Manual for LoD2 Modelling by Integrating AHN and BAG data

# Application

This manual explains the working of the tools (executables) as designed by Biao Xiong for 3D4EM. In appendix C there is an instruction on how to install and compile the corresponding source codes. This tool set is to reconstruct LoD2 building models by combining laser points and cadastral footprint maps. The input data are laser points and 2D cadastral maps, and output data are 3D building models. Point cloud from image dense matching could also be used, but modelling results depends point quality. Until now most image point cloud are not as good as laser points.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Open source | Lastools | FME |
| Selection of point cloud data |  |  | Select laser points within building polygons |
| Removing double points |  | *Lasduplicate* |  |
| Conversion of laz/las to laser format | *laz2xyzlaser* | *Las2txt, ascii2laser* | *laz2xyzlaser* |
| Convert map to asci format | *shp2pcm* | | |
| Segment point clouds into planar patches | *growsurfaces* | | |
| LoD2 creation | *3DBuilding* | | |
| Converting 3D models to 3D shapes | *pcm2shp* | | |
| Evaluate 3D models | *Evaluate3DBuilding* | | |

Table 1 Overview of LoD2 building modelling tools, with three different data preparation possibilities.

# Requirement

* 1. System Requirement:

Windows, tested on win 8 and win 10

* 1. Data requirement:

Laser points are in las or laz format, the file should be small enough so can be loaded into memory. It is therefore advised to work with maximum 7 Million points. Only building points are needed; if the las file contains information on the class buildings or has already been filtered into ground/nonground, it is preferred only to use “buildings” or “nonground” points respectively. The point density is expected to be above 5 points/m2.

Cadastral footprint maps are in Esri shape format, 1:500-1:2000 scales are recommended.

The laser points and footprint maps are assumed to be in the same coordinate system.

* 1. License requirement:

Three alternative methods for the preparation stage are given here, each with their own license requirements.

1. Open source solution. All tools are open source software with GNU General Public License. Triangle.c is used in the libraries of all open source software; this code may be freely redistributed under the condition that the copyright notices are not removed, and no compensation is received. See appendix b, for the complete copyright statement of triangle.c.
2. Lasduplicate and las2txt from lastools tool set can be used directly to remove double points and convert laz/las to ascii. Please license from martin.isenburg@rapidlasso.com to use lasduplicate commercially. See appendix A for lastools license options.
3. FME is a useful program to handle a variety of data preprocessing steps, such as selecting the area of interest. Also, clipping the laz file to the corresponding map data can be done efficiently in FME. If you have FME it is preferred to clip the laser data to the building polygons. The output can be a laz or las file. The advantage is that the size of the point cloud file is reduced significantly.

Libraries CGAL, GDAL and PPrepair are used by 3DBuilding, shp2pcm and pcm2shp tools.

# Installation

Copy the folder MappingLib to the place where you want, for example C:\Program Files, and add the path “your\_folder”\MappingLib to the system path.

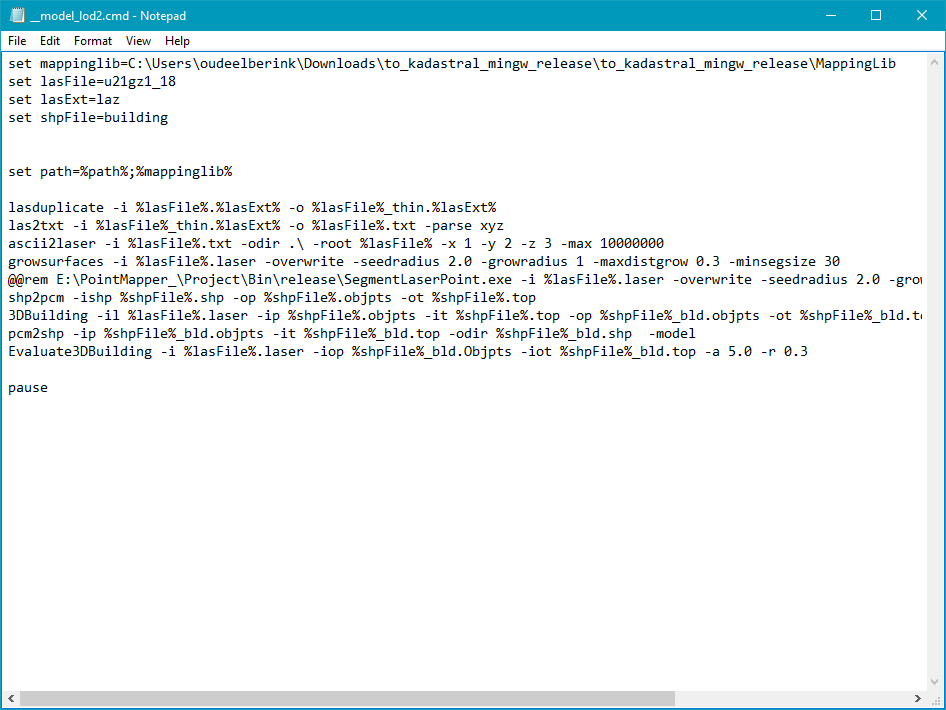
# Usage

* 1. \_\_model\_lod2.cmd is the batch file to model lod2 building models. It should be put in the same folder of las and shp files.
  2. Three parameters in \_\_model\_lod2.cmd file can be changed according to user data.

1. Set mappinglib to the path of the executable file, for example: mappinglib=..\MappingLib
2. Set lasFile to the name of las file, for example lasFile=bag77
3. Set shpFile to the name of shp file, for example shpFile =bag77
   1. Run \_\_model\_lod2.cmd, the created \*\_bld.shp is the destination result.
   2. MappingLib folder has the excecutable programs.

In the following figure a screenshot is given of a batchfile (using the Lastools preprocessing steps). The blue arrow shows the “mapping lib” folder where the newest exe’s are.

Orange, grey and yellow: specify the laser file name, extension (las/laz) and building shape file of the 2D map.



# Description of the tools

The default parameters work for AHN2 and AHN3 data, and work for most cases. For advanced users, a new set of parameters will get better results. You will set the parameters in batch file \_\_model\_lod2.cmd. For a better understanding how the tools work, please see to the papers, which are described under each section.

The batch is to process dataset that is small enough to be loaded into memory. FME could be used to partition large dataset into small ones, which then can be processed by this toolset. A similar partition by FME has been used by 3D TOP10NL project.

* 1. **lasduplicate.** This tool is from lastool set, to remove double points. It is called by:

lasduplicate -i %lasFile%.las -o %lasFile%.las

See link <https://rapidlasso.com/lastools/> for more information about lastools.

* 1. **las2txt.** This tool is from lastool set, to transform laser point format from .las to raw ascii format. It is called by:

las2txt -i %lasFile%.las -o %lasFile%.txt -parse xyz

In the parameter list: -i is input data, in las or laz format, -o is output data, in txt format, and -parse show which information is parsed. Here xyz presents only parse x,y, and z coordination. See link <https://rapidlasso.com/lastools/> for more information about lastools.

* 1. **ascii2laser.** It is to transform laser point from asci to laser format (ITC internal format). It is called by:

ascii2laser -f \*.txt -odir .\ -root 0 -x 1 –y 2 -z 3 -max 5000000

In the parameter list: -f resent input file/files, -odir presents output folder, -root presents root name of output file, and parameters -x, -y, -z means parse x, y, and z coordinate. -max presents the maximum points that one output file can has. Because we what all points of input file should be stored in one output file, the -max parameters should always larger than the number of points of input file.

* 1. **laz2xyzlaser.** It is to transform laser point from laz or las to laser format (ITC internal format). It is called by:

laz2xyzlaser -i %lasFile%.%lasExt% -o %lasFile%.laser [–td 0.01]

In the parameter list: -i resent input file, -o output file, optional one can set a thinning distance, e.g. 0.01 m. That means that all points within 1 cm from another point are removed. This avoids potential crashing in the reconstruction phase. Default thinning distance is 0, meaning that there is no thinning.

* 1. **shp2pcm.** This tool is to transform cadaster map from .shp format to .objpts and .top format (ITC internal format). It is called by:

shp2pcm -ishp %shpFile%.shp -op %shpFile%.objpts -ot %shpFile%.top

In the parameter list: -ishp is input data, in shp format, -op is output data, in objpts format, and -ot is output data in top format. The combination of .objpts and .top presents a cadaster data.

* 1. **growsurfaces.** This tool is to segment laser points into planar patches. It is called by:

growsurfaces -i %lasFile%.laser -overwrite -seedradius 2.0 -growradius 1 -maxdistgrow 0.3 –minsegsize 30

In the parameter list: -i is input data, in laser format, -overwirte means the result will replace input data, -seedradius is seed neighbourhood radius, -growradius is growing search radius, -maxdistgrow is maximum distance of point to surface, -minsegsize is minimum segment to be kept. The algorithm is explained in the paper:

*Vosselman, G., 2012. Automated planimetric quality control in high accuracy airborne laser scanning surveys. ISPRS J. Photogramm. Remote Sens. 74, 90–100.*

* 1. **3DBuilding.** This tool is to reconstruct 3D building model from laser data and cadaster maps.

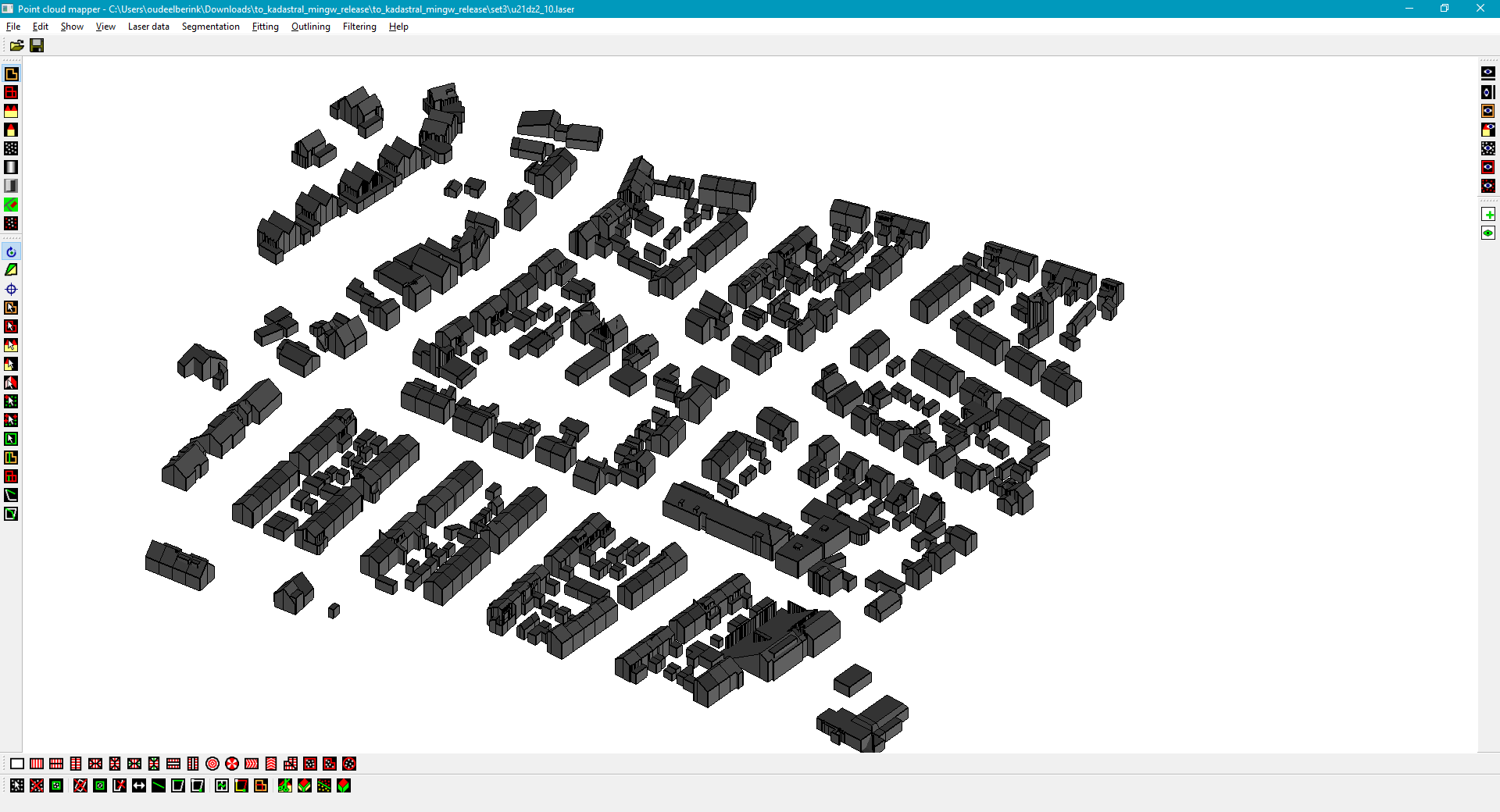
3DBuilding -il %lasFile%.laser -ip %shpFile%.objpts -it %shpFile%.top -op %shpFile%\_bld.objpts -ot %shpFile%\_bld.top

In the parameter list: -il is input data, in laser format, -ip and -it are cadaster maps in objpts and top format. -op and -ot are created building model in objpts and top format. This method derive structure points and boundaries to find inner corners and boundaries. The idea is introduced in the paper:

*Xiong, B., Oude Elberink, S.J. and Vosselman, G. (2014) Building modeling from noisy photogrammetric point clouds. In: Annals of the Photogrammetry, Remote Sensing and Spatial Information Science, Vol. II-3, 2013, ISPRS technical Commission III, Symposium, 5-7 september, 2014, Zurich, Switserland.: open access / ed. by K. Schindler and N. Paparoditis. Zurich: ISPRS, 2014. eISSN 2194-9050 pp. 197-204*

In this paper the building outlines are extracted from point clouds for the situation that no BAG polygons are available. The toolset is made in 3D4EM project that the main work in 2016 was the integration of the decomposed BAG outlines into this method, as described in:

*Xiong, B, Oude Elberink, S.J. and Vosselman, G. (2016) Footprint map partitioning using airborne laser scanning data. In: Proceedings of the XXIII ISPRS Congress : From human history to the future with spatial information, 12-19 July 2016, Prague, Czech Republic. Peer reviewed Annals, Volume III-3, 2016 / edited by L. Halounova, ... [et al.]. ISPRS, 2016. ISSN: 2194-9050. pp. 241-247.*



*Screenshot of “test data set 3” on detailed building models aligned with BAG polygons.*

* 1. **pcm2shp.** This tool is to transform vector data from ITC internal format (.objpts and .top) to shp file. It is called by:

pcm2shp -ip %shpFile%\_bld.objpts -it %shpFile%\_bld.top -odir %shpFile%\_bld.shp -model

In the parameter list: -ip and -it present input vector data in objpts and top format, -odir present the output file, in shp format.

* 1. **Evaluate3DBuilding.** This tool is to evaluate the quality of created models. A quality report will be exported. It is called by:

Evaluate3DBuilding -i %lasFile%.laser -iop %shpFile%\_bld.Objpts -iot %shpFile%\_bld.top -a 5.0 -r 0.3

In the parameter list: -i is input laser data, in laser, -iop and -iot are input building model data. -a presents area threshold and -r presents residual threshold. A building will be verified to be erroneous if it has more than an area of points that is r far away from created models. The evaluation idea is introduced in the paper:

*Xiong, B., et al. "Flexible building primitives for 3D building modeling."ISPRS Journal of Photogrammetry and Remote Sensing 101 (2015): 275-290.*

A report will be exported as follows:

Total Blds: 15

Bad Blds: 20 percent

Area threshold: 5.00 m2, Residual threshold: 0.5 m

Average Residual: 0.34 m

Bad area(m) per building:

0.000

0.000

0.000

7.795

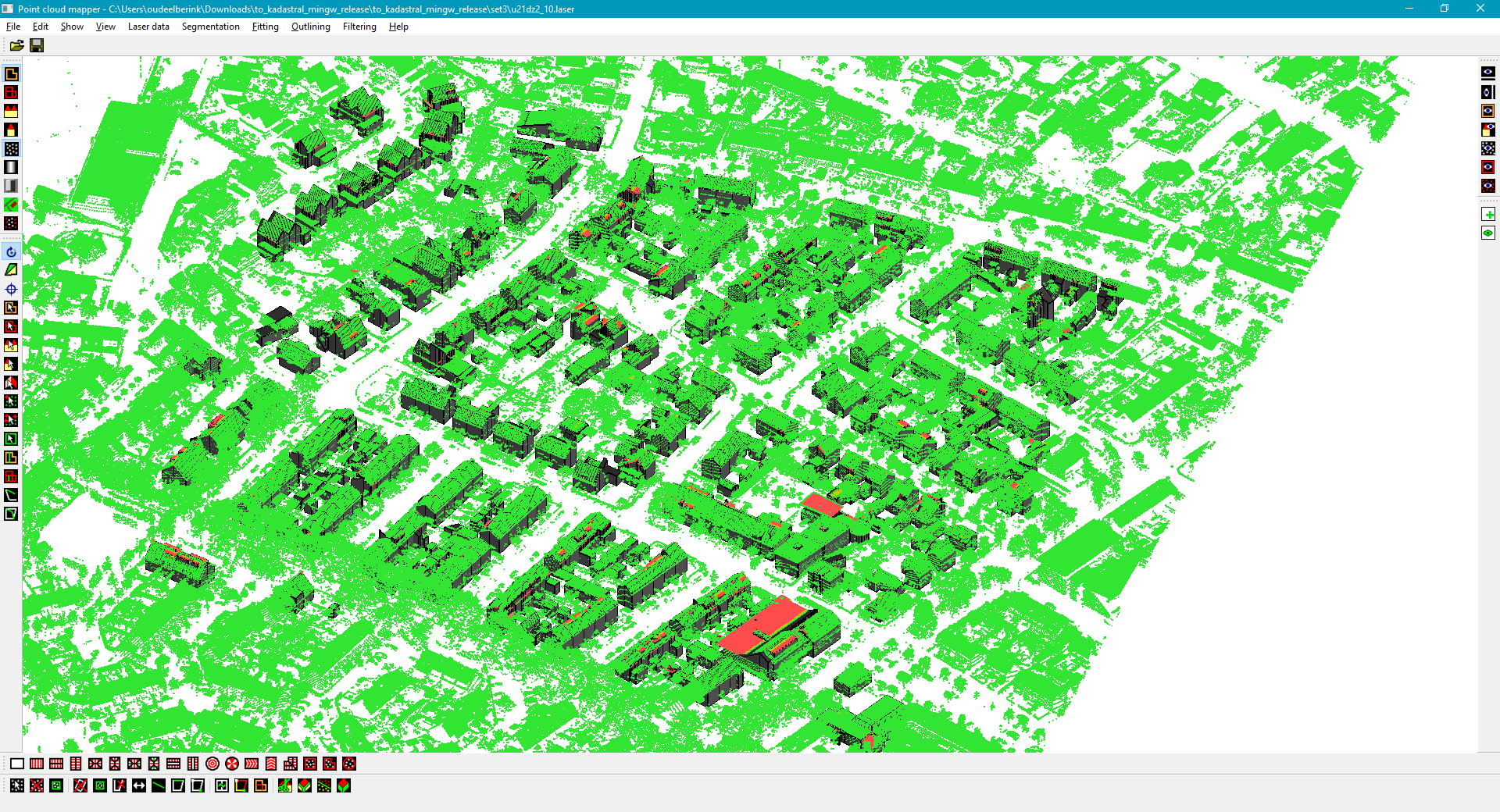
18.405

33.177

…

This report says the total building number is 15, and 20% buildings are evaluated to be wrong. The area threshold is 5.0 m2, and residual threshold is 0.5m. Average residual for all points is 0.34m. The average residuals for all buildings are listed at the end of the table.

Next to this report, residual values are stored per point in the input file. This way one can visualize the evaluation results by overlaying the points on the 3D model, and visualize the residuals, as shown in the figure below.



Appendix A: Lastools licence

From

<http://www.cs.unc.edu/~isenburg/lastools/LICENSE.txt>

--------------------------------------------------------------------------

LICENSE AGREEMENT:

This software is distributed WITHOUT ANY WARRANTY and without even the

implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

--------------------------------------------------------------------------

There are TWO parts to LAStools. One part is OPEN source (LGPL 2.1). The

other part is CLOSED source and requires licensing for most commercial

or government use. If you're unsure please email 'martin@rapidlasso.com'

before using LAStools. For education and evaluation purposes you can use

and test LAStools as much as you need to ... (-:

Personal note: The creator of LAStools and LASzip is currently not able

to attend LiDAR conferences or other scientific meetings in the US over

a somewhat unfortunate, kind of silly, and in many ways absurdly funny

incident involving "laser chickens" and a US Nuclear Weapons Lab. If you

have enough "pull" to rectify this ridiculous situation we would like to

hear from you ... (-;

--------------------------------------------------------------------------

This part of LAStools is "free" open-source LGPL (see COPYING.txt). To

purchase a different licensing model contact 'license@rapidlasso.com'.

\* laszip

\* lasindex

\* lasvalidate

\* lasliberate

\* lasinfo

\* las2las

\* lasdiff

\* lasmerge

\* las2txt & txt2las

\* lasprecision

\* LASzip (with static linking exception)

\* LASlib (with LASzip) - the API used by LAStools.

These two are also "free" to use but not available in open source.

\* lasview

\* laspublish

--------------------------------------------------------------------------

The following LAStools require licensing for any commercial, government,

or production use. However, they may be used "freely" for all non-profit

personal, non-military educational, or non-profit humanitarian purposes.

Note that the output of the unlicensed version can be slightly distorted

after certain point limits are exceeded. Control output in the console

(aka "the black window") informs the user whenever this happens.

\* blast2dem

\* blast2iso

\* lasground & lasground\_new

\* lasheight & lastrack

\* lasclassify

\* lasgrid & lascanopy

\* lasboundary

\* lascontrol

\* lasoverlap

\* lasoverage

\* lasduplicate

\* lassplit

\* lasreturn

\* las2tin

\* las2iso

\* las2dem

\* lasthin & lasnoise

\* lassort

\* lastile

\* lasplanes

\* lascolor

\* lasclip

\* las2shp & shp2las.

--------------------------------------------------------------------------

If you want to use LAStools please contact us for licensing. You can

(a) license the existing executables for production

(b) license the existing executables for inclusion in your software

You can also contract with rapidlasso GmbH to create LAStools tailored

to your particular needs.

--------------------------------------------------------------------------

COPYRIGHT:

(c) 2007-2015, rapidlasso GmbH, GERMANY - fast tools to catch reality

Appendix B Triangle.c

Version 1.3 \*/

/\* July 19, 1996 \*/

/\* \*/

/\* Copyright 1996 \*/

/\* Jonathan Richard Shewchuk \*/

/\* School of Computer Science \*/

/\* Carnegie Mellon University \*/

/\* 5000 Forbes Avenue \*/

/\* Pittsburgh, Pennsylvania 15213-3891 \*/

/\* jrs@cs.cmu.edu \*/

/\* \*/

/\* This program may be freely redistributed under the condition that the \*/

/\* copyright notices (including this entire header and the copyright \*/

/\* notice printed when the `-h' switch is selected) are not removed, and \*/

/\* no compensation is received. Private, research, and institutional \*/

/\* use is free. You may distribute modified versions of this code UNDER \*/

/\* THE CONDITION THAT THIS CODE AND ANY MODIFICATIONS MADE TO IT IN THE \*/

/\* SAME FILE REMAIN UNDER COPYRIGHT OF THE ORIGINAL AUTHOR, BOTH SOURCE \*/

/\* AND OBJECT CODE ARE MADE FREELY AVAILABLE WITHOUT CHARGE, AND CLEAR \*/

/\* NOTICE IS GIVEN OF THE MODIFICATIONS. Distribution of this code as \*/

/\* part of a commercial system is permissible ONLY BY DIRECT ARRANGEMENT \*/

/\* WITH THE AUTHOR. (If you are not directly supplying this code to a \*/

/\* customer, and you are instead telling them how they can obtain it for \*/

/\* free, then you are not required to make any arrangement with me.) \*/

/\* \*/

/\* Hypertext instructions for Triangle are available on the Web at \*/

/\* \*/

/\* http://www.cs.cmu.edu/~quake/triangle.html

Appendix C

Compilation instructions on the

LoD2 building modelling tools

1. Install Devcpp

Download and install DevCpp program, which is an IDE with gcc compiler. Tested version is Dev-Cpp 5.4.0 - MinGW64 - tdm gcc 4.7 1. If a portable package is used, we need to find and automatically configure compilers by:

*Menu 🡪 Compiler Options 🡪 and click the bottom “find and automatically configure compilers”.*

DevCpp should have MinGW, and MSYS. If the MSYS is not updated to version 1.07, download and install it from here:

<https://sourceforge.net/projects/mingw/files/MSYS/Base/msys-core/msys-1.0.11/>

In dvecpp\_path\MinGW64\bin folder, copy make.exe, and rename make.exe to mingw32-make.exe.

Add system path *dvecpp\_path\MinGW64\bin, and dvecpp\_path\bin*

Add system environment

MAPPING\_LIB\_DIR= D:\eos\_mapping\Library\lib64

DEVCPP\_DIR= D:/Dev-Cpp

1. Compile GMP and MPFR
   1. GMP

Follow instruction:

<http://suchideas.com/journal/2007/07/installing-gmp-on-windows/>

<https://git.archlinux.org/svntogit/packages.git/tree/trunk/PKGBUILD?h=packages/gmp>

In short, in MSYS of MinGW, redirect to the source code folder and run:

**./configure --prefix=/d/dev-cpp --enable-static --disable-shared**

**make**

**make check**

**make install**

1. Tips:

When you find problem “/c/Program: No such file or directory”. That seems to indicate that configure is not happy with the fact that Mingw is installed to a path that contains a space.

* 1. MPFR

Follow the instruction:

<https://github.com/emphasis87/libmpfr-msys2-mingw64>

In short, Compile in MSYS of MinGW, redirect to the source code folder and do:

./configure --prefix=/d/dev-cpp --with-gmp-include=/D/Dev-Cpp/include --with-gmp-lib=/D/Dev-Cpp/lib --enable-static --disable-shared  
make clean

make > build.log

make check

make install

1. Install boost

Follow the instruction:

<http://www.joshuaburkholder.com/blog/2013/04/23/how-to-compile-boost-using-mingw/>

In short:

In bootstrap.bat, change

set toolset=msvc to set toolset=gcc

Pass in mingw to bootstrap.bat:

bootstrap.bat gcc

Call b2.exe:

b2.exe toolset=gcc --build-type=complete address-model=64 architecture=x86

1. Install CGAL
2. 4.1 Add boost and GMP to system variables

On my computer it is:

BOOST\_LIBRARYDIR D:\OpenSource\boost\boost\_1\_54\_0\lib

BOOST\_INCLUDEDIR D:\OpenSource\boost\boost\_1\_54\_0

Boost\_Dir D:\OpenSource\boost\boost\_1\_54\_0

GMP\_Dir D:\OpenSource\CGAL\CGAL-4.4\auxiliary\gmp

1. 4.2 Download CGAL, and compile with CMAKE

Download and install CMAKE: <https://cmake.org/download/>, tested version 3.6.3.

Download CGAL: <https://github.com/CGAL/cgal/releases>, tested version 4.9.

Open CMAKE, and locate the path of CGAL source file. Configure and generate make file. Then in console window, re-direct to the source file path, and compile by make.exe with commend “make”.

1. Install GADL

<https://trac.osgeo.org/gdal/wiki/FAQInstallationAndBuilding#CanIbuildGDALwithCygwinorMinGW>

<https://trac.osgeo.org/gdal/wiki/BuildingWithMinGW>

./configure --prefix=$PATH\_TO\_MINGW\_ROOT --host=mingw32 \

--without-libtool --without-python $YOUR\_CONFIG\_OPTIONS

make

make install

Linking issues:

Edit GNUmakefile on lines 6-12 and replace $(GDAL\_ROOT) with . (a dot).

Problem: gdal.h:42 fatal error:cpl\_port.h: no such file or directory

Solve: <https://www.oschina.net/question/997483_2151503>

Still think this is bug. gdal\_wrap.cpp should #include "gdal/cpl\_port.h" instead of just "cpl\_port.h"

For MSVC:

<https://trac.osgeo.org/gdal/wiki/BuildingOnWindows>

First you should check the [basic options](https://trac.osgeo.org/gdal/wiki/BuildingOnWindows#BasicOptions) in nmake.opt especially the settings for the VC variant to use and for the installation directory.

C:\Program Files\Microsoft Visual Studio\VC98\bin\VCVARS32.BAT

C:\GDAL> nmake -f makefile.vc MSVC\_VER=1900 DEBUG=1

C:\GDAL> nmake /f makefile.vc install

C:\\GDAL> nmake /f makefile.vc devinstall

1. Compile mapping library

You can compile in dev-cpp IDE and

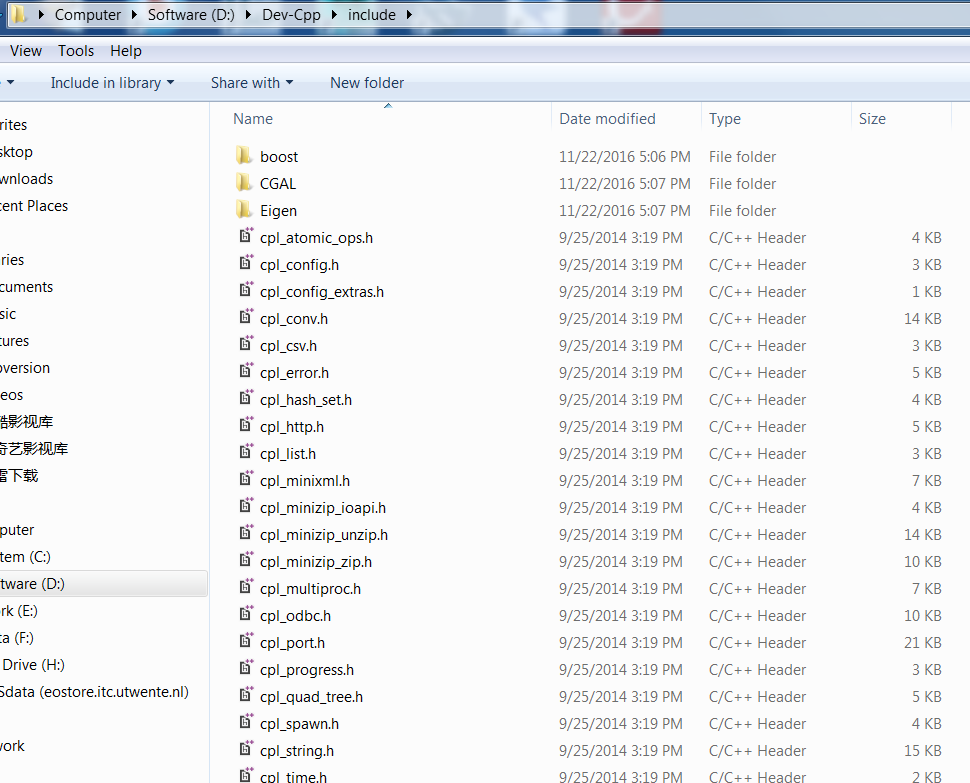
1. Add all the compiled library compiled to Dev-Cpp path

Header files should be in “yourlocation\Dev-Cpp\include\”

Achieve library files should be in “yourlocation\Dev-Cpp\lib\”

Dynamic library files should be in “yourlocation\Dev-Cpp\bin\”

Boost, CGAL, Eigen have their folders, and GDAL, BMP, MPDF



1. Add Dev-cpp path to system variables

DEVCPP\_DIR = “yourlocation\Dev-Cpp”

1. Source code package
2. Compile

You can open the project file \*.devcpp by DEVCPP.exe, and compile the source code in IDE. Or compile them via the make file in a console window.

Compile mapping library, start with photogrammetry, then laserscan, buildings.

Compile tools such as inferenceEngine, 3DBuilding, and Evaluate3DBuilding.

1. Use 3DBuilding for LoD2 modelling

See more information on the running of the tools “Manual for LoD2 Modelling by Integrating AHN and BAG data” (the first part of this document).