MATLAB 3D Rendering

Surface Rendering

盧家鋒 Chia-Feng Lu, Ph.D.
Department of Biomedical Imaging
and Radiological Sciences, NYCU
alvin4016@nycu.edu.tw

Contents

- Surface rendering and processing

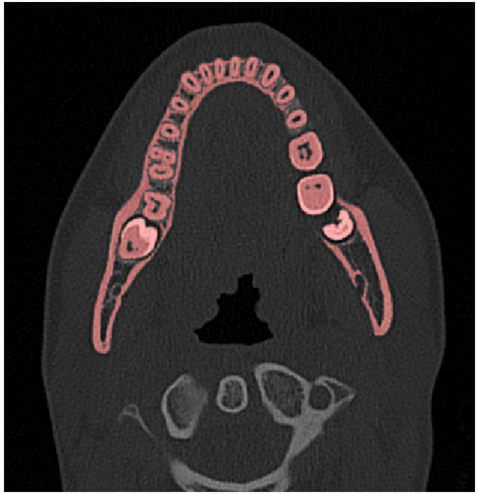
Please download the handout and materials from (Week 12)

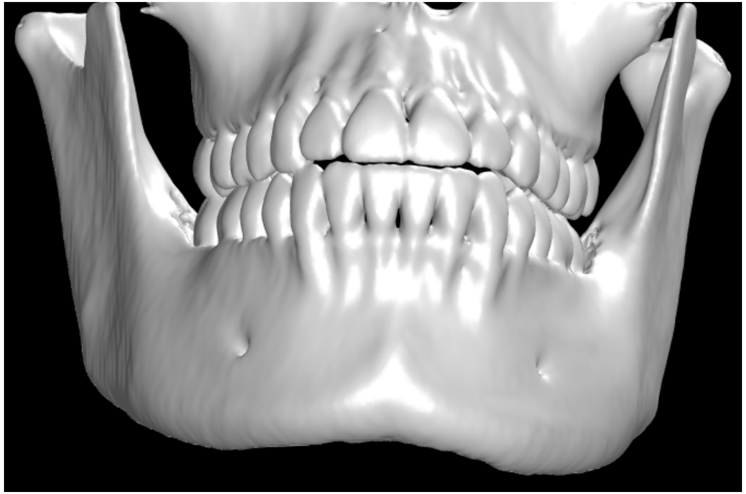
http://cflu.lab.nycu.edu.tw/CFLu_course_matlabimage.html

3D Rendering

- Surface Rendering
  - A binary rendering technique (A pixel is within a certain threshold or not).
  - Threshold can be rendered into surfaces(isosurface)

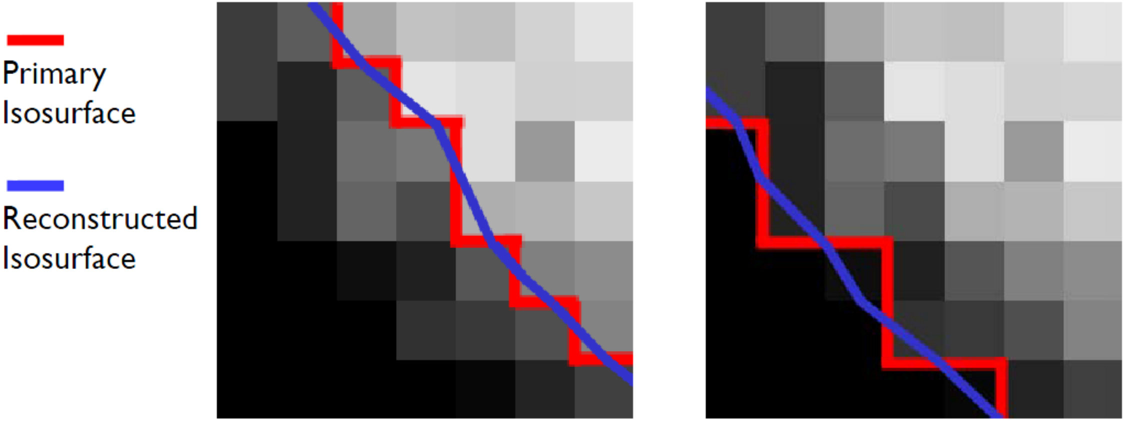
roipoly + thresholding





Threshold based rendering

- Pixels at tissue interfaces can be determined by a specific threshold.

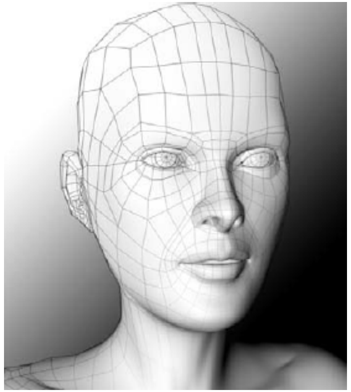


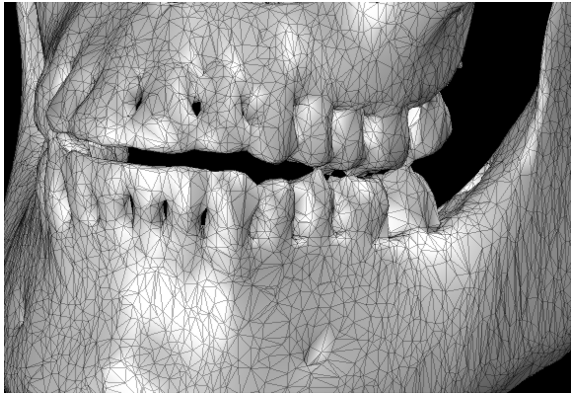
high threshold

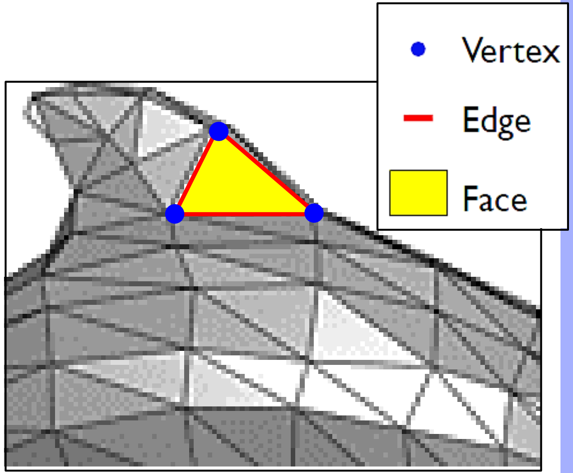
low threshold

Polygonal modeling of surface

- Polygons consists of **vertices**, **edge** and **faces**.
- **Marching cubes algorithm**: One of the first surface rendering algorithms



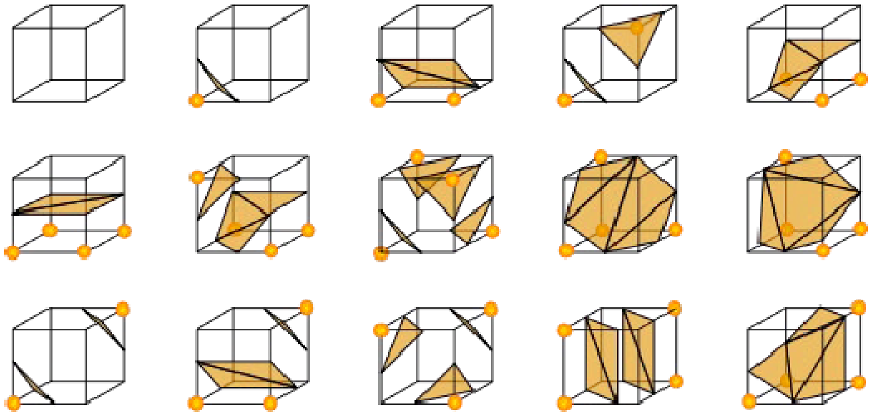




Marching cubes algorithm

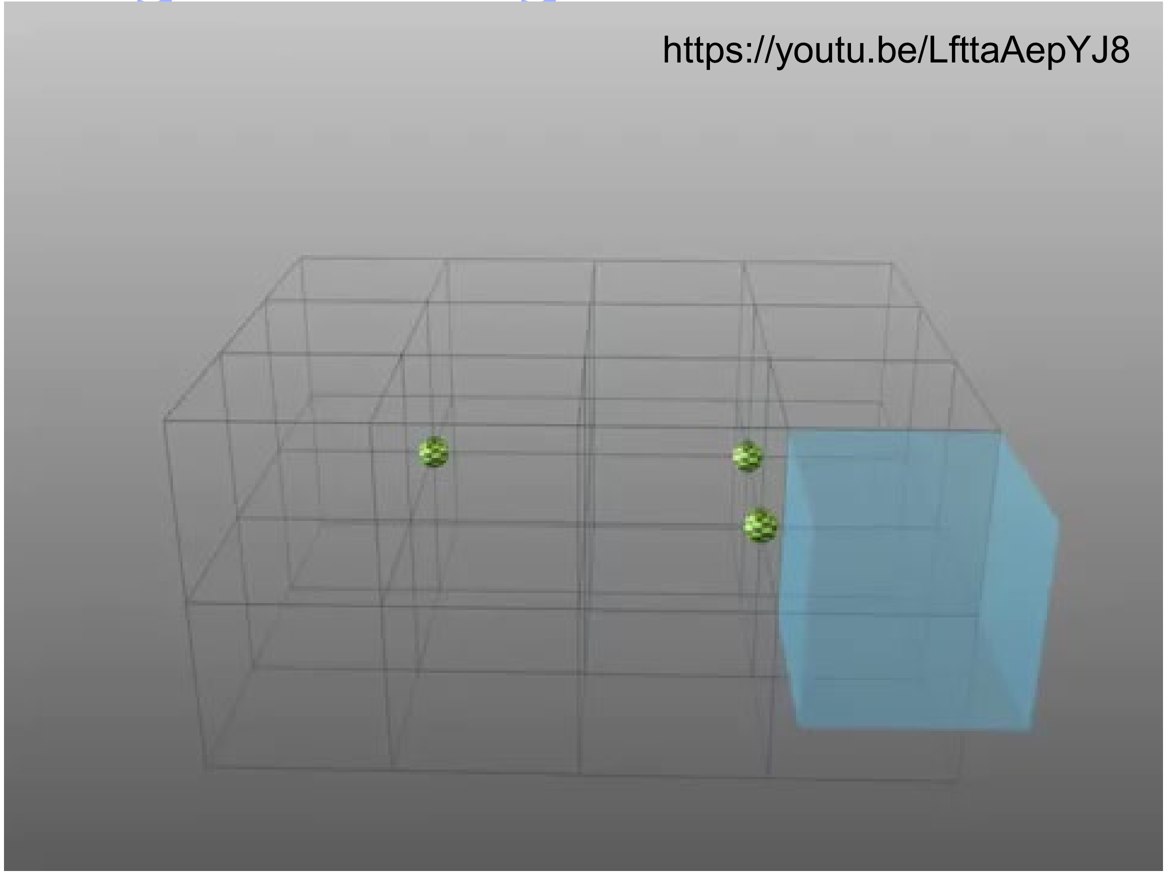
- Surfaces are arranged in triangles
- Algorithm calculates where surface crosses the voxel, "marching" from one cube to the other.

15 unique cases



Lorensen et al, Computer Graphics, 1987.

Marching cubes algorithm



https://youtu.be/LfttaAepYJ8

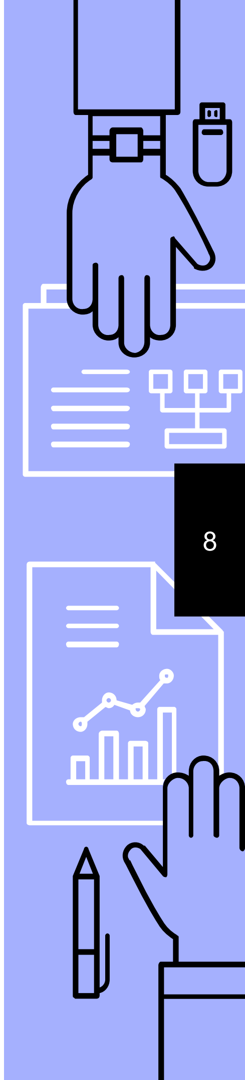
Matlab isosurface

isosurface Isosurface extractor.

[F,V] = isosurface(V,ISOVALUE)

computes isosurface geometry for data V at isosurface value ISOVALUE.

The struct FV contains the faces and vertices of the isosurface and can be passed directly to the PATCH command.



Take a Cube as an Example

- A=zeros(10,10,10);
- A(2:9,2:9,2:9)=1;

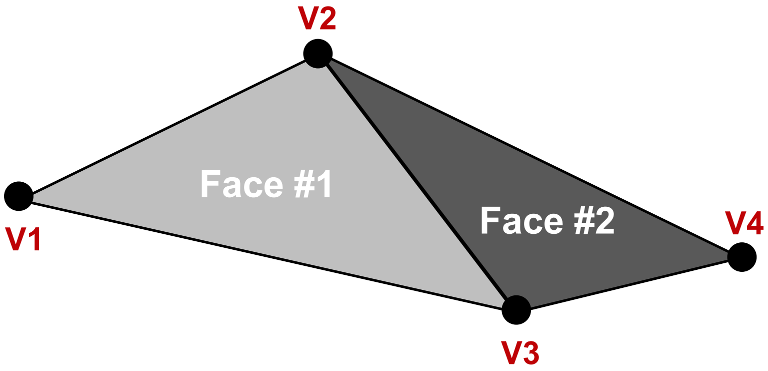
- [F,V] = isosurface(A,0);

F

1	2	3

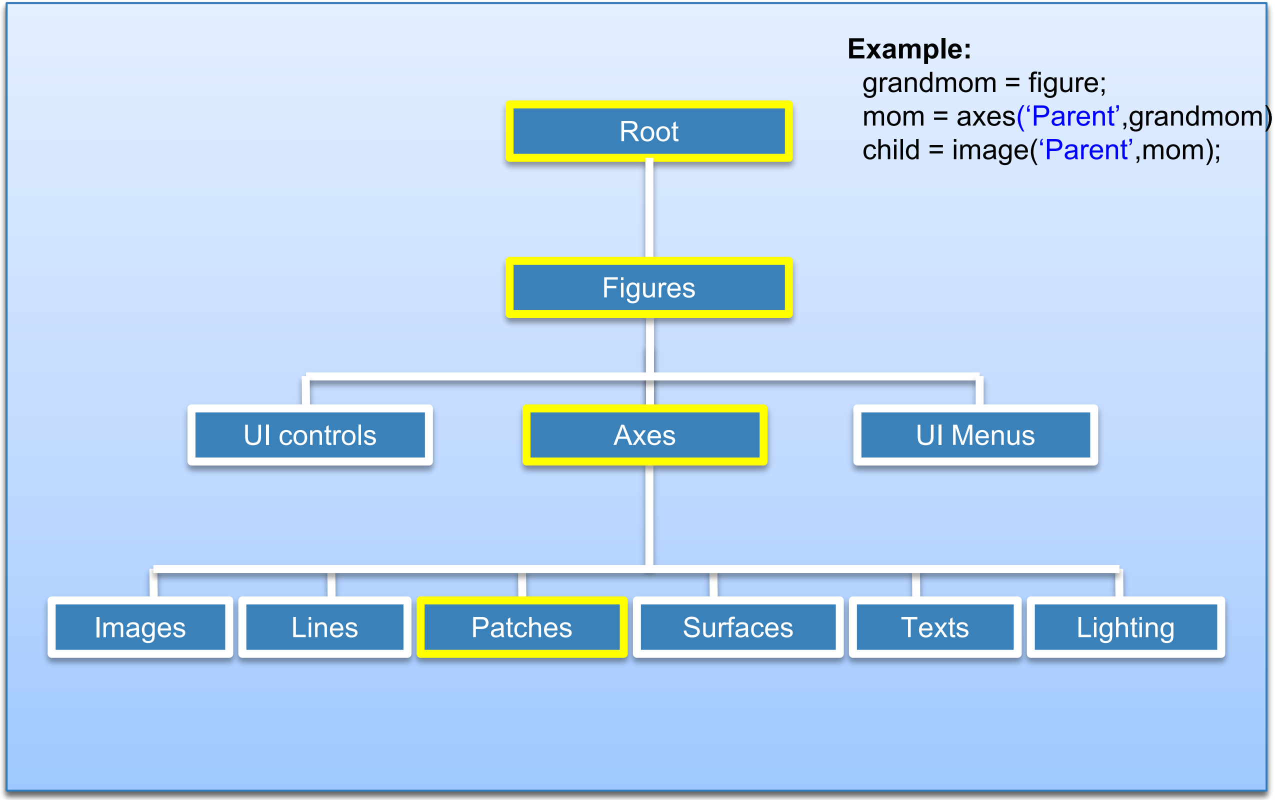
V

4	10	2
5	2	1
5	1	2



Each face is composed of 3 vertices.

Hierarchical Relations of Objects



Example:

```
grandmom = figure;
mom = axes('Parent',grandmom);
child = image('Parent',mom);
```

patch

patch Create one or more filled polygons

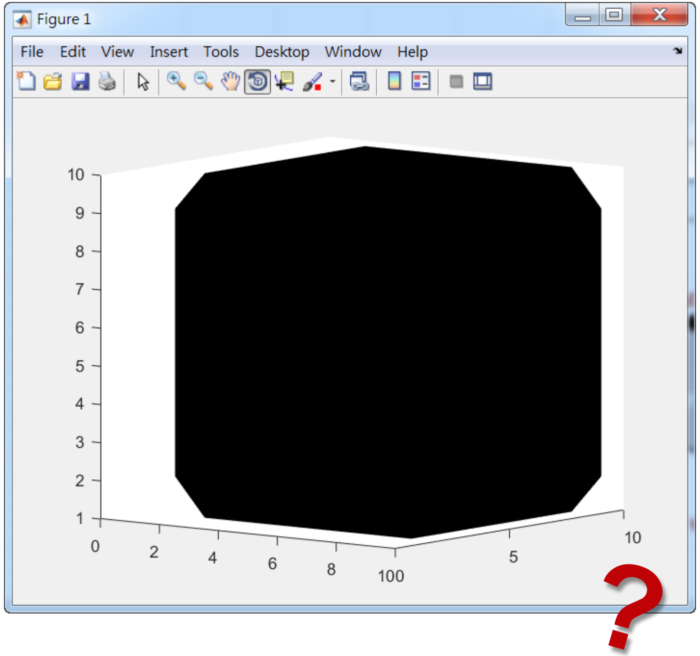
- patch('Faces',F,'Vertices',V) creates one or more polygons where V specifies vertex values and F defines which vertices to connect.

patch

- A=zeros(10,10,10);
- A(2:9,2:9,2:9)=1;

- [F,V] = isosurface(A,0);

- figure
- patch('Faces',F,'Vertices',V)



Light it up! ⇔ shadow

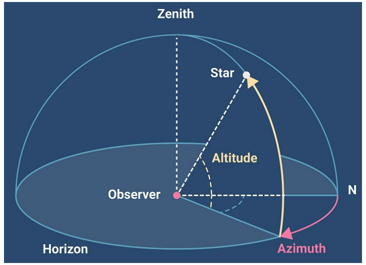
- patch('Faces',F,'Vertices',V,'FaceColor',[0 1 0])
- **lighting** gouraud
- **camlight**(0,0)

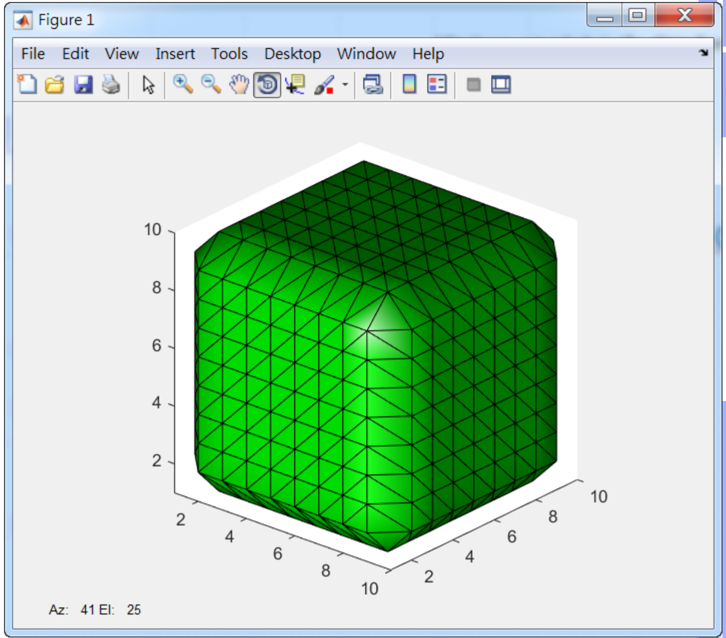
% azimuth (horizontal rotation) and

% vertical elevation (both solid angles

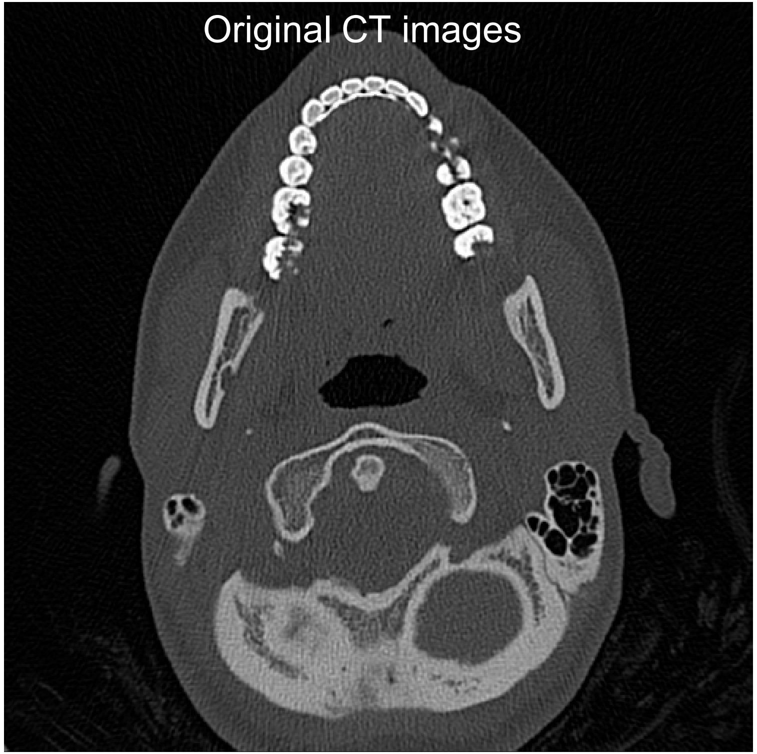
% in degree)

- axis equal

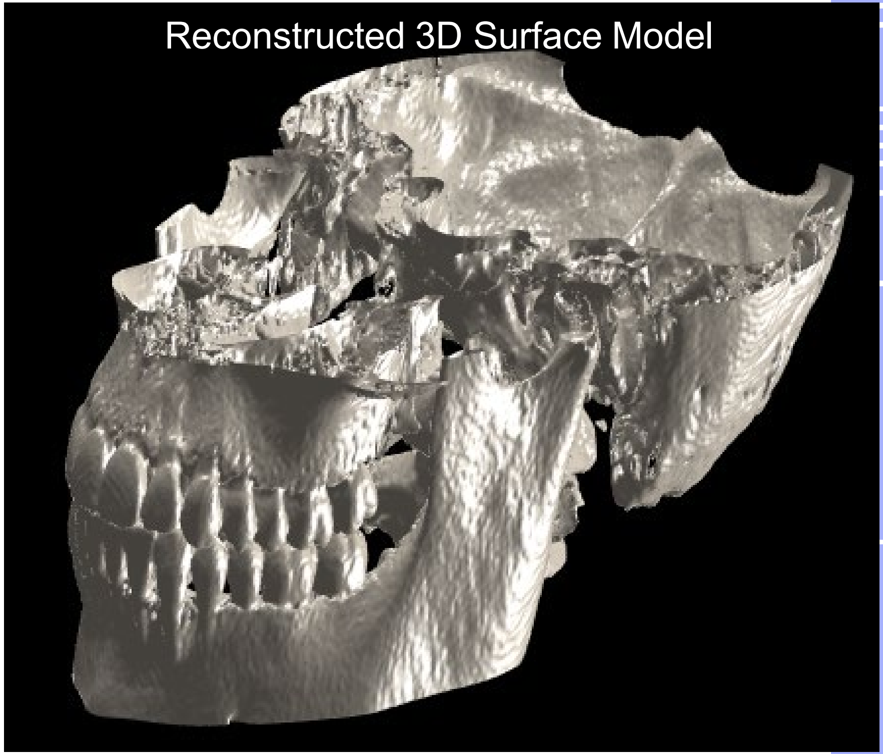




Dental CT Images

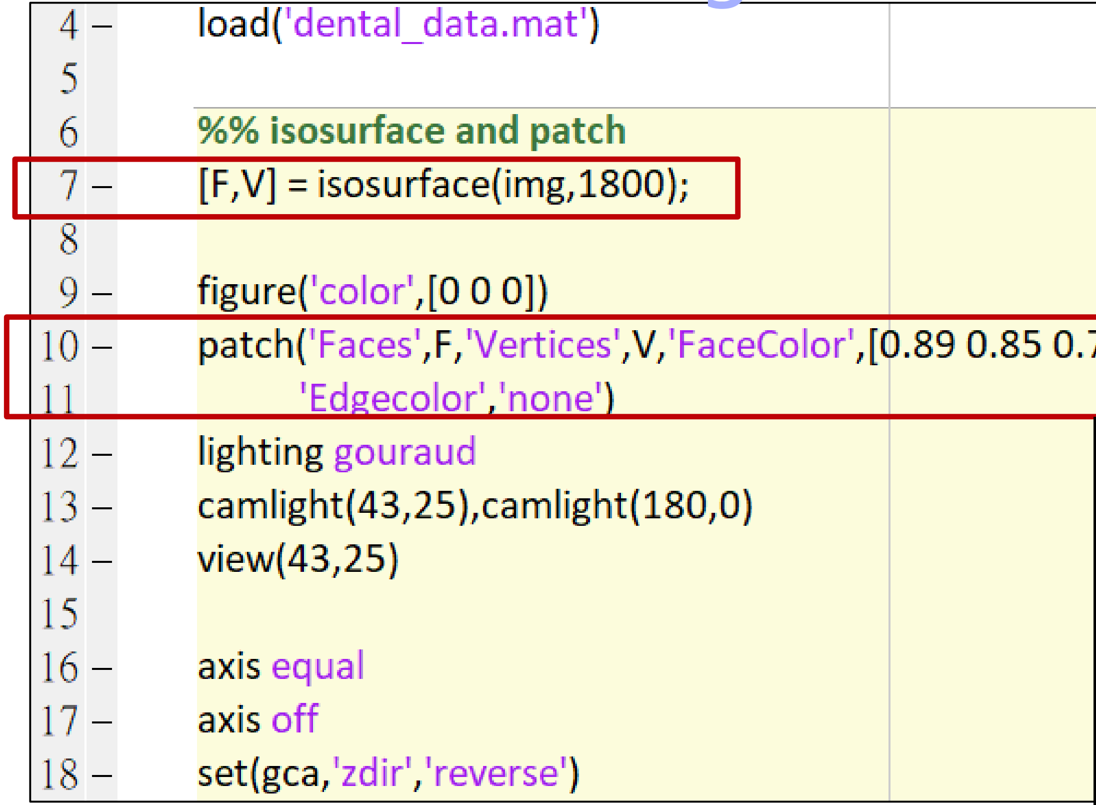


Original CT images



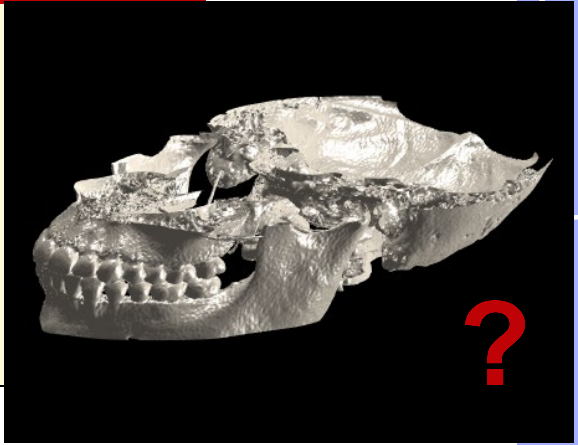
Reconstructed 3D Surface Model

Surface Rendering of Dental CT



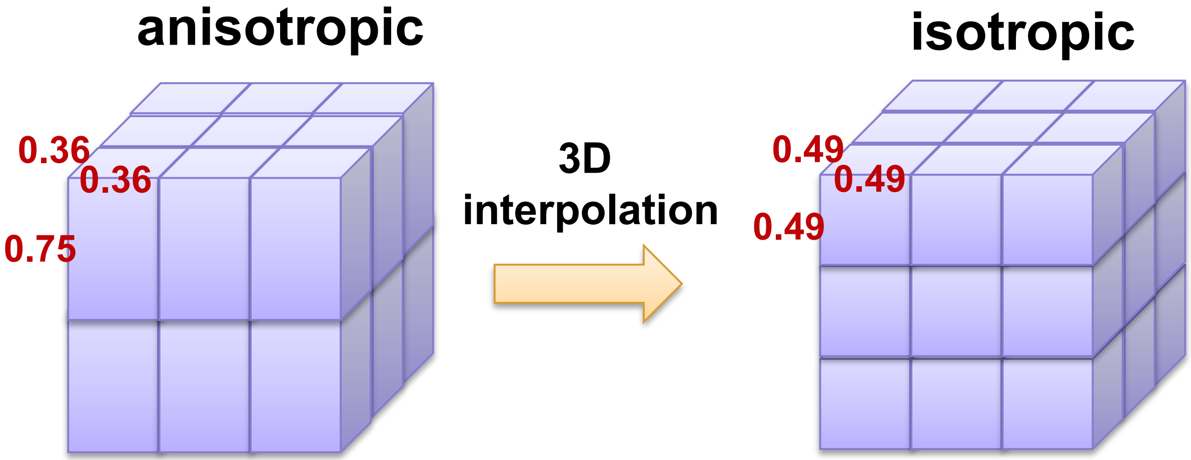
```
load('dental_data.mat')

%% isosurface and patch
[F,V] = isosurface(img,1800);

figure('color',[0 0 0])
patch('Faces',F,'Vertices',V,'FaceColor',[0.89 0.85 0.79],...
    'Edgecolor','none')
lighting gouraud
camlight(43,25),camlight(180,0)
view(43,25)

axis equal
axis off
set(gca,'zdir','reverse')
```

Unreal ratio

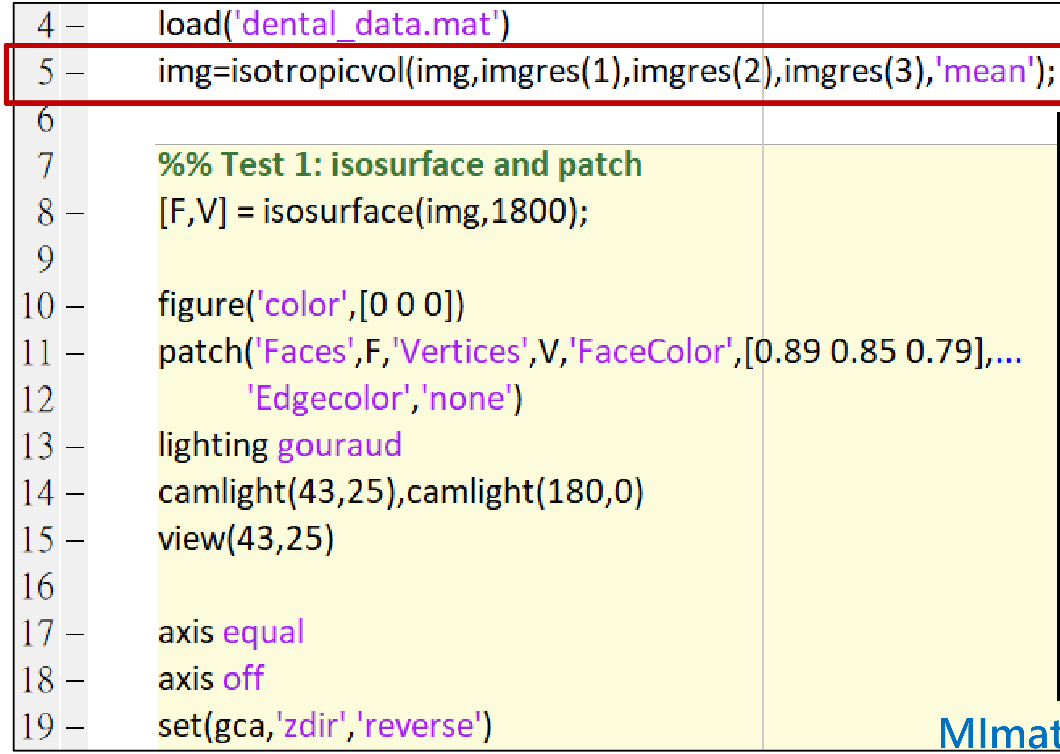


Isotropic Voxels

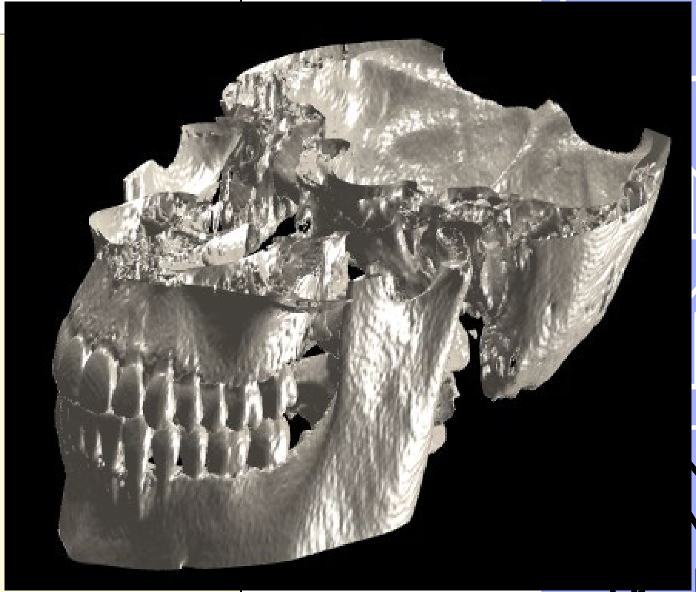
- img=isotropicvol(img,0.36,0.36,0.75,'mean');



Test 1 – isotropicvol + isosurface

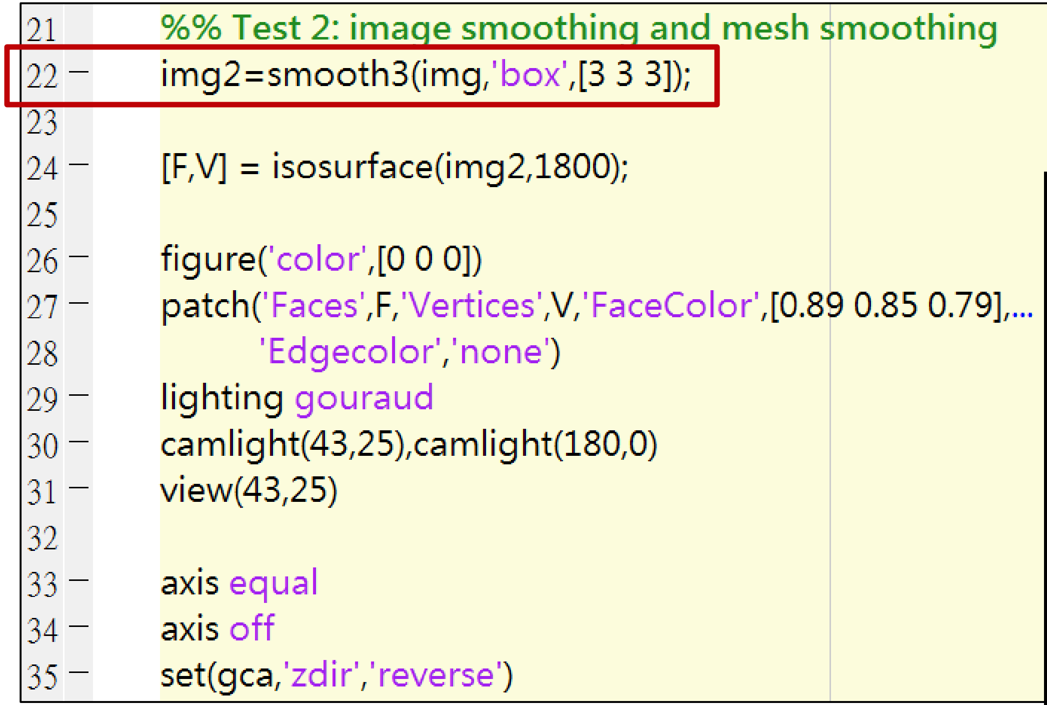


```
load('dental_data.mat')
img=isotropicvol(img,imgres(1),imgres(2),imgres(3),'mean');

%% Test 1: isosurface and patch
[F,V] = isosurface(img,1800);

figure('color',[0 0 0])
patch('Faces',F,'Vertices',V,'FaceColor',[0.89 0.85 0.79],...
    'Edgecolor','none')
lighting gouraud
camlight(43,25),camlight(180,0)
view(43,25)

axis equal
axis off
set(gca,'zdir','reverse')
```

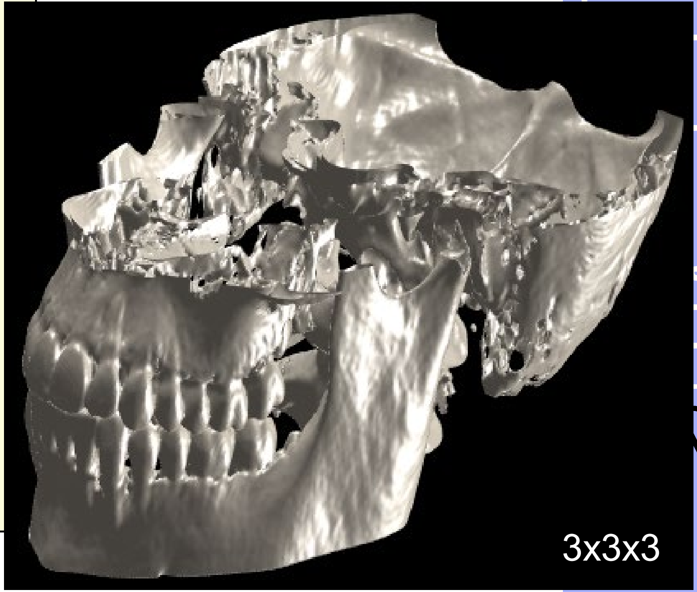


MImaterials_L12\Dental_model.m

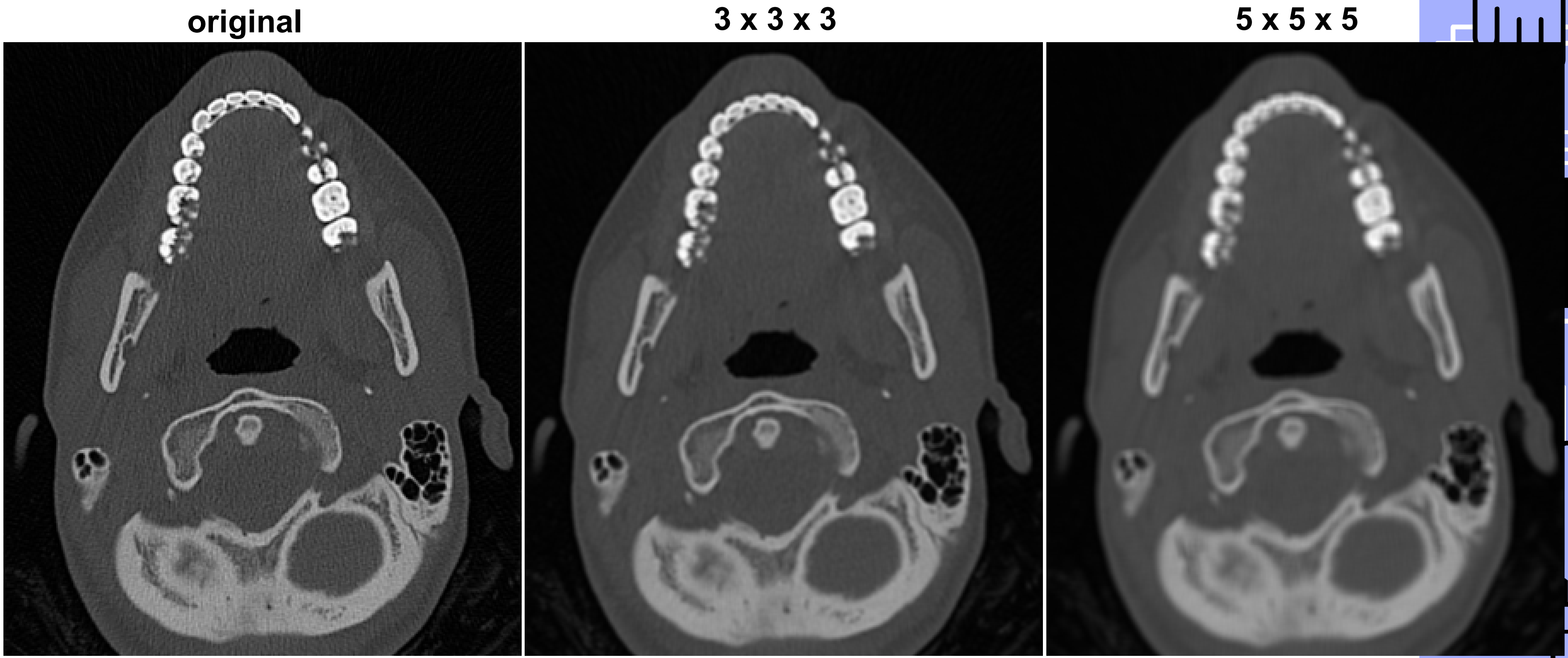
Test 2 – image smoothing



```
%% Test 2: image smoothing and mesh smoothing
img2=smooth3(img,'box',[3 3 3]);

[F,V] = isosurface(img2,1800);

figure('color',[0 0 0])
patch('Faces',F,'Vertices',V,'FaceColor',[0.89 0.85 0.79],...
        'Edgecolor','none')
lighting gouraud
camlight(43,25),camlight(180,0)
view(43,25)

axis equal
axis off
set(gca,'zdir','reverse')
```



Test 2 – image smoothing

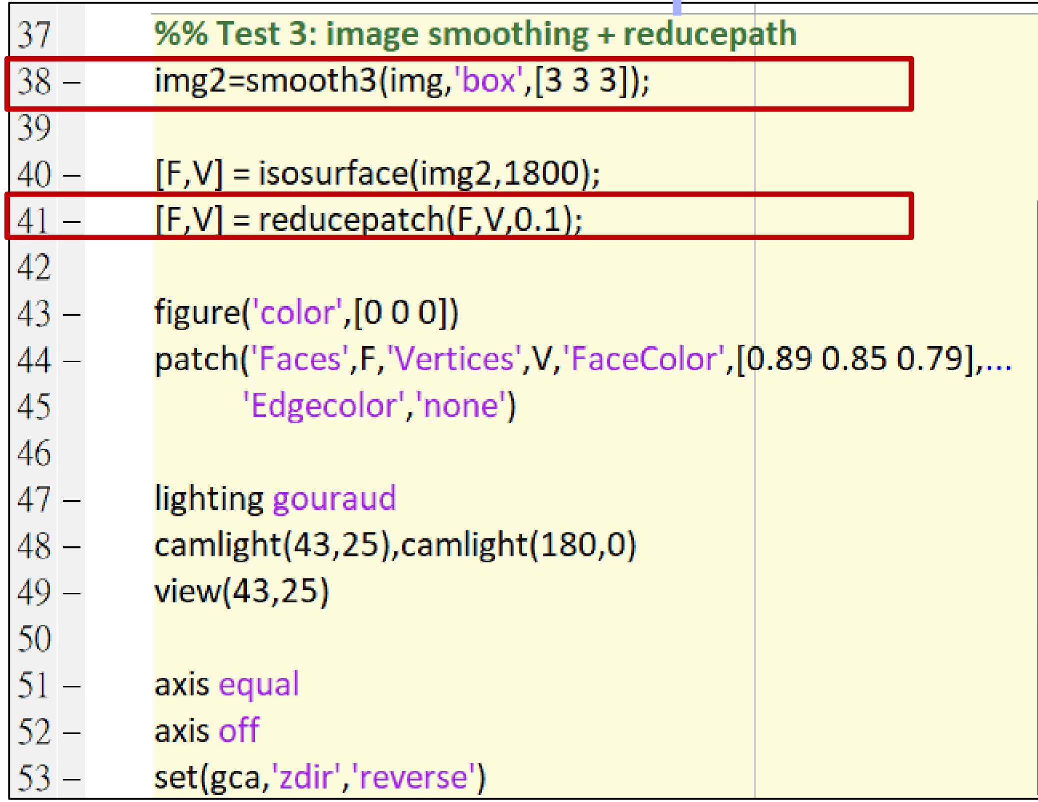


original

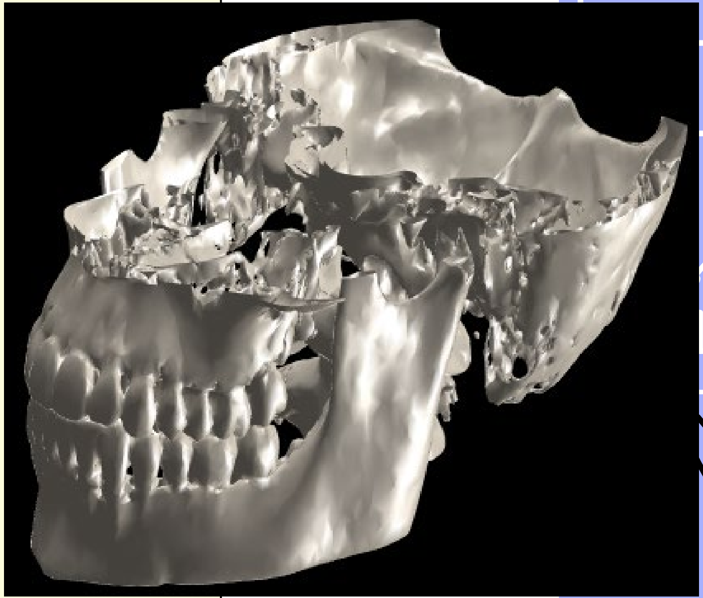
3 x 3 x 3

5 x 5 x 5

Test 3 – image smoothing + reducepatch

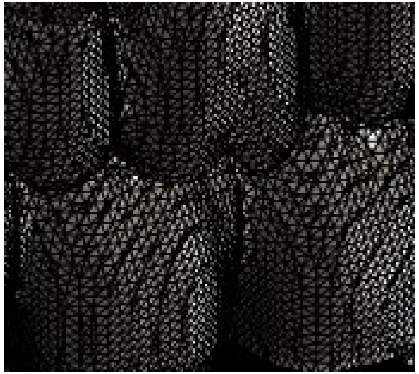


```
%% Test 3: image smoothing + reducepath
img2=smooth3(img,'box',[3 3 3]);

[F,V] = isosurface(img2,1800);
[F,V] = reducepatch(F,V,0.1);

figure('color',[0 0 0])
patch('Faces',F,'Vertices',V,'FaceColor',[0.89 0.85 0.79],...
        'Edgecolor','none')

lighting gouraud
camlight(43,25),camlight(180,0)
view(43,25)

axis equal
axis off
set(gca,'zdir','reverse')
```

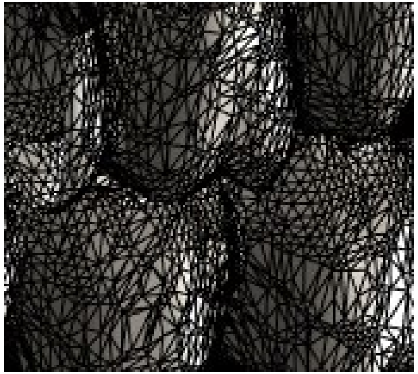


Test 3 – image smoothing + reducepatch

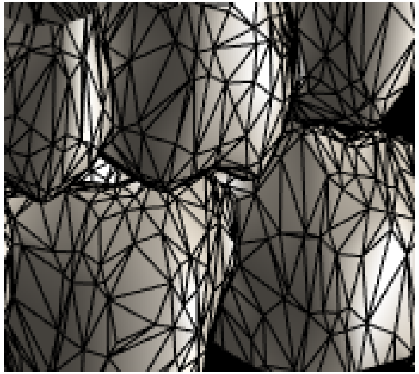
Full patch



reducepatch to 50%

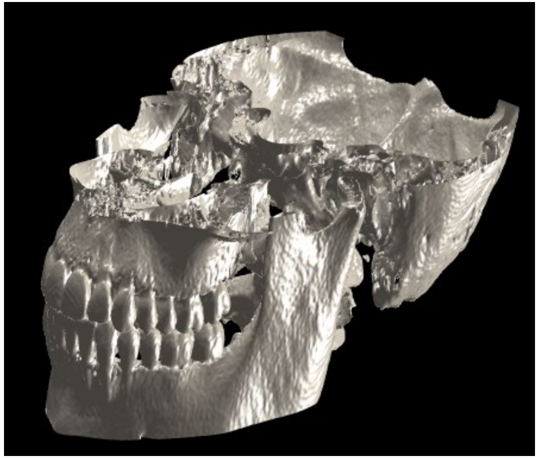


reducepatch to 10%

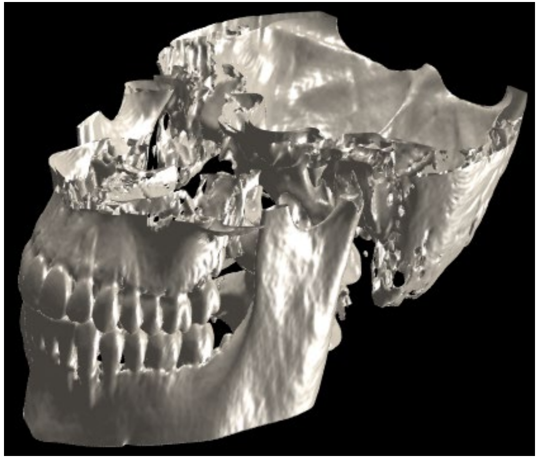


Comparisons between rendering models

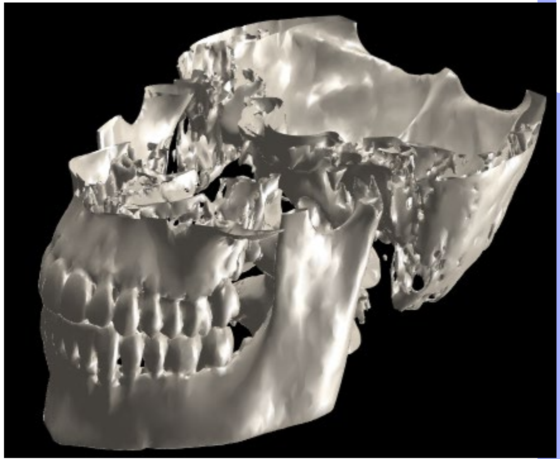
<Test 1>
Direct surface rendering



<Test 2>
Image smoothing



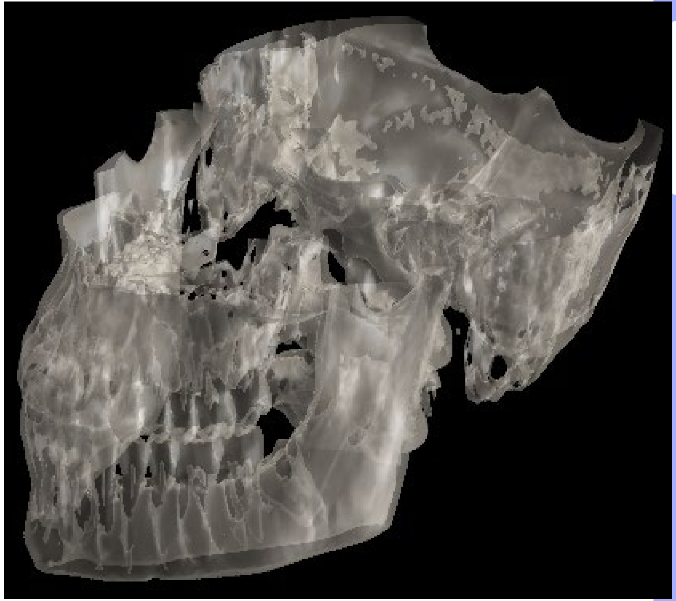
<Test 3>
Image smoothing + reducepatch



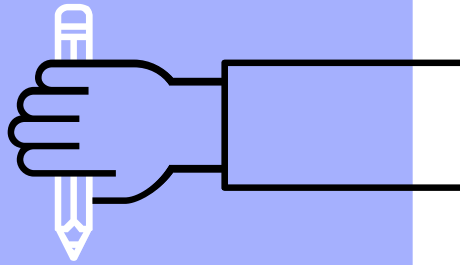
MImaterials_L12\Dental_model.m

Homework

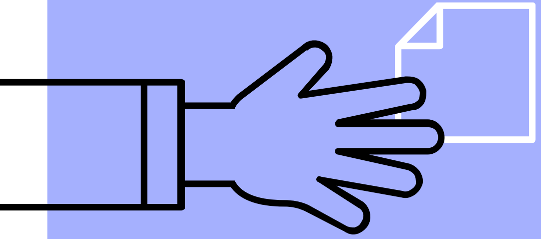
- Please create a surface model with **the transparency of 40%** as shown in the figure.



Please modified from Test 3 section in
MImaterials_L12\Dental_model.m



THE END



alvin4016@nycu.edu.tw