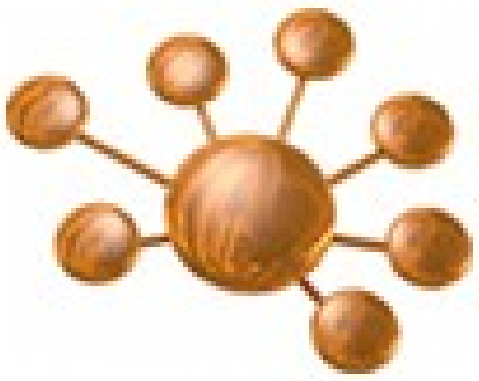
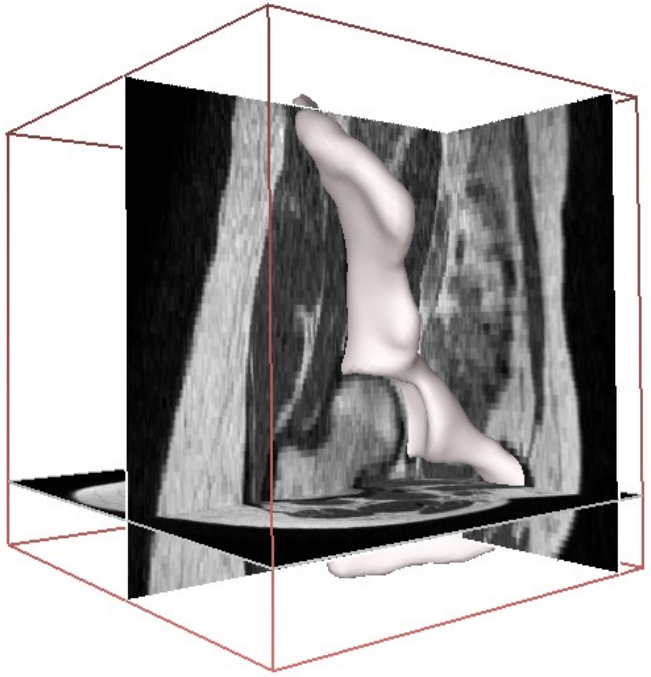
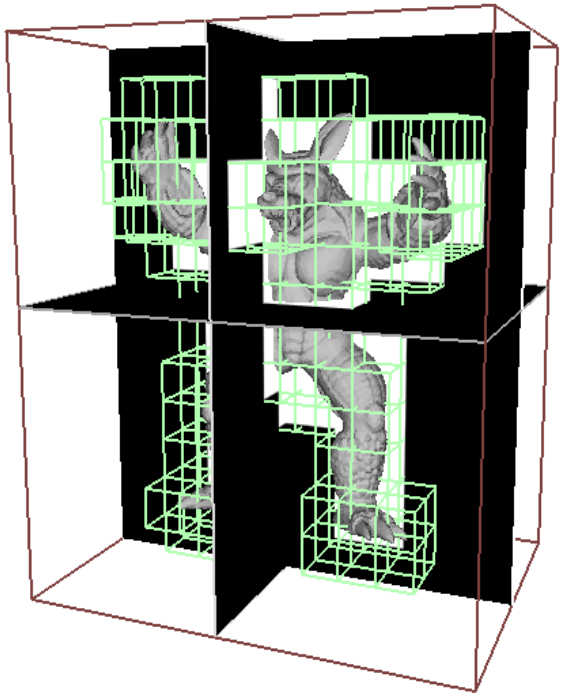


Image Plugin

Benjamin Gilles

16/10/12





Overview

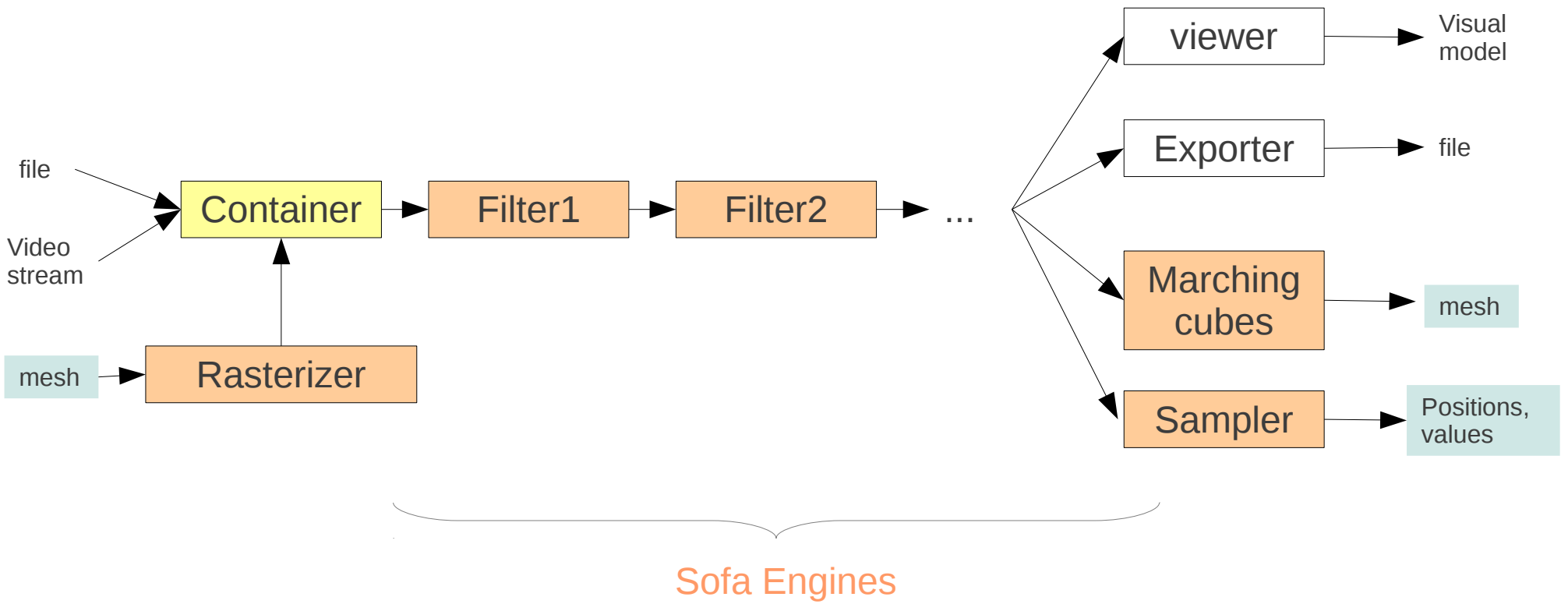

file
Video stream
Container
Filter1
Filter2
...
mesh
Rasterizer
viewer
Visual model
Exporter
file
Marching cubes
mesh
Sampler
Positions, values
Sofa Engines

Image data

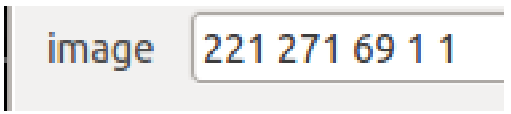
- Based on CImg opensource library http://cimg.sourceforge.net/
  - Import/export in standard formats : bmp, jpeg, mpg, hdr, mhd+raw ...

Handles image orientation, position, pixel size

- Templated on the pixel type
  - ImageC = Image<char>, ImageUC = Image<unsigned char>, ImageD = Image<double>, ImageB = Image<bool>, etc.

- Most components templated on the image type :
  - e.g. <ImageViewer template='ImageD' />

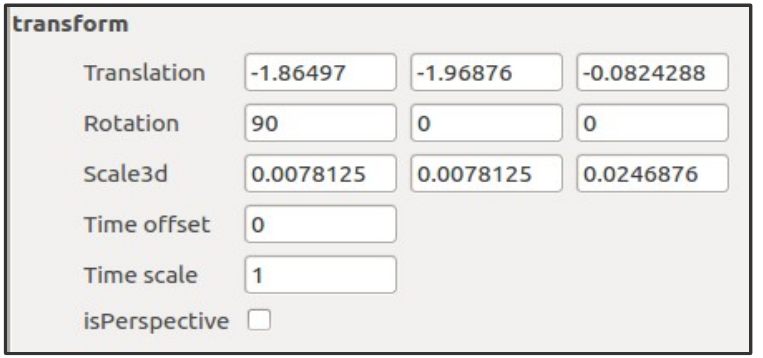
- Five dimensions : x, y, z, channels, t

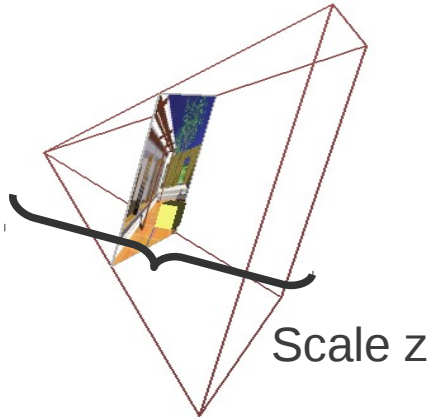


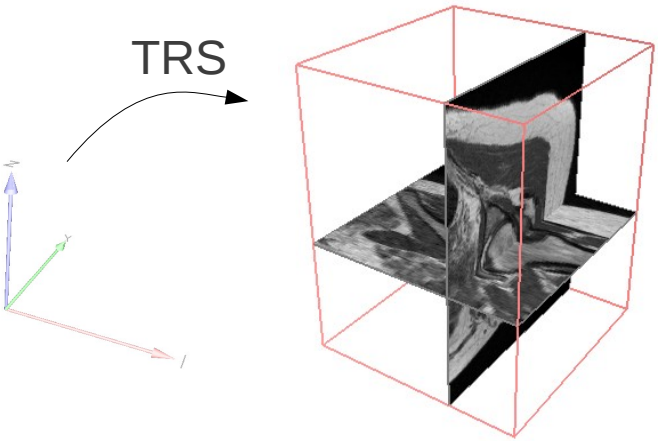
- Shared memory → no overhead when using data links
  - e.g. <ImageFilter inputImage= '@ container.image' />

Transform data

- Each Image is associated to a transformation
  - Encasulated into a single data to simplify linking across components
  - One type to minimize the number of instanciations
  - Linear transformation in the spatio-temporal domain
  - Can be turned into a perspective transformation
    - Pinhole camera intrinsics :
      - fx = scalez /(2 × scalex), fy = scalez /(2 × scaley)
      - cx = (dimx − 1)/2, cy = (dimy − 1)/2

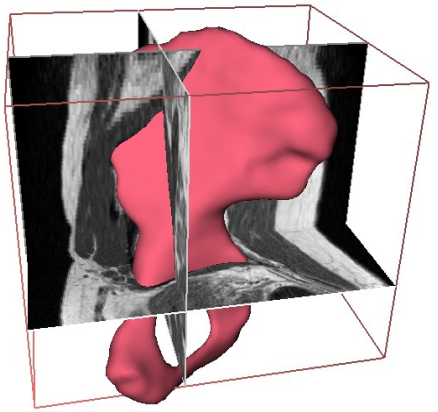


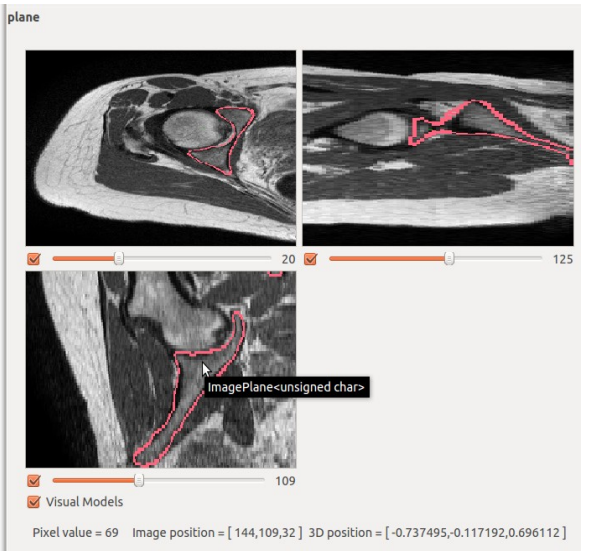




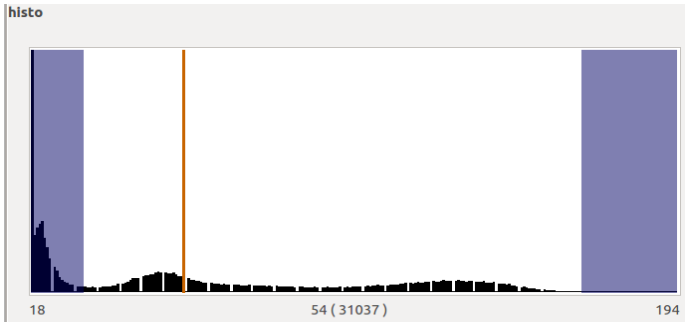
Viewer

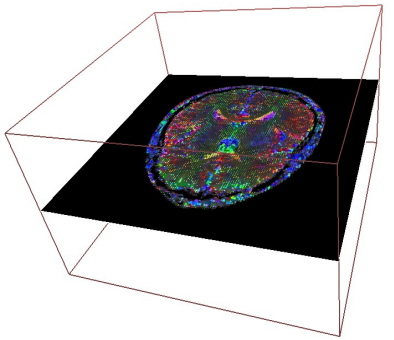
- 'Plane' data to tune multi-planar rendering (MPR)
  - e.g. <ImageViewer plane=' 125 109 20' />
  - Visual model « slice through » visualization
  - Zoom using 'ctrl' key





- 'Histo' data to tune window/level





- Vector/tensor visu for multi channel images

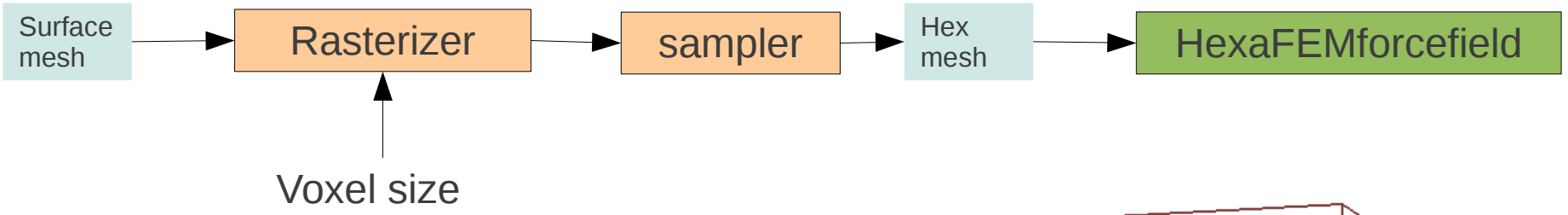
Filtering

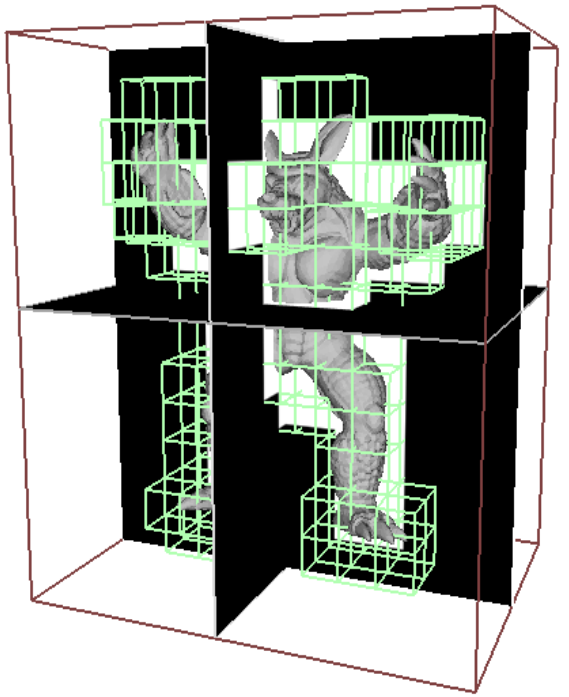
- A single engine for all the standard filters
  - Blur, crop, threshold, distance, resample, etc...
  - e.g. smoothing :

```
<Node>
    <ImageContainer  name="image" filename="data/pelvis_f.raw" />

    <ImageFilter  name="engine"  filter="1" param="2" inputImage="@image.image" inputTransform="@image.transform" />

                                                                                    src="@image"

    <ImageViewer       image="@filter.outputImage" transform="@image.outputTransform" />

                                                                src="@engine"

</Node>
```

Generation of volumetric meshes

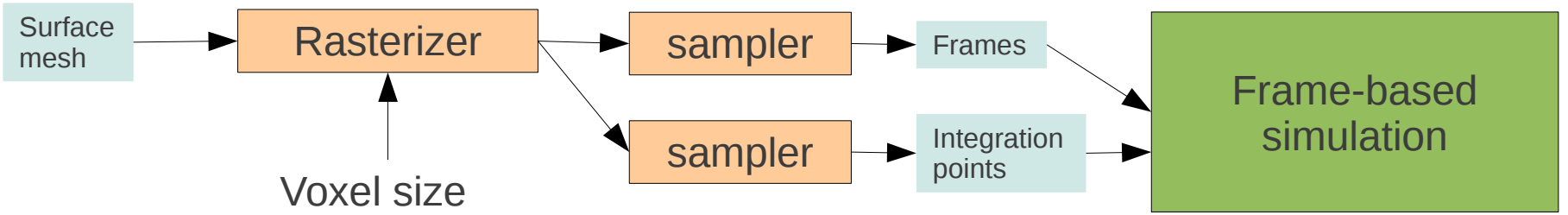
sampler_HexaFEM.scn

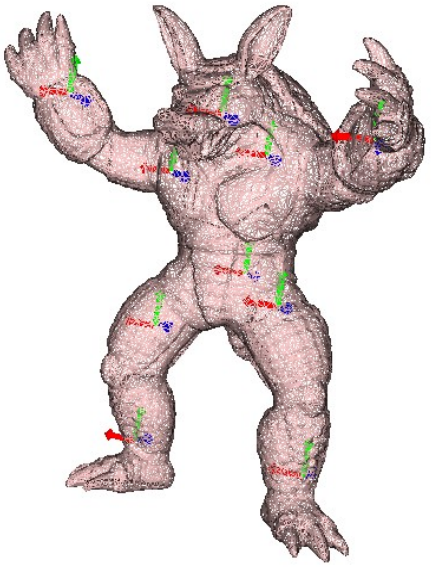




Generation of frame model

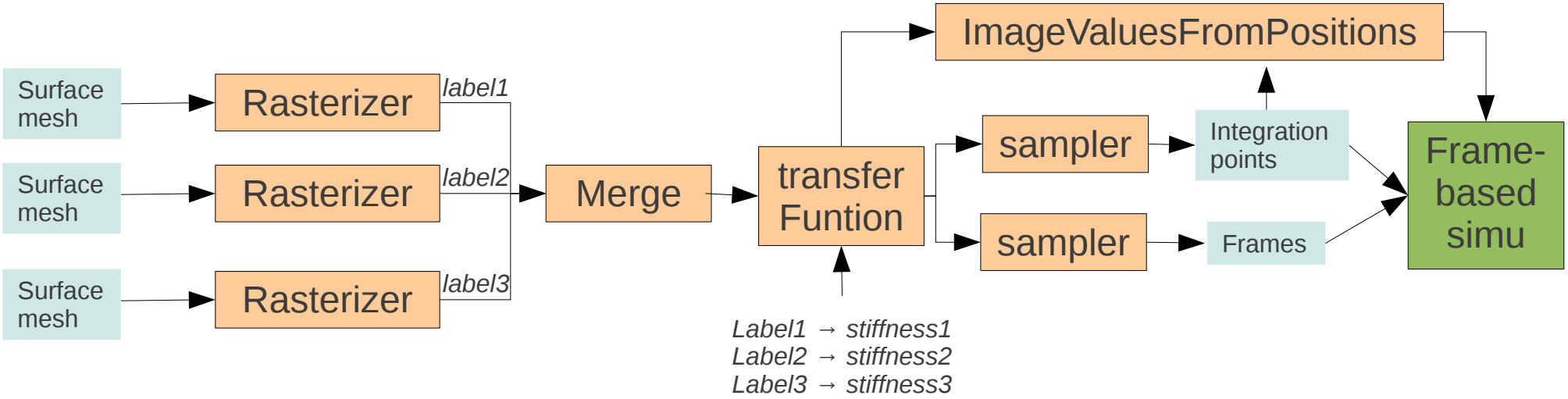
MeshToImage_Frame.scn

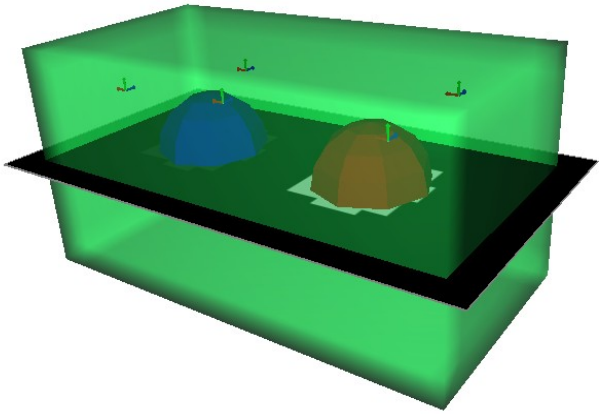




Representing heterogeneous materials

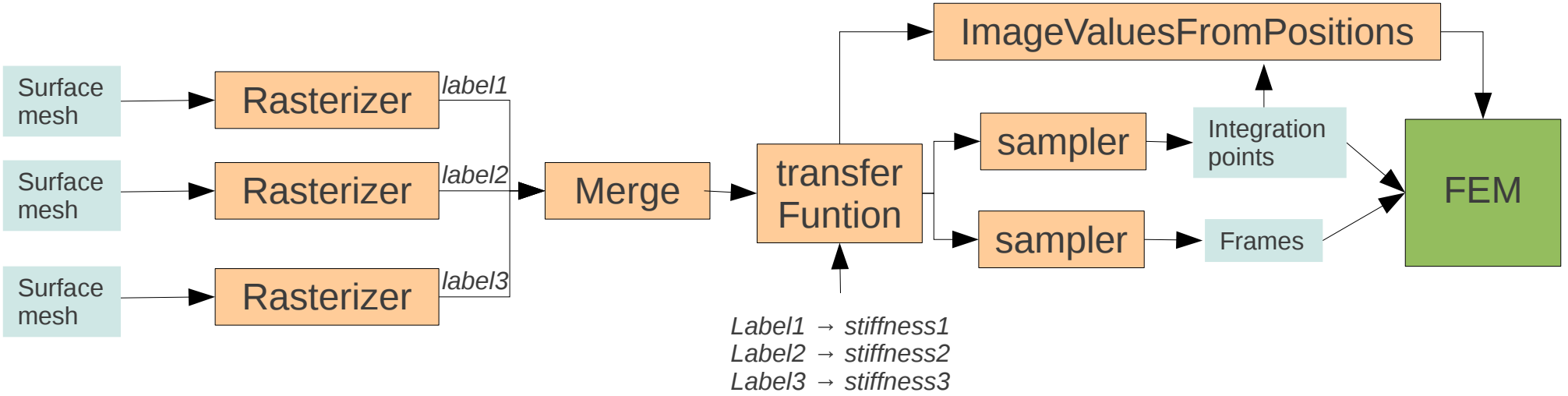
MeshToImage_Frame2.scn

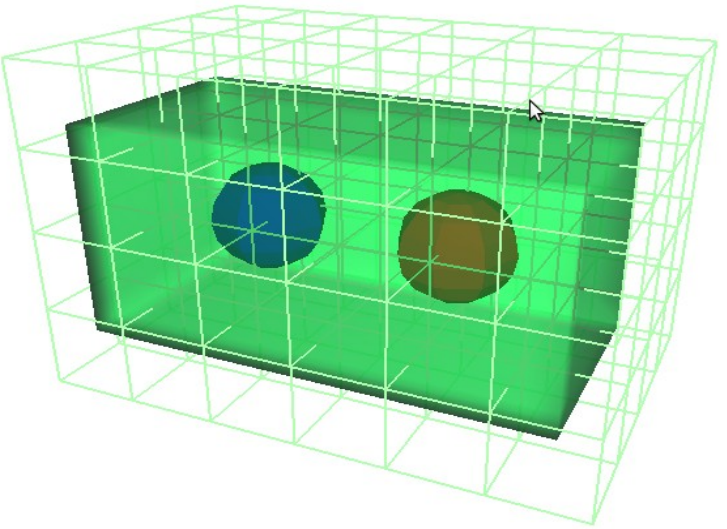




Representing heterogeneous materials

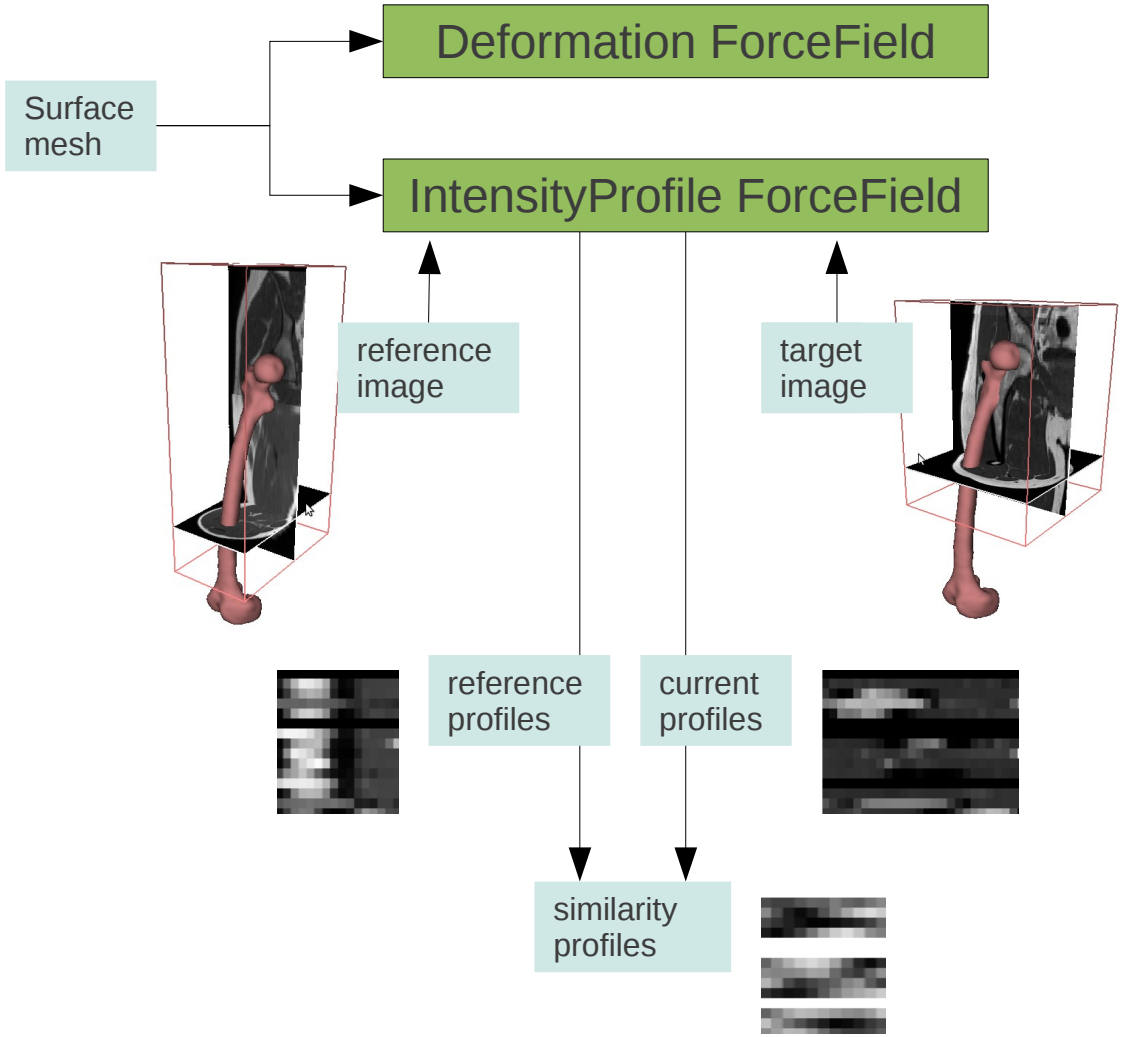
MeshToImage_Hexa.scn

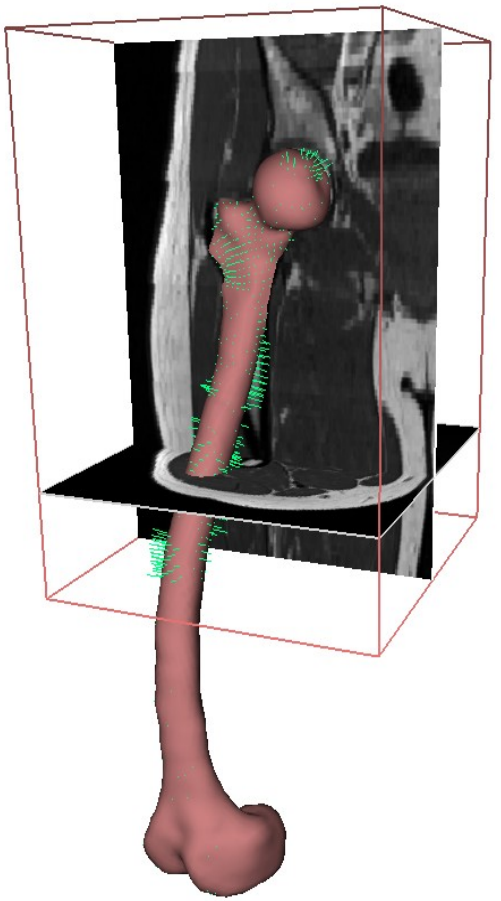




Registration

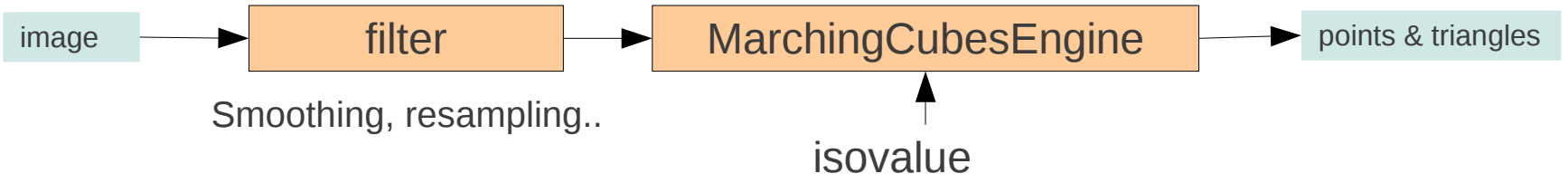
Registration/imageregistration.scn

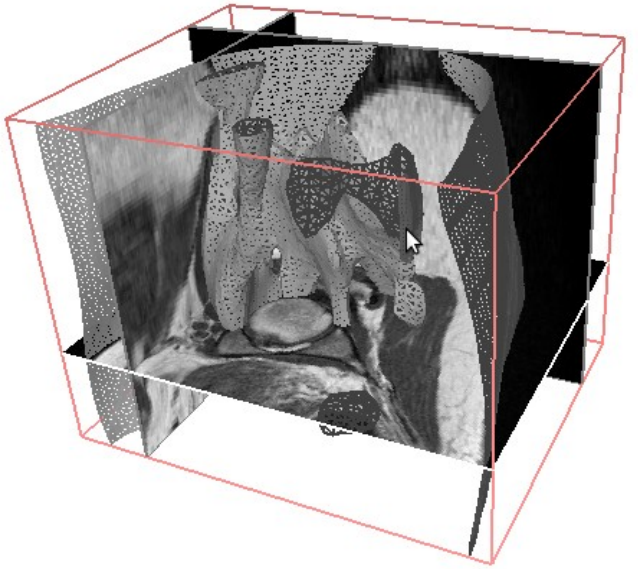




Generation of iso-surfaces

marchingCubes.scn





Generation of a complete simulation

Flexible/plate.scn

image

transfer Funtion

ImageValuesFromPositions

sampler

Integration points

sampler

Frames

Frame-based simu

Label1 → stiffness1
Label2 → stiffness2
Label3 → stiffness3

MarchingCubesEngine

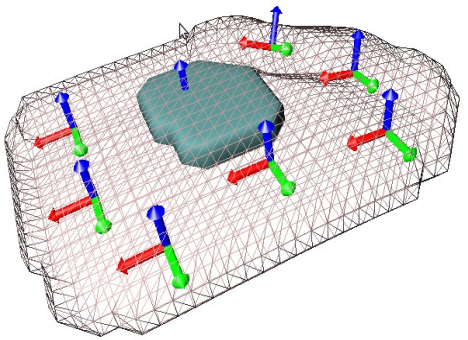
Surface mesh 1

isovalue1

MarchingCubesEngine

Surface mesh 2

isovalue2



Video streaming

testCam.scn

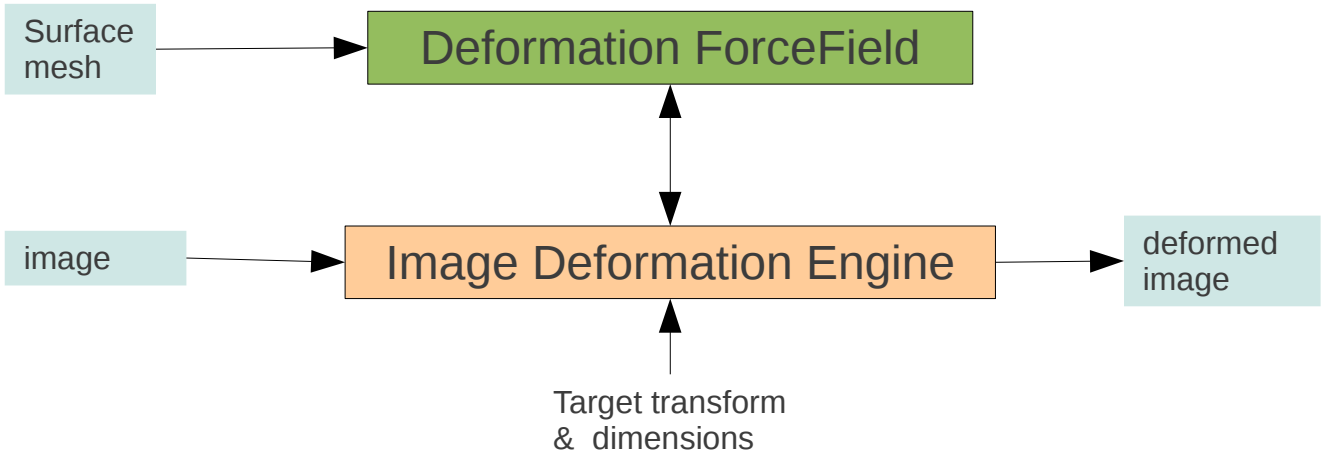
- Opencv camera
- ImageAccumulator : creates 4d+t data and handles synchro

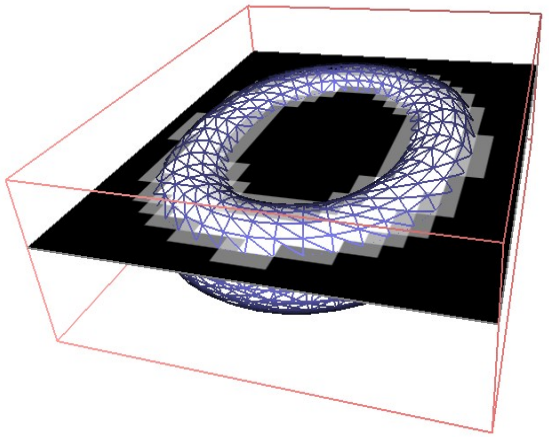
kinect.scn

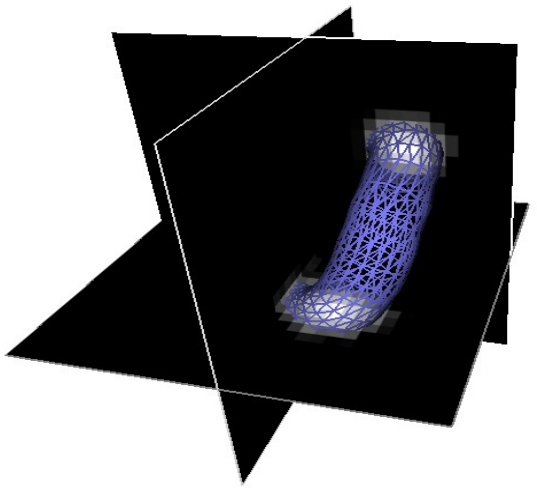
- Kinect + depthMapToMeshEngine
  - Mix simulation and 4d captures for validation, interaction, etc.

Image deformation

Flexible/imageDeformation.scn







Future work

- Volume rendering, simulation of X-ray images
- 2d/3d textures using image types
- Image based collision models
- Cutting
- More registration methods
- More transformations
- Deformation models in Eulerian setting
- Anisotropic materials