Template of Manipulator Short project: Skull tumor surgery

Authors:

Pablo Montón Gimeno

Mauro García Lorenzo

Team: L

Link: https://drive.matlab.com/sharing/205cab5a-35f7-4f76-98c6-daea26e86ecd

Shared Link with the teacher:

I spect: 1) Pdf file, 2) Videos demostrating your successful task and your mlx file.

Notes. For better undestanding you can split the videos in the meaninful task.

Remember use the options of serial/link plot:

'workspace' for centering in the surgery task

'zoom' ... nice puma ratio aspect

'trail' .. to see the trajectory

etc..

See all at:

>> help SerialLink/plot

Table of Contents

The Robotic environment (10%)

Think that later on the enviroment will move to any place in a Univers Reference Frame {U}

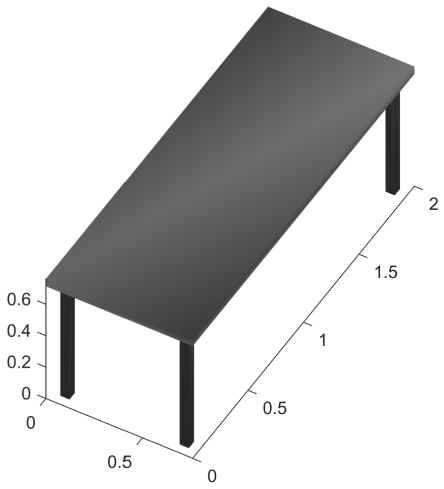
Use: 'c = uisetcolor' to chose your prefered colors

Operating table.

It can be raised, lowered, and tilted in any direction, and an auxiliary table for the tools. Define: Vertices and Faces and use 'patch' functions to model it. See help patch to find and example.

Think that later on the enviroment will move to any place in a Univers Reference Frame {U}

Expected results

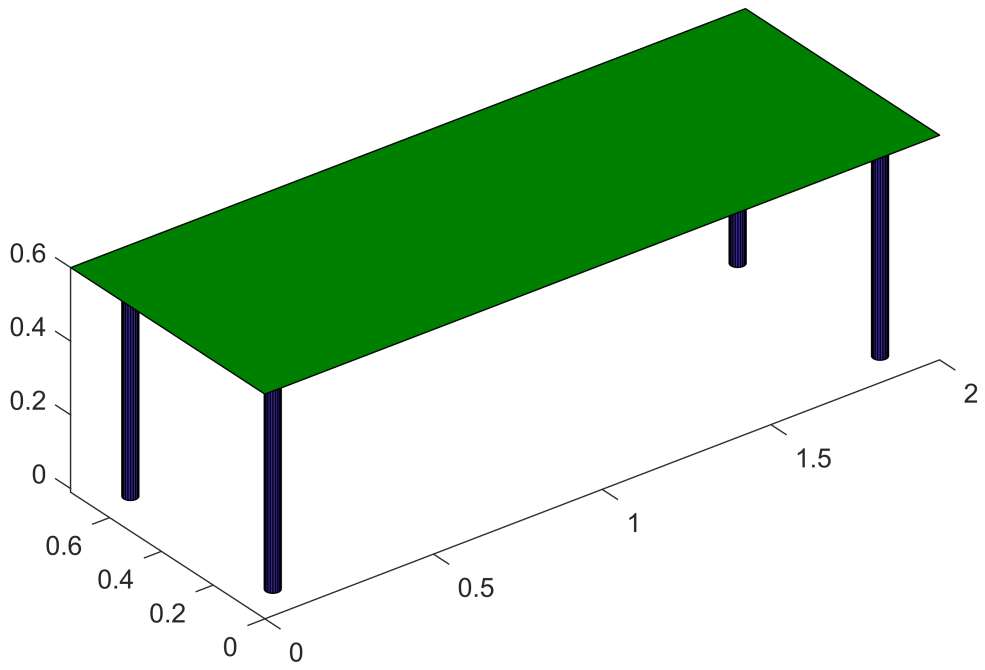


put your code Here

```
clear
clf
v = [0 0 0.6; 0 0.75 0.6; 2 0.75 0.6; 2 0 0.6];
f = [1 2 3 4];
view(3)
patch('Faces',f,'Vertices',v,'FaceColor',[0 0.5 0])
r = 0.02;
[X,Y,Z] = cylinder(r);
h = 0.6;
Z = Z*h-0.01;
hold on
```

```
surf(X+0.1,Y+0.1,Z)
surf(X+0.1,Y+0.65,Z)
surf(X+1.9,Y+0.1,Z)
surf(X+1.9,Y+0.65,Z)

axis equal
```



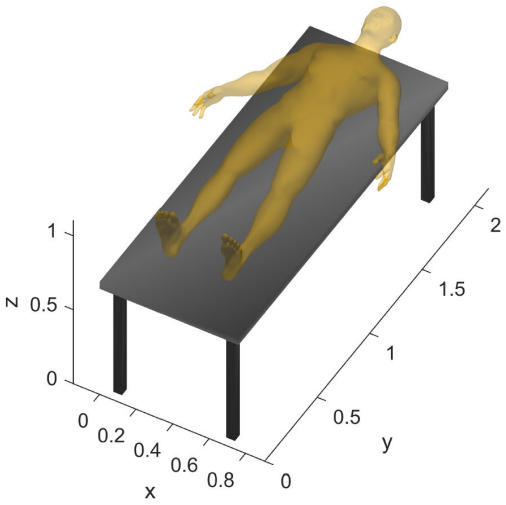

```

```

3D model of a human body

Situate the human model on the operating table.

Expected results

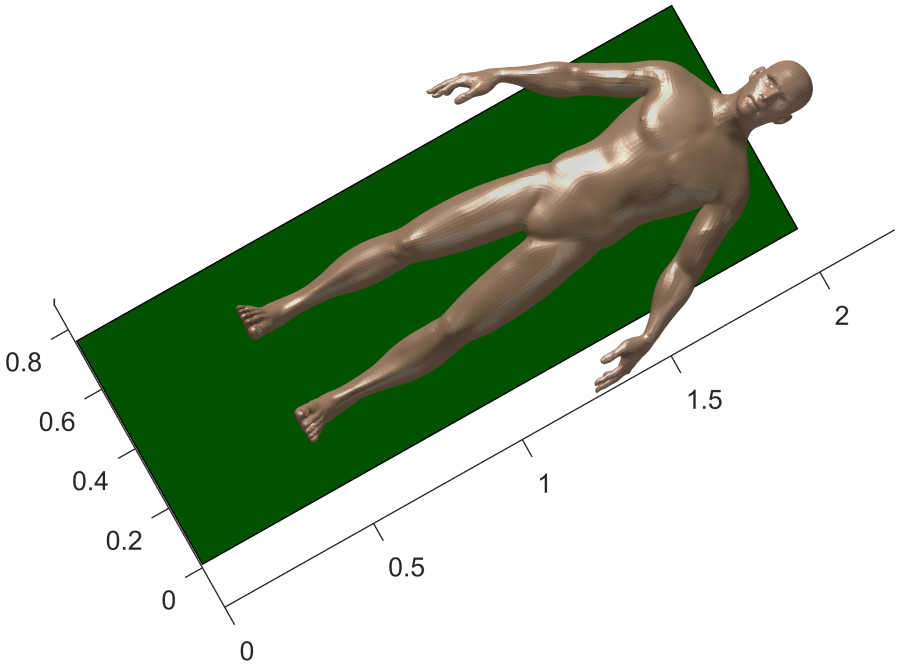


put your code Here

```
load('F_V_HumanBody.mat')
view(3)
Vh(:,4) = 1;
Vh1 = transl(0.45,0.375,0.77)*trotz(-pi/2)*trotx(-pi/2)*Vh'
```

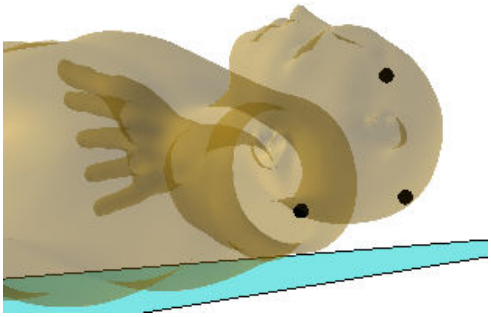
```
Vh1 = 4×24461
    1.7610    1.7649    1.7738    1.7706    1.7789    1.7786    1.7805    1.7844 ···
    0.4893    0.5018    0.4941    0.4857    0.4896    0.4833    0.4987    0.4912
    0.6268    0.6315    0.6250    0.6226    0.6216    0.6196    0.6248    0.6209
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
```

```
patch('Vertices',Vh1(1:3,:)','Faces',Fh,'FaceColor', [0.82 0.64 0.52], 'EdgeColor' , 'none', 'F
axis equal
light('Position',[0.2 0.6 0.4]);
view([-29.413 89.040])
```



Fiducials

The Radiology Department before to take a Computer Tomography (CT) of the brain, fix three fiducials in the head of the patient for registering purpose, visit: https://en.wikipedia.org/wiki/Fiducial



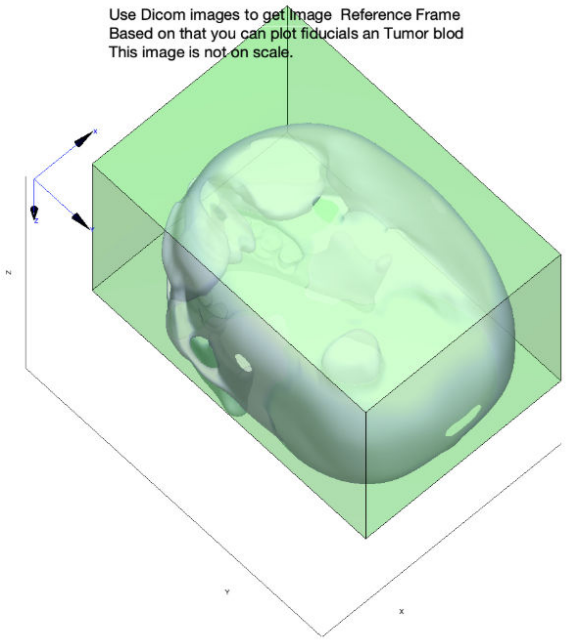
Dicom image vs Image Reference frame {I}

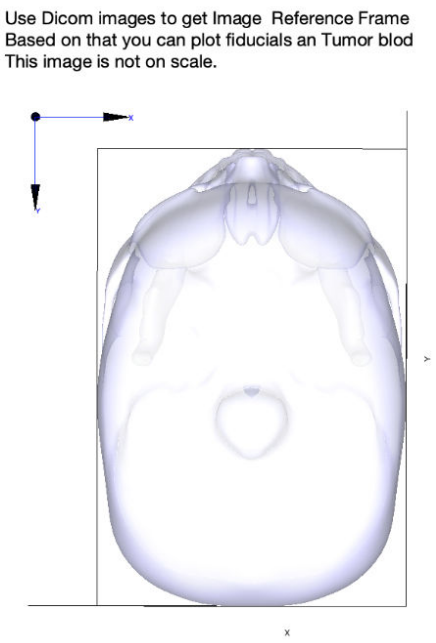
Get familiar with Dicom Images, Visit: https://www.imaios.com/en/Imaios-Dicom-Viewer#!

Use a container Box of the skull to infer the Image Reference Frame {I}

See: 6_Plot_Box_Cone.mlx and 7_Help_Image_RF_Containig_Box.fig to inspire yourselt

Expected results





put your code Here

```
clear
clf
load('F_V_Skull.mat')
view(3)
patch('Vertices',Vs,'Faces',Fs,'FaceColor', [0.5 0.5 0.5], 'EdgeColor' , 'none','FaceAlpha', 0.
%,'FaceColor', [0.5 0.5 0.5], 'EdgeColor' , 'none','FaceAlpha', 0.7
axis equal
max = max(Vs)
```

```
max = 1×3
    0.0765    0.1077    0.2127
```

```
min = min(Vs)
```

```
min = 1×3
   -0.0765   -0.1194    0.0004
```

```
maxX = max(1)
```

```
maxX = 0.0765
```

```
maxY = max(2)
```

```
maxY = 0.1077
```

```
maxZ = max(3)
```

```
maxZ = 0.2127
```

```
minX = min(1)
```

```
minX = -0.0765
```

```
minY = min(2)
```

```
minY = -0.1194
```

```
minZ = 0.072
```

```
minZ = 0.0720
```

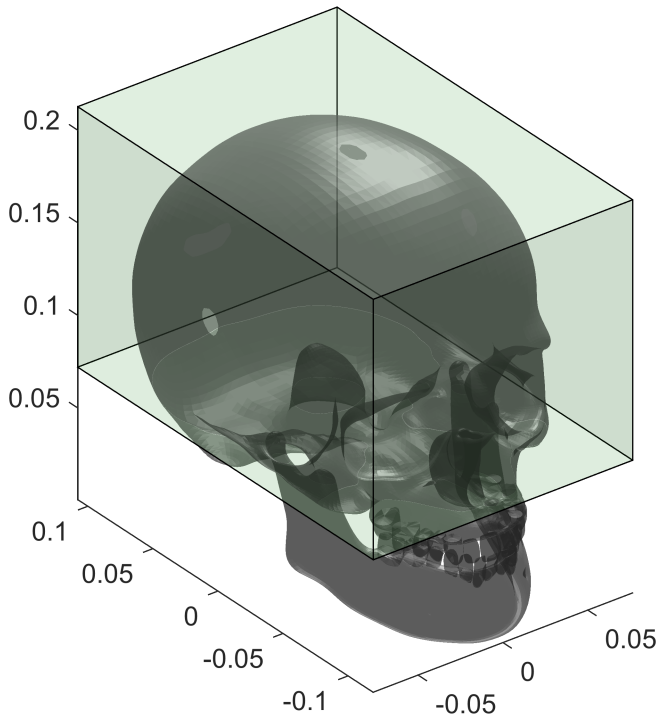
```
v= [minX minY minZ;maxX minY minZ;maxX maxY minZ;minX maxY minZ;minX minY maxZ;maxX minY maxZ;m
```

```
v = 8×3
   -0.0765   -0.1194    0.0720
    0.0765   -0.1194    0.0720
    0.0765    0.1077    0.0720
   -0.0765    0.1077    0.0720
   -0.0765   -0.1194    0.2127
    0.0765   -0.1194    0.2127
    0.0765    0.1077    0.2127
   -0.0765    0.1077    0.2127
```

```
f = [1 2 6 5;2 3 7 6;3 4 8 7;4 1 5 8;1 2 3 4;5 6 7 8]
```

```
f = 6×4
     1     2     6     5
     2     3     7     6
     3     4     8     7
     4     1     5     8
     1     2     3     4
     5     6     7     8
```

```
patch('Vertices',v,'Faces',f,'FaceVertexCData',hsv(6),'FaceColor','g','FaceAlpha',0.1)
light('Position',[1 1 1]);
```



```
altura = maxZ-minZ
```

```
altura = 0.1407
```

```
ancho = maxY-minY
```

```
ancho = 0.2271
```

```
hondo=maxX-minX
```

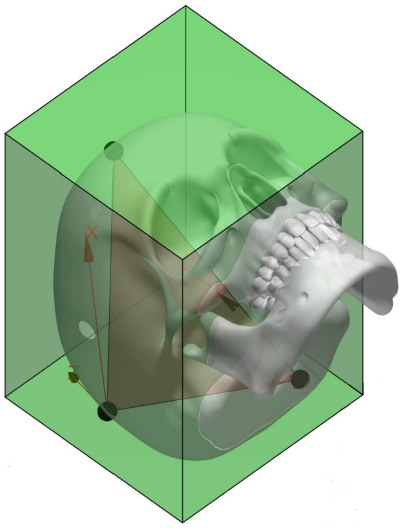
```
hondo = 0.1530
```

Fiducials wrt {I}

Use the Dicom images to place the fiducial relative to Image Reference Frame {I}.

See: 5_Skull_pose_estimation.mlx and use the skull to make the exercise.

Expected results



put your code Here

```

```

FINAL VERSION

```
hold on
%MAXIMOS DICOM [0-MAX]
maxDicomX = 222
```

```
maxDicomX = 222
```

```
maxDicomY = 256
```

```
maxDicomY = 256
```

```
maxDicomZ = 112*1.4
```

```
maxDicomZ = 156.8000
```

```
%CENTRO CALAVERA
centerSkullX = minX + ((maxX-minX)/2)
```

```
centerSkullX = -1.9279e-06
```

```
centerSkullY = minY + ((maxY-minY) / 2)
```

```
centerSkullY = -0.0059
```

```
centerSkullZ = minZ + ((maxZ-minZ) / 2)
```

```
centerSkullZ = 0.1423
```

```
centerSkull = [centerSkullX, centerSkullY, centerSkullZ]
```

```
centerSkull = 1×3
   -0.0000   -0.0059    0.1423
```

```
%CAJA CALAVERA DICOM
minXDic = 41
```

```
minXDic = 41
```

```
maxXDic = 206
```

```
maxXDic = 206
```

```
minYDic = 27
```

```
minYDic = 27
```

```
maxYDic = 234
```

```
maxYDic = 234
```

```
minZDic = 0
```

```
minZDic = 0
```

```
maxZDic = 122 * 1.4
```

```
maxZDic = 170.8000
```

```
%FIDUCIALS
pFid1 = [122, 61 , 131.6]
```

```
pFid1 = 1×3
  122.0000   61.0000  131.6000
```

```
pFid2 = [185, 209 , 91]
```

```
pFid2 = 1×3
   185   209    91
```

```
pFid3 = [71, 212 , 44.8]
```

```
pFid3 = 1×3
   71.0000  212.0000   44.8000
```

```
%DIBUJO CAJA CEREBRO DICOM SPACE
hold on
v2= [minXDic minYDic minZDic;maxXDic minYDic minZDic;maxXDic maxYDic minZDic;minXDic maxYDic mi
```

```
v2 = 8×3
   41.0000   27.0000         0
  206.0000   27.0000         0
  206.0000  234.0000         0
   41.0000  234.0000         0
   41.0000   27.0000  156.8000
  206.0000   27.0000  156.8000
  206.0000  234.0000  156.8000
   41.0000  234.0000  156.8000
```

```
f2 = [1 2 6 5;2 3 7 6;3 4 8 7;4 1 5 8;1 2 3 4;5 6 7 8]
```

```
f2 = 6×4
     1     2     6     5
     2     3     7     6
     3     4     8     7
     4     1     5     8
     1     2     3     4
     5     6     7     8
```

```
%patch('Vertices',v2,'Faces',f2,'FaceVertexCData',hsv(6),'FaceColor','g','FaceAlpha',0.1)
%scatter3(pFid1(1), pFid1(2) ,pFid1(3))
%scatter3(pFid2(1), pFid2(2) ,pFid2(3))
%scatter3(pFid3(1), pFid3(2) ,pFid3(3))
%axis equal

%CENTRO DICOM
centerDicomX = (maxXDic - minXDic) / 2
```

```
centerDicomX = 82.5000
```

```
centerDicomY = (maxYDic - minYDic) / 2
```

```
centerDicomY = 103.5000
```

```
centerDicomZ = (maxZDic - minZDic) / 2
```

```
centerDicomZ = 85.4000
```

```
centerDicom = [centerDicomX, centerDicomY, centerDicomZ]
```

```
centerDicom = 1×3
   82.5000  103.5000   85.4000
```

```
%DIFERENCIA ENTRE CENTROS
difCenterDicSkl = centerDicom - centerSkull
```

```
difCenterDicSkl = 1×3
```

```
   82.5000  103.5059   85.2577
```

```
pFid1Dif = pFid1 - difCenterDicSkl
```

```
pFid1Dif = 1×3
   39.5000  -42.5059   46.3423
```

```
pFid2Dif = pFid2 - difCenterDicSkl
```

```
pFid2Dif = 1×3
  102.5000  105.4941    5.7423
```

```
pFid3Dif = pFid3 - difCenterDicSkl
```

```
pFid3Dif = 1×3
  -11.5000  108.4941  -40.4577
```

```
hold on
%scatter3(pFid1(1),pFid1(2),pFid1(3))
%scatter3(pFid2(1),pFid2(2),pFid2(3))
%scatter3(pFid3(1),pFid3(2),pFid3(3))

%scatter3(pFid1Dif(1),pFid1Dif(2),pFid1Dif(3))
%scatter3(pFid2Dif(1),pFid2Dif(2),pFid2Dif(3))
%scatter3(pFid3Dif(1),pFid3Dif(2),pFid3Dif(3))
v= [minX minY minZ;maxX minY minZ;maxX maxY minZ;minX maxY minZ;minX minY maxZ;maxX minY maxZ;m
```

```
v = 8×3
   -0.0765   -0.1194    0.0720
    0.0765   -0.1194    0.0720
    0.0765    0.1077    0.0720
   -0.0765    0.1077    0.0720
   -0.0765   -0.1194    0.2127
    0.0765   -0.1194    0.2127
    0.0765    0.1077    0.2127
   -0.0765    0.1077    0.2127
```

```
f = [1 2 6 5;2 3 7 6;3 4 8 7;4 1 5 8;1 2 3 4;5 6 7 8]
```

```
f = 6×4
     1     2     6     5
     2     3     7     6
     3     4     8     7
     4     1     5     8
     1     2     3     4
     5     6     7     8
```

```
patch('Vertices',v,'Faces',f,'FaceVertexCData',hsv(6),'FaceColor','g','FaceAlpha',0.1)
light('Position',[1 1 1]);
dicomToSkullX = (maxX - minX) / (maxXDic - minXDic)
```

```
dicomToSkullX = 9.2727e-04
```

```
dicomToSkullY = (maxY - minY) / (maxYDic - minYDic)
```

```
dicomToSkullY = 0.0011
```

```
dicomToSkullZ = (maxZ - minZ) / (maxZDic - minZDic)
```

```
dicomToSkullZ = 8.2350e-04
```

```
dicomToSkull = [dicomToSkullX ,dicomToSkullY, dicomToSkullZ]
```

```
dicomToSkull = 1×3
    0.0009    0.0011    0.0008
```

```
%pFid1Aux = (pFid1 .* dicomToSkull)
%pFid2Aux = (pFid2 .* dicomToSkull)
%pFid3Aux = (centerSkull + (pFid3 .* dicomToSkull) / 2)

%scatter3(pFid1Aux(1),pFid1Aux(2),pFid1Aux(3))
%scatter3(pFid2Aux(1),pFid2Aux(2),pFid2Aux(3))
%scatter3(pFid3Aux(1),pFid3Aux(2),pFid3Aux(3))

%DICOM * PROP DICOM->SKULL
centerDicSk = centerDicom .* dicomToSkull
```

```
centerDicSk = 1×3
    0.0765    0.1136    0.0703
```

```
pFid1S = pFid1 .* dicomToSkull
```

```
pFid1S = 1×3
    0.1131    0.0669    0.1084
```

```
pFid2S = pFid2 .* dicomToSkull
```

```
pFid2S = 1×3
    0.1715    0.2293    0.0749
```

```
pFid3S = pFid3 .* dicomToSkull
```

```
pFid3S = 1×3
    0.0658    0.2326    0.0369
```

```
difCentDicSk2 = centerDicSk - centerSkull
```

```
difCentDicSk2 = 1×3
    0.0765    0.1194   -0.0720
```

```
pFid1S = pFid1S - difCentDicSk2
```

```
pFid1S = 1×3
    0.0366   -0.0525    0.1804
```

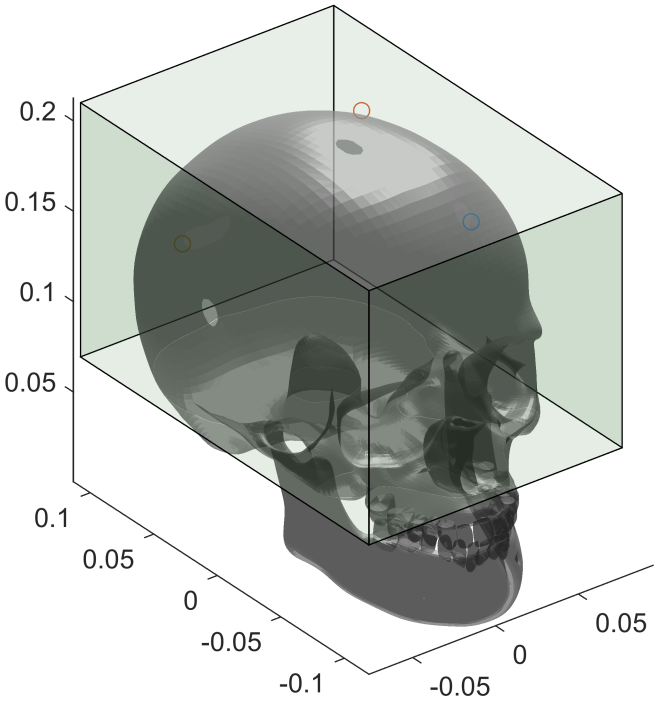
```
pFid2S = pFid2S - difCentDicSk2
```

```
pFid2S = 1×3
    0.0950    0.1099    0.1469
```

```
pFid3S = pFid3S - difCentDicSk2
```

```
pFid3S = 1×3
   -0.0107    0.1132    0.1089
```

```
scatter3(pFid1S(1),pFid1S(2),pFid1S(3))
scatter3(pFid2S(1),pFid2S(2),pFid2S(3))
scatter3(pFid3S(1),pFid3S(2),pFid3S(3))
```

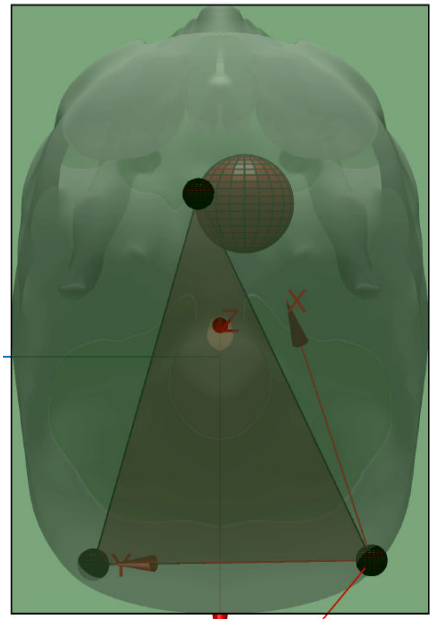


Tumor points wrt {I}

Use the Dicom images to get the points of the outer perimeter of the tumor relative to Image Reference Frame {I}.

You can simplify the tumor information by defining the center of mass and estimate an equivalent diameter.

Expected results



put your code Here

```
hold on
v2= [minXDic minYDic minZDic;maxXDic minYDic minZDic;maxXDic maxYDic minZDic;minXDic maxYDic mi
```

```
v2 = 8×3
   41.0000   27.0000         0
  206.0000   27.0000         0
  206.0000  234.0000         0
   41.0000  234.0000         0
   41.0000   27.0000  156.8000
  206.0000   27.0000  156.8000
  206.0000  234.0000  156.8000
   41.0000  234.0000  156.8000
```

```
f2 = [1 2 6 5;2 3 7 6;3 4 8 7;4 1 5 8;1 2 3 4;5 6 7 8]
```

```
f2 = 6×4
     1     2     6     5
     2     3     7     6
     3     4     8     7
     4     1     5     8
     1     2     3     4
     5     6     7     8
```

```
%hold on
%patch('Vertices',v2,'Faces',f2,'FaceVertexCData',hsv(6),'FaceColor','g','FaceAlpha',0.1)
%scatter3(pFid1(1), pFid1(2) ,pFid1(3))
%scatter3(pFid2(1), pFid2(2) ,pFid2(3))
%scatter3(pFid3(1), pFid3(2) ,pFid3(3))
axis equal

tumorMaxZ = 90*1.4
```

```
tumorMaxZ = 126.0000
```

```
tumorMinZ = 64*1.4
```

```
tumorMinZ = 89.6000
```

```
tumorMaxX = 148
```

```
tumorMaxX = 148
```

```
tumorMinX = 113
```

```
tumorMinX = 113
```

```
tumorMaxY = 115
```

```
tumorMaxY = 115
```

```
tumorMinY = 80
```

```
tumorMinY = 80
```

```
difTZ = tumorMaxZ - tumorMinZ
```

```
difTZ = 36.4000
```

```
difTX = tumorMaxX - tumorMinX
```

```
difTX = 35
```

```
difTY = tumorMaxY - tumorMinY
```

```
difTY = 35
```

```
radioTumor = ((difTZ + difTY + difTX) / 3) / 2
```

```
radioTumor = 17.7333
```

```
tumorCenterDicomX = tumorMinX + difTX/2
```

```
tumorCenterDicomX = 130.5000
```

```
tumorCenterDicomY = tumorMinY + difTY/2
```

```
tumorCenterDicomY = 97.5000
```

```
tumorCenterDicomZ = tumorMinZ + difTZ/2
```

```
tumorCenterDicomZ = 107.8000
```

```
tumorCenterDicom = [tumorCenterDicomX,tumorCenterDicomY,tumorCenterDicomZ]
```

```
tumorCenterDicom = 1×3
  130.5000   97.5000  107.8000
```

```
tumorCenterDicomSk = tumorCenterDicom .* dicomToSkull
```

```
tumorCenterDicomSk = 1×3
    0.1210    0.1070    0.0888
```

```
tumorCenterDicomSk = tumorCenterDicomSk - difCentDicSk2
```

```
tumorCenterDicomSk = 1×3
    0.0445   -0.0125    0.1608
```

```
hold on
% Make unit sphere
[x,y,z] = sphere;
% Scale to desire radius.
radius = [radioTumor * dicomToSkullX,radioTumor * dicomToSkullY, radioTumor * dicomToSkullZ];
x = x * radius(1);
y = y * radius(2);
z = z * radius(3);
% Translate sphere to new location.
offsetx = tumorCenterDicomX * dicomToSkullX - difCentDicSk2(1)
```

```
offsetx = 0.0445
```

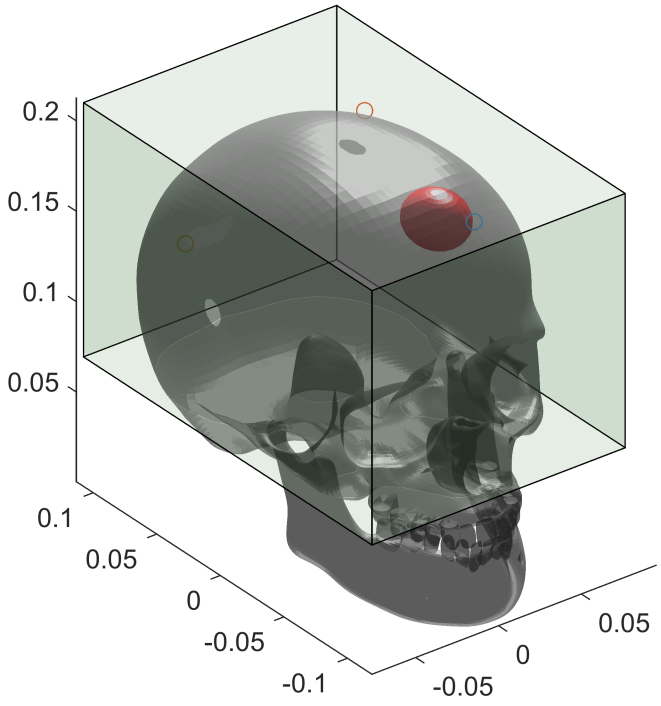
```
offsety = tumorCenterDicomY * dicomToSkullY - difCentDicSk2(2)
```

```
offsety = -0.0125
```

```
offsetz = tumorCenterDicomZ * dicomToSkullZ - difCentDicSk2(3)
```

```
offsetz = 0.1608
```

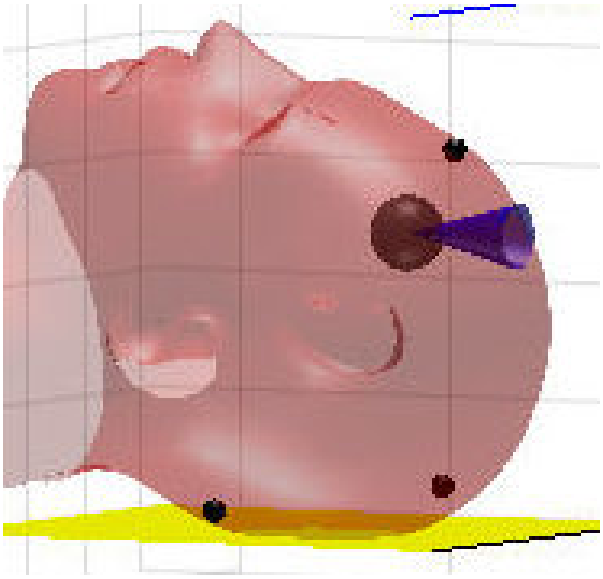
```
% Plot as surface.
surf(x+offsetx,y+offsety,z+offsetz,'FaceColor','r','EdgeColor','none')
axis equal;
```



Fiducials and Tumor wrt Human Reference Frame

Place fiducial and tumor in the head of the human. You will have to re-do the containing box secction.

Expected results



First approach (10%)

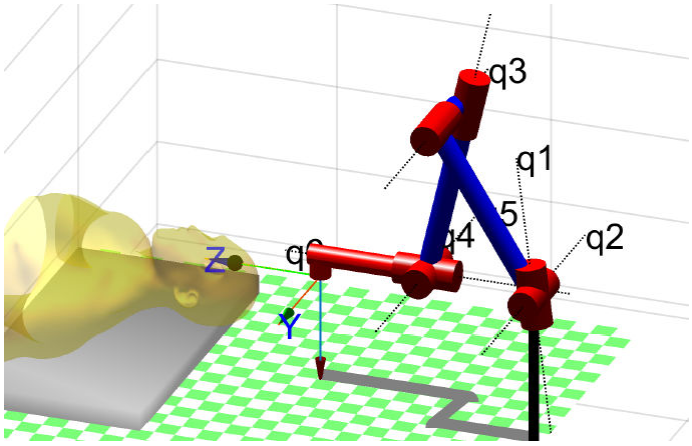
Asume that the ZX plane of the Robot is aligned with the plane of symmetry of the human body.

Robot manipulator

Consider the best position of manipulator to be nearby the operating table to warranty that the head is in the reachable work space. Use a Puma 560. Use p560.teach to play.

Use: p560.base & p560.tool to locate the Puma and add the tools.

Expected results

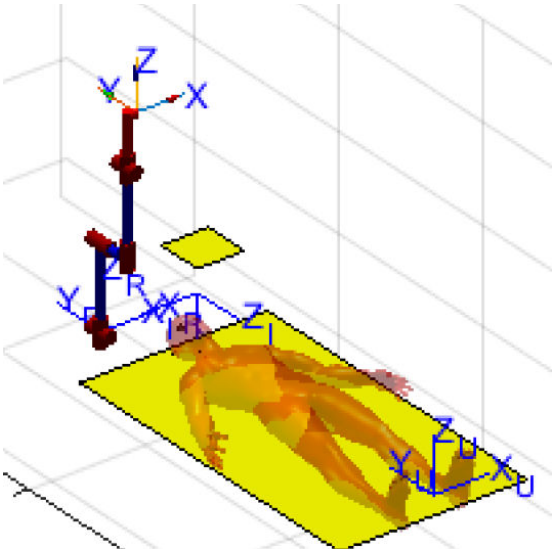



```
%% put your code Here
```

Reference Frames

Display all necessary reference frame. Use best scale to see it.

- {U} Univers [0 0 0]
- {R} Robot
- {I} Image
- {Tb} Table_body
- {Tt} Table tool
- {EE} End Efector
- others
- ...



```
%% put your code Here
```

Transformations

Enumerate the transformation you will need.

```
%% put your code Here
```

Tumor points in Robot Frame.

Remember the Transform compound exercise

```
%% put your code Here
```

Second approach: (25%)

Modify your code to repeat the exercise if the table with the patient is given as happend in the Rosa video.

To know the head relative pose with respect to the Puma Robot ...

See: '4_Sckeching_Key_ideas_students.mlx' , '5_Skull_pose estimation.mlx' for inspiration and Second_approach_SPM.pdf

```
open('3_Second_approach_Patient_pose.fig')
```

```
Error using open (line 87)
File '3_Second_approach_Patient_pose.fig' not found.
```

Surgery (55%)

Biospy

Prepare a script that perform a biopsy. Zoom in the scene and record a video with the best view.

Use a tool that has the following Transformation: transl(0.05 0 0.25)

Let us see the 'trail' option of plot to visualize the trajectory.

```
%% put your code Here
```

Trepanation

Prepare a script that perform trepanation. Zoom in the scene and record a video with the best view.

Use a tool that has the following Transformation: transl(0 0 0.2)

Let us see the 'trail' option of plot to visualize the trajectory. Place a 45º cone on top of the trepanation to better understand. See: 6_Plot_Box_Cone.mlx. You will have to scale it. Play with transparency.

```
%% put your code Here
```

Tumor burning

Prepare a script that perform tumor burning with the laser. Zoom in the scene and record a video with the best view.

You ought to think in an algorithm, that in order, fill up the tumor's equivalent sphere with small burning spheres of 4m diameter.

Use a tool that has the following Transformation: transl(0 0 0.2)

```
%% put your code Here
```