List of projects for the course Computer vision and 3D Graphics 2016/2017

Simone Milani

A.A. 2016/2017

Description

Here's the list of the projects for the course Computer vision and 3D Graphics 2015/2016 A.A. 2014/2015.

The output of each project's work must be a short report (at least 4-5 pages without considering the source code or MATLAB script), which will be presented to the teacher during the oral exam.

The report must be sent to the teacher at least 1 day before the oral exam.

Project 1

Overview the camera calibration pipeline for a fish-eye lens.

Given the dataset of images at `http://www.dei.unipd.it/~sim1mil/materiale/CVprojects/proj1.zip`, compute calibration parameters using the toolbox
`https://sites.google.com/site/scarabotix/ocamcalib-toolbox`

Implement in MATLAB an immersive vision system using the obtained parameters and the available pictures.

Programming language: MATLAB.

Project 2

Build a 3D viewer of 3D models using Google Cardboard and a cellular phone. Assuming that each eye is a camera watching 3D reality, generate an Android/iOS application (it depends on your device) which display the 3D model to the viewer.

Programming language: Android/iOS or MATLAB/C/C++ (if simulator is used in place of phone).

Project 3

Overview different methods for point cloud registration. Implement Horn and ICP strategies ; then compare them in terms of computational complexity and accuracy.

Dataset: you can use the images (RGB + depth) used for lab. experience 4.

Programming language: MATLAB or C++.

Project 4

Camera calibration and localization for sport videos. Starting from the work

Dirk Farin, Susanne Krabbe, Wolfgang Effelsberg, Peter H. N. de With, "Robust Camera Calibration for Sport Videos using Court Models", SPIE Storage and Retrieval Methods and Applications for Multimedia, vol. 5307 p. 80-91, January 2004, San Jose (CA), USA pdf

You can check some references on the web page
`http://www.dirk-farin.net/projects/sportsanalysis/index.html`.

Perform the analysis on frames taken from a Rugby match. Dataset is available at `http://www.dei.unipd.it/~sim1mil/CVcourse/RBSvideo.zip`

Programming language: MATLAB or C++.

Project 5

Fast SIFT descriptors matching. Starting from the paper

Marius Muja and David G. Lowe. 2012. Fast Matching of Binary Features. In Proceedings of the 2012 Ninth Conference on Computer and Robot Vision (CRV '12). IEEE Computer Society, Washington, DC, USA, 404-410. pdf

implement the described FAST SIFT matching strategy using some of the datasets SIFT software is available at
`http://www.cs.ubc.ca/~lowe/keypoints/`

Datasets to be used are available at `http://www.robots.ox.ac.uk/~vgg/data/data-aff.html`

Programming language: MATLAB or C++.

Project 6

Augmented reality. Starting from the homography computation strategy presented in the course, implement an augmented reality application that takes an input image, looks for a specific planar pattern (marker in it), and replaces it with a 3D model. Some models are available at
`http://graphics.stanford.edu/data/3Dscanrep/`

Programming language: MATLAB or C++ or Android.

Project 7

Given a set of point clouds obtained from Kinect v.2 3D camera, implement a statistical outlier removal filter that takes into account color information as well. Some documentation about SOR is available at `http://pointclouds.org/documentation/tutorials/statistical_outlier.php#statistical-outlier-removal`. Compare its performance with the standard SOR.

Dataset: you can use the images (RGB + depth) used for lab. experience 4.

Programming language: MATLAB or C++.

Project 8

Plane segmentation using RGBD data. Implement a segmentation strategy fro RGBD signals that clusters pixels according to their normals. References can be found on paper

Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. 2012. Real-time plane segmentation using RGB-D cameras. In Robot Soccer World Cup XV, Thomas Rfer, Norbert Michael Mayer, Jesus Savage, and Ulu Saranl? (Eds.). Springer-Verlag, Berlin, Heidelberg 306-317. pdf

Programming language: MATLAB or C++.

Project 9

Given a set of 3D points and a pair of camera projection matrices:

1. render the 3D scene to the two 2D images;
2. undistort and rectify images if required;
3. compute stereo disparity and estimate 3D model.

Programming language: MATLAB or C++.

Project 10

Stereo disparity computation from projected random dot pattern. Given a couple of stereo images from Aquifi camera, compute disparity using a multi-layer hierarchical approach, i.e., computing the disparity on a reduced resolution version of the two images and re-using the results of this computation at the higher resolution layer.

Use up to three decomposition layers and measure the complexity reduction and the accuracy obtained.

Dataset is available at `http://www.dei.unipd.it/~sim1mil/CVcourse/imagesStereo.zip`

Programming language: MATLAB or C++.

Project 11

Performances of SIFT descriptors matching. Compare different descriptors and matching performances. Consider

- SIFT
- SURF
- BRIEF

and the image sets available at `http://www.robots.ox.ac.uk/~vgg/data/data-aff.html`.
Compare the performance in terms of matching percentages, computational complexity, and accuracy in the estimation of the homography.

Programming language: MATLAB or C++.

Project 12

Object tracking. Track the x,y,z coordinates and the orientation of a known object (geometry is known) given a single view video.

You will be provided with 3D model and images to match.

Programming language: MATLAB or C++.

Project 13

Stereo disparity computation from projected random dot pattern. Given a couple of stereo images from Aquifi camera, compute disparity and reconstruct the 3D point cloud.

Try also adapting the size of the block to the local characteristics of the image, i.e., using a vertical or horizontal block depending on the local edge information (extracted from the side RGB camera).

Dataset is available at `http://www.dei.unipd.it/~sim1mil/CVcourse/imagesStereo.zip`

Programming language: MATLAB or C++.

Project 14

Compression of 3D models for augmented reality application. Given $256 \times 256 \times 256$ voxel 3D model, implement an oct-tree coding strategy to code its geometry. For more information see

Meagher D (1982) Geometric modeling using octree encoding. Computer Graphics and Image Processing 19(2):129?147

Programming language: MATLAB or C++.

Project 15

3D modeling of real scene from collections of unordered pictures.

Using the VisualSfM [A] or the Bundler software [B], test different image inclusion orderings to generate the final 3D model. Possible strategies:

- decreasing number of matching points [B];
- hierarchical clustering [C].

[A] VisualSFM software (available at `http://ccwu.me/vsfm/`)
[B] Bundler (available at `http://www.cs.cornell.edu/~snavely/bundler/`)
[C] SAMANTHA (code and documentation available at `http://www.diegm.uniud.it/fusiello/demo/samantha/`)

Programming language: MATLAB or C++.