
The Damped Bundle Adjustment Toolbox

v0.1 for Matlab

Niclas Börlin
Department of Computing Science
Umeå University
niclas.borlin@cs.umu.se

November 13, 2013

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Limitations	2
1.3	Legal	2
1.4	Scientific publications	2
2	Installation	2
3	Usage	2
3.1	Demos	2
3.1.1	loadplotdemo	2
3.1.2	romabundledemo	2
3.1.3	camcalibdemo	2
3.2	Using your own data	2
3.2.1	Enabling text export from Photomodeler	2
3.2.2	Export from Photomodeler	3
3.2.3	Loading into Matlab	3
A	Camera model	3

1 Introduction

1.1 Purpose

Matlab toolbox with freely available code for bundle adjustment. Intention to be state-of-the-art.

1.2 Limitations

What it can do.

What it cannot do.

1.3 Legal

Licensing.

1.4 Scientific publications

Refer to any of the papers...

2 Installation

1. Download the package file `dbat_0.1.zip`.
2. Unpack the package into a directory `dbat`.
3. Inside Matlab, do the following initialization:

```
cd dbat % the directory where you installed the files.  
dbatSetup % set paths, etc.
```
4. Test the installation by executing `loadplotdemo`.
5. If `loadplotdemo` runs without errors and generates a figure with a camera network, the installation is ok.

3 Usage

3.1 Demos

3.1.1 loadplotdemo

3.1.2 romabundledemo

3.1.3 camcalibdemo

3.2 Using your own data

3.2.1 Enabling text export from Photomodeler

.
Some versions of Photomodeler do not have the text file export option enabled by default. In that case, follow the following steps to enable it:

1. Right-click on the main window toolbar, select *Customize toolbar....*
2. In the *Commands* tab, select the *File* category.

3. Drag the *Export Text File...* command to a toolbar of your choice.
4. Now you should be able to export your project as a text file by clicking on the *Export Text File* button.

3.2.2 Export from Photomodeler

To import a Photomodeler project into the toolbox, the following steps are valid in Photomodeler Scanner 2012:

1. Export the project using *Export Text File*. If the *Export Text File* command is not available, follow the instructions in Section 3.2.1.
2. After export, open the *Project/Cameras...* dialog and select the camera that was used in your project.
3. Open the generated text file in a text editor.
 - (a) On the 2nd line (usually reading 0.00005 20), append the width and height in pixels of your images, e.g. to 0.000500 20 5616 3744.
 - (b) Inspect the 4th line. For instance, the original data in *roma.txt* was (some trailing zeros removed):
 24.3581 18.1143 12.0 35.96404 24.0 0.00022 -0.0 0.0 0.0 0.0
 The values correspond to the following camera parameters:
 focal pp_x pp_y format_w format_h K1 K2 K3 P1 P2.
 Notice that most of the significant digits of K1–K3 were lost in the text export.
 - (c) Update the parameter values on the 4th line with values from the camera dialog *for each parameter with a larger number of significant digits in the dialog*. This usually means all parameters except *format_w*. In the *roma.txt* test case, the 4th line was modified to:
 24.3581 18.1143 12 35.96404 24 2.174e-4 -1.518e-7 0 0 0.

3.2.3 Loading into Matlab

1. In matlab, run step 3 from Section 2 if not already done.
2. Set the variable `fName` to the text export file name
`fName='c:/path/to/exported/file.txt';`, or select it using
`[f,p]=uigetfile('*.txt');` `fName=[f,p];`
3. Run the `loadplotdemo` script. A figure with your camera network, aligned with the first camera and rotated to have +Z 'up', should now have been generated.

A Camera model