RGB+D Cameras: Image, 3D point clouds and camera model

Suggested readings
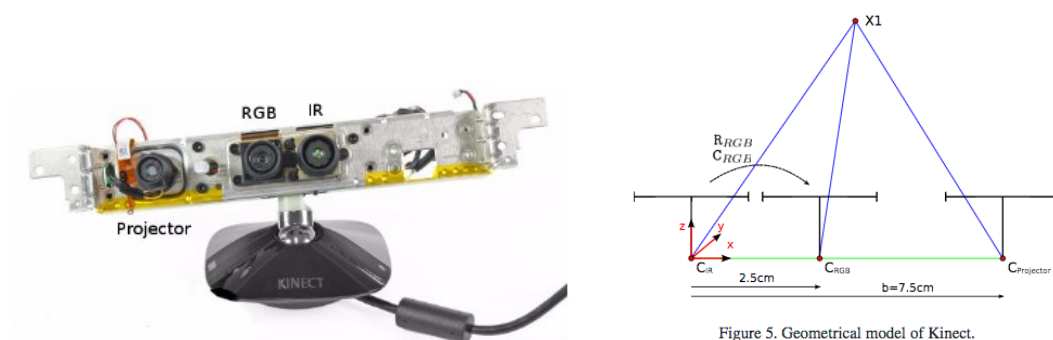
"Kinect" (depth) cameras provide 2 type of images: RGB and DEPTH

(a) RGB image   (b) Depth image
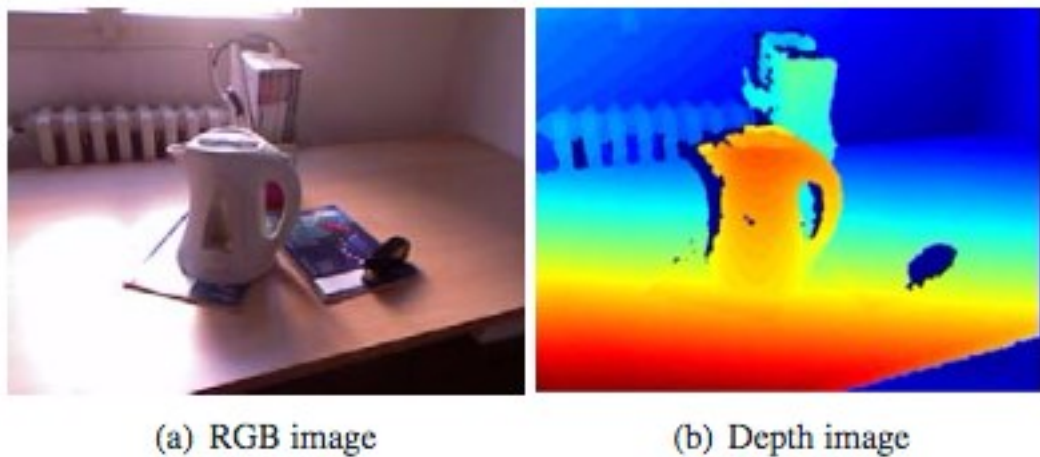
Q: What coordinate is the "depth" ? How is it obtained ?
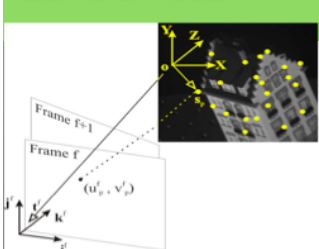

Figure 5. Geometrical model of Kinect.

Tasks:
1. See RGB and Depth images
   a. display and interpret images
   b. display R,G,B in separate figures
2. Display depth
   a. depth as image

b. depth as a surface
c. point cloud, mesh
d. depth image vs. point cloud

Camera model -
- How to generate an image from 3D points?
- How to compute 3D positions from images/depth


Camera model

$$M = K [R \mid t]$$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Tasks:
1. Given u,v and depth compute XYZ
2. Generate image projection from point cloud
3. Rotate and Translate point cloud
4. Project Point cloud in the RGB image. What about the reverse: click rgb and get the 3D!
5. Compute the camera for image 14
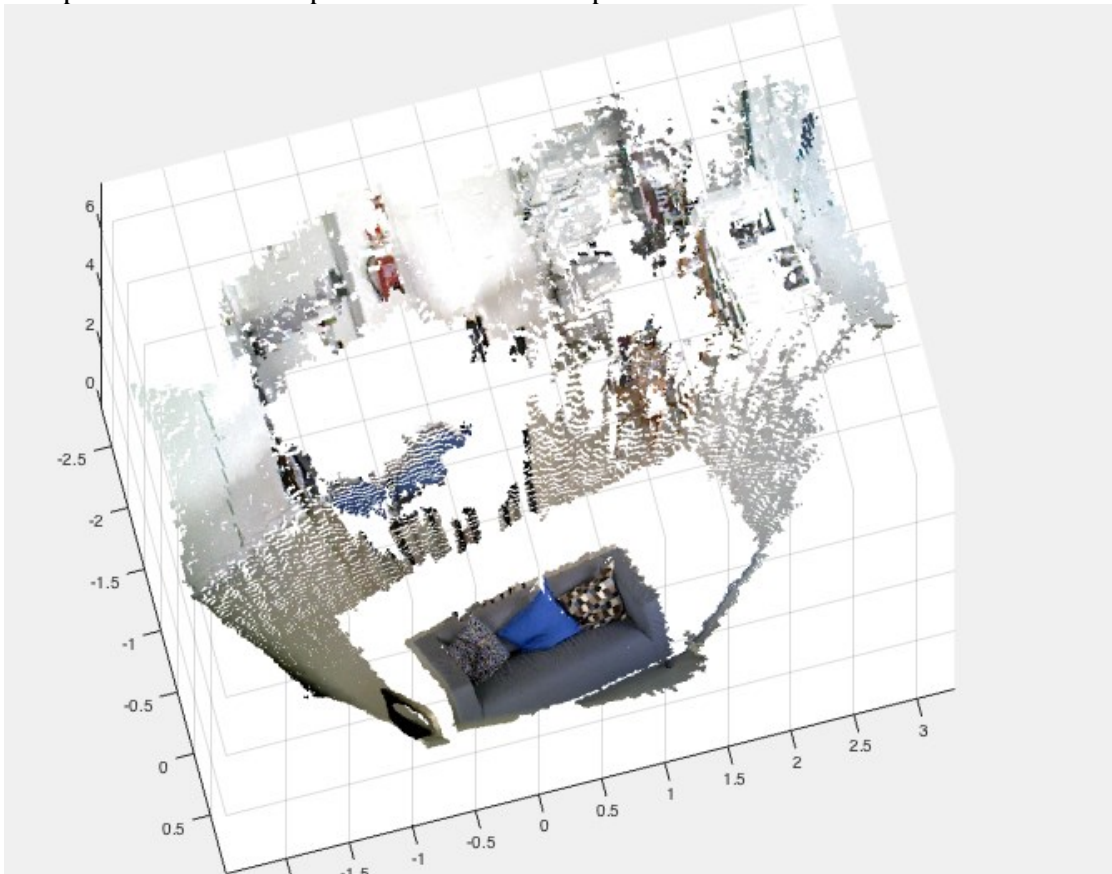6. Next class ...compute the pose !(you are going to need this)

In short

From these two images
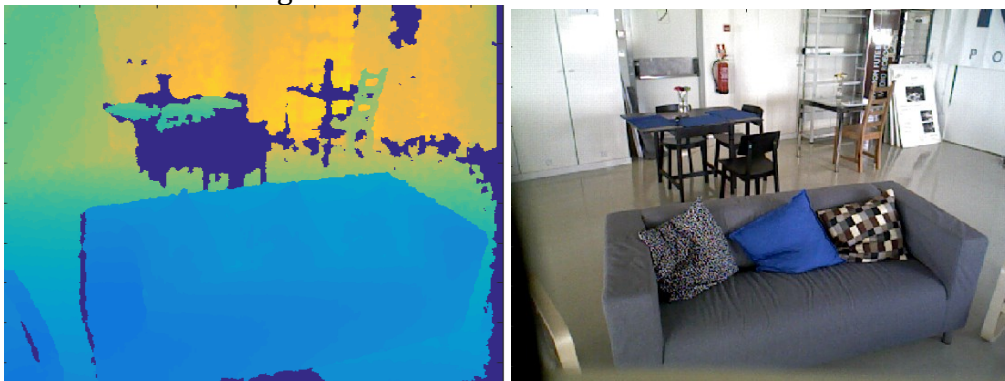
and the intrinsic and extrinsic camera parameters you should be able to "superimpose" the rgb over the depth
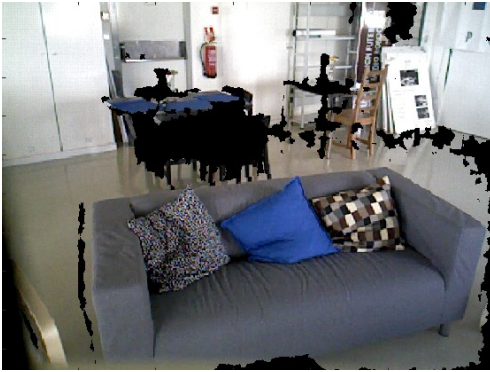
Note that now each pixel of this image is in line with the depth! So, you can compute XYZ for each pixel and create this point cloud

Auxiliary code to implement these steps

```
im1=imread('rgb_image_10.png');
im2=imread('rgb_image_17.png');
load depth_10.mat;
dep1=depth_array;
load depth_17.mat;
dep2=depth_array;
load calib_asus.mat;
%load pts_10_14.mat
```

```matlab
K=Depth_cam.K;
Krgb=RGB_cam.K;


%image 10
xyz=get_xyzasus(dep1(:),[480 640],1:480*640,K,1,0);
imd=get_rgbd(xyz,im1,R_d_to_rgb,T_d_to_rgb,RGB_cam.K);
pc=pointCloud(xyz,"Color",reshape(imd,480*640,3));
pc=pcdownsample(pc,"random",.4);
showPointCloud(pc);
```