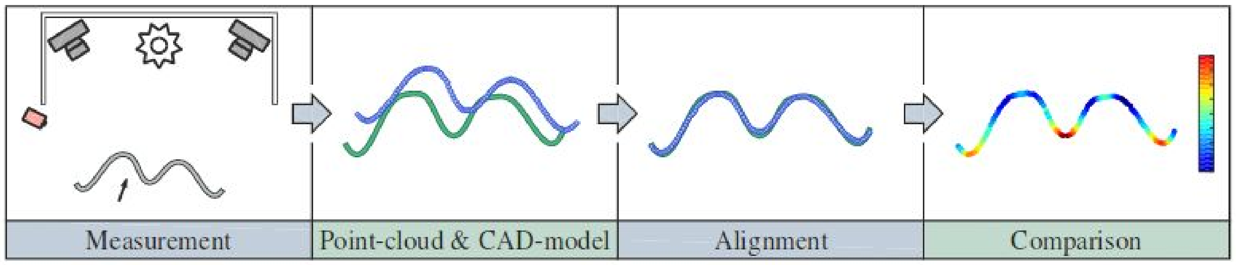
BTP-2 Mid-Term Report

Vision Aided 3D Inspection

Shwetha Krishnamurthy (14ME10072)

Introduction

During the course of last semester, I had worked on object detection in two dimensions using SIFT (Scale Invariant Feature Transform) algorithm. This proved to be helpful in the case of 2D feature matching and comparison. As a continuation, I have tried to extend the feature matching to three dimensions, without the use of a depth-sensor, by the use Structure From Motion, which is essentially an algorithm to construct the point cloud of the object to be inspected on the conveyor belt from a video feed, hence eliminating the use of a dedicated inspection station (no stoppage for inspection in the assembly line). Once the point cloud was created, Poisson's Surface Reconstruction was used to create a mesh of the object. In order to perform the comparison with the CAD model to inspect the fitness of the produced good, I have used Iterative Closest Point Registration technique for alignment of the CAD model with the generated point cloud, and hence generation of a rotation and translation matrices, along with the fitness factor, which is just the average vector difference between two corresponding points on the point cloud and the template.

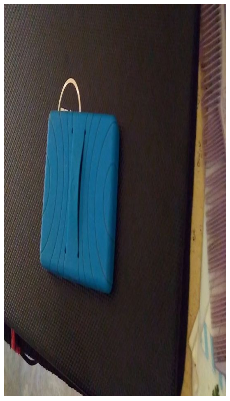
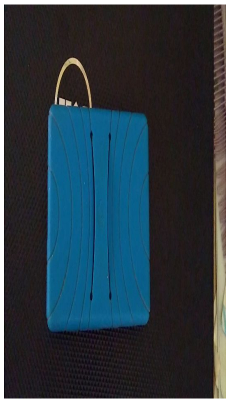


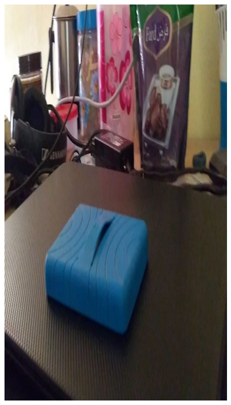
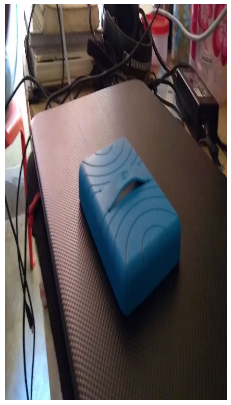
Basic workflow is shown in the above picture.

Process Summary

The steps for the vision aided inspection are:

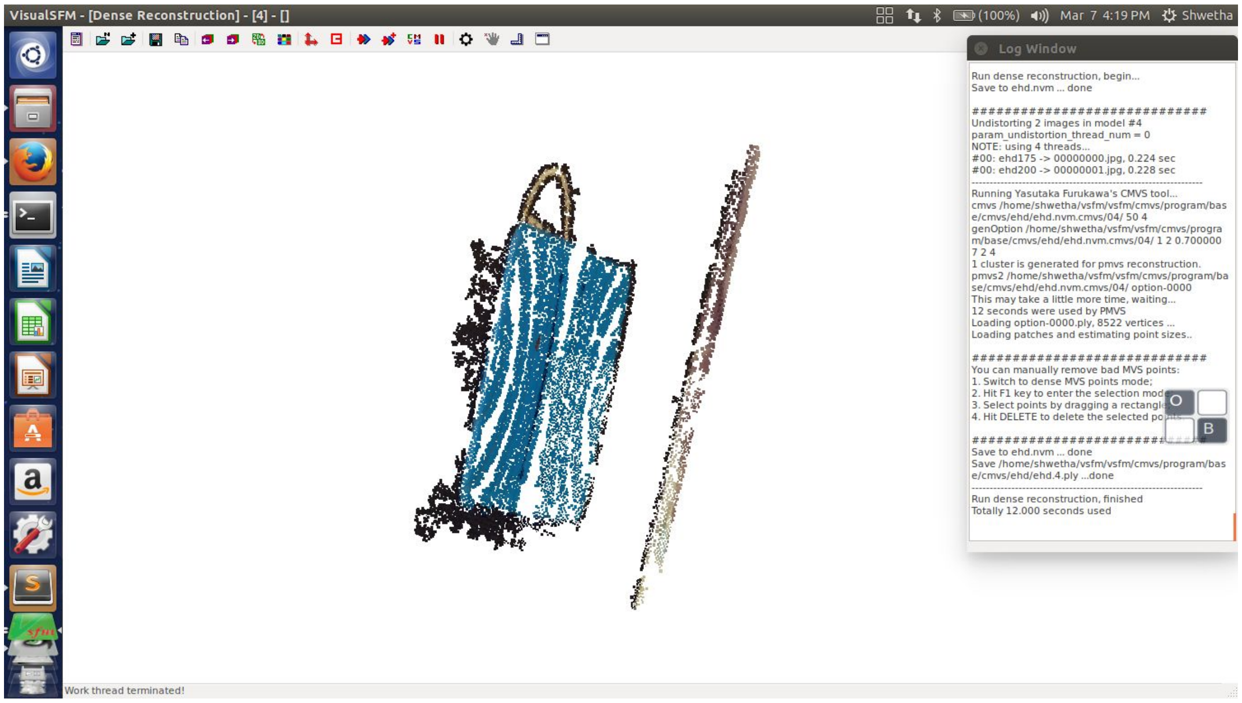
1. **Capturing Video:**
   The first step to the proposed process is capturing of the video. For Structure From Motion algorithm to work perfectly, we need snapshots of the object from various different viewpoints. For this purpose, I have exploited the motion of the conveyor belt to capture snaps from the video filmed of the object moving on the conveyor belt, the camera being held stationary. But, in order to test the Structure From Motion algorithm, I made a video of an External Hard Disk in my room itself, while keeping the object stationary, and moving the camera (my phone camera) and capturing a video from all sides. Following are the images captured from the video.

2. **Applying Structure From Motion:**
   Once the video is captured, I wrote a code in C++ (link to the code in my GitHub) to get the snaps. I have considered every 25th shot as an estimate to cover maximum viewpoints. Next, I did some research on how SFM can be implemented, and it turns out we have a well made software called Visual SFM by Changchang Wu, which implements SIFT parallely on a GPU for pairwise feature matching for sparse reconstruction, followed by CMVS by Yasutaka Furukawa to get dense reconstruction point cloud from the sparse reconstruction (links to the chapter; CS231a report). This process took a total of eleven minutes on my machine (Intel i3

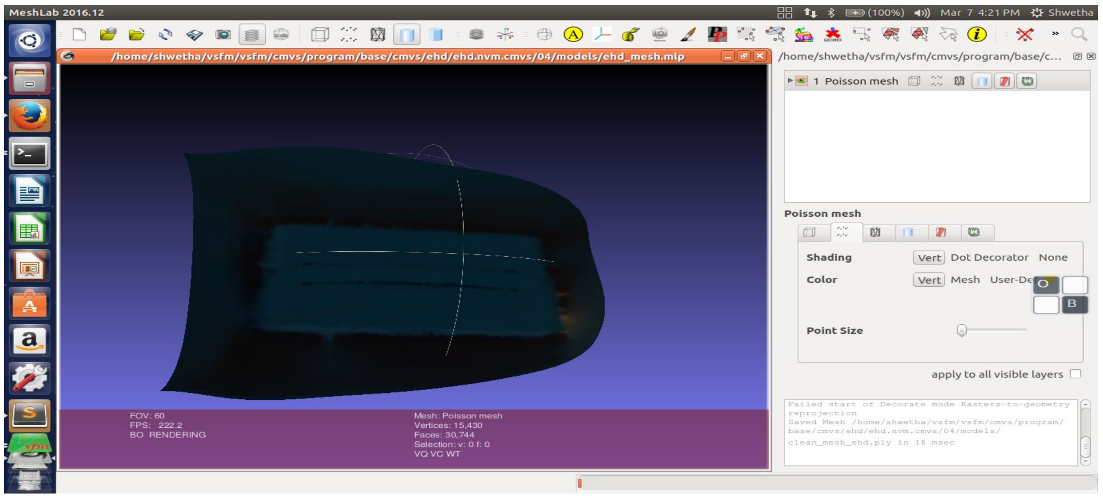
core, 4 microprocessors, no GPU), which I feel will get significantly reduced if run on a more powerful GPU enabled machine. The SFM algorithm and CMVS are explained in detail in Annexure 2. The dense point cloud obtained from VSFM software is shown below.



```
Run dense reconstruction, begin...
Save to ehd.nvm ... done

################################
Undistorting 2 images in model #4
param_undistortion_thread_num = 0
NOTE: using 4 threads...
#00: ehd175 -> 00000000.jpg, 0.224 sec
#00: ehd200 -> 00000001.jpg, 0.228 sec
----------------------------------------------------------------
Running Yasutaka Furukawa's CMVS tool...
cmvs /home/shwetha/vsfm/vsfm/cmvs/program/bas
e/cmvs/ehd/ehd.nvm.cmvs/04/ 50 4
genOption /home/shwetha/vsfm/vsfm/cmvs/progra
m/base/cmvs/ehd/ehd.nvm.cmvs/04/ 1 2 0.700000
7 2 4
1 cluster is generated for pmvs reconstruction.
pmvs2 /home/shwetha/vsfm/vsfm/cmvs/program/ba
se/cmvs/ehd/ehd.nvm.cmvs/04/ option-0000
This may take a little more time, waiting...
12 seconds were used by PMVS
Loading option-0000.ply, 8522 vertices ...
Loading patches and estimating point sizes..

################################
You can manually remove bad MVS points:
1. Switch to dense MVS points mode;
2. Hit F1 key to enter the selection mod
3. Select points by dragging a rectangl
4. Hit DELETE to delete the selected po

################################
Save to ehd.nvm ... done
Save /home/shwetha/vsfm/vsfm/cmvs/program/bas
e/cmvs/ehd/ehd.4.ply ...done
----------------------------------------------------------------
Run dense reconstruction, finished
Totally 12.000 seconds used
```

3. **Applying Poisson's Surface Reconstruction:**

This step is essential to create a mesh of the dense point cloud formed in the previous step. For this purpose, I have used [Point Cloud Library]() and applied the algorithm to obtain the mesh. This step can be alternatively performed in Meshlab,


MeshLab 2016.12
(100%)
Mar 7 4:21 PM
Shwetha
/home/shwetha/vsfm/vsfm/cmvs/program/base/cmvs/ehd/ehd.nvm.cmvs/04/models/ehd_mesh.mlp
/home/shwetha/vsfm/vsfm/cmvs/program/base/c...
1 Poisson mesh
Poisson mesh
Shading
Vert
Dot Decorator
None
Color
Vert
Mesh
User-De
Point Size
apply to all visible layers

```
FOV: 60
FPS:   222.2
BO  RENDERING

Mesh: Poisson mesh
Vertices: 15,430
Faces: 30,744
Selection: v: 0 f: 0
VQ VC WT

Failed start of Decorate mode Rasters-to-geometry
reprojection
Saved Mesh /home/shwetha/vsfm/vsfm/cmvs/program/
base/cmvs/ehd/ehd.nvm.cmvs/04/models/
clean_mesh_ehd.ply in 18 msec
```

where the color data of the mesh is preserved, which is not possible in PCL. The algorithm is detailed in Appendix 3 (link to paper). The mesh obtained from Meshlab, with the color information retained is shown in the screenshot.

4. **Applying Iterative Closest Point Registration:**
   Next step is to align the generated mesh with the point cloud of the CAD model. For testing purposes, I have used a point cloud data from the Internet , i.e a point cloud of the template and the object. Using ICP registration technique (link to paper) I generated the rotation and the translation matrix, and also got a fitness score.

Work To Be Done

Integrating the steps in one C++ Program

Next, I need to use Socket Programming to access the VisualSFM software from a C++ program and use it in a programmable fashion.

Point Cloud from a CAD Model

For comparison step, I need to convert the CAD file into one of the PCL compatible files.

Testing

The accuracy of the point cloud generation needs to be improved with more advanced optimisation techniques.