Project1 Reconstruction

Assignment

1. Reconstruct a 3D volume by putting the 2D *FrameData* stack by stack and display it.
2. Reconstruct a 3D volume using forward mapping method and display it.
3. (Optional) Reconstruct a 3D volume using inverse mapping method and display it.
4. Discuss the difference of the reconstruction results using different methods.

- **Data description:**

1. *FrameData.mat*

   Each frame consisted of 320*240 pixels. You can use *imagesc(frameData{1…n})* in Matlab to display the 2D frame. (The resolution is 0.296*mm* in both horizontal and vertical directions. This information is for reference only. In fact, the rectangle coordinates given by *GpsData* already contain this resolution information).

2. *GpsData.mat*

   Think that each frame is a rectangle, the location information provided is the coordinates of the four vertices of the rectangle. The unit is millimeter.

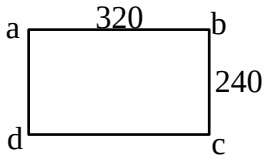
```
val(:,:,1) =
        a         b         c         d
x   215.8689  268.1450  314.0332  261.7571
y    14.0057   77.2119   76.1429   12.9367
z  -137.4250  -90.0855 -139.3471 -186.6866

val(:,:,2) =

   215.8688  268.1450  314.0334  261.7572
    14.0035   77.2213   76.1605   12.9428
  -137.4250  -90.0855 -139.3471 -186.6866
```

frame 1

frame 2



3. *CalibrationPoints.mat*

   The coordinates of four calibration points in the GPS transmitter space and screen space have been provided. You can use these four points to get transformation matrix. The unit is millimeter.

```
val(:,:,1) =
        a         b         c         d
x   215.8689  268.1450  314.0332  267.3782
y    14.0057   77.2119   76.1429   77.1386
z  -137.4250  -90.0855 -139.3471  -89.7278

val(:,:,2) =

  -105.3426  -10.6910  -10.9811  -10.9858
  -577.7535 -577.9899 -645.2327 -577.1995
   246.8376  243.6969  240.2480  243.5991
```

Coordinate in GPS transmitter space

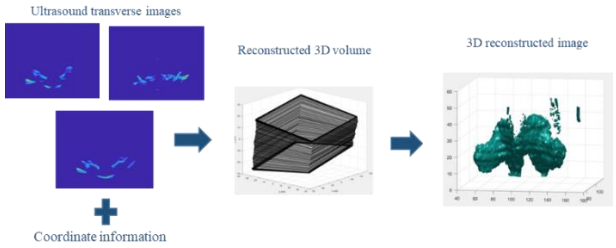
Coordinate in world space

4. Voxel size

   The voxel size in the volume is recommended to set to 0.5*mm* in width (*x* direction) and 0.5*mm* height (*y* direction), and 0.5*mm* in depth (*z* direction).

   In order to accelerate reconstruction, the voxel size in the volume can be to set to 1*mm* in width (*x* direction) and 1*mm* height (*y* direction), and 1*mm* in depth (*z* direction) firstly.

The detail of step 2/3:



The procedure of reconstruction.

(1) Load data.

(2) Coordinate Transformation to seek the relationship between pixel and voxel.

(3) Reconstruction to find the intensity value for each voxel

(4) Display the 3D image.

- **The process of transformation:**

In this project, the coordinates of four calibration points in the GPS transmitter space and screen space have been provided. You can use these four points to get transformation matrix. For forward mapping method, you need to transform all the points in GPS transmitter space to screen space before reconstruction. And for inverse mapping method, you need to find the corresponding pixels in the GPS transmitter space for each voxel in the screen space.

Tips: You can use *plot3()* function in Matlab to display the volume according to *GpsData*. Since the coordinate of four vertices in each frame have been given, you can draw all rectangles in one space.

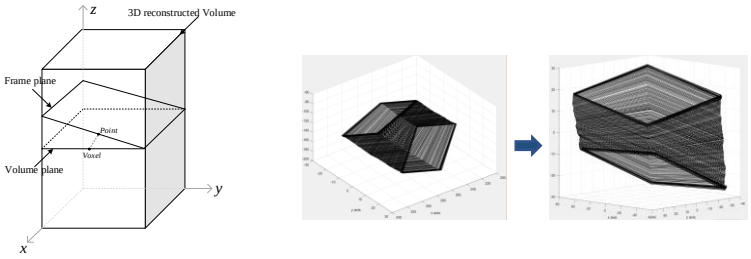
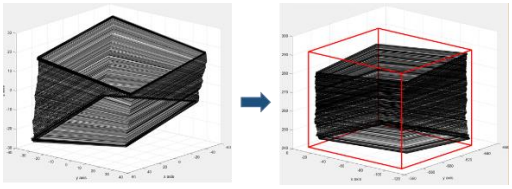


Fig.2. Coordinate transformation. Volume plane is a plane within the reconstructed volume and the frame plane is the original planes to be reconstructed in the data file of "*FrameData*".

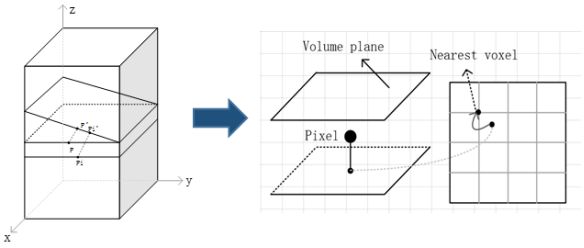
- **The process of reconstruction:**
  - **Forward mapping method**
    - Rebuild a regular volume according to the *GpsData*. (see Fig. 3. (a))
    - Find the nearest voxel for each pixel. (see Fig. 3. (b))
    - Assign the value of pixel to the voxel.
    - Display the 3D matrix by using Matlab function *isosurface()*.

Tips:

1. How to determine the reconstruction volume size?
2. When the regular volume is rebuilt, in order to facilitate reconstruction, you can recast the origin according to the volume. (Please note that pixels and voxels should have the same origin)
3. How to find the nearest voxel for each pixel?
4. How to determine the value of the voxel if multiple pixels correspond to one voxel?



(a)



(b)

Fig.3. Forward mapping method. (a) Rebuild a regular volume. The red lines show the volume. (b) Find the nearest voxel and assign the value of the pixel to the voxel.

- **Inverse mapping method**
  - Rebuild a regular volume according to the *GpsData*. (see Fig. 4. (a))
  - Project the voxel onto the nearest frame plane and get the projection points on the plane. (See Fig. 4(b))
  - Assign the value of pixel to the voxel by using the nearest neighbor interpolation method in frame plane.
  - Display the 3D matrix by using Matlab function *isosurface()*.

Tips:

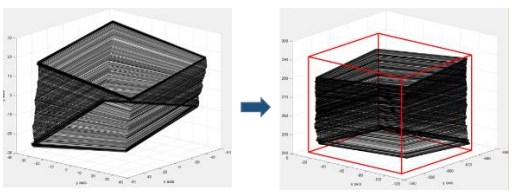
(1) Project the voxel onto the surrounding frame planes. The shortest distance gives the nearest frame plane. So, how to determine the surrounding frame planes?

(2) The 3D point-to-plane distance can be calculated as follows:

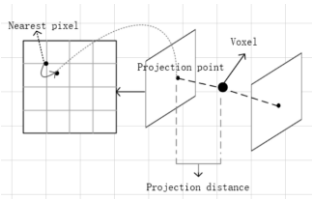
$$d = |\overrightarrow{AB} \cdot \frac{\vec{n}}{|\vec{n}|}|$$

Where, $\overrightarrow{AB}$ denotes a vector from point *A* to *B*, *A* is a point outside the plane, *B* is any point in the plane. $\vec{n}$ is the normal vector of the plane.

(3) Here, frame plane is a finite plane in the volume. So, not all voxels need to project onto the frame plane.



(a)



(b)

Fig.4. Inverse mapping method. (a) Rebuild a regular volume. The red lines show the volume. (b) Implement the nearest neighbor interpolation in volume plane.