

## CS-532 Final Project List

What should you turn in: A single zip file containing: 1) a pdf file with all results and a discussion of your solution 2) all your resulting images files and code. In addition to the above you should prepare a short (~5min) presentation describing your work, which you may be required to present at some point.

### **Project 1) Multi-view Plane Sweep**

The goal of the assignment is to implement the multi-view plane sweep algorithm for depthmap generation. To this end you shall download the datasets fountain-P11, Herz-Jesu-P8, entry-P10 from the Strecha MVS evaluation website

<https://icwww.epfl.ch/~marquez/multiview/denseMVS.html>

and use the provided camera pose and calibration information provided for each image

Fountain-P11



Herz-Jesu-P8



entry-P10

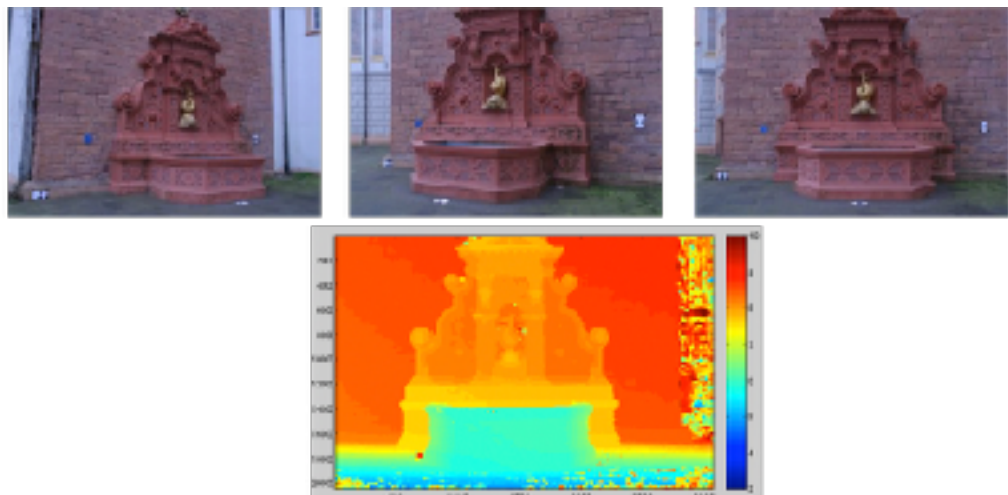


- a) Select three images from each of the datasets/scenes and generate for each a depth map by sweeping a plane orthogonal to the each cameras optical axis. Show the resulting depth maps.

- b) Report the accuracy of the generated depth maps compared to the available ground truth for the fountain dataset from Homework 1 data
  1. Report the average pixel error for each of the depth map
  2. Generate an error map (an image where the magnitude of the estimation error is stored at the pixel position) using Matlab's "jet" colormap for visualization
  3. Plot the cumulative error distribution for each depth map
- c) Recompute the depthmaps from a) using multiple sweeping directions. You will need to propose and justify in the report your selection of sweeping direction
- d) Evaluate the accuracy of the depthmaps generated in c) in the same way as specified in b)

Present in your report the attained depthmaps and accuracy reports, explain the plane induced homography used for planesweeping and discuss your choices for photo- consistency measure, depth sampling strategy and multiview cost aggregation

Fountain-P11



**Bonus:** Use multiple photo consistency measures and compare performance

NOTES:

- The 3x4 projections matrices for each image are found in the file ####.png.P
- The matrices/vectors for rotation, translation and intrinsic parameters for each image are found in the file ####.png.camera
- The P matrix can be composed as follows (in Matlab code):  $P = K * [R' \quad -R' * t']$

## Project 2) Image Overlay Augmented Reality

The goal of this project is for you to estimate pairwise camera motion of each successive frame of a video sequence and to overlay the rendering of a simple 3D model onto each of the frames as commonly done in AR applications. The main challenge will be implementing camera motion estimation based on descriptions found in the computer vision research literature.

**Development:** You are expected to integrate a system that:

- 1) Detects Harris corner as 2D features
- 2) Matching should be implemented with photo consistency measures utilized in class
- 3) Assuming a flat supporting surface, estimates an homography between successive frames
- 4) Estimates 3D camera motion in the form of a [ Rotation | Translation] matrix



- 5) Determines the rendering of your 3D model from the estimated camera position
- 6) Overlays the rendering into the image frame for which the motion was estimated.

**Data Capture:** A multi-image dataset observing a flat surface (similar to the image pair one included below) will be uploaded to Canvas and will be the one used for your submitted results. You are welcome to use the ground truth camera motion and images of any of the Strecha Datasets having dominant flat surfaces for testing/development purposes.

**Rendered 3D Model:** The edge outline of a simple tetrahedron or cube will be sufficient for rendering. The vertices of the bottom face of the Tetrahedron (or cube) should be aligned with the estimated plane. Position the 3D model so that the principal axis of the FIRST camera in the sequence, passes through the model's centroid. Scale the 3D model accordingly.

**Estimating Motion From Homography:** Implement one of the methods described in [1] Malis, Manuel Vargas. Deeper understanding of the homography decomposition for vision- based control. [Research Report] RR-6303, INRIA. 2007, pp.90 which can be downloaded at <https://hal.inria.fr/inria-00174036v3/document>

There is no provided internal calibration matrices for the images. Hence, you will need to specify a reasonable set of parameters and explain your choice in the report.

Notes:

- 1) You may use third-party feature extraction and matching for partial credit (e.g. VLFEAT <http://www.vlfeat.org> ). As a last resort, you may commit this step and use manual feature selection, at the expense of deduction on your grade.
- 2) When doing feature matching it may be helpful to consider multiple putative matches
- 3) Pose estimation should be performed in a robust manner (i.e RANSAC)
- 4)

**Bonus:** If you test/develop on the Strecha dataset, be sure to include accuracy benchmarks of camera motion estimation (vs. ground truth) in your final report.

**Bonus:** Compare the accuracy of the SVD vs analytic motion-from-homography solutions in presented in [1] w.r.t. the Strecha ground truth datasets.

**Output:** A video of the original image frames with the overlaid 3D model.

### **Project 3) Multi-view PatchMatch**

The goal of the assignment is to implement the multi-view patch algorithm for depthmap generation. To this end you shall download the datasets fountain-P11, Herz-Jesu-P8, entry-P10 from the Strecha MVS evaluation website

<https://icwww.epfl.ch/~marquez/multiview/denseMVS.html>

and use the provided camera pose and calibration information provided for each image.

Use the Patchmatch sample and propagation scheme alternating among the four image directions (left-to-right, top-to-bottom, right-to-left, bottom to top) and report the progress after each propagation direction.

- a) Select three images from each of the datasets/scenes and generate for each a depth map  
Show the resulting depth maps after each iteration.
- b) Report the accuracy of each generated depth map compared to the available ground truth, by
  1. Report the average pixel error for each of the depth map
  2. Generate an error map (an image where the magnitude of the estimation error is stored at the pixel position) using Matlab's "jet" colormap for visualization
  3. Plot the cumulative error distribution for each depth map

Notes:

- Start from a random depth initialization for each pixel in the depth map

- Use any window size photo-consistency measure you deem adequate. Justify your design choice

**Bonus:** Use multiple photo-consistency measures and compare performance