Nikhil Chandra (22405492)
Muroor Vikas Rao (62805822)
CS 211A: Fall 2015

HW 3

Part 1: Configuring the Image Sets

We divide the images into two sets each consisting of 5 images. Each set is created in such a way such that every part of the horse is viewed. Following are the images from the two sets.
The first set of images.

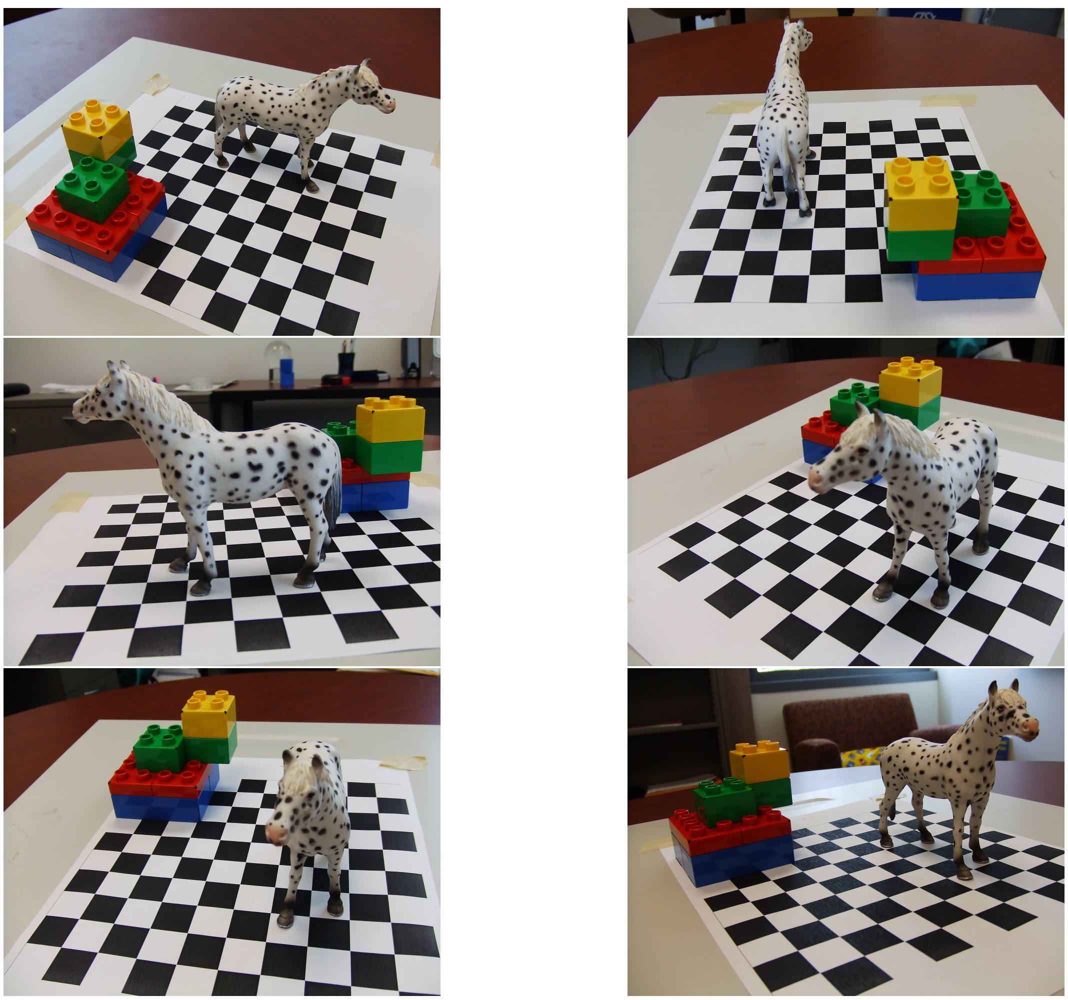


Figure 1: Image Set A

The second set of images.

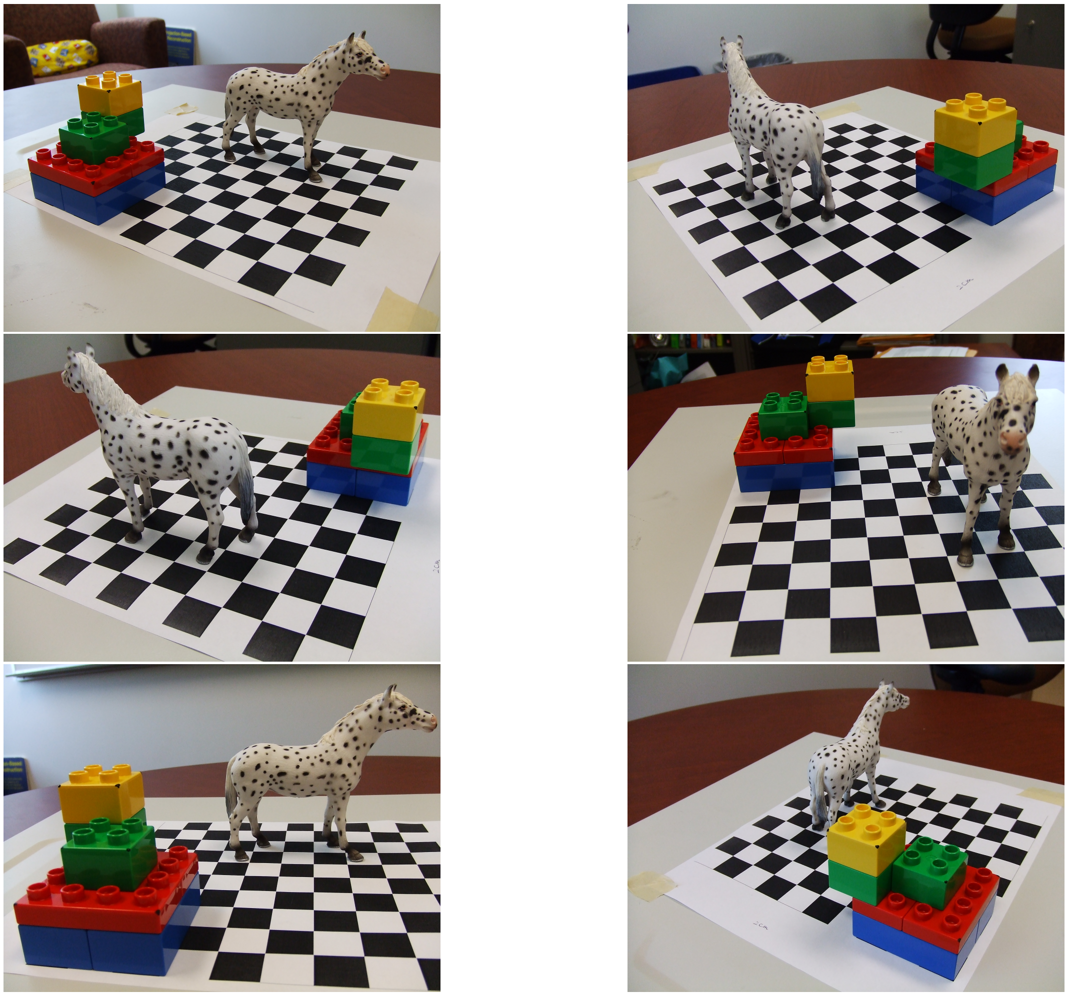


Figure 2: Image Set B

The two sets can be found in the folder. **HW3/data/1 and HW3/data/2**
Following are the image numbers in the both the sets

Set A	Set B
DSCF4177	DSCF4179
DSCF4182	DSCF4183
DSCF4186	DSCF4184
DSCF4188	DSCF4193
DSCF4190	DSCF4199
DSCF4195	DSCF4201

Part 2: Camera Calibration

We then calibrate the camera by mapping 3-D World Coordinates to the 2-D Images for all the images in each set. Below we show some of the images we took for Camera Calibration.

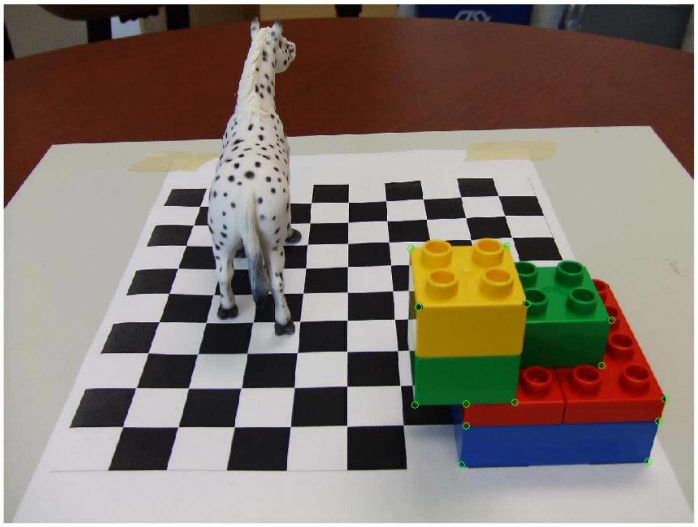


Figure 3: Calibration points for Image in Set A

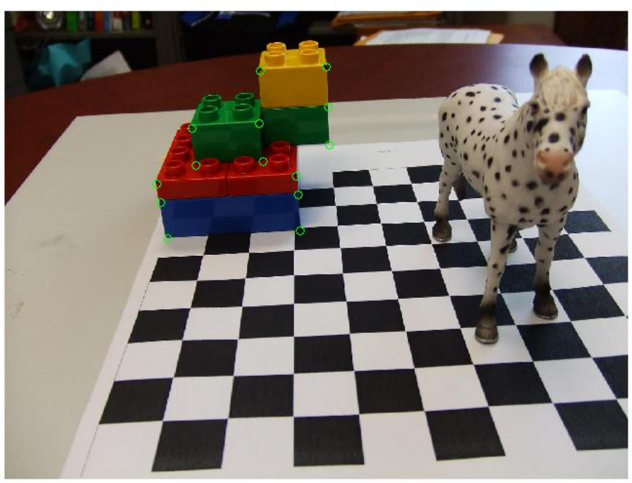


Figure 4: Calibration points for Image in Set B

We then average over all the 6 intrinsic matrix obtained for each set to get the average intrinsic matrix.

To keep everything in homogeneous coordinates the matrix displayed here is 4x4 whereas the intrinsic matrix is only 3x3.

Although both the sets have images the reason why the intrinsic parameters may be different is because the focal length must have changed thus leading to a different intrinsic paramters.

On looking at the matrix we observe that it has a skew and also the image plane is at a certain offset from origin. Also the scale parameters across both the axis are approximately the same and hence the pixel can be considered as a square.

	1	2	3	4
1	2.4098e+03	62.3069	1.5897e+03	0
2	0	2.5043e+03	874.8112	0
3	0	0	1	0
4	0	0	0	1

Figure 5: Intrinsic Matrix Set A

	1	2	3	4
1	1.6166e+03	77.6767	1.6320e+03	0
2	0	1.6849e+03	1.2021e+03	0
3	0	0	1	0
4	0	0	0	1
5				

Figure 6: Intrinsic Matrix Set B

We next plot the extrinsic parameters for each set A and B. We make use of an existing function **plotcube** that lets us plot cubes of different sizes and color. This helps us visualize and interpret if our camera positions are correct or not.

Looking at the visualizations of Set A where we look at 5 different cameras we observe that all the extrinsic parameters indeed match the images in the set. Thus the extrinsic parameters have been plotted accurately. We further include the image of the horse computed using **Depth Estimation** in the same view for better understanding of the scene.

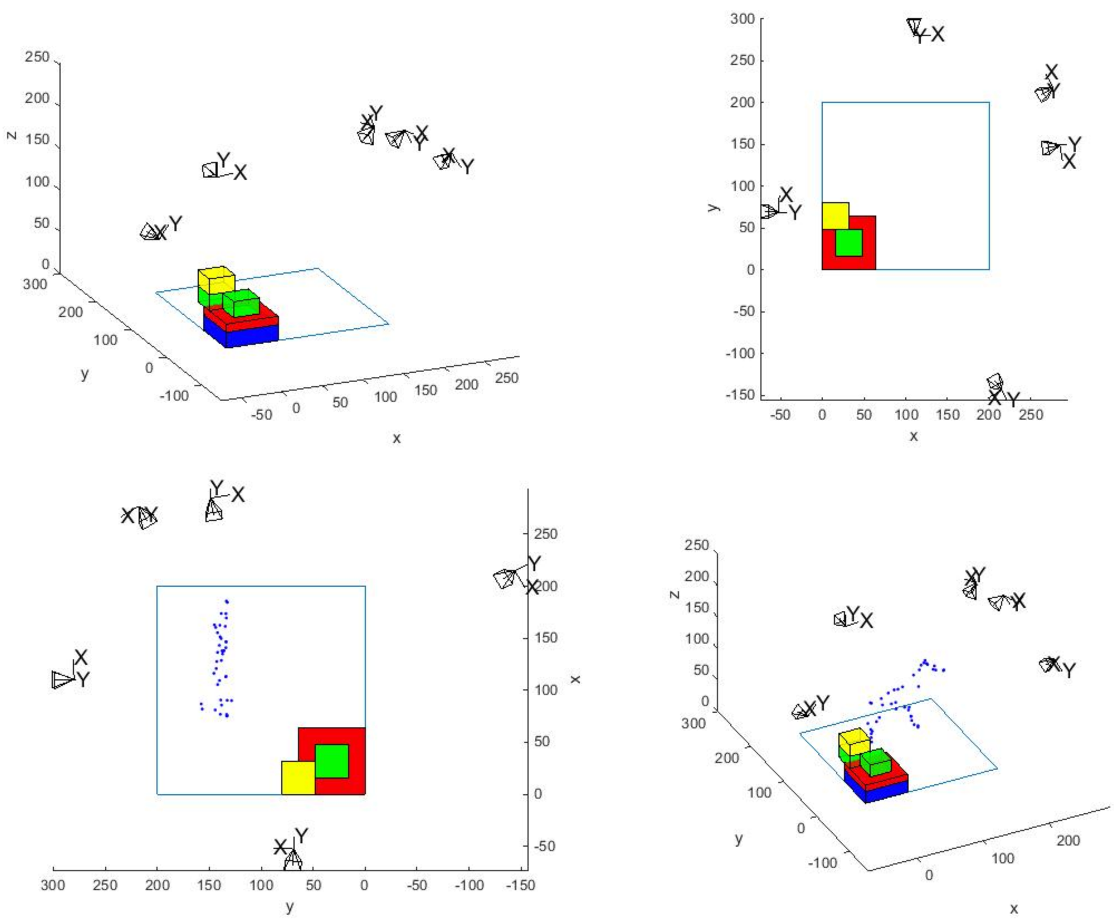


Figure 7: Depth Estimation (Camera Position in Set A)

Looking at Set B consisting of 6 images we plot the 6 different camera views.

In this case we observe that 4 of the cameras are accurate in representing the correct scene while 2 cameras are somehow being included inside the checkerboard. This may be because of the noise/inaccurate extraction of the calibration matrix. We also show along the horse position obtained as done in the previous set.

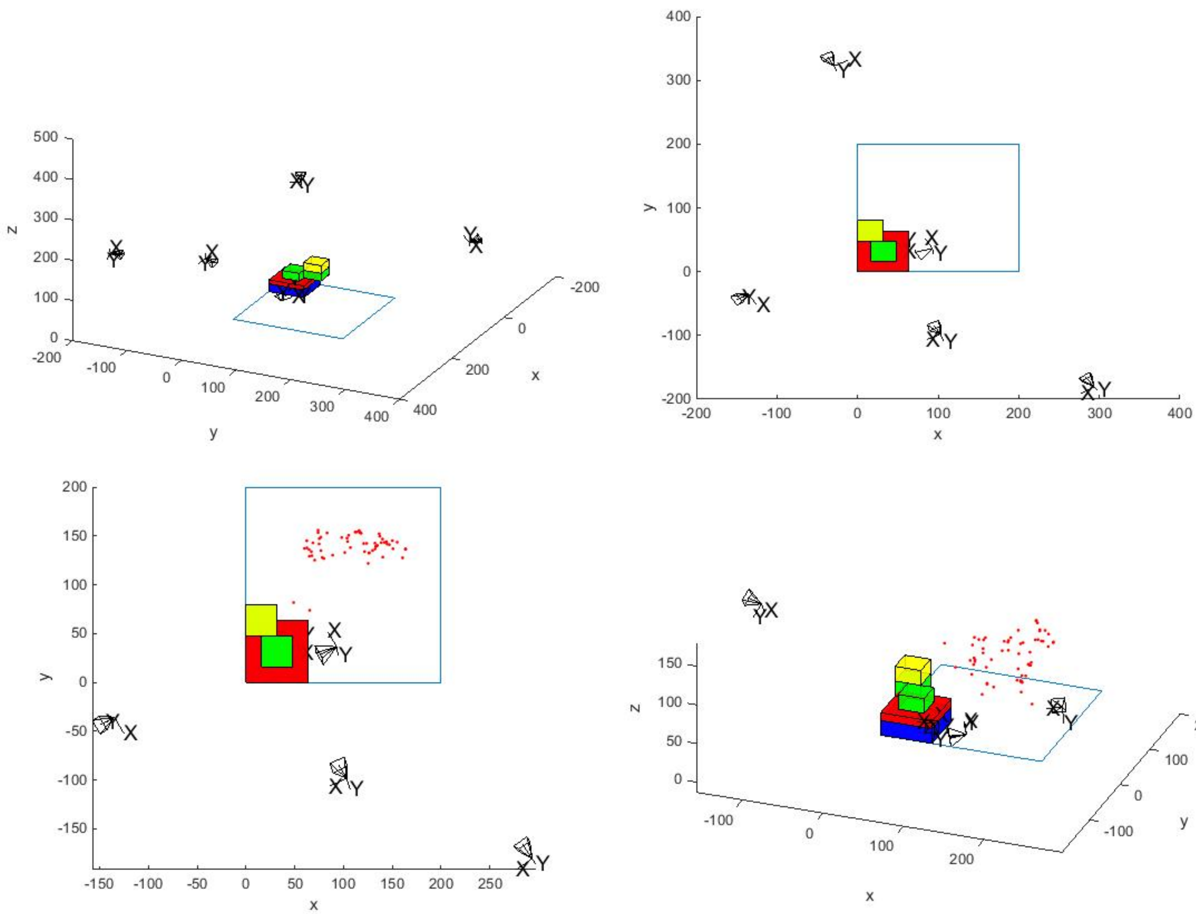


Figure 8: Depth Estimation (Camera Position in Set B)

NOTE : The 3-D to 2D correspondence points for the images can be found in the following folder.

Set A. HW3/data/1/CDSCFXXXX.txt

Set B. HW3/data/2/CDSCFXXXX.txt

Here XXXX refers to the names of the other images in a particular set.

Part 3: Epipolar Lines

For computing epipolar lines we fix on the images (DSCF477 in case of Set A and DSCF499 in case of Set B) and compute pairwise matrix with the rest of the images in the set.

For this correspondence are computed across the images in the set. Thus we have 5 Fundamental Matrix for each set of 6 images.

Below is the Matlab program to compute the Fundamental Matrix. After the computation we plot epipolar lines for the images in a particular set.

```
function [ fund_matrix ] = computeFundamentalMatrix(mPointsA, mPointsB )

% Extracting the correspondences U an V for image A and image B
aU = mPointsA(:,1);
aV = mPointsA(:,2);
bU = mPointsB(:,1);
bV = mPointsB(:,2);

% Write the equation in the form of A.x=0
A = zeros(length(aU),9);
for x = 1:length(aU)
    [u,v] = deal(aU(x),aV(x));
    [up,vp] = deal(bU(x),bV(x));
    A(x,:) = [u*up, v*up, up, u*vp, v*vp, vp, u, v, 1];
end

% solve Ax=0 using SVD
[~,~,V] = svd(A);
f = V(:,end);
F = reshape(f,[3,3])';

% Enforce det(F)=0 condition using SVD
[U,S,V] = svd(F);
S(3,3) = 0;
fund_matrix = U*S*V';

end
```

NOTE : The 2D correspondence points for the images can be found in the following folder.

Set A. HW3/data/1/C79XX.txt

Set B. HW3/data/2/C79XX.txt

Here XX refers to the names of the other images in a particular set.

We observe that the epipolar lines are accurate except for one image in which that point is not visible and hence the epipolar line for that image is not plotted.

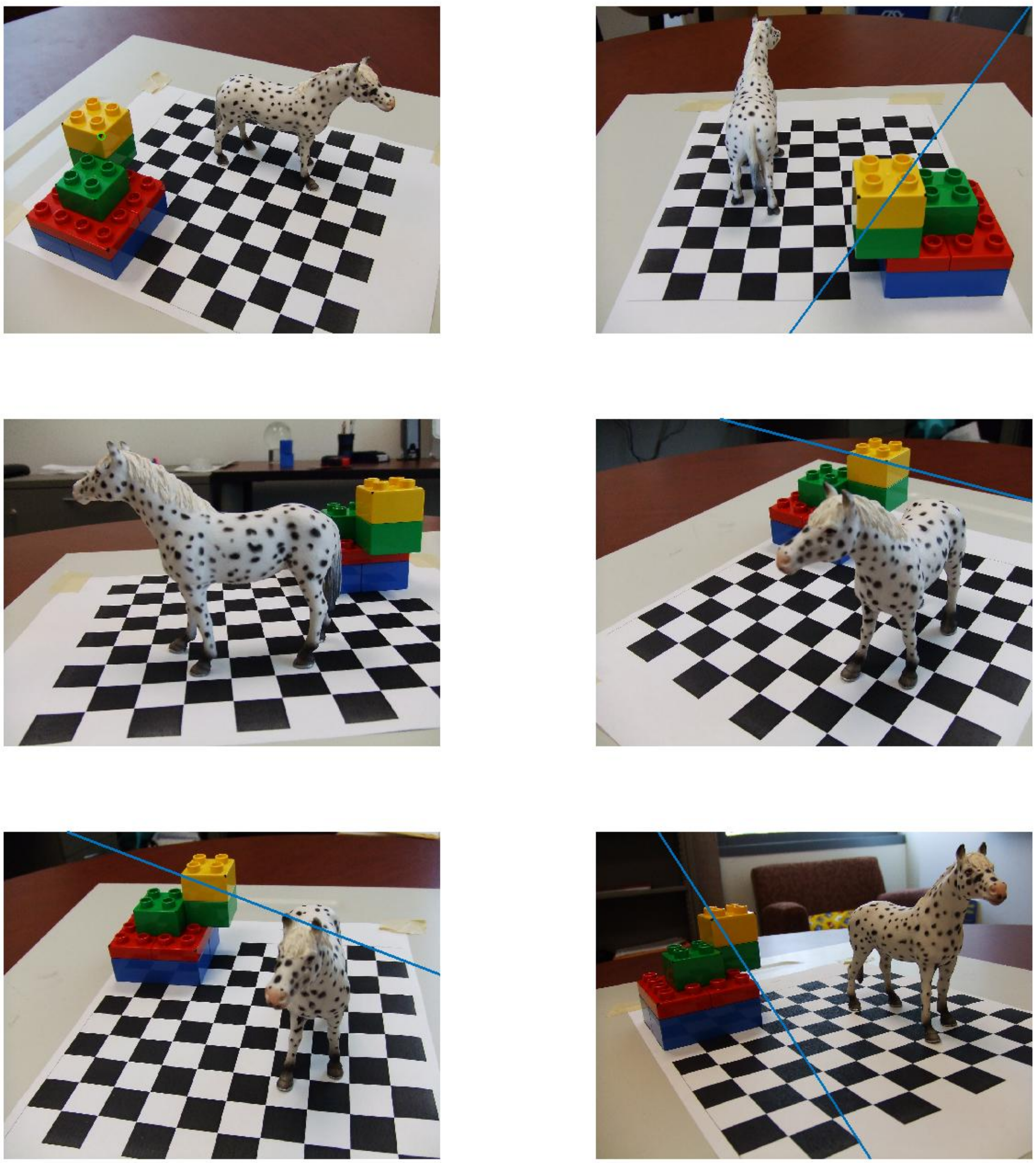


Figure 9: Epipolar Lines in Set A for 1 point

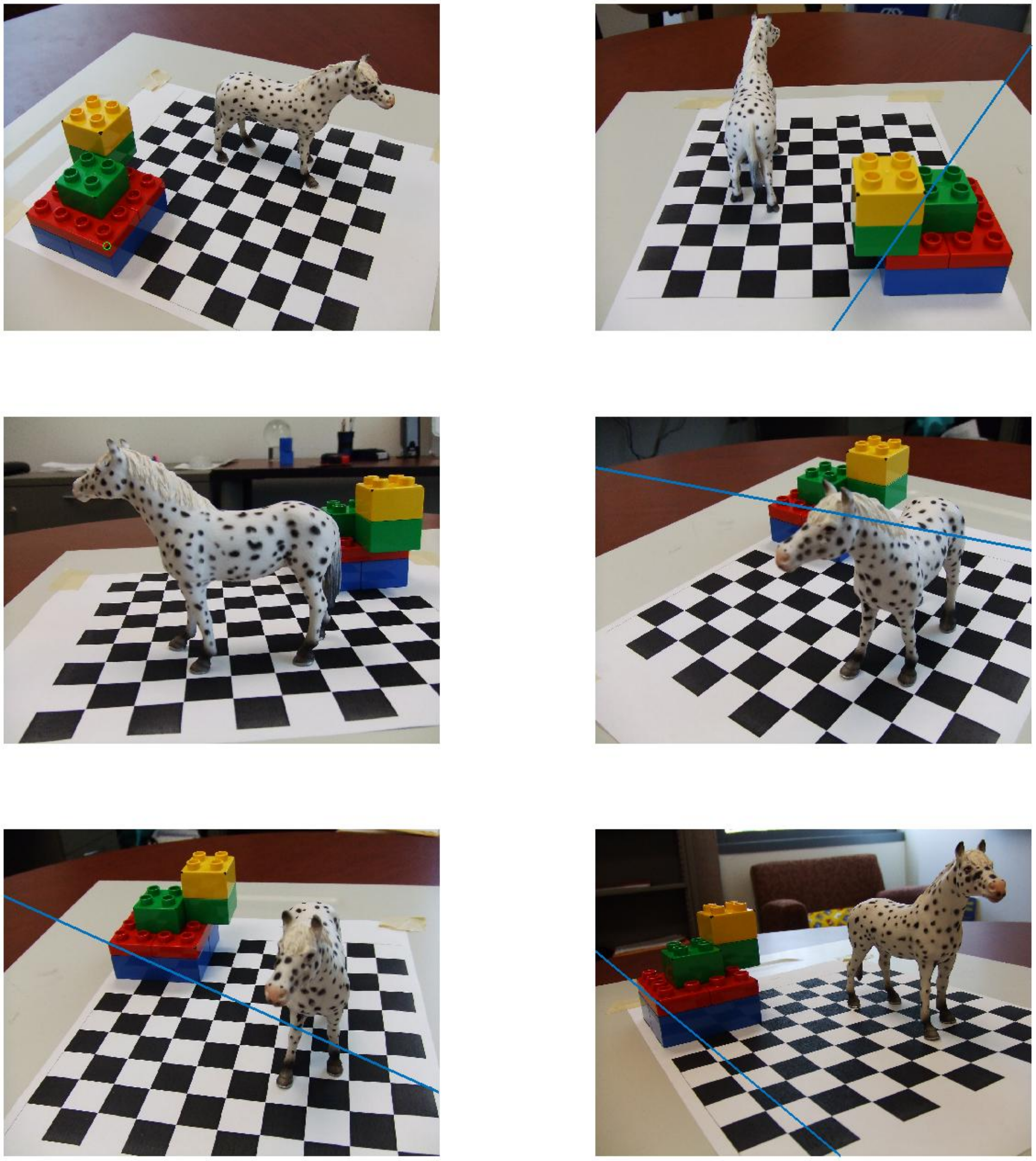


Figure 10: Epipolar Lines in Set A for 1 point

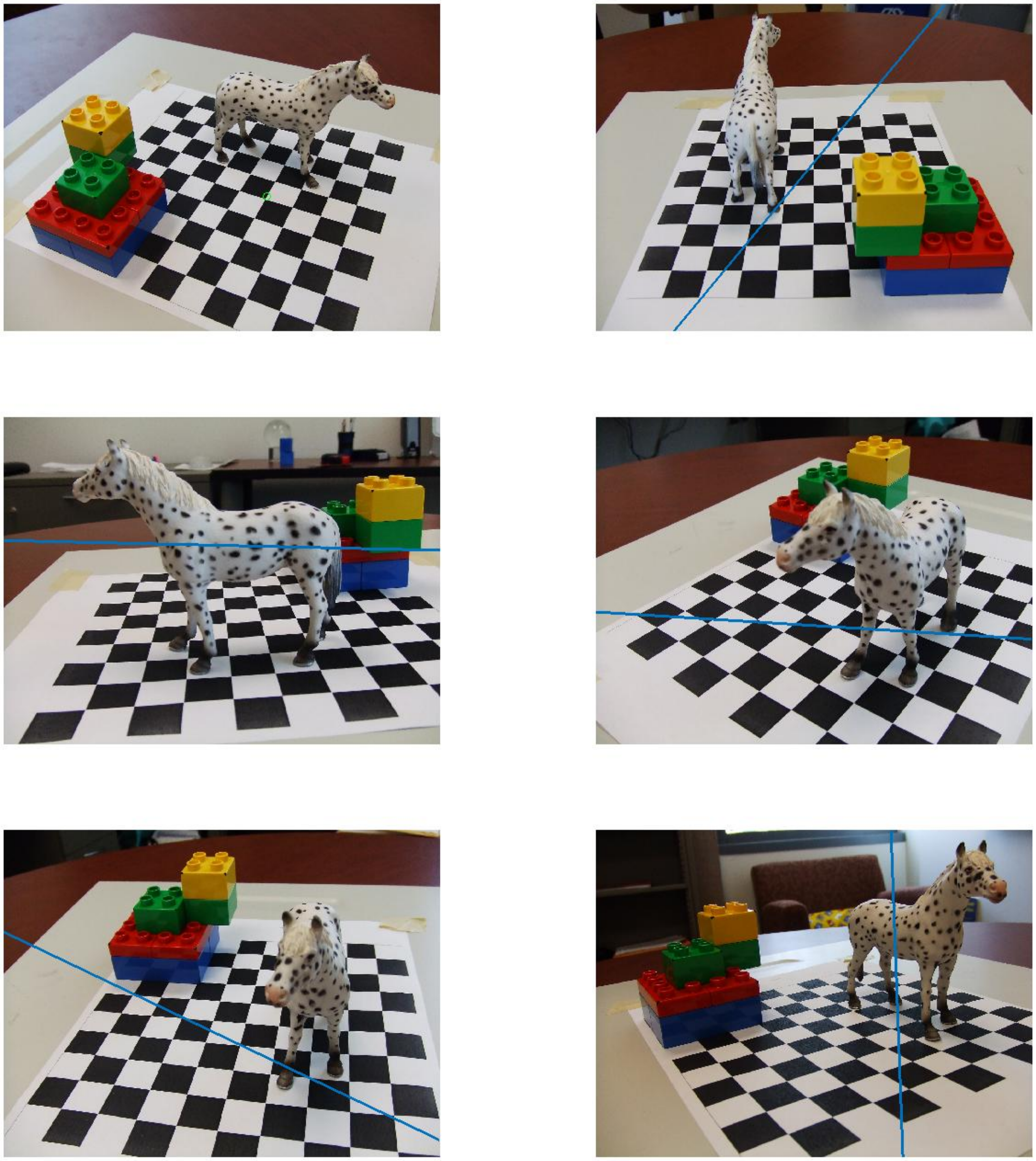


Figure 11: Epipolar Lines in Set A for 1 point

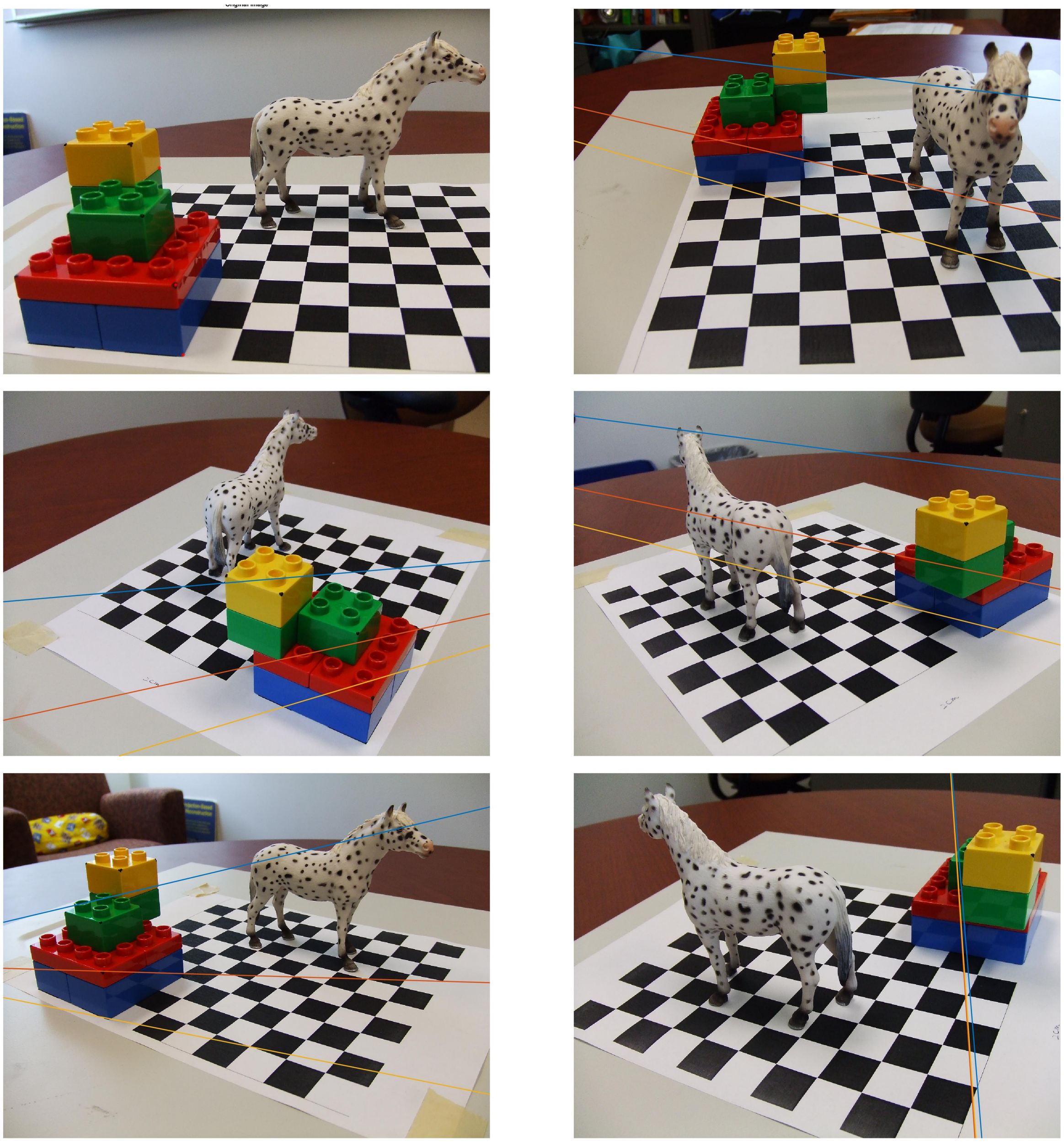


Figure 12: Epipolar Lines in Set B for 3 random points

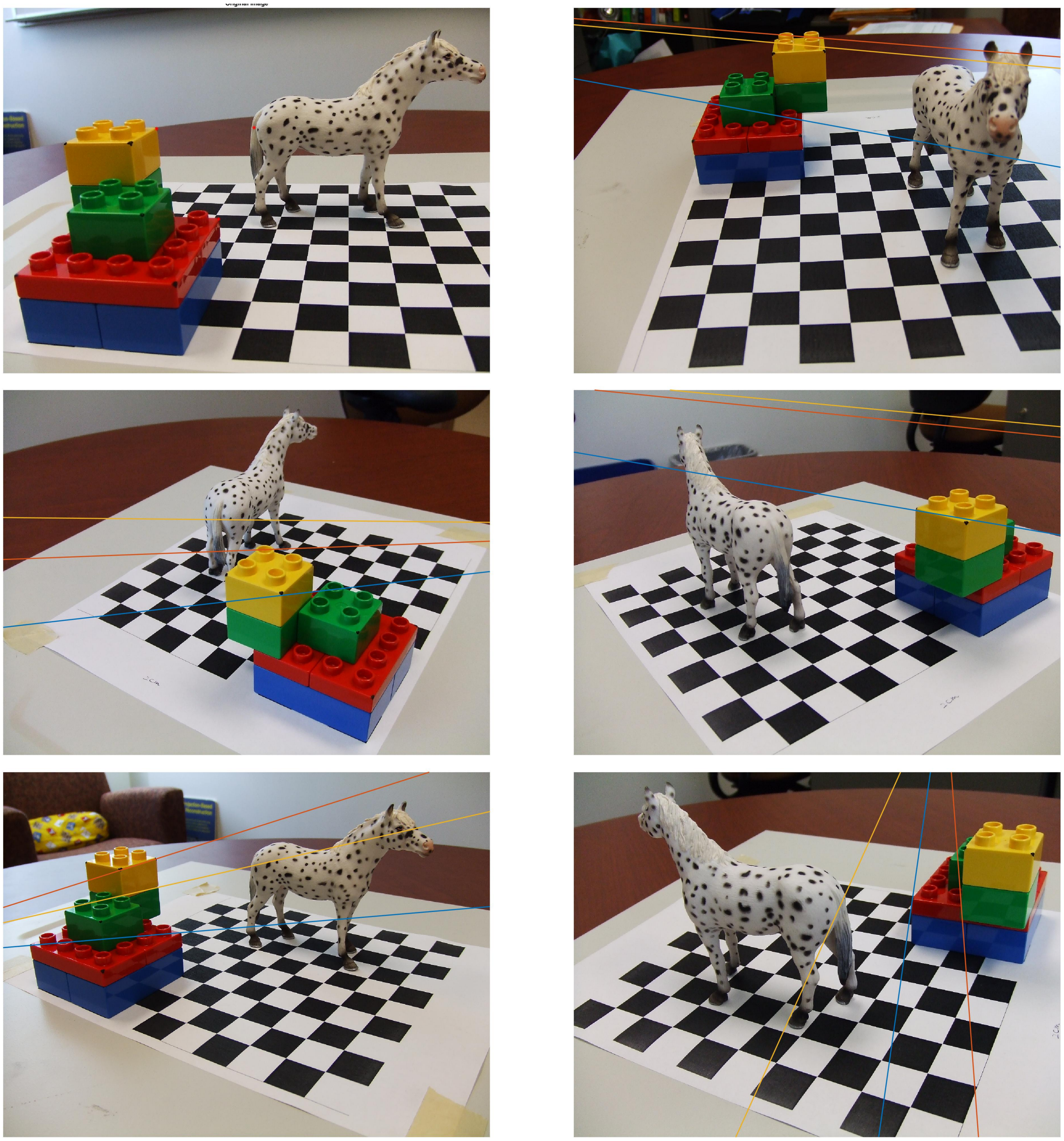


Figure 13: Epipolar Lines in Set B for 3 random points

Part 4: Depth Estimation

For the final part of estimating the depth we solve the linear equation for stereoscopic view using the equation given in class notes.

Initially we find out 60 correspondences among all the images in each set. Thus we over-constrained the linear equation and solve it using SVD.

We finally plot these points in the 3-d plane and observe the results.

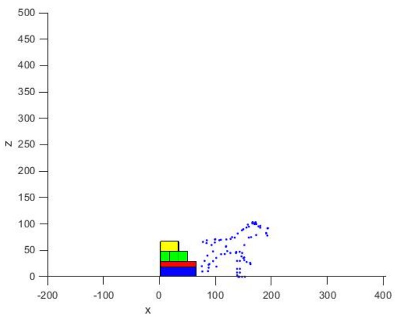
```
function [ XYZPoints ] = depthEstimation(CalibMatrix,camCo)
% camCo is an m x 2 x noOfCameras containing the m correspondence.
% CalibMatrix is the calibration matrix for numOfCameras (3 x 4 x noOfCamers)
[~,~,numCameras] = size(CalibMatrix);
% [~,numPts,numData] = size(data);
[numPts,~,~] = size(camCo);
for p = 1 : numPts
    %% Set up a linear system of equations for solving depth equation
    linsys = zeros(numCameras,4);
    for cam = 1 : numCameras
        row = 2 * cam - 1;
        % Writing the equation in the form of Ax = 0;
        % u-coordinate row
        linsys(row,1) = camCo(p,1,cam)*CalibMatrix(3,1,cam) - CalibMatrix(1,1,cam);
        linsys(row,2) = camCo(p,1,cam)*CalibMatrix(3,2,cam) - CalibMatrix(1,2,cam);
        linsys(row,3) = camCo(p,1,cam)*CalibMatrix(3,3,cam) - CalibMatrix(1,3,cam);
        linsys(row,4) = camCo(p,1,cam)*CalibMatrix(3,4,cam) - CalibMatrix(1,4,cam);
        % v-coordinate row
        linsys(row+1,1) = camCo(p,2,cam)*CalibMatrix(3,1,cam) - CalibMatrix(2,1,cam);
        linsys(row+1,2) = camCo(p,2,cam)*CalibMatrix(3,2,cam) - CalibMatrix(2,2,cam);
        linsys(row+1,3) = camCo(p,2,cam)*CalibMatrix(3,3,cam) - CalibMatrix(2,3,cam);
        linsys(row+1,4) = camCo(p,2,cam)*CalibMatrix(3,4,cam) - CalibMatrix(2,4,cam);
    end

    %Solve the linear system with SVD
    [U,S,V] = svd(linsys);
    % Solution is the vector V with lowest singular value
    P = V(:,end);
    % Finding out original coordinates from 4-D Homogenous Coordinates
    XYZPoints(:,p) = P ./ P(end);

end
```

NOTE : The 2D correspondence points among the images can be found in the following folder. We calculate the correspondence for about 130 points i.e 65 points in each set

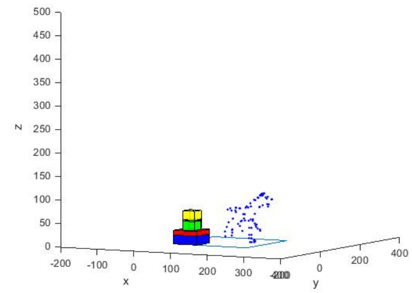
Set A. HW3/data/1/mPointsAll.txt

Set B. HW3/data/2/mPointsAll.txt

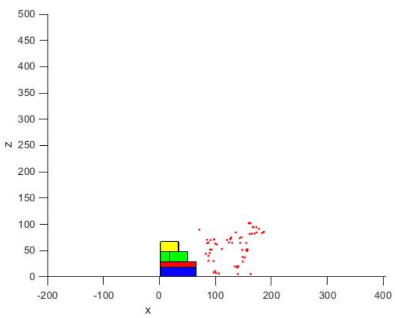
On plotting the 3-D points we observe that our results indeed represent a horse. Thus we were successfully able to map 2-D Camera Coordinates to 3-D World Coordinates. The plot in blue is for Set A and the one in red is for set B.



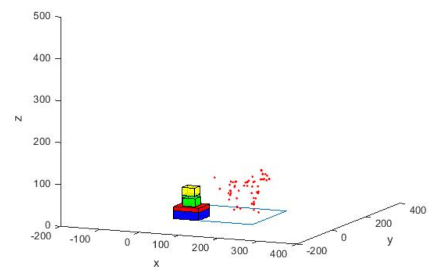
(a) Set A



(b) Set A



(c) Set B



(d) Set B

Figure 14: Depth Estimation for Set A and Set B

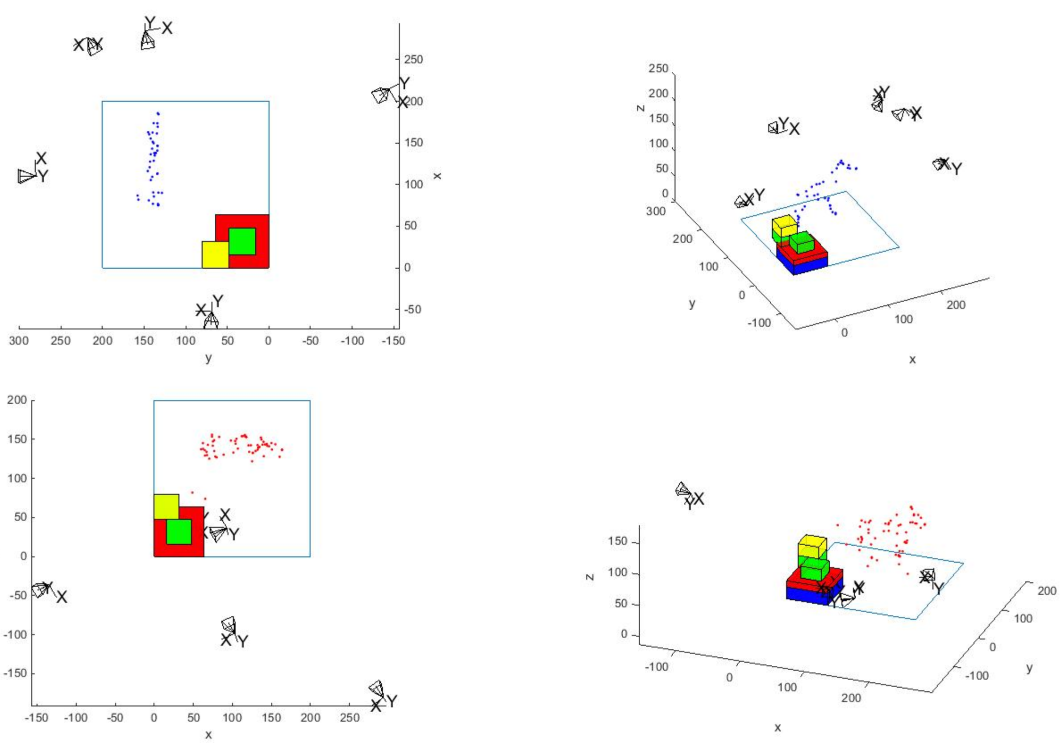


Figure 15: Depth Estimation and Camera Position

Next we merge the sets A and B. We see only one horse even after merging the data. Although we could have plotted more points but these are sufficient to get an idea that its indeed a horse.

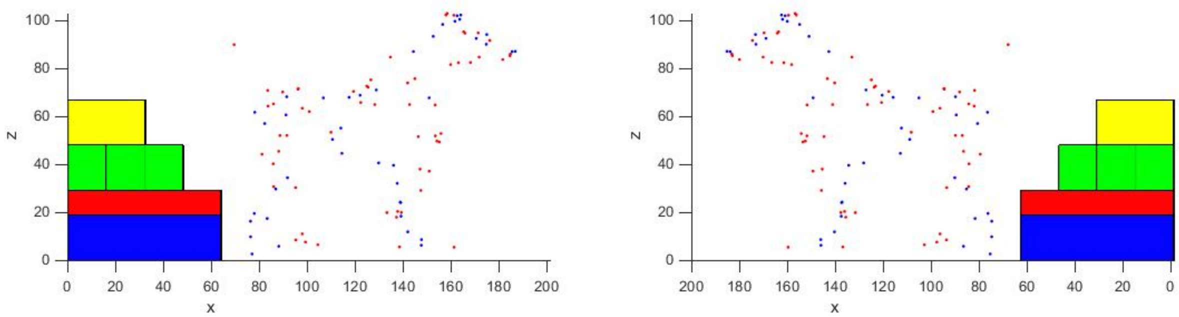


Figure 16: Depth Estimation (Both Sets Combined)

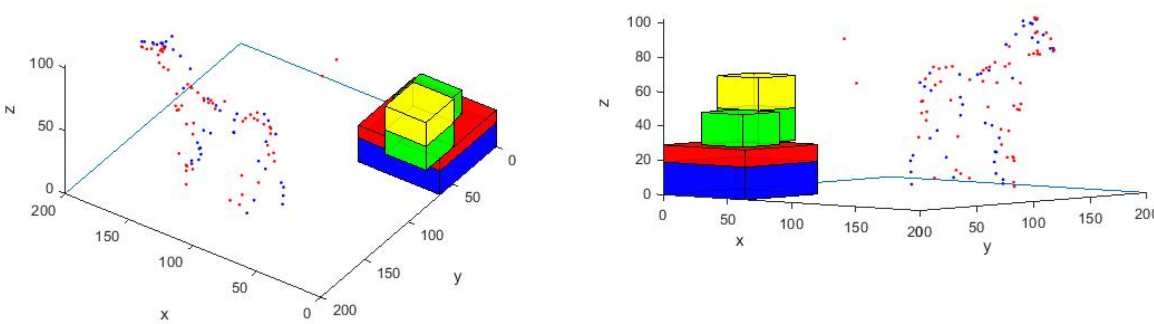


Figure 17: Depth Estimation (Both Sets Combined)

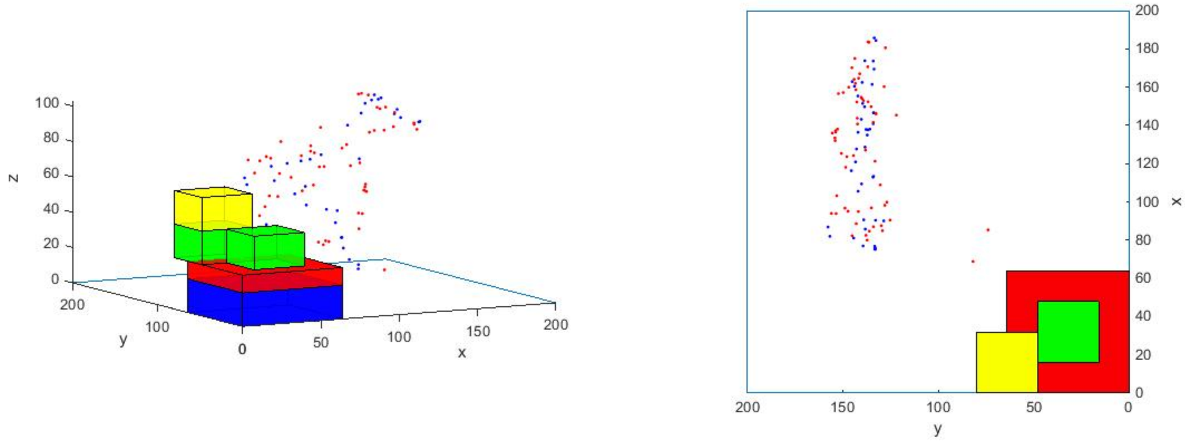


Figure 18: Depth Estimation (Both Sets Combined)

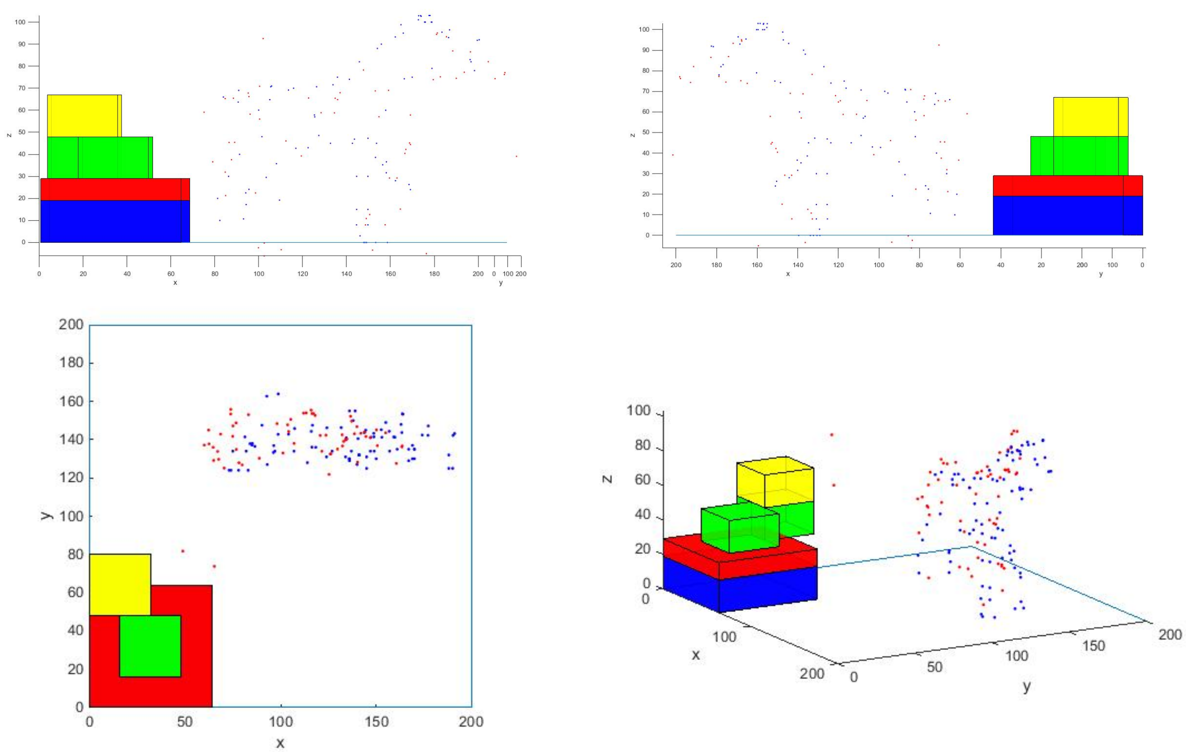


Figure 19: Depth Estimation (Both Sets Combined)