3D Reconstruction from accidental motion

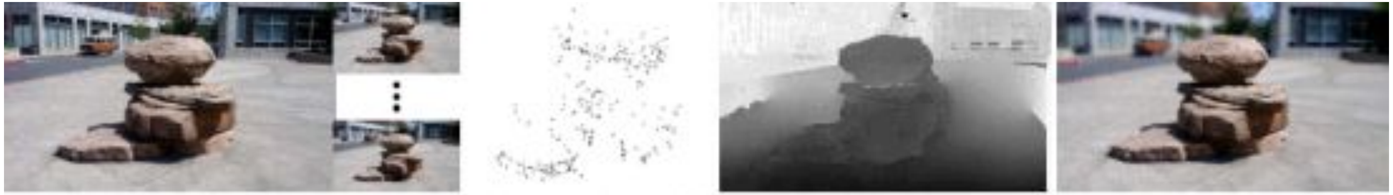
Team: SfM

Aim

3D scene reconstruction from a set of initial frames of a video capture by exploiting accidental motion

Input: Sequence of frames part of a video

Output: 3D reconstruction depth map of a reference frame



(a) Input image sequence (b) Foreground points of SfM (c) Dense reconstruction (d) Synthetic aperture

Problem formulation

- Given an image sequence of N_C images and N_P 2D projections of 3D points seen from every camera, we try to estimate the ground-truth 3D coordinates of the corresponding real-world points using bundle adjustment.
- The first frame is considered as reference frame and all the 3D points are parameterized by the inverse depth relative to the reference frame. All the camera poses are estimated using bundle adjustment with random depth initialization.
- Using the estimated camera poses, the 3D scene is densely reconstructed to solve for a smooth depth map. A conditional random field energy function is minimized using to regularize the depth estimation.
- Long-range connections between pixels are incorporated to pass information to a pixel effectively better than adjacent pixel connections.

Key words

Bundle Adjustment: It refers to the problem of solving for poses and location of pixel values for a given estimated initial poses and location of 3D points.

Accidental motion: When one intends to hold a camera still, there is inevitable motion due to hand shaking or heart beating, especially when a lightweight camera like a smartphone, is used. This type of motion is called *accidental motion*.

Steps in solution

1. **Initialization**
2. **Optimization**
3. **Dense reconstruction**

Pipeline

KLT Tracking

We first track features between all the frames using KLT tracking

- Find Shi tomasi corners. The main difference between shi tomasi corners and harris corners is the change in scoring function.

$$R = min(\lambda_1, \lambda_2)$$

- We then filter out corners by finding homography matrix between the reference frame (frame 0) and every other frame in the video sequence.
- We select those corners that are inliers for more than 95 % of camera frames found by estimating homography matrix.
- We find optical flow over all the images of the sequence.

Initialization

To find the minima a good initialization for the Bundle adjustment optimization is very important.

- Given a **sequence of images** the **first image is kept as reference image** and all the other images are initialized with **zero rotation and translation with respect to this image as the motion is accidental and very small**.
- The projections of 3D points are found by feature tracking using KLT (Kanade-Lucas-Tomasi) function over the sequence.
- The 3D points are initialized by its inverse depth.

Bundle Adjustment

- Given a set of images depicting a number of 3D points from different viewpoints, bundle adjustment can be defined as the problem of simultaneously refining the 3D coordinates describing the scene geometry, the parameters of the relative motion, and the optical characteristics of the camera(s) employed to acquire the images, according to an optimality criterion involving the corresponding image projections of all points. Bundle Adjustment problem requires good initial estimate of camera poses and points to solve the reprojection error.
- Reprojection error is the error between the projected 3D point on the image frame and the observed pixel.
- Bundle adjustment optimizes for both 3D point locations and camera poses.

Bundle adjustment optimization

In our case the loss function is the L2 norm of the reprojection error of 3D points with respect to the pixel values computed by tracking corner pixels. We use ceres solver to solve the Bundle Adjustment problem. The cost function is given as follows:

$$
\begin{aligned}
F &= \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} \|p_{ij} - \pi(R_i P_j + T_i)\|^2, \\
&= \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} (\frac{e_{ij}^x + f_{ij}^x w_j}{c_{ij} + d_{ij} w_j})^2 + (\frac{e_{ij}^y + f_{ij}^y w_j}{c_{ij} + d_{ij} w_j})^2,
\end{aligned}
$$

$$
\begin{aligned}
\alpha_{ij}^x &= x_j - \theta_i^z y_j + \theta_i^y, \\
b_{ij}^x &= T_i^x, \\
\alpha_{ij}^y &= y_j - \theta_i^x + \theta_i^z x_j, \\
b_{ij}^y &= T_i^y, \\
c_{ij} &= -\theta_i^y x_j + \theta_i^x y_j + 1, \\
d_{ij} &= T_i^z, \\
e_{ij}^x &= p_{ij}^x c_{ij} - \alpha_{ij}^x, \\
f_{ij}^x &= p_{ij}^x d_{ij} - b_{ij}^x, \\
e_{ij}^y &= p_{ij}^y c_{ij} - \alpha_{ij}^y, \\
f_{ij}^y &= p_{ij}^y d_{ij} - b_{ij}^y. \text{ (3)}
\end{aligned}
$$

Dense reconstruction

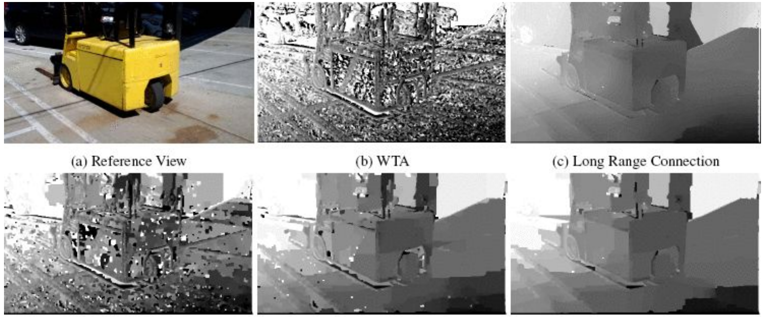
After getting structure from motion our next goal is to reconstruct the object. As the images are taken from the same viewpoint we can reconstruct the object only from a common viewpoint. The depth at each pixel estimated by the optimization tends to be noisy. To smoothen out the variation between points we adopt a CRF framework to solve a smooth depth map.

$$E(\mathrm{D}) = E_p(\mathrm{D}) + \alpha E_s(\mathrm{D}).$$

$$E_p(\mathrm{D}) = \sum_{i \in \mathcal{I}} \mathrm{P}(i, \mathrm{D}(i)),$$

$$E_s(\mathrm{D}) = \sum_{i \in \mathcal{I}, j \in \mathcal{I}, i \neq j} \mathrm{C}(i, j, \mathrm{I}, \mathrm{L}, \mathrm{D}),$$

$$\mathrm{C}(i, j, \mathrm{I}, \mathrm{L}, \mathrm{D}) = \rho_c(\mathrm{D}(i), \mathrm{D}(j)) \times \exp\left(-\frac{\|\mathrm{I}(i) - \mathrm{I}(j)\|^2}{\theta_c} - \frac{\|L(i) - L(j)\|^2}{\theta_p}\right),$$



(a) Reference View

(b) WTA

(c) Long Range Connection

(d) Less first-order Smoothness

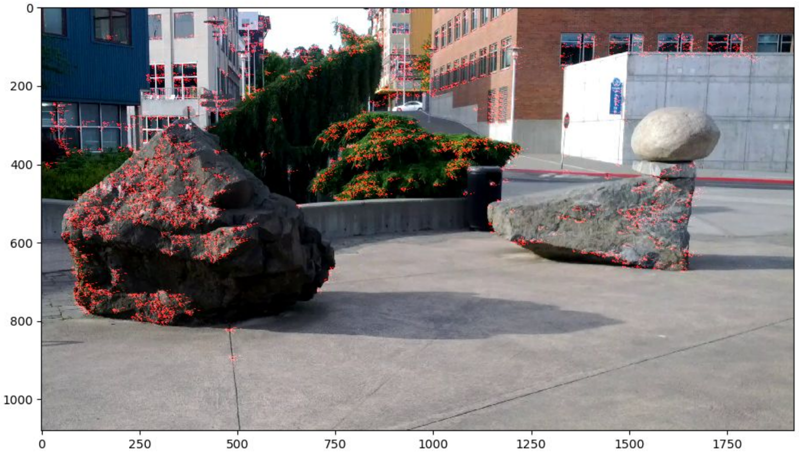
(e) First-order Smoothness

(f) More first-order Smoothness

Results

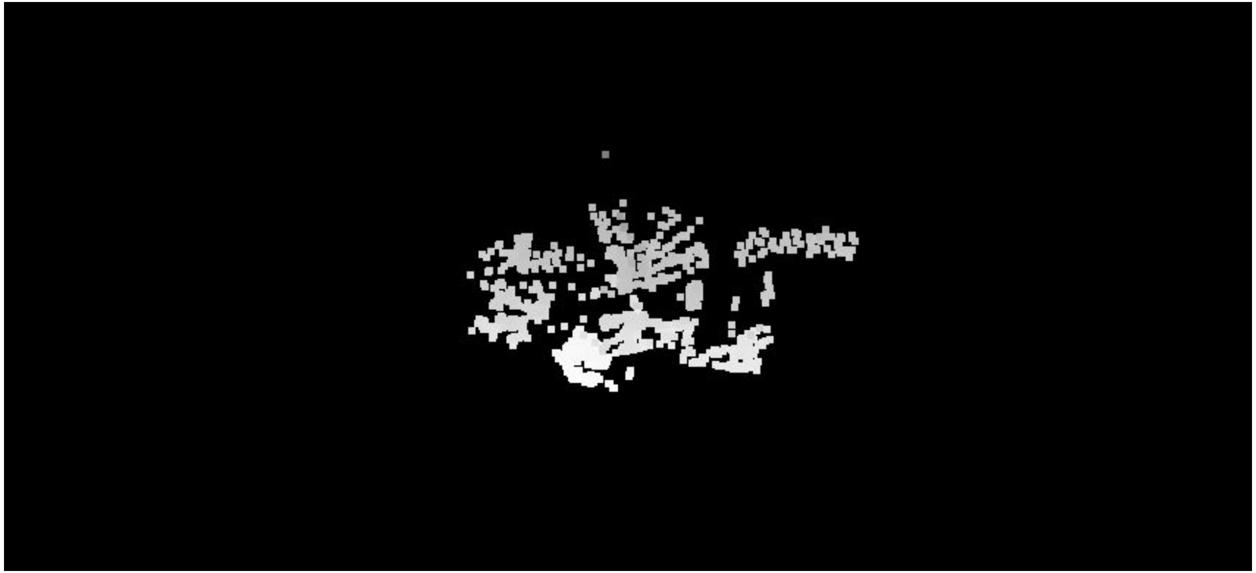
Results

Optical flow



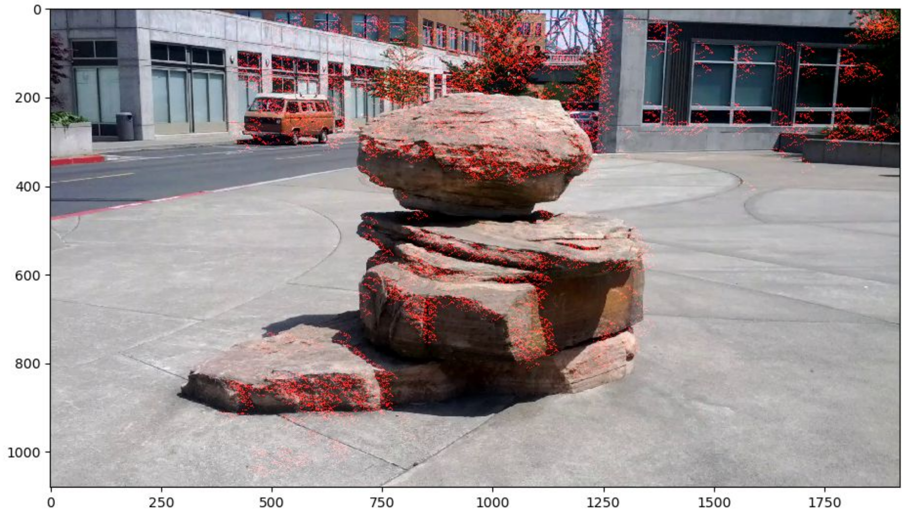
Point cloud

Depth map obtained



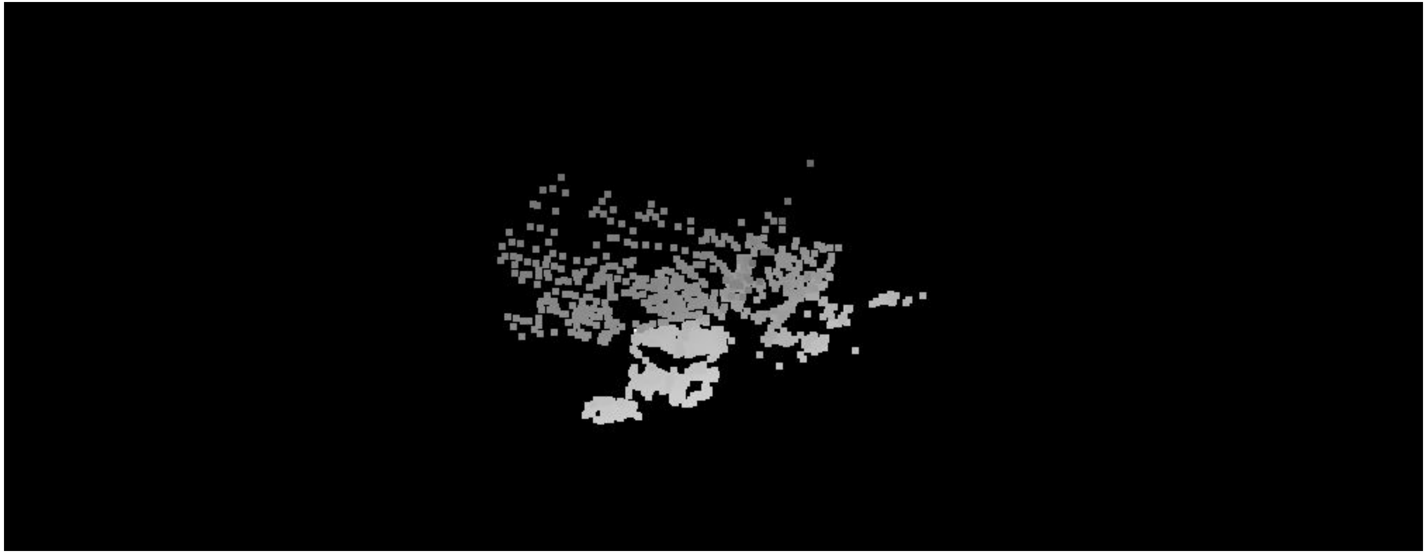
Results

Optical flow



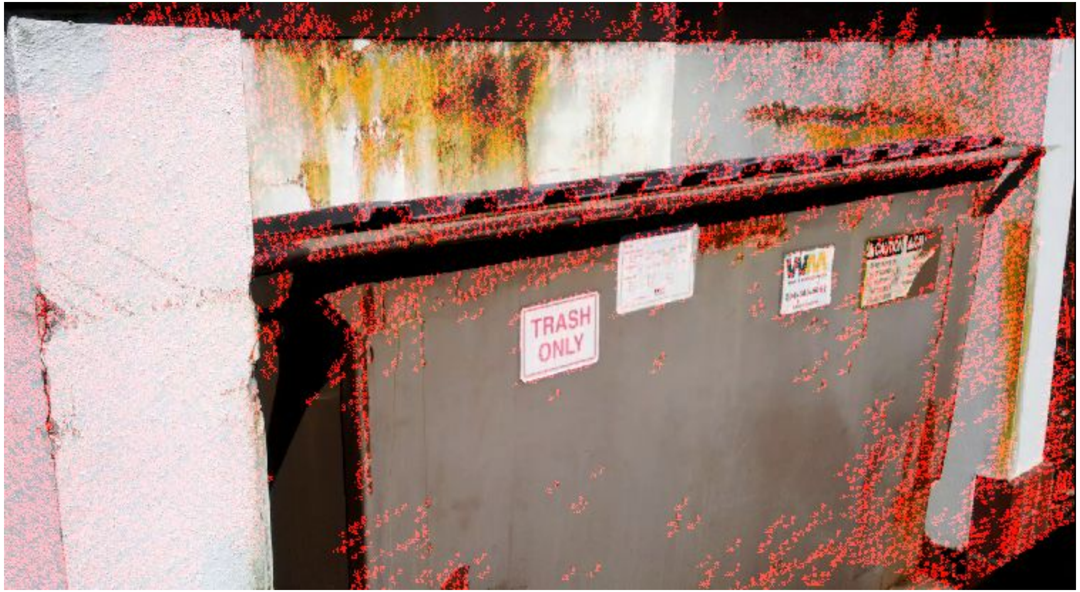
Point cloud

Depth map obtained

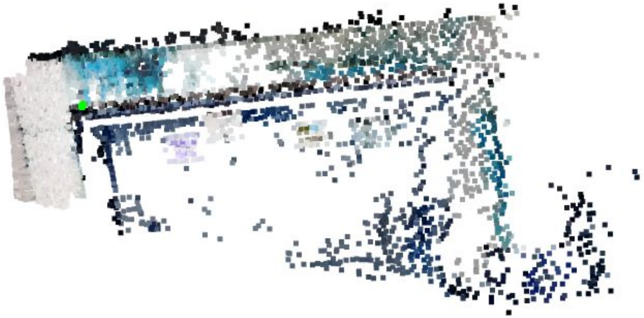


Results

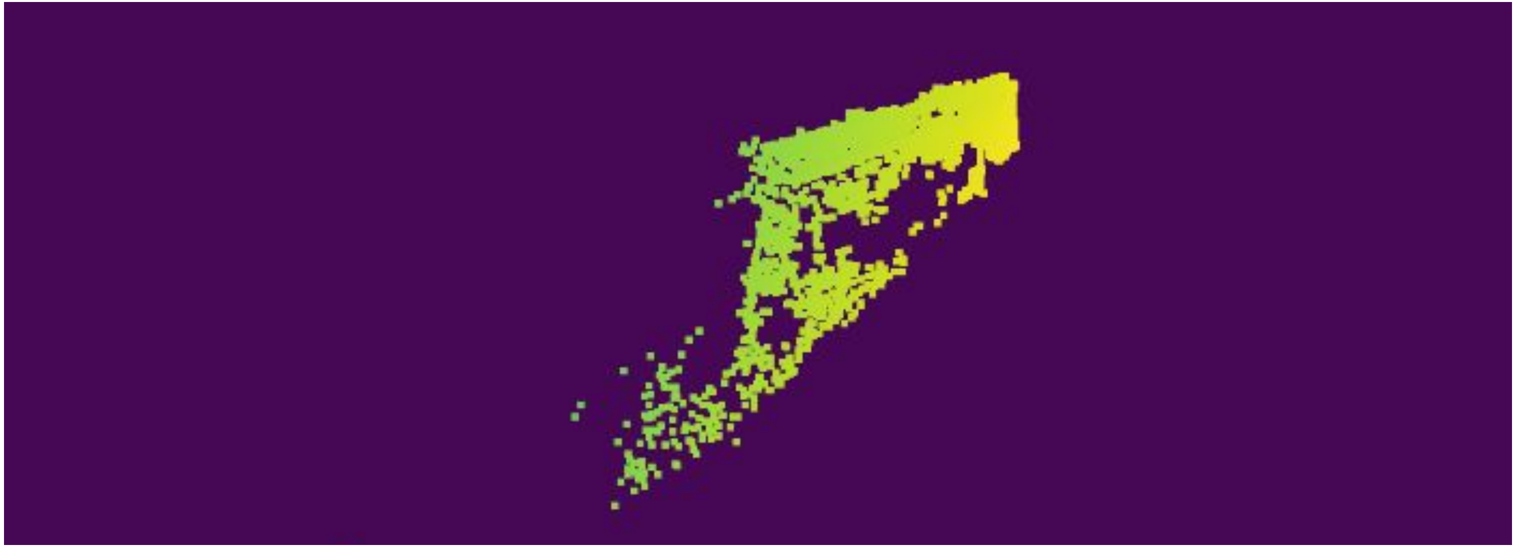
Optical flow



Point cloud

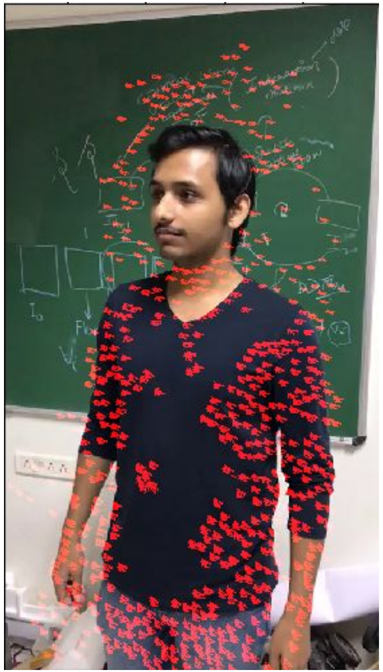


Depth map obtained

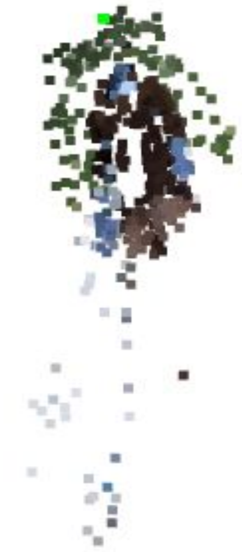


Results obtained on our photos

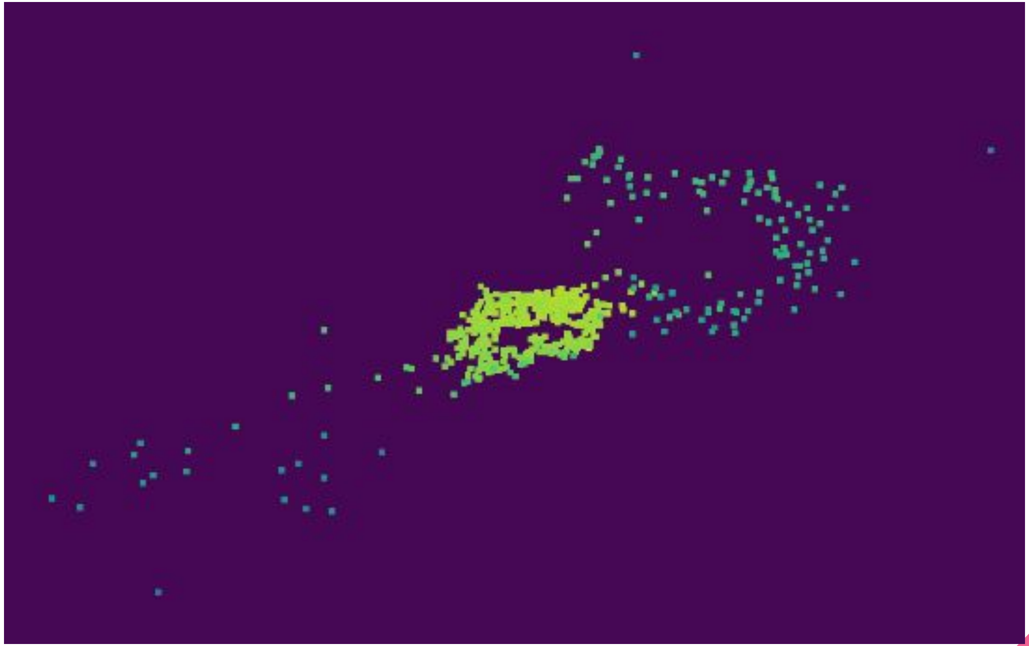
Optical flow



Point cloud

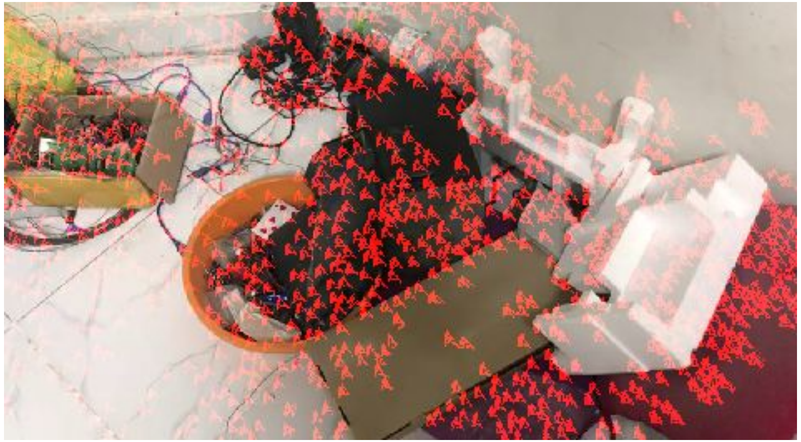


Depth map obtained (colorized)

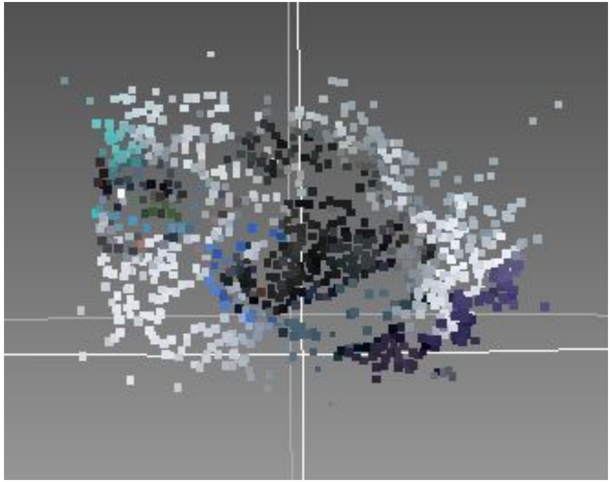


Results

Optical flow



Point cloud



End Of Presentation