Structured Light Surface 3D Reconstruction

EE5176: Term Project

Mukhesh Pugalendhi Sudha EE18B114
Amalan S EE20D408
Tanvi Vinay Kulkarni EE20S046

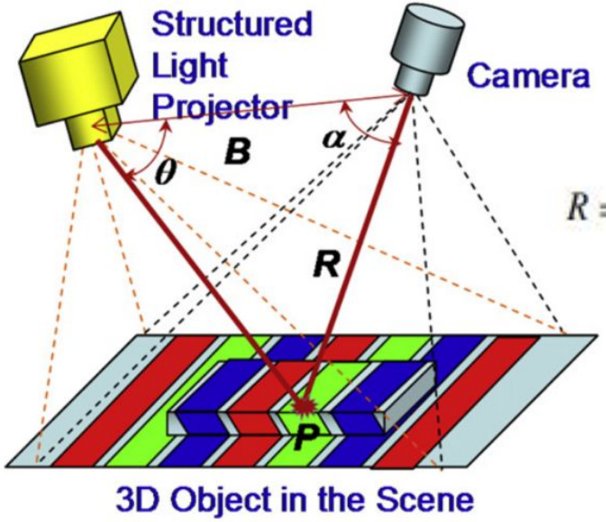
Problem Statement :

3D Surface Imaging using Structured Light

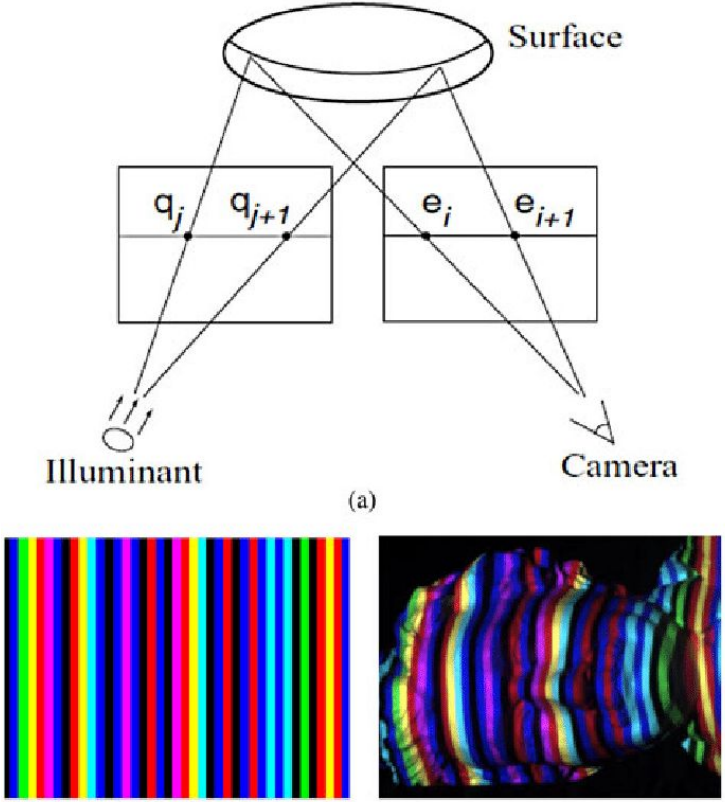
Proposed Approach:

Structured light is projected on a nonplanar surface which distorts the geometric patterns of the projected light as seen from the camera.
These distortions are used as a cue to obtain a depth map for the surface!

Approach:



$$R = B\frac{\sin(\theta)}{\sin(\alpha+\theta)}.$$

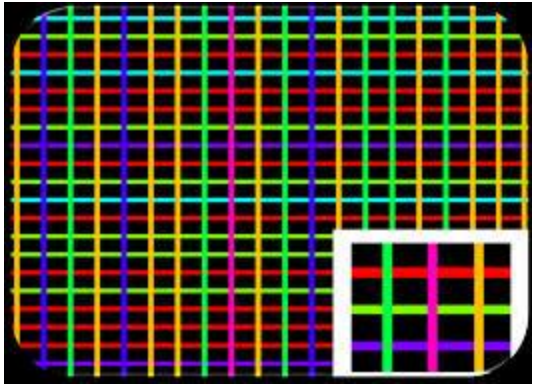


(a)

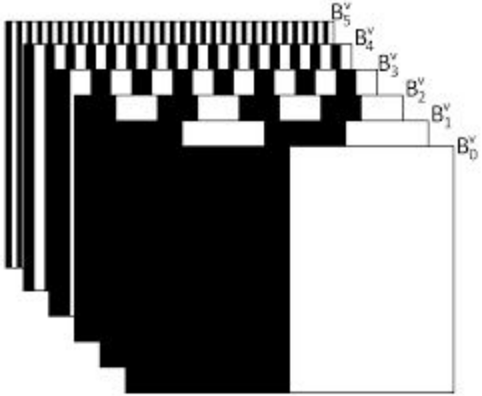
(b)

(c)

Types of projection patterns



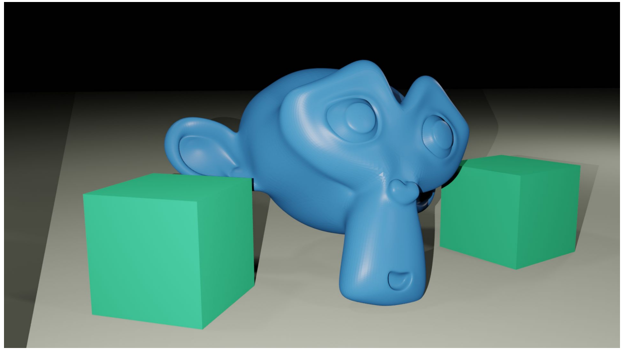
Single shot
Color Coded Grids

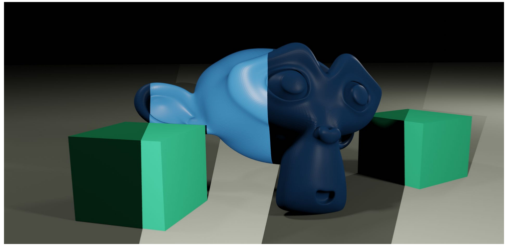


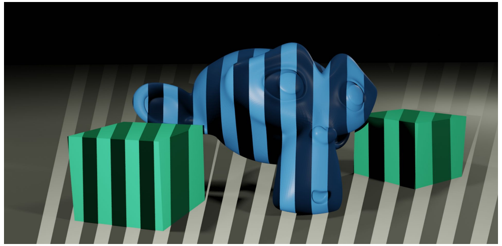
Multi shot
Sequential Binary Coded Pattern

Problem 1- Structured light simulation with blender

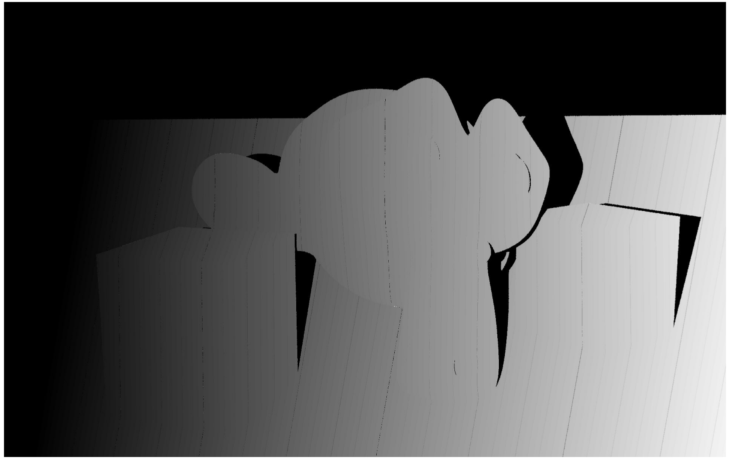
Simulation Results for sequential binary coded pattern :



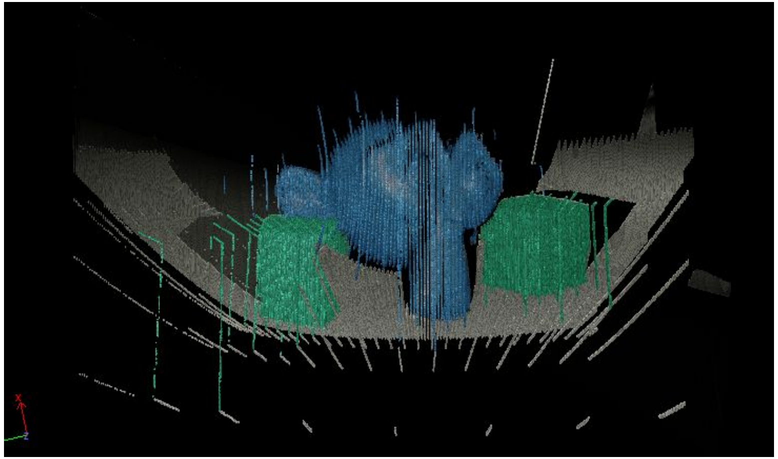




Obtaining depth from multishot pattern projection

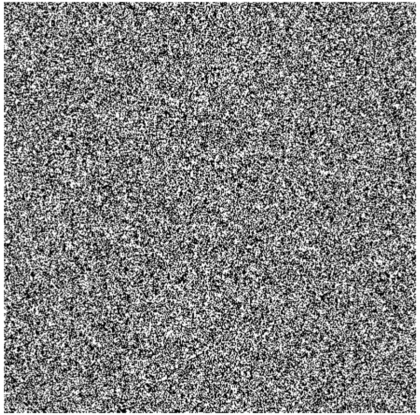


Pixel-wise code decoded from Sequential binary coded images

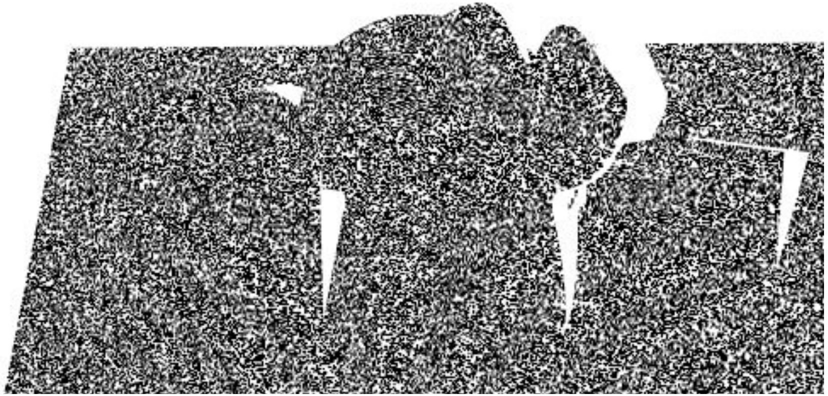


Reconstructed 3d point cloud

Obtaining depth from single shot pattern projection



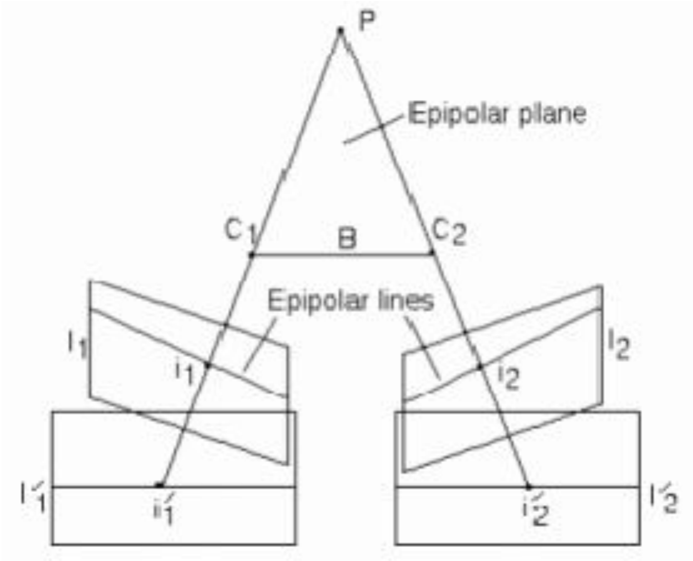
Random Pattern used for projection



Pixel-wise code decoded from the single shot image

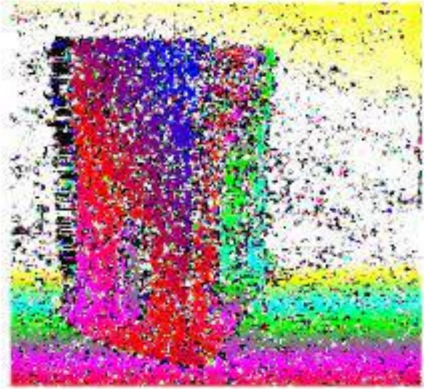
Bottlenecks:

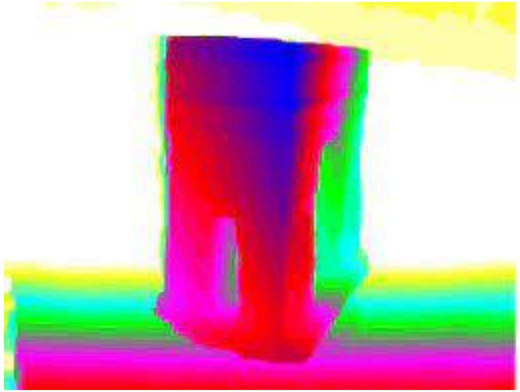
Rectification of stereo images to make the epipolar lines parallel



Subsequent Work Plan for next 2 weeks

Use a deep network to obtain an estimated disparity map with a random projected pattern as per the paper “Connecting the Dots: Learning Representations for Active Monocular Depth Estimation network weights” by Riegler et. al.





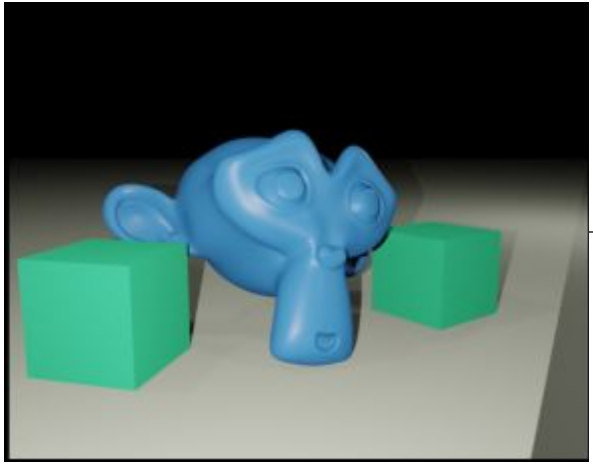
Individual Contributions

Team Member	Contribution
1. Amalan	Explored blender and set up the scene for multishot
2. Mukhesh	Implemented functions to find correspondence
3. Tanvi	Find out depth map for multishot

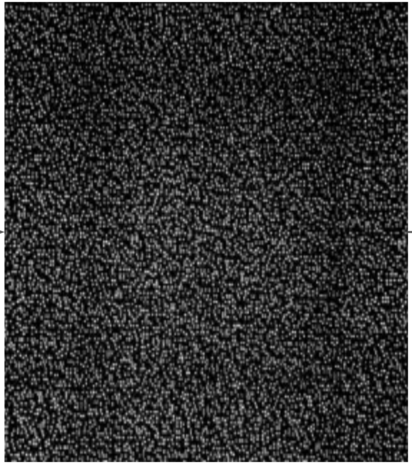
Thank you

Review Problem 1

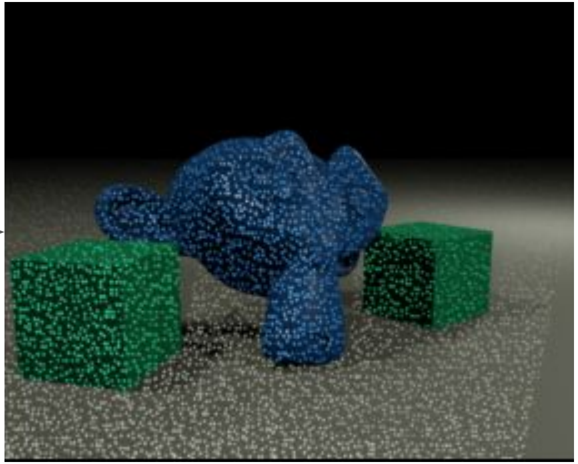
Blender Simulation of random pattern projection and 3D reconstruction using stereo matching



3D Scene Setup



Random Pattern used for projection



Pattern Overlay on 3D Scene

Random pattern reference: “Connecting the Dots: Learning Representations for Active Monocular Depth Estimation network weights” by Riegler et. al.

Structured Light Surface 3D Reconstruction

EE5176: Term Project

Phase 2 - 06.05.2020

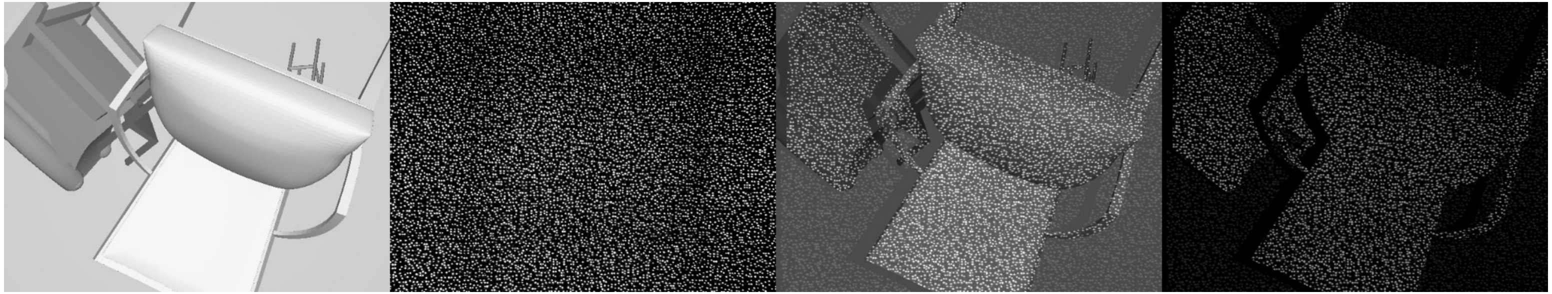
Amalan S EE20D408
Mukhesh Pugalendhi Sudha EE18B114
Tanvi Vinay Kulkarni EE20S046

Link to repo

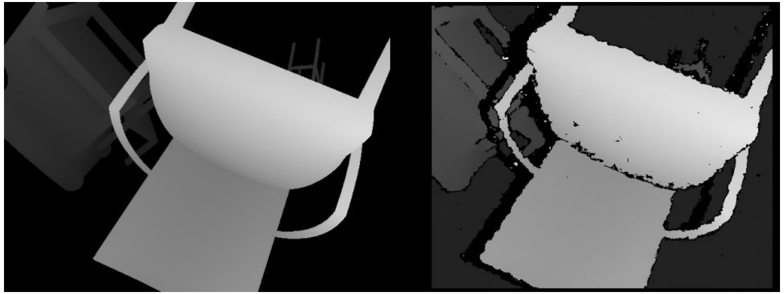
Review Problem 1

Single-shot Random pattern projection and 3D reconstruction using stereo matching

Scene *Projection pattern* *Image captured* *Extracted code*



Ground truth *Stereo matching*



Problem Statement 2

Compare results of state-of-the-art deep network with those of traditional single-shot methods

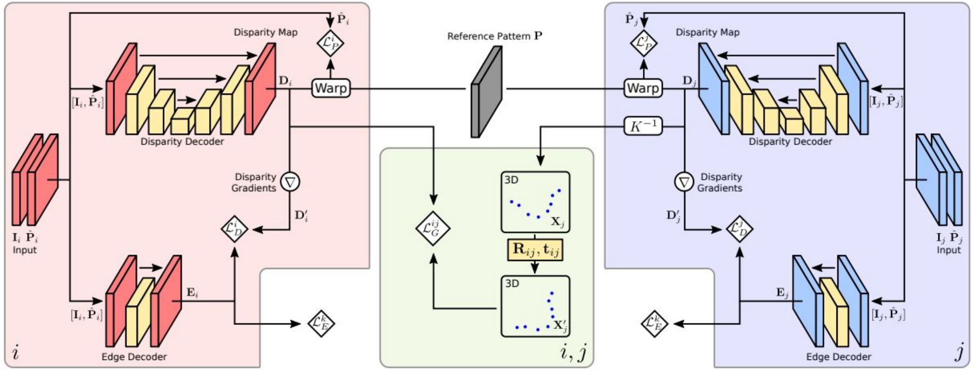
Network reference: “Connecting the Dots: Learning Representations for Active Monocular Depth Estimation network weights” by Riegler et. al.

About the Paper

Simple convolutional architecture to get high-quality disparity estimates

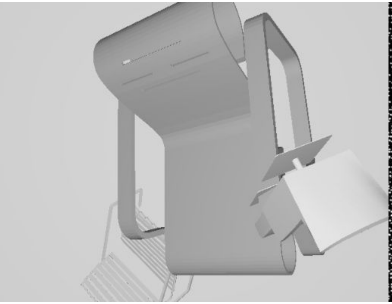
Model trained in a supervised fashion with a combination of photometric, geometric, warp and edge losses

LCN layer for patchwise normalization to extract the code from image captured under structured light



Deep Learning Framework Results: 1/3

Scene



Projection pattern

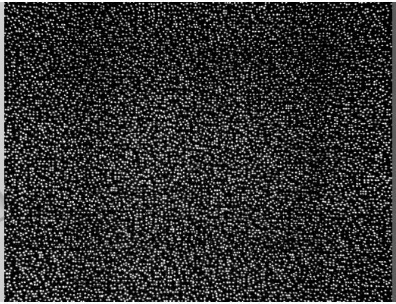
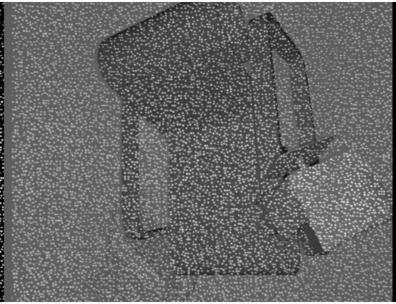
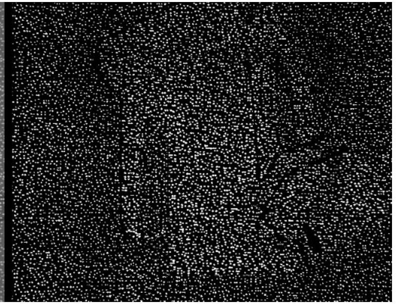


Image captured

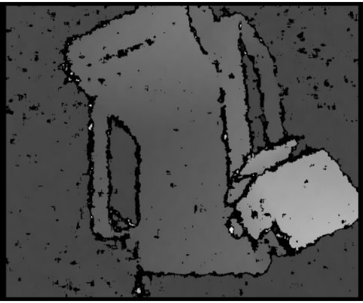


Extracted code

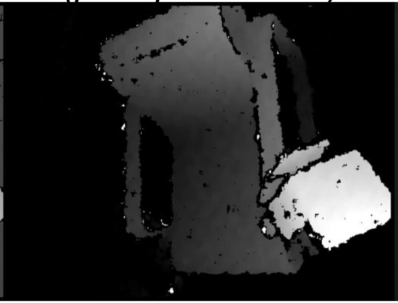


Ground truth

Stereo matching



Stereo matching (post processed)

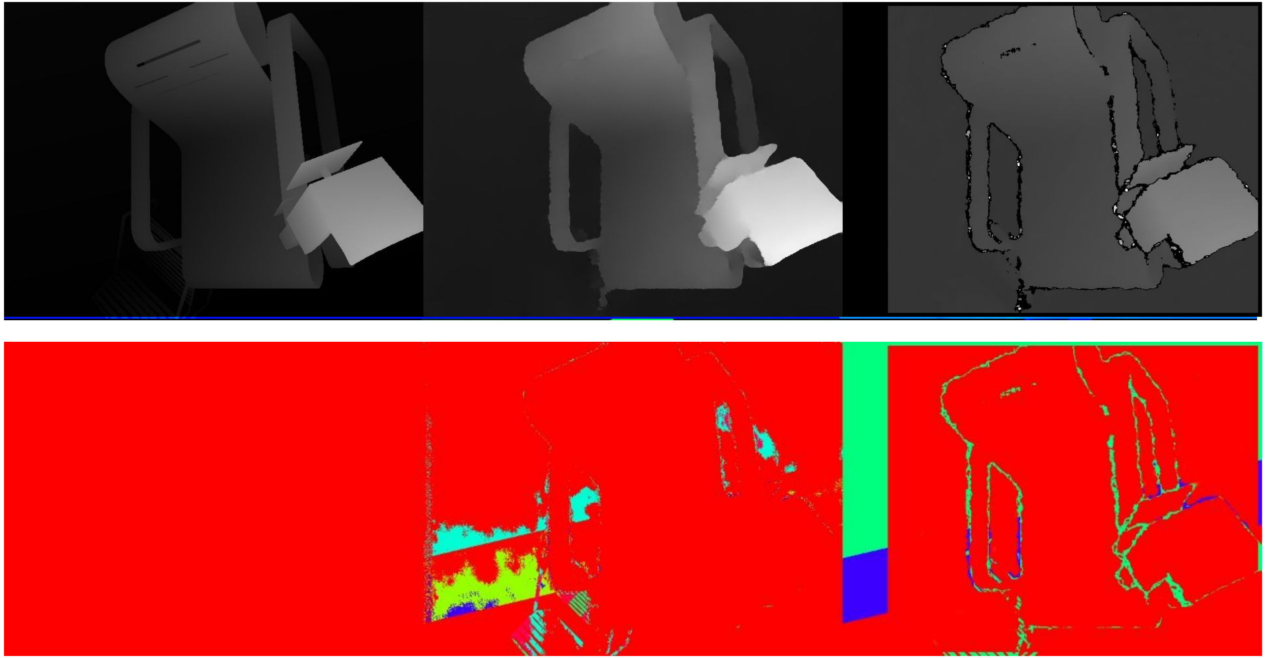


Connecting the dots

Deep Learning Framewrok Results: 1/3

Outputs

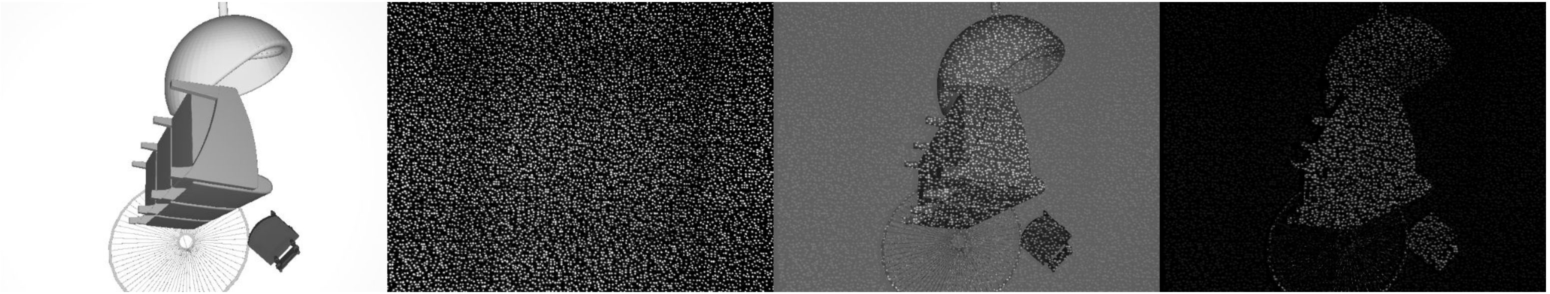
Errors



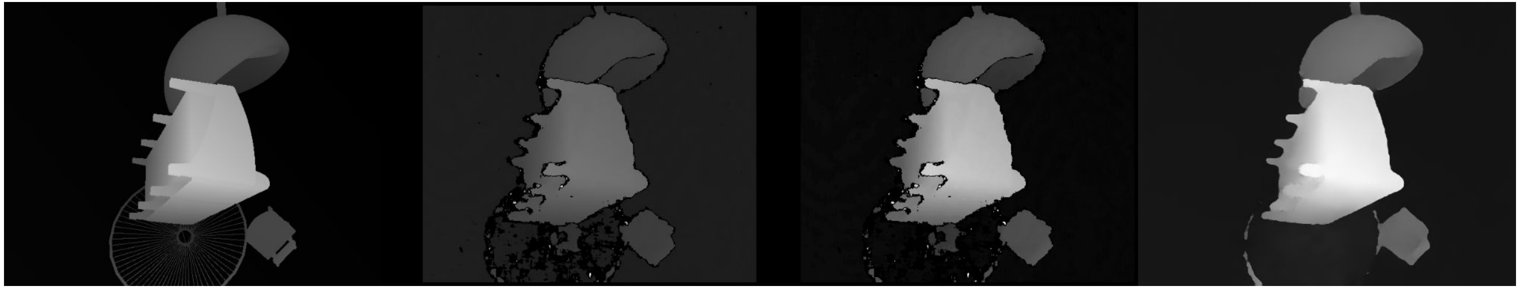
Ground truth *Connecting the dots* *Stereo matching*

Results 2/3

Scene | *Projection pattern* | *Image captured* | *Extracted code*



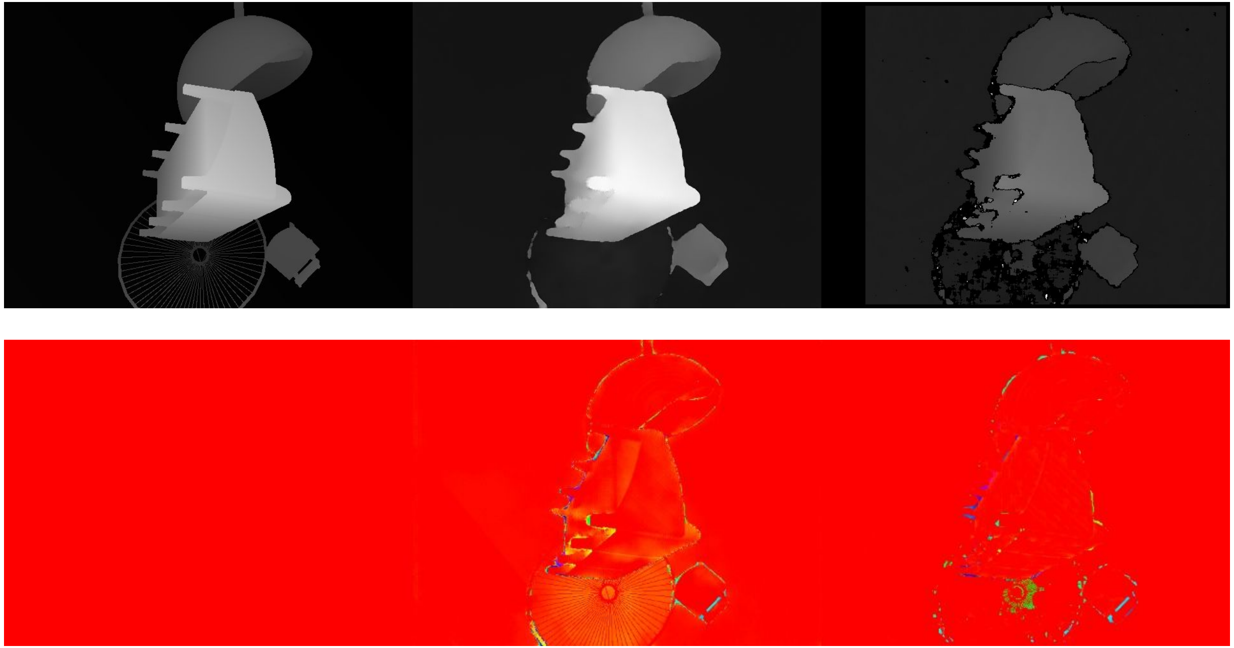
Ground truth | *Stereo matching* | *Stereo matching (post processed)* | *Connecting the dots*



Deep Learning Framewrok Results: 2/3

Outputs

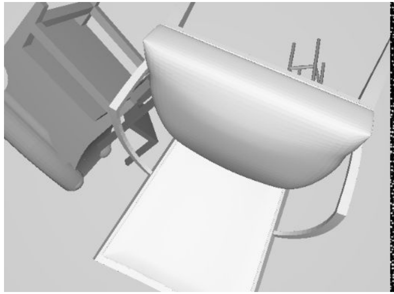
Errors



Ground truth *Connecting the dots* *Stereo matching*

Results 3/3

Scene



Projection pattern

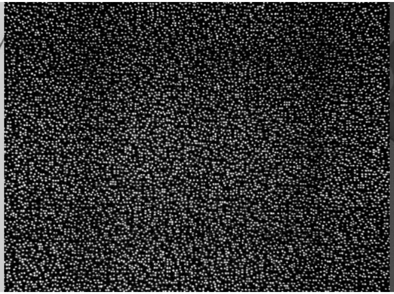
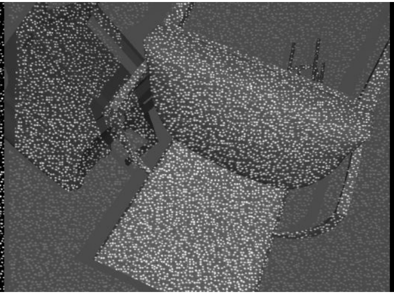
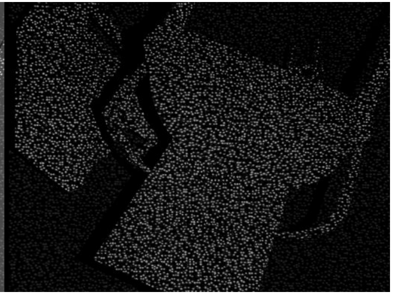


Image captured

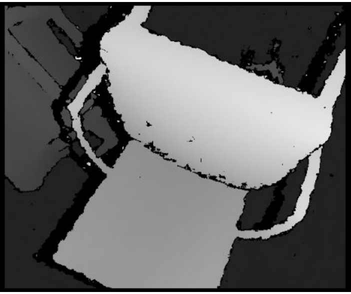


Extracted code

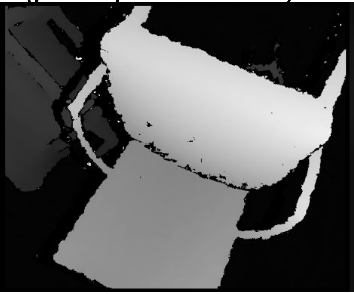


Ground truth

Stereo matching



Stereo matching (post processed)

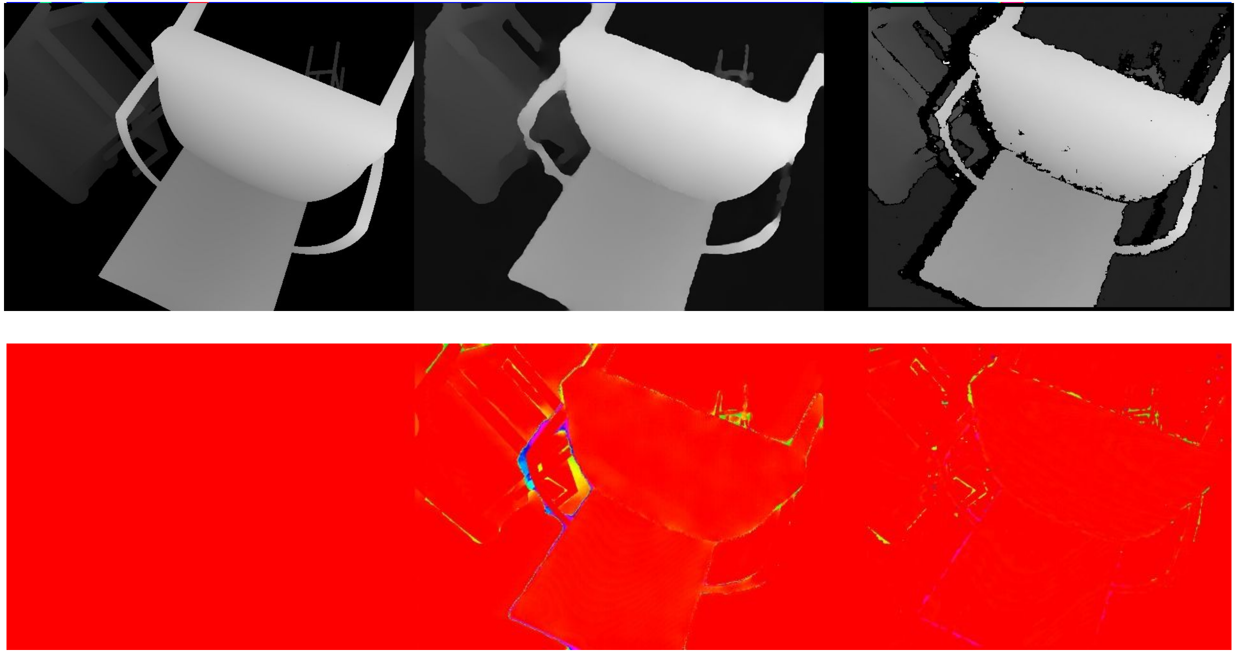


Connecting the dots

Deep Learning Framework Results: 3/3

Outputs

Errors



Ground truth *Connecting the dots* *Stereo matching*

Bottlenecks

- Tracing implicit code base was time consuming and version specific pytorch and other libraries were needed to be installed
- Inability to use custom rendered data to produce results with the trained network

Subsequent Work Plan for next 2 weeks

- Try to improve the network output for our input scene
- Use an optimized pattern to train the network for better results
- Explore alternate ideas

Individual Contributions

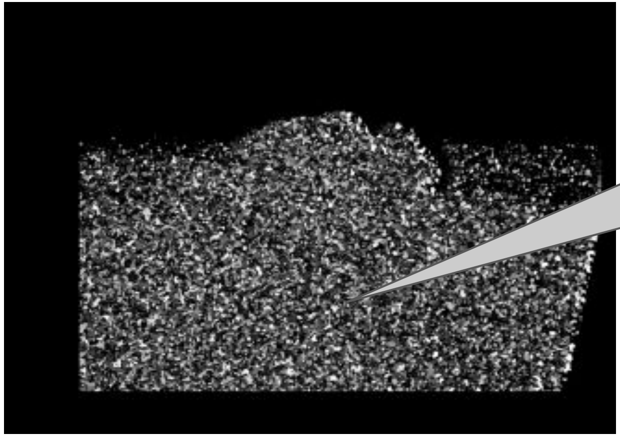
Team Member	Contribution
1. Amalan	● Got the repo shared by the authors working in GTX 1050 Ti (required >2.5GB RAM) ● Found the settings required in blender to obtain a comparable render as required by the network
2. Mukhesh	● Explored the code base to extract important functions and classes needed for inference ● Inverse projected to 3D for visualization
3. Tanvi	● Explored 3D reconstruction using block matching ● Explored stereo rectification for custom input data and unconstrained camera and projector setup

Thank you

Problem Statement 2- Deep Learning V/S Block Matching

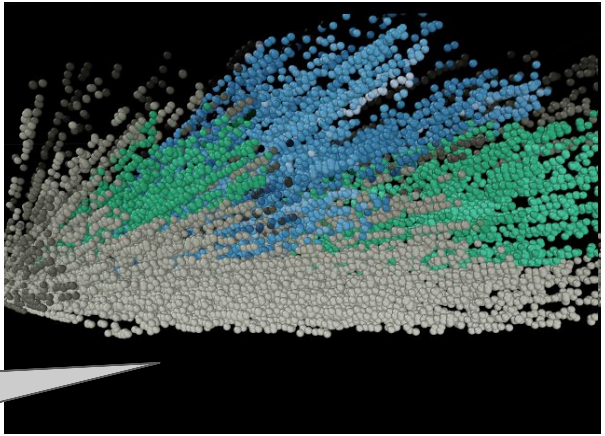
<u>Block Matching Results:</u>

- traditional method
- Uses stereo block-matching
- Works well with higher image resolutions



Does not show the correct disparity

Incoherent 3D point cloud



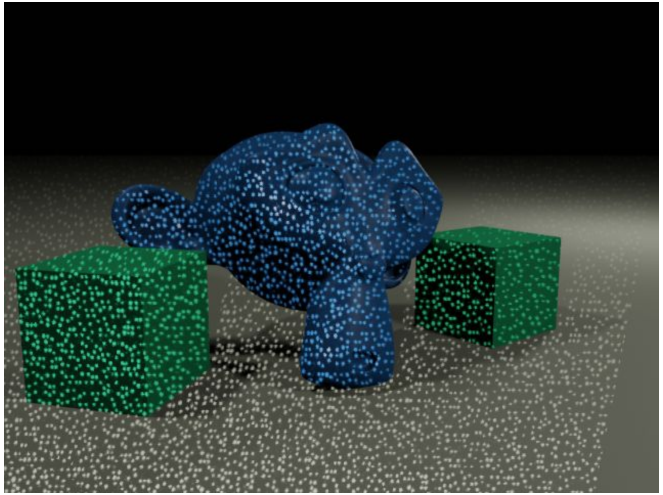
Disparity Map

Reconstructed 3D Point Cloud

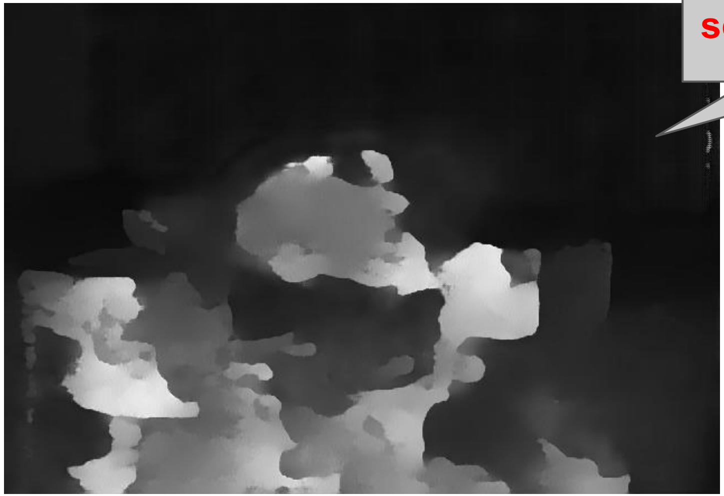
Deep Learning Framework Results:

- State-of-the-art method based on trained neural network as presented in paper "Connecting the Dots: Learning Representations for Active Monocular Depth Estimation network weights" by Riegler et. al.

Does not work well with custom input scene



Pattern Overlay on 3D Scene



Obtained Disparity Map

Structured Light Surface 3D Reconstruction

EE5176: Term Project

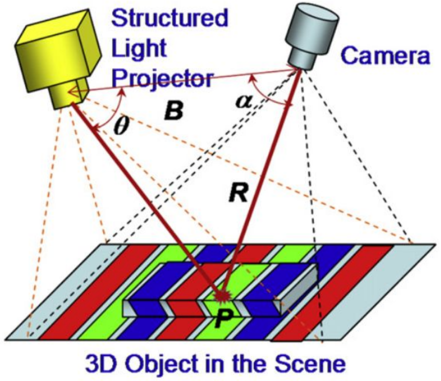
Phase-3 Final Presentation

Amalan S EE20D408
Mukhesh Pugalendhi Sudha EE18B114
Tanvi Vinay Kulkarni EE20S046

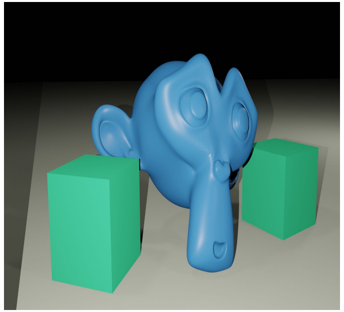
Network reference: “Connecting the Dots: Learning Representations for Active Monocular Depth Estimation network weights” by Riegler et. al.

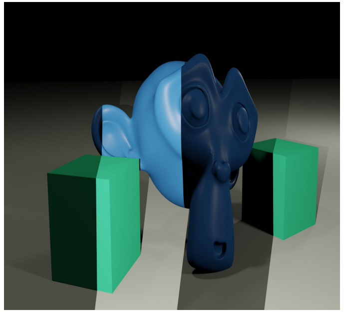
Story so far...

3D Reconstruction with multishot patterns

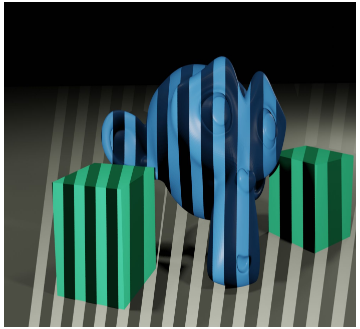


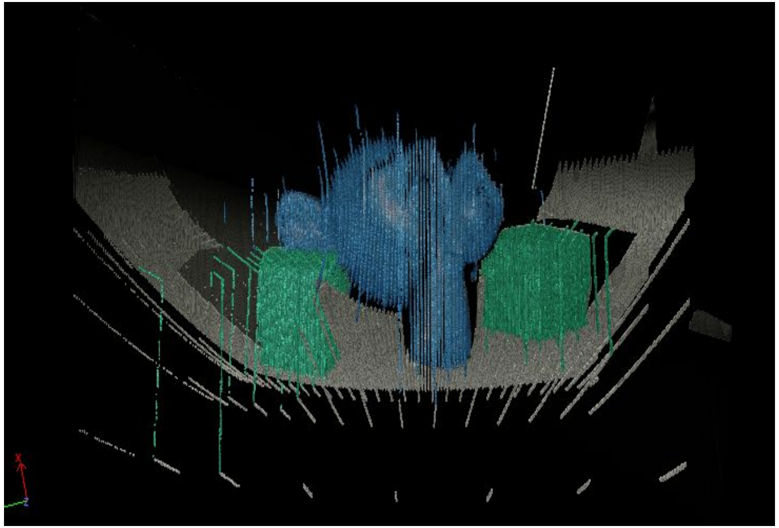
$$R = B\frac{\sin(\theta)}{\sin(\alpha+\theta)}.$$





Multishot patterns



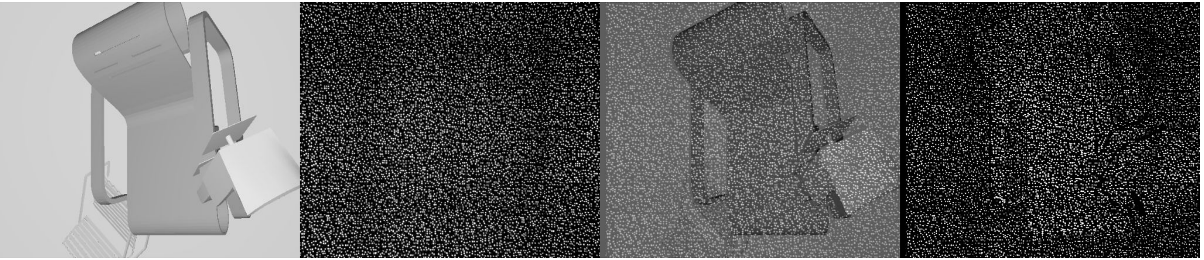


Reconstructed 3D point cloud

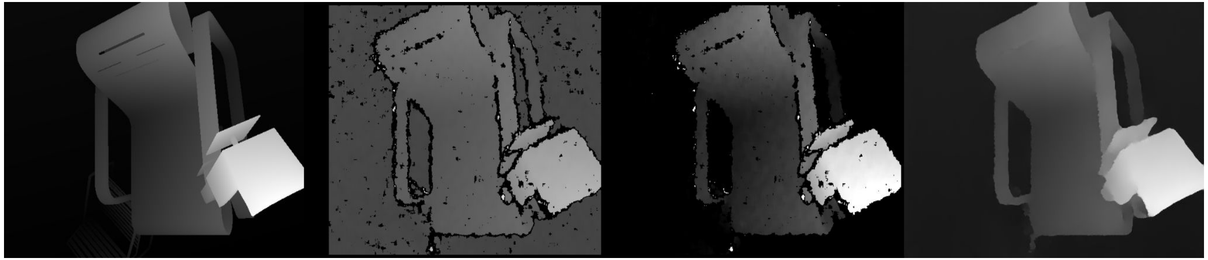
Story so far...

3D Reconstruction with single-shot pattern

Scene — *Projection pattern* — *Image captured* — *Extracted code*



Ground truth — *Stereo matching* — *Stereo matching (post processed)* — *Connecting the dots*



Proposed work for Phase 3

- Evaluate inferences on a custom rendered scene
- Train network with 1 pattern on the custom rendered scene
- Train network with 3 random patterns with an objective to find a better disparity map

Network inferences on custom rendered scene

Scene

Projection pattern

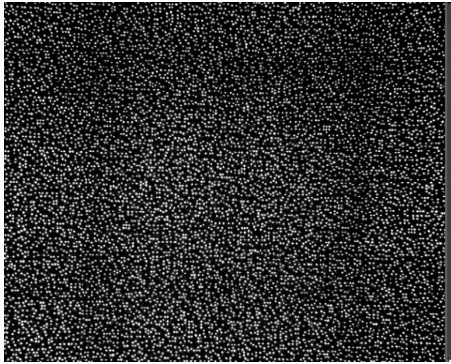
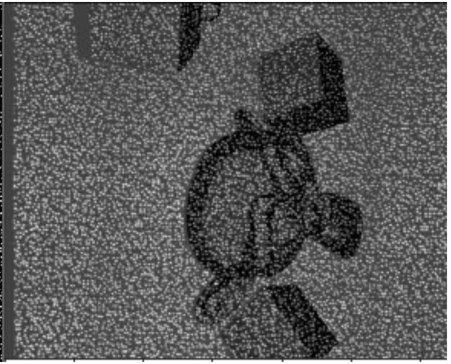
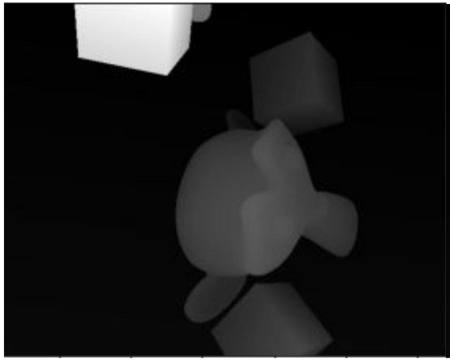


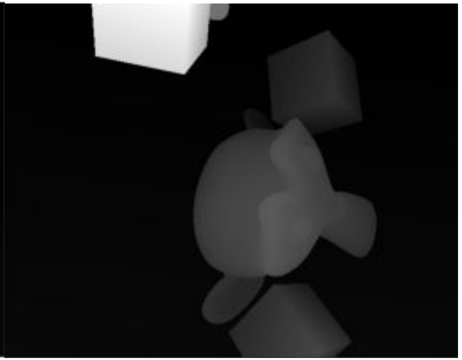
Image captured



Ground truth



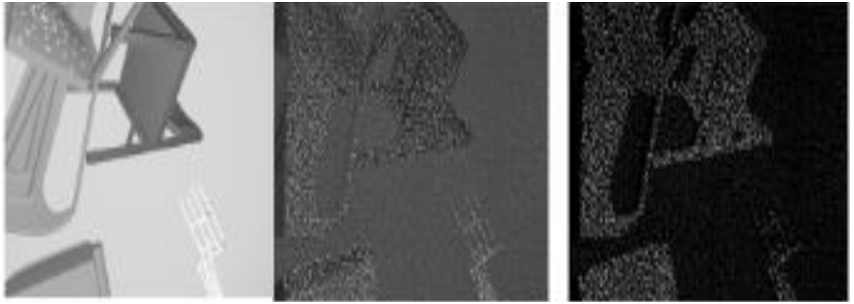
Trained network output



Bottlenecks

- In training network with single pattern, the input rendered scene could not be captured in the warped pattern(same issues for kinect pattern and our random pattern)

Expected Input and warped input images



Our Input and warped input images

Bottlenecks

- In training with 3 random patterns, lack of DL understanding to use 3 input images and their respective losses and blending them into one disparity map, also the pattern dependent loss calculation was ambiguous with 3 patterns
- Acquiring a workstation with 12GB GPU and compilation issues were time consuming tasks

Individual Contributions

Team Member	Contribution
1. Amalan	● Tried network training with single pattern ● Obtained workstation and resolved compilation issues
2. Mukhesh	● Generated 3 random dot patterns ● Tried to use 3 patterns during training
3. Tanvi	● Imported custom rendered scene from blender and created input dataset ● Tried network training with single pattern

Thank you