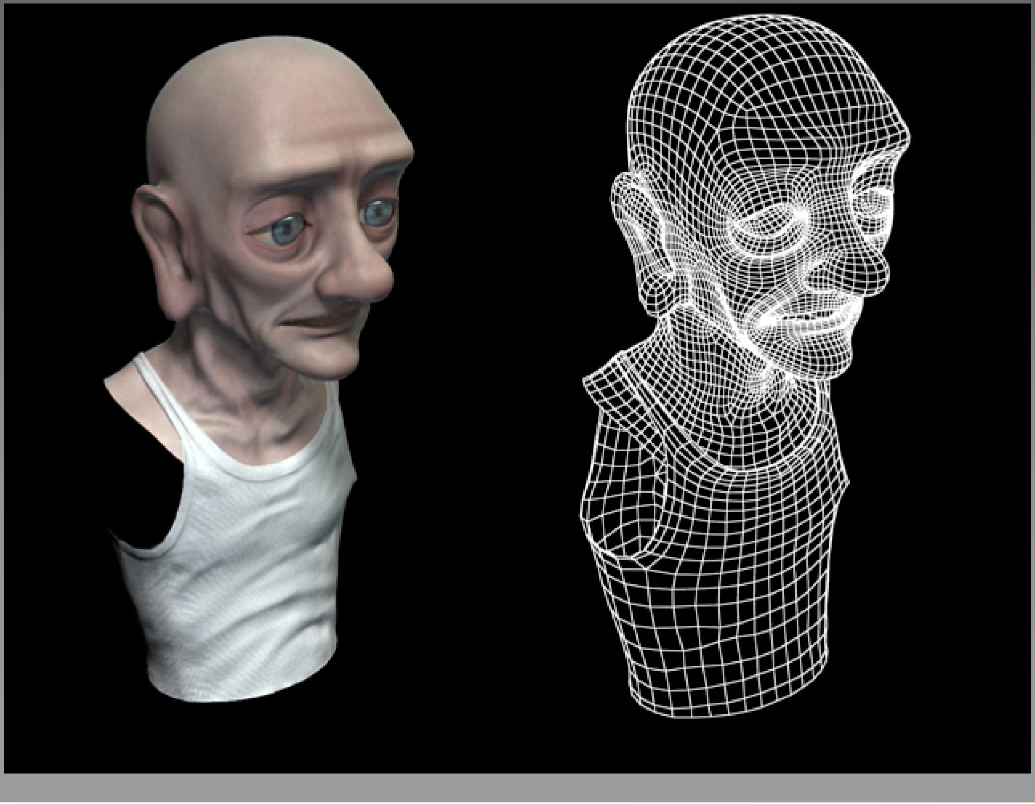
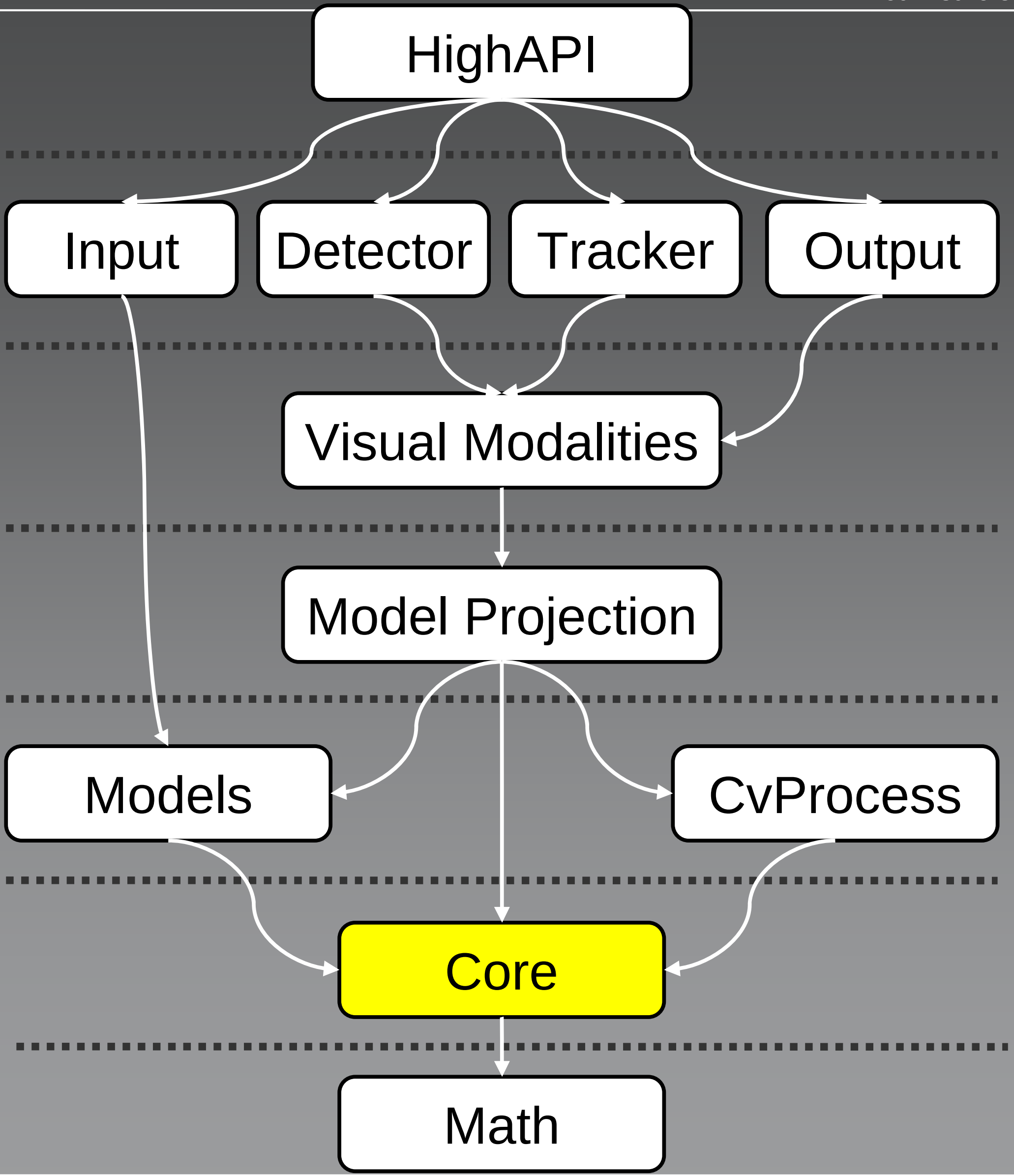
OpenTL – Tutorial 3

- Shape and Appearance representation





HighAPI

Input

Detector

Tracker

Output

Visual Modalities

Model Projection

Models

CvProcess

Core

Math

ShapeAppearance class purpose

- Single container for visible model props
- Store texture information
- Store geometry information
- Basic model related computations
  - Enclosing circle (“circumcircle”)
  - Bounding limits
  - Primitive generation (rectangles, cubes, contours, ellipses, ellipsoids, ...)

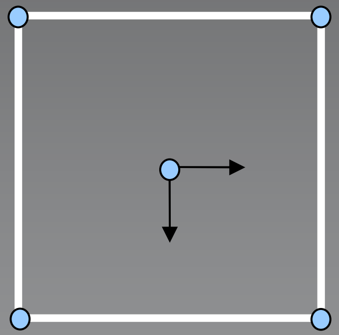
Appearance representation

- Each texture is represented by a cvdata::Image
- Access through a texture/material name
- Textures can be mapped to model faces

- Registering a texture
  - initAppearance(vector<Image*>* images,vector<char*>* names)

Shape representation in object space

- Internal storage: unique vertices and edges
- Object origin



-

Important functions

- Constructor
  - opentl::core::cvdata::ShapeAppearance
- Initialization of model primitives
  - initRectangle(width, height)
  - initEllipse(diagX, diagY, #segments)
  - …

Task – Select and register a image region as a texture

- Use OpenCV GUI to select a ROI
- Initialize a 2d rectangle as the object model
- Initialize the appearance with this ROI and set a name for this texture
- Show the texture in a cvdata::Image