AutoFuzSlpPos-2016
User Manual

Liang-Jun ZHU Cheng-Zhi QIN

zlj@lreis.ac.cn qincz@lreis.ac.cn

State Key Laboratory of Resources and Environmental Information System,

Institute of Geographic Sciences and Natural Resources Research,

Chinese Academy of Sciences

Nov 24, 2016

1 Introduction ---------- 2

2 Compilation and Configuration ---------- 2

2.1 Code structure ---------- 2

2.2 Compile on Windows ---------- 3

2.3 Compile on Linux/Unix ---------- 4

2.4 Configuration ---------- 5

3 Run AutoFuzSlpPos ---------- 6

1 Introduction

AutoFuzSlpPos is an automatic approach with only one required input data (i.e., a gridded DEM of the study area) to deriving fuzzy slope positions based on the prototype-based inference method proposed by Qin *et al*. (2009. Quantification of spatial gradation of slope positions. Geomorphology 110, 152-161.). AutoFuzSlpPos consists three major parts, i.e., preparing topographic attributes, extracting typical locations, and calculating similarity for each slope position.

Current version of AutoFuzSlpPos is developed on the basis of TauDEM parallelized framework (http://hydrology.usu.edu/taudem/taudem5/index.html) and programmed using C++ and Python language. The program is capable with Windows and Linux/Unix, e.g., Windows 7, CentOS 6.2, and Ubuntu 14.04. The prerequisites environment of the compilation and configuration of AutoFuzSlpPos is as follows:

- For PC, Microsoft Visual C++ Redistributable (e.g., VS 2010), Microsoft MPI V6 or later, CMAKE
- For Linux/Unix, GCC 4.7+, MAKE, and MPICH or OpenMPI
- Python 2.7+ packaged with Numpy 1.6+ and GDAL 1.9+.

2 Compilation and Configuration

2.1 Code structure

The source code files consist of two parts: 1) the files of C++ source code located in .../source-code/code/src, and 2) python scripts located in .../source-code/code/py_main. The former will be compiled as separated functional executable files, such as “selecttyplocslppos” which is used for extracting typical locations and setting parameters for fuzzy inference of each slope position. The latter is to organize the whole workflow with a configurable script for users’ customizing optional parameters (see Table 1).

Table 1 Main function of the Python scripts

Script file	Functionality
main.py	The entrance of AutoFuzSlpPos.
Config.py	Parse the configuration file (*.ini) prepared by user.
Nomenclature.py	Predefined filenames.
TauDEM.py	Functions based on TauDEM and the extension functions, e.g., SelectTypLocSlpPos.
Util.py	Some fundamental functions, e.g., functions for the Input/Output of raster data.
PreProcessing.py	Preprocessing for topographic attributes, such as relative position index (RPI), profile curvature.
SelectTypLoc.py	Prepare input files for typical location extraction.
FuzzySlpPosInference.py	Prepare input files for fuzzy inference of each slope position.

2.2 Compile on Windows

The MPI library used for PC is Microsoft MPI V6 or later. Users can download the installation files from https://msdn.microsoft.com/en-us/library/bb524831(v=vs.85).aspx.

Firstly, make sure that CMAKE and nmake (installed with Visual Studio, such as VS2010) have been installed on the PC. It is highly recommended to check the MPI Library path in …/source-code/code/src/CMakeLists.txt to make sure they are correct for user's environment; the default is as follows:

```
include_directories("C:/Program Files (x86)/Microsoft SDKs/MPI/Include")
```

```
link_directories("C:/Program Files (x86)/Microsoft SDKs/MPI/Lib/x86")
```

```
link_libraries("C:/Program Files (x86)/Microsoft SDKs/MPI/Lib/x86/msmpi.lib")
```

Then, open “**Visual Studio Command Prompt**” from Start menu (as administrator), and run following commands:

```
cd <Build_Path>
cmake -G "NMake Makefiles" <Source_Path>
nmake
```

e.g.,

```
cd C:\AutoFuzSlpPos\build
cmake -G "NMake Makefiles" C:\source-code\code\src
nmake
```

Then, the executable files will be compiled and saved in `<Build_Path>`.

2.3 Compile on Linux/Unix

Unlike the MPI version for PC, the implementation of MPICH (http://www.mpich.org/downloads/) is adopted for Linux/Unix platform.

Firstly, check the MPI Library path in `.../source-code/code/src/makefile` to make sure that it is correct for current environment, e.g.:

```
MPI_DIR=/home/zhulj/mpich/
```

Then, assign the installation path, e.g.:

```
INSTALLDIR=/home/zhulj/AutoFuzSlpPos/exec
```

Run the following commands:

```
cd <Source_Path/src>

make
```

The executable files will be generated in `INSTALLDIR`.

2.4 Configuration

A script program of Python language is implemented to organize the workflow to deriving fuzzy slope positions.

User can configure the environment of AutoFuzSlpPos through the configuration file with the extension of *.ini, e.g. `.../source-code/data/Jamaica_windows.ini` for Windows and `../source-code/data/Jamaica_cluster.ini` for Linux cluster.

Besides the required path of the DEM of the study area (i.e., `rawdem`), the paths of the compiled executable files of AutoFuzSlpPos and workspace to store the results should be given correctly, for instance:

```
exeDir = /home/zhulj/AutoFuzSlpPos/exec

rootDir = /home/zhulj/AutoFuzSlpPos/Demo
```

Note that, if the path of MPI is in the ENVIRONMENT PATH in the system, the `mpiexeDir` could be set as “None”, otherwise it should be explicitly assigned, such as `mpiexeDir = /home/zhulj/mpich/bin`. The “hostfile” is used to specify the hosts on which the MPI jobs will be submitted. If user does not know how to prepare the hostfile, just leave it as `hostfile = None`. One possible example is as follows:

```
hostfile = /home/zhulj/AutoFuzSlpPos/exec/dgpm
```

```
dgpm-cluster.public:1
dgpm-compute-1.local:12
dgpm-compute-2.local:12
dgpm-compute-3.local:12
dgpm-compute-4.local:12
```

Next, the AutoFuzSlpPos with default parameter settings is ready to run for the specific study area. Other optional parameters are briefly introduced in the configuration file (*.ini).

3 Run AutoFuzSlpPos

Use the following command to run AutoFuzSlpPos:

```
python .../source-code/code/py_main/main.py -ini <configuration file path> [-proc <process number> -root <workspace path>]
```

where:

<configuration file path> is the full path of the *ini file, e.g.

```
/home/zhulj/AutoFuzSlpPos/data/Jamaica/Jamaica_dgpm.ini
```

<process number> is the process number for MPI, which can overwrite the inputProc defined in configuration file.

<workspace path> it the workspace path to store the results, which can overwrite rootDir the defined in configuration file.

Table 2 gives a brief introduction to the result files.

Table 2 Brief introduction to the result files of AutoFuzSlpPos

Result Folder	Introduction
FuzzySlpPos	Similarity maps of each slope position, as well as the hardened map of slope positions and the corresponding maximum similarity map.
DinfpreDir	Intermediate files in preparing topographic attributes
Params	GRID of topographic attributes, including RPI (Relative Position Index), profile curvature, slope gradient, and HAND (Height Above the Nearest Drainage), by default.
Config	Configuration files of the extraction of typical locations and fuzzy inference for each slope positions
TypLoc	Typical locations of each slope positions
Log	Log files which record information such as runtime, etc..

Note that in the current implementation a system of five basic slope positions is used, i.e., ridge (RDG), shoulder slope (SHD), backslope (BKS), footslope (FTS), and valley (VLY).