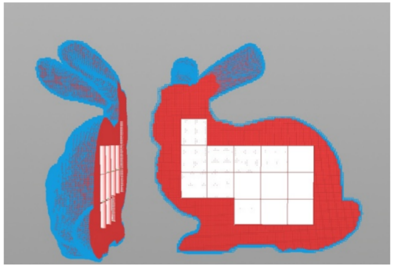
- Representation of **shape** - geometry
- Representation of **material** – appearance

3D model shape representation

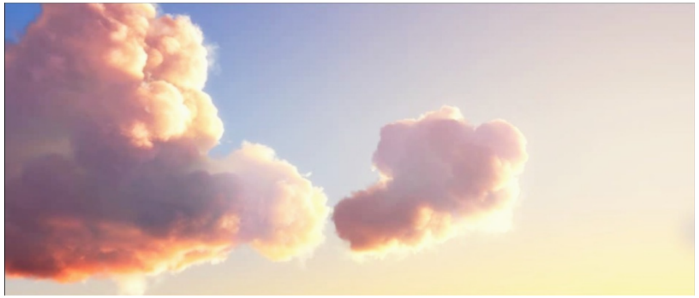
Shape representation: surface and volume

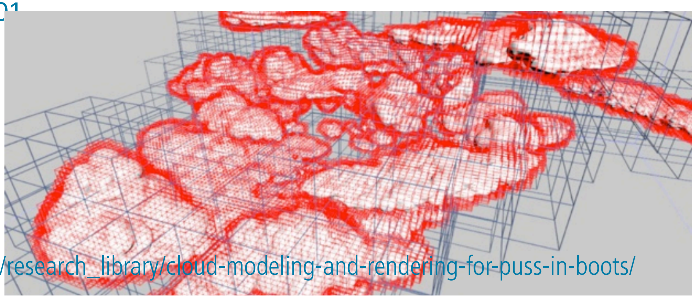
- Shape representation can be used to describe:
  - Surfaces
  - Volumes
- Decision on shape representations depends on phenomena that we are modeling





Production examples at dreamworks, OpenVDB:
https://artifacts.aswf.io/io/aswf/openvdb/openvdb_production_2015/1.0.0/openvdb_production_2015-1.0.0.pdf

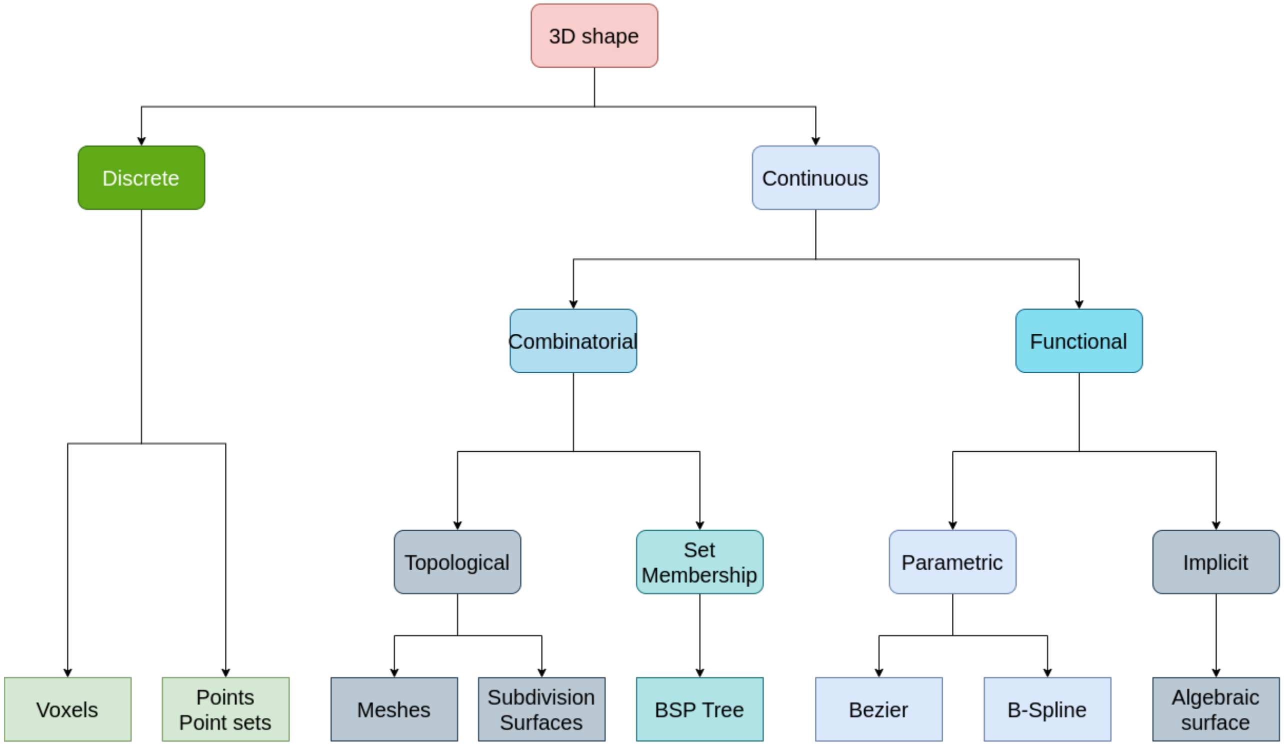




https://research.dreamworks.com/research_library/cloud-modeling-and-rendering-for-puss-in-boots/

Object shape representations

- **Points**
  - Point clouds
  - Particles and Particle systems
- **Surfaces**:
  - Polygonal mesh
  - Subdivision surfaces
  - Parametric surfaces
  - Implicit surfaces
- **Volumetric objects/solids**
  - Voxels
  - Space partitioning data-structures
- **High-level structures**
  - Scene graph



Based on:
https://www.cs.princeton.edu/courses/archive/spring22/cos426/lectures/Lecture-5.pdf

Shape representation 1: polygons

- To define a surface, simplest way is to connect points to form a **polygon**. In computer graphics, for computation tractability, we often use co-planar – all points lying on the same plane - polygons and especially **triangles**.
  - Triangle is widely used surface shape representation **primitive.**
  - Very simple and holds great properties for easy calculation, often used for efficient rendering purposes.
  - It is approximation method: curved surfaces are approximated with triangles
  - Not suitable for modeling, thus different shape representations are introduced. For rendering purposes, often all representations are turned to triangles before/during rendering stage - using very elaborated method called **triangulation**.
- Triangle is basic building block for creating more complex shapes. And modeling is all about creating complex shapes using basic building blocks.

<IMAGE: POINTS, TRIANGLES, COMPLEX SHAPES>

Shape representation 1: polygons

- Some shapes do not have flat surfaces! Polygonal shape representation will always have flat surfaces*.
- If we would like to represent curved surface, we would need smaller triangles which would fit the curved surface better. In this process we are placing more points on to surface (sampling). Generally, converting smooth surface to triangulated polygon representation – a **triangulated mesh** – is called **tessellation**.
- Main point to take away is that we can represent any object using triangle polygons. Those objects will never be perfect representations of a real world, but triangles of which they consists of can be small enough to display objects in high quality taking in account restrictions of raster display.
  - Amount of details increase realistic representation but also complexity of rendering. Computer graphics often deals with trade-off between visual quality and speed**.

* There are methods in rendering which make surface looks smooth although is represented using polygons. This method is called smooth shading and we will discuss it later.

** Finding good tradeoff between visual quality and object complexity is big research field in computer graphics. Note also that distance of viewing plays important role in amount of detail that it has to be represented.

Shape representation 2: parametric surfaces

- Simple shapes can be described mathematically (parametrically), e..g., spheres, disks, planes

Shape representation 2: parametric surfaces

- Although polygonal meshes are widely used and popular (e.g., games and film) there are other representations that are more suitable for modeling purposes
- One of these are **parametric surfaces -** used to design manufactured objects and often used in CAD software.
- Foundation of those representations are still **points**, but those points define a **control mesh** from which perfect curved surface can be generated using **analytical description**.
  - Note that this kind of representation is much more compact than polygonal mesh.
- Modeling using control points is very beneficial for curved objects. When it comes to rendering, this representation can not be rendered directly. As discussed, process called **tessellation** must be performed prior rendering.

<IMAGE: CURVED SURFACES>

Shape representation 3: subdivision surfaces

- Polygon modeling can be used for fast creation of simple shapes
- Then, **subdivision surfaces** method can be employed to alternatively refine simple polygon into smooth shapes
- <example>

Shape representation 4: voxels

- Polygonal or parametric surfaces are describing surfaces, to represent volume it is required to representing space inside such volume
  - Example: simulation can be used for shape modeling, e.g., smoke simulation. For this purposes, it is required to represent 3D space in which simulation is performed as grid of cells which are called **voxels**. <example>
  - Each voxel (a cell) is a small volume of space on which computation is performed. Simulation is performed in series of steps. Each step is recorded and transformed to triangulated mesh for rendering.
  - This way, **animated** mesh can be generated.
- Any kind of visualizations/simulations/interaction requiring volumetric representation can be well represented with **voxels**.
  - https://sites.google.com/site/letsmakeavoxelengine/home/water
  - <example: medical images>
- Voxels are also widely used for any kind of modeling purposes where it is required to describe object's interior – e.g., digital sculpting.
  - Example: zbrush

Shape representation 5: implicit

- Constructing complex shapes can be also done using basic primitives (box, sphere, etc.) and series of operations (add, substract, etc.). This kind of modeling is called **Constructive Solid Geometry**.
- An representation for such basic primitives is called **implicit** since they are completely analytically defined. - **SDFs**
- This enables fast and flexible modeling system, but when it comes to rendering, they need to be converted to triangulated mesh – similarly as for curved surfaces. Algorithm in this case is called marching cubes.
- Often used in engineering and CAD modeling environments

<CSG AND IMPLICIT SHAPES MODELING>

Shape representation 6: sweeping surfaces

- Surfaces that are generated from a section curve positioned along a path*
- Blender:
  - https://docs.blender.org/manual/en/latest/modeling/meshes/tools/spin.html
  - https://docs.blender.org/manual/en/latest/modeling/meshes/editing/edge/screw.html

* https://www.ironcad.com/support/OnlineHelp/3D_Design_Environment/Part_Design_Process/Surface_Design_Process/Sweep_Surface1.htm

Shape representation 7: fractals

- Complex, natural phenomena such as clouds, mountains, trees, vegetation, galaxies can be represented using fractals
- Mandelbrot: https://www.mandelbulb.com/
- Irregular geometrical shapes with symmetrical property
- Generative art: http://blog.hvidtfeldts.net/

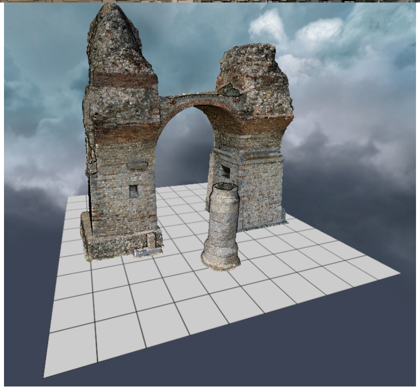
  TODO

Shape representation 8: particle systems

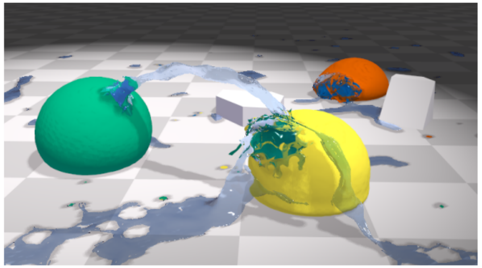
- Particles (points, images, simple shapes) which have dynamic behavior described by rules.
- examples

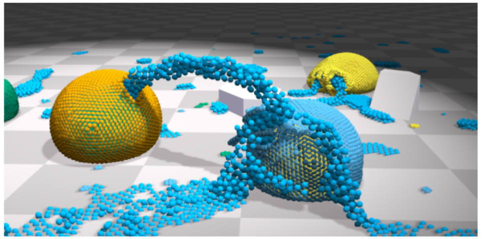
Shape representation 9: points and point clouds

- Point clouds, Scans
- Unified physics simulation using points representation:



https://github.com/potree/potree





Unified physics for real-time applications: https://mmacklin.com/uppfrta_preprint.pdf

Shape representation 10: image based modeling

- TODO

Equivalence of representations

- Each shape representation has capacity to describe shape of any geometric object
- Different representations exist because certain tasks are more tractable with particular representation. For example:
  - Rendering process
  - 3D modeling
  - Simulation of objects
  - Animation
  - Acquisition/digitalization of objects from a real world
- Different datastructures used for representations determine algorithms for processing

3D models: various representations?

Similar as in, e.g., drawing where drawing tools and shapes needed to be chosen, 3D models creation is based on set of representations and operations from which we must choose. Those representations are on the one hand flexible enough to be used for any shape and understandable to a computer, that is rendering program:

- From the user-creation point of side: we need a representation of shape and material which is intuitive to handle (to author)
- From the rendering point of side: we need to make sure that shape and material representation can be used for rendering process
  - Shape representation will be used for determining visibility problem*: what objects we see from certain point of view
  - Material representation will be used for determining the shading problem**: how the objects that we see appear

* As we will get to know more complex material representations we will see that this is not completely true. Visibility of objects will also depend on material. Imagine glass cup on the top of a book – book will be visible since glass is transparent.

** Shape and material information are both important for calculating how objects appear. We separate them to show that you can first purely focus on modeling a 3D shape and then model material afterwords. Also, separating objects in shape and material is desired for rendering architectures so that are decoupled.

3D models: various representations?

- Different tasks require different representations - **acquisition**:
  - Reconstructing from photographs: photogrammetry
  - Creating isosurfaces from CAT or MRI
  - Transforming data into surfaces or volumes
  - Sampling real world objects using scanners, digitizers or other sensing devices
    - Example: kinect

3D models: various representations?

Different tasks require different representations – **modeling**:

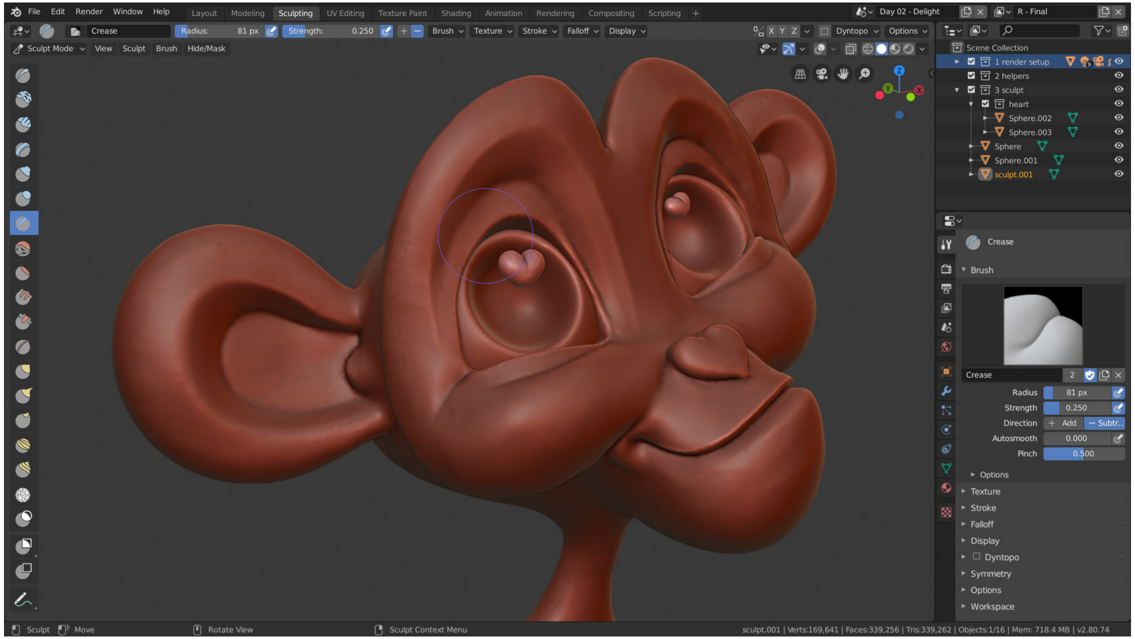
- Manual modeling
- Procedural modeling

3D models: various representations?

Different tasks require different representations – **modeling**:

- Sculpting

Jasmine, Zbrush, http://pixologic.com/zbrush/gallery/?year=2018

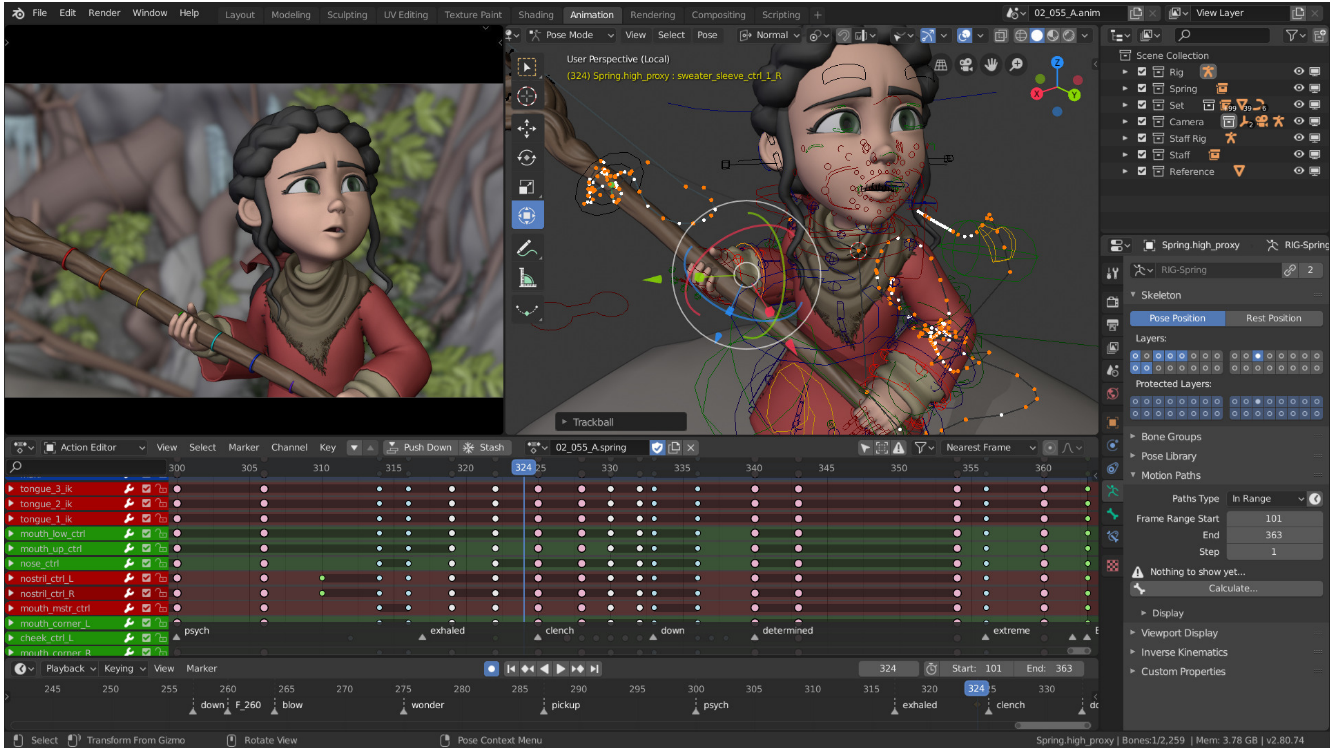


Blender, https://www.blender.org/features/sculpting/

3D models: various representations?

Different tasks require different representations – **animation**:

- Manual, rigging

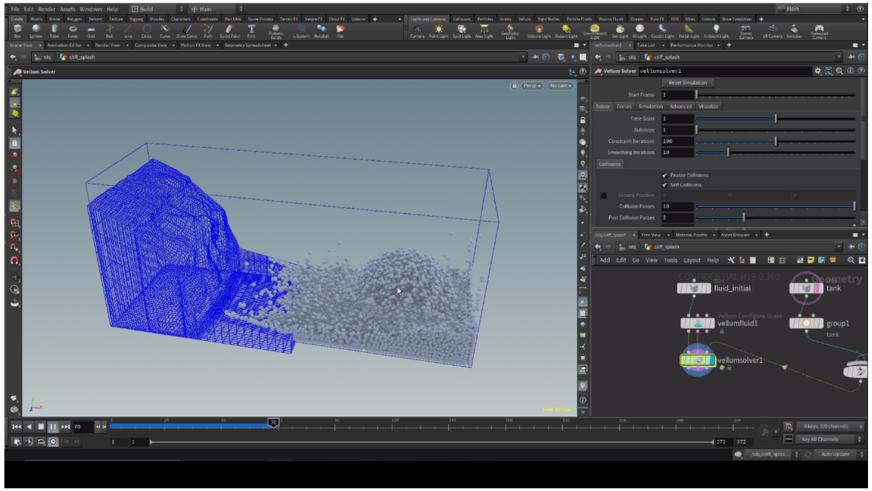


https://www.blender.org/features/animation/

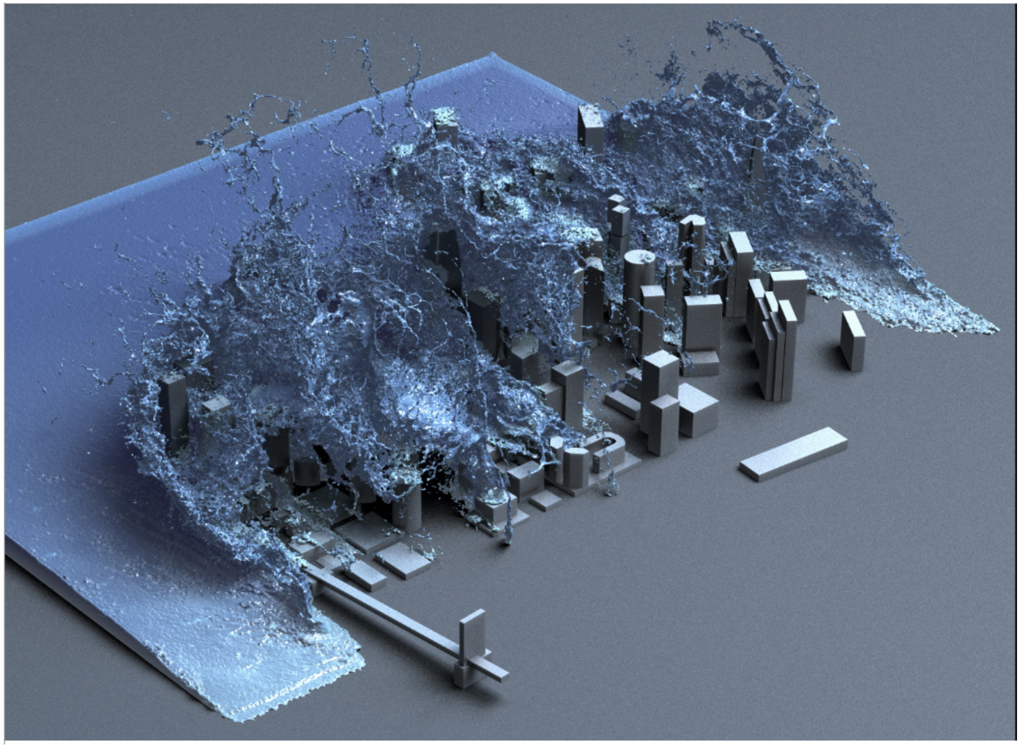
3D models: various representations?

Different tasks require different representations – **animation**:

- procedural, simulation



Houdini, https://www.sidefx.com/tutorials/h19-vellum-fluids-starter-pack/



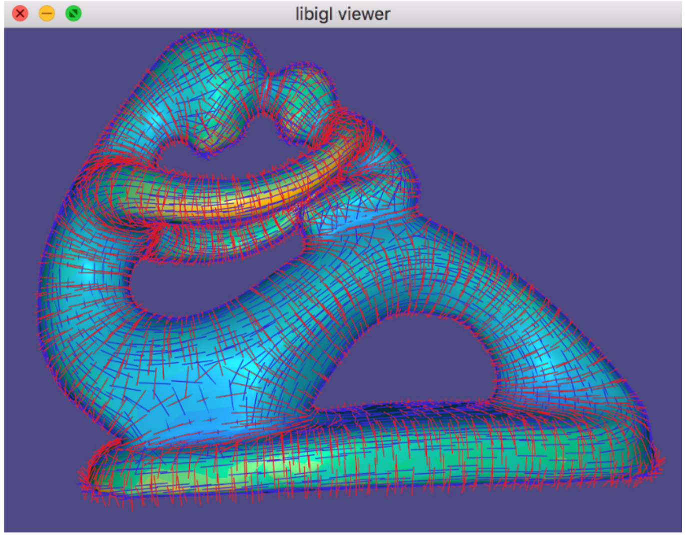
OpenVDB, https://people.csail.mit.edu/kuiwu/gvdb_sim.html

3D models: various representations?

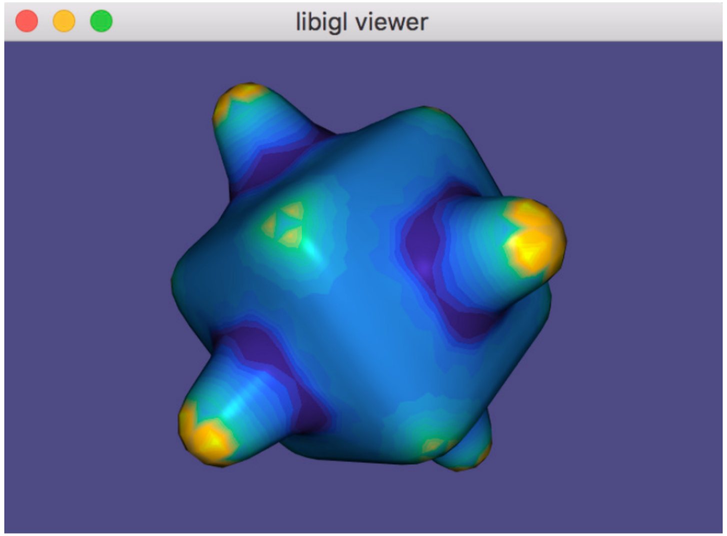
- Different tasks require different representations – **rendering**:
  - Efficient visibility solving

3D models: various representations?

- Different tasks require different representations – **analysis**:
  - Curvature
  - Smoothness



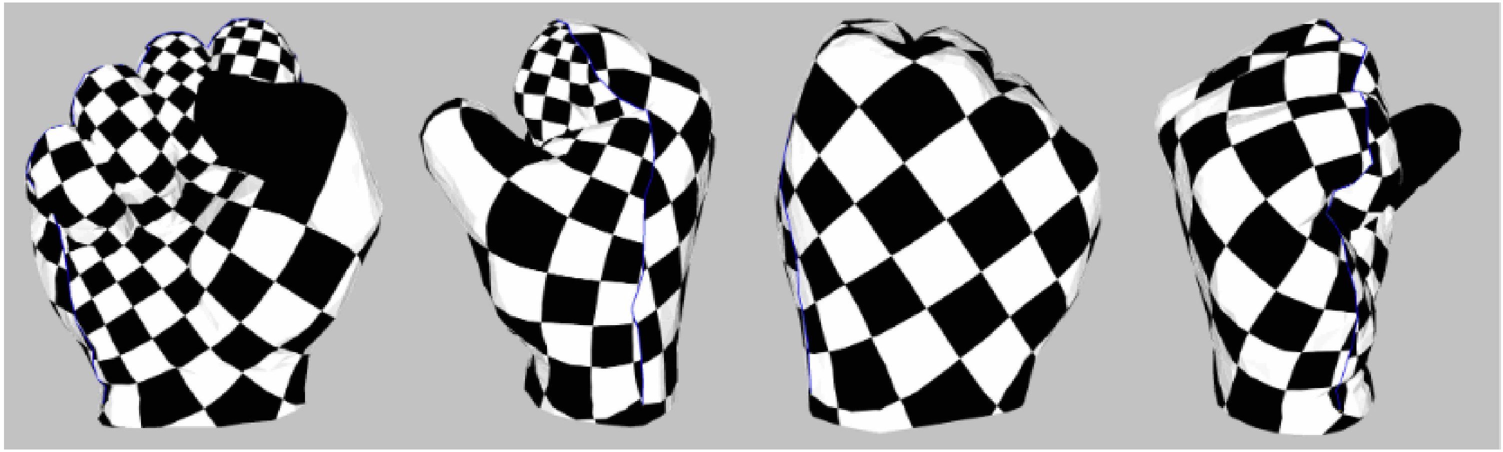
Libigl: principled curvature directions vectors

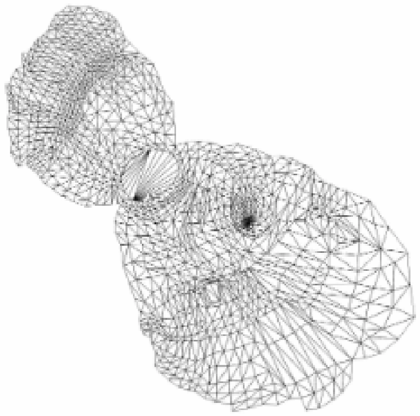


Libigl: Gaussian curvature visualization using pseudocolor

3D models: various representations?

- Different tasks require different representations – **texturing**:
  - Parameterization





CGAL: Least Squares Conformal Maps

3D models: various representations?

- OTHER examples

Materials

- How light interacts with objects?
- TODO

Lights

- How are light sources simulated
- TODO

Cameras

- How are cameras simulated
- TODO

Complex scene

<**IMAGE**: An motivation image that we will understand by the end of the lecture.>

Literature

- https://github.com/lorentzo/IntroductionToComputerGraphics/wiki/Foundations-of-3D-scene-modeling