Document for Linear SFM C/C++ Source Code

Liang Zhao
Hamlyn Centre for Robotic Surgery
Department of Computing, Faculty of Engineering
Imperial College London, United Kingdom

Shoudong Huang and Gamini Dissanayake
Centre for Autonomous Systems
Faculty of Engineering and Information Technology
University of Technology Sydney, Australia

liang.zhao@imperial.ac.uk
March 28, 2015

Abstract

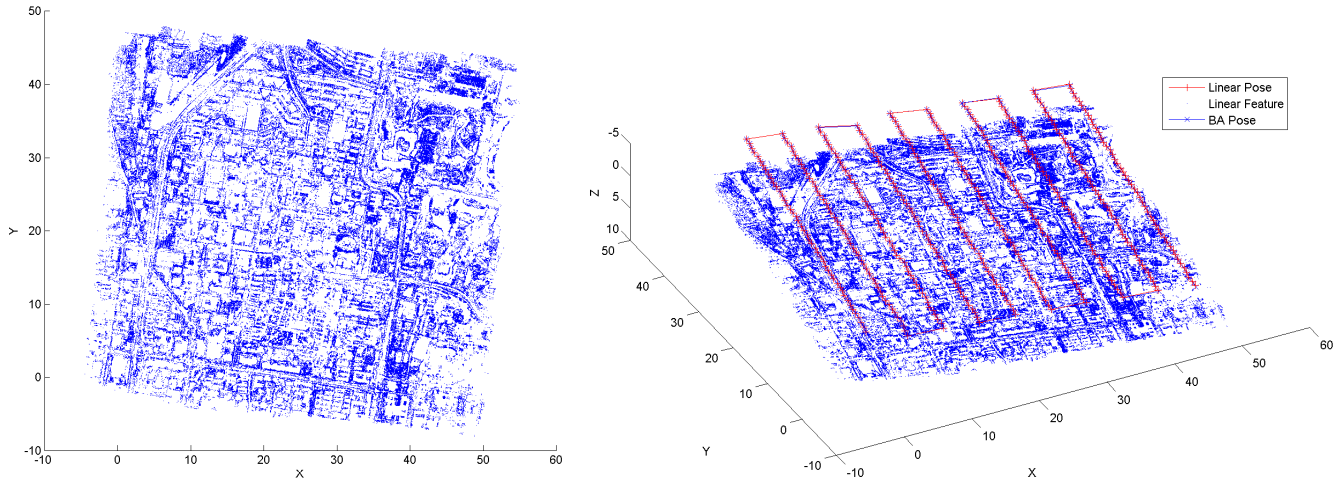
This document provides some details about how to use the C/C++ source code of Linear SFM algorithm to solve structure-from-motion (SFM) problems for ordered monocular and stereo image sequences. The code is written by Liang Zhao (liang.zhao@imperial.ac.uk), Shoudong Huang (Shoudong.Huang@uts.edu.au) and Gamini.Dissanayake (Gamini.Dissanayake@uts.edu.au). The code with some experimental datasets are available on GitHub. Please contact Liang Zhao if you have any questions/comments about the code.

I. Introduction

Linear SFM is a novel hierarchical approach to solving structure-from-motion (SFM) problems. The algorithm begins with small local reconstructions based on nonlinear bundle adjustment (BA). These are then joined together in a hierarchical manner using a strategy that requires solving a linear least squares optimization problem followed by a nonlinear transform. The algorithm can handle ordered monocular and stereo image sequences. Two stereo images or three monocular images are adequate for the initial reconstructions. Bulk of the computation involves solving a linear least squares problem and therefore the proposed algorithm avoids three major issues associated with most of the nonlinear optimization algorithms currently used for SFM; the need for a reasonably accurate initial guess, the need for iterations, and the possibility of being trapped in a local minimum. Also, by summarizing all the original observations into the small local reconstructions and associated information matrices, the proposed linear SFM manages to preserve all the information contained in the observations.

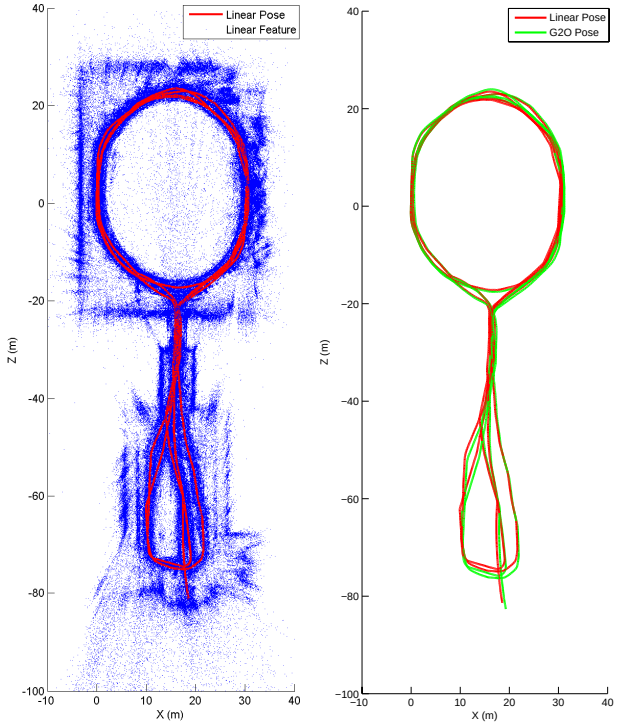
The input of Linear SFM is a sequence of initial reconstructions. Each initial reconstruction is built with three monocular images with two common camera poses, or with two stereo images with one common camera poses between two adjacent initial reconstructions, built by BA, together with the corresponding information matrix.

The output of Linear SFM algorithm is a global reconstruction estimates as well as the corresponding information matrix. For convenience, the result of the global reconstruction is transformed into the coordinate frame of the first camera pose.



(a) Top view (b) 3D view

Fig. 1. Linear SFM result of photogrammetric College Monocular dataset.



(a) Pose Feature (b) Comparison

Fig. 2. Result of New College stereo dataset.

II. Start to Run

Please run the code on x64 platform.

A. Datasets

The code are accompanied by some experimental datasets, for both monocular and stereo SFM. Thus, user can run the code directly by using these demo datasets.

1) Monocular Datasets:

- Aerial Photogrammetric Village Datasets, with 88 initial reconstructions
- Aerial Photogrammetric College Datasets, with 466 initial reconstructions

2) Stereo Datasets:

- New College Datasets, with 3499 initial reconstructions

B. Start to Run the C/C++ Code in Linux

1) Install:

Linear SFM C/C++ in Linux requires cmake, Eigen, SuiteSparse. On Ubuntu/Debian these dependencies are resolved by installing the following packages. If one want to improve the efficiency of solving sparse linear systems, we recommend to download and install GotoBLAS.

- sudo apt-get install cmake
- sudo apt-get install libeigen3-dev
- sudo apt-get install libsuitesparse-dev

We recommend a so-called out of source build which can be achieved by the following command sequence.

- cd LinearSFM_C/linux
- mkdir build
- cd build
- cmake ..
- make

```
- sudo make install
```

The binaries will be placed in bin and the libraries in lib which are both located in the top-level folder.

2) Run:

The demonstrated datasets can be run by one of the following command lines

```
- LinearSFM -path RS468_C -p Pose_RS468.txt -f Feature_RS468.txt -num 466 -type Monocular
- LinearSFM -path NC3500_C -p Pose_NC3500.txt -f Feature_NC3500.txt -num 3499 -type Stereo
- LinearSFM -path RS90_C -p Pose_RS90.txt -f Feature_RS90.txt -num 88 -type Monocular
```

If the user wants to know all the commands, one can type

```
- LinearSFM -help
```

and the following help information will be on the command window

LinearSFM General Options:

-path	Set Data Path.
-st	Set Path to Save Final State Vector.
-p	Set Path to Save Final Poses.
-f	Set Path to Save Final Features.
-num	Set Initial Reconstruction Number.
-type	Set Data Type, including Monocular and Stereo.

C. Start to Run the C/C++ Code in Windows

The CHOLMOD and GotoBLAS2 software packages are used in the C/C++ code in Windows to solve linear equations.

We recommend the user to use Microsoft Visual Studio 2010 or higher version to compile the code. The user can open and run solution "LinearSFM.sln" in folder "windows/LinearSFM" to get the results of the demonstrated datasets. We provide class "CLinearSFM" to accomplish the whole LinearSFM functions.

The user can easily set all the inputs in "_tmain()" function to run a specific dataset by the following steps:

1) Open the "LinearSFM.sln" Solution:

2) Select the Dataset:

Uncomment the dataset one want to run, as well as the files to save the result of poses and features and comment all the other datasets as follows

```
//Stereo Dataset

//New College Dataset
//szData = "../../../DataForC/NC3500_C"; nMapCount = 3499;
//szPose = "../../../Pose_NC3500.txt";
//szFeature = "../../../Feature_NC3500.txt";

//Monocular Dataset

//Aerial Photogrammetric Village
szData = "../../../DataForC/RS90_C"; nMapCount = 88;
szPose = "../../../Pose_RS90.txt";
szFeature = "../../../Feature_RS90.txt";

//Aerial Photogrammetric College
//szData = "../../../DataForC/RS468_C"; nMapCount = 466;
//szPose = "../../../Pose_RS468.txt";
//szFeature = "../../../Feature_RS468.txt";
```

3) Select to Run Monocular or Stereo:
If the dataset one selected is monocular, then uncomment

```
//Monocular
ptr.runMono( szSt, szPose, szFeature, szData, nMapCount );
```

If stereo, uncomment

```
//Stereo
ptr.runStereo( szSt, szPose, szFeature, szData, nMapCount );
```

4) Compile and Run:

III. Academic Usage and Paper Reference

The paper details the Linear SFM algorithm is

- L. Zhao, S. Huang and G. Dissanayake, "Linear MonoSLAM: A Linear Approach to Large-Scale Monocular SLAM Problems," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1517-1523, 2013.

If this code is used for academic work, please reference this paper.
If you are interested in the paper for further understanding about the Linear SFM algorithm, please send an email to the first author.