Projet long : POPART
Specifications

Clients :

Simone Gasparini, VORTEX team, IRIT-ENSEEIHT : simone.gasparini@enseeiht.fr
Fabien Castan, Mikros Image, Paris : faca@mikrosimage.eu

Project name :

Projet Long POPART

Project team :

Matthias Benkort, ENSEEIHT : matthias.benkort@etu.enseeiht.fr
Nicolas Gaborit, ENSEEIHT : nicolas.gaborit@etu.enseeiht.fr
Arthur Manoha, ENSEEIHT : arthur.manoha@etu.enseeiht.fr
Matthieu Pizenberg, ENSEEIHT : matthieu.pizenberg@etu.enseeiht.fr
Louis Viot, ENSEEIHT : louis.viot@etu.enseeiht.fr

Project manager :

Arthur Manoha, ENSEEIHT : arthur.manoha@etu.enseeiht.fr

Table of contents :

1 Project outline

This project aims at developing a tool to ease the task of collecting a dataset of pictures of a scene for 3D reconstruction. The project is casted in the context of the European Project POPART, in which the team VORTEX and Mikros Image are involved. POPART aims at developing tools for the real-time pre-visualization of special effects during the shooting of a movie. In such context, the preliminary task is to collect a dataset of images of the scene where the movie will take place and perform an accurate 3D reconstruction of the scene, that will later be used for real-time camera localization purpose.

2 Main objectives

The objective of the project is to develop a graphical tool covering the whole process of on-set 3D reconstruction of a scene. Thanks to a GUI (Graphical User Interface) the tool allows the user to :

- Automatically fetch photos being taken by a camera
- Select photos as input data for the 3D reconstruction algorithm
- Visualize the progress and result of the 3D reconstruction
- Visualize the distribution of camera viewpoints

3 Users

Simone Gasparini and Fabien Castan will be the users of the application, along with people from the cinema industry.

4 Functional description

4.1 Fetch and manage input pictures

1. The application must be able to retrieve pictures from a connected camera.
2. The camera can be connected by USB or Wi-Fi.
3. Pictures can be retrieved from a local folder too.
4. The application must be able to listen for incoming pictures and download them from the camera.
5. All the retrieved pictures should be stored in a user-accessible folder.
6. The images used for reconstruction can be selected from a thumbnail list, on a viewpoint map, or in a 3D representation of these viewpoints.

4.2 Run a 3D reconstruction algorithm

1. A 3D reconstruction algorithm can be run for a specified set of pictures.
2. Given a set of pictures in a specific folder corresponding to a scene, an algorithm doing the 3D reconstruction of the scene already exists in OpenMVG.
3. The algorithm can be started automatically when new pictures are available.
4. The application must show a progress status of the algorithm, indicating which pictures have been taken into account and which ones have been rejected.
5. The reconstruction algorithm should present at least two configurations : a fast configuration (parameters are set for a fast but not so precise algorithm) and a best configuration (slower but more accurate).

4.3 Access and visualize the results of the 3D reconstruction

1. The reconstruction will generate these output files :
    a. A .ply file describing a colored point cloud that represents the modeled scene.
    b. A file listing the location and orientation of the camera for each of the input pictures.
    c. Some intermediate files describing the point cloud (without colors) that represent the scene at different steps of the reconstruction.
2. The resulting files of the reconstruction must be accessible in a specific output folder.
3. The application must be able to render the generated 3D point cloud.
4. The list of camera viewpoints should be visible as 2D markers on a geographic map.

4.4 Graphical User Interface

These are the general guidelines for the final Graphical User Interface (GUI). A more concrete description will be provided by the graphical mock up and the GUI specification document.

1. The GUI must allow to manage the input pictures (fetching and selection).

2. The GUI must allow to run and monitor the reconstruction algorithm.
3. The rendering of the output 3D cloud point should be displayed within the GUI.
4. The GUI must permit to rotate and zoom on the 3D model being rendered.
5. The map containing the camera locations should be displayed within the GUI.

5 Other constraints

5.1 Libraries

1. The reconstruction algorithm must use the OpenMVG library.
2. Interaction with the camera should be made with the gPhoto library.
3. The GUI must use the Qt library.

5.2 Language

The whole project needs to be done in English. Therefore, the documentation and the code (variable names, comments, etc.) must be written in English.

5.3 Programming Language

The application should be developed in Python3 as long as there is no compatibility problem. C++ can be used for code dealing with 3D rendering and camera access, or other code which is not likely to change over time.

5.4 Open source

The source code must be open source and available on GitHub.

5.5 Performances

There is no constraint concerning performances in time or memory for the reconstruction algorithm.

5.6 Platforms

The application must run on Linux.

5.7 Documentation

The application will come with a short user manual and an architecture documentation.

5.8 Ground Truth data

The client will provide datasets allowing us to verify our program (for example images of a scene along with 3D reconstruction of the scene).

6 Optional functionalities

6.1 Incremental algorithm

It consists in adding new functionalities to the OpenMVG library. The algorithm should allow to add incrementally new photos to the reconstruction algorithm. Currently, when images are added, the algorithm needs to start over with all images.

6.2 Feedback to the user to improve the quality of the reconstruction

That improvement consists in being able to analyze the reconstruction and automatically suggest to the user areas of the scene where there is a need of new images.

7 Planning

The project should be completed on Friday 13th March 2015.

Date	*Client*	*Project Manager*