

Project acronym: POPART	Title: Initial 3D Visual Database library
Project number: 644874	Work Package: WP1
Work package: 3D Visual Database	Version: 1
Deliverable number and name:	Date: 01/04/2015
D1.1: Initial 3D Visual Database library	Author: Simone Gasparini
Type: <input type="checkbox"/> Report <input type="checkbox"/> Demonstrator, pilot, prototype <input type="checkbox"/> Website, patent filings, videos, etc. <input checked="" type="checkbox"/> Other: source code	Co-Author(s): To: Albert Gauthier, Project Officer
Status: <input type="checkbox"/> Draft <input type="checkbox"/> To be reviewed <input type="checkbox"/> Proposal <input checked="" type="checkbox"/> Final / Released to EC	Confidentiality: <input checked="" type="checkbox"/> PU – Public <input type="checkbox"/> CO – Confidential <input type="checkbox"/> CL – Classified
Revision: Final	
Contents: D1.1: Initial 3D Visual Database library.	

Deliverable description

The goal of this deliverable is to provide some preliminary tools to create and manage a 3D visual database. In this implementation, the visual database is created in an off-line step from a collection of views of the scene. These views are processed by a structure-from-motion pipeline (OpenMVG) in order to calibrate the cameras and recover the relative poses of the cameras and the 3D reconstruction of the scene in terms of points cloud.

This preliminary version of the database is composed of a tree structure, namely a vocabulary tree as described in the seminal work of [Nistér 2006]. This approach is normally used in image retrieval tasks, in order to find the images in a dataset that are visually similar to a given query image. In the case of image localization, the query image is the current camera frame and the goal is to find the visually closest view among the set of views.

From the structure-from-motion pipeline all the SIFT features from the set of views are collected, and an hierarchical k-means clusterization process is run over this SIFT feature set. For each new image to query, the relevant SIFT features are extracted and propagated down the obtained tree structure: a IDF-TF mechanism allows to associate each feature to a score, the sum of the scores allows to recover the most similar image(s). Matching the recovered image with the closest view allows to recover the 2D-3D point association and hence the pose of the camera.

Code description

In this preliminary version of the 3D visual database a set of libraries have been developed in order to provide an interface for the vocabulary creation and management (load/save from/to file), as set of computer vision functions for camera pose estimation and other accessory libraries to manage data structure and their conversions. Some preliminary shell scripts are also provided to create the database and test the image retrieval algorithm and the pose estimation.

The code has some dependencies from some common external libraries, in particular

- `glew` ≥ 1.6 (`libglew1.6-dev`)
- `OpenGL`, `Glut`
- `Boost 1.55` (`[core, thread, system, math, filesystem, serialization, thread, exception, unit_test_framework]-dev`)
- `GSL` ≥ 0.16 (`libgsl1.6-dev`)
- `Eigen3` (`libeigen3-dev`)
- `Lapack 3.5`

Also at the moment of writing the code depends on this other open source libraries

- `OpenCV 2.4.9` for image, matrix data structures and computer vision algorithms
- `SSBA 2.0` for the bundle adjustment.

As part of the further development of the library we are planning to use `Ceres-solver` for the bundle adjustment part.

All the instructions for compiling the code, retrieve the dependencies are explained in the BUILD file in the root directory of the repository

Code organization

The code is organized in two main directory:

- the directory `library` contains all the libraries that constitutes the core of the project, in particular for managing the vocabulary tree, the computer vision algorithms for pose estimation and the unity tests.
- The directory `application` contains the set of scripts that allows the creation of the vocabulary tree, its management, and the test of the algorithm for image localization based on image query.

The code is available at:

<https://github.com/poparteu/cameraLocalization>