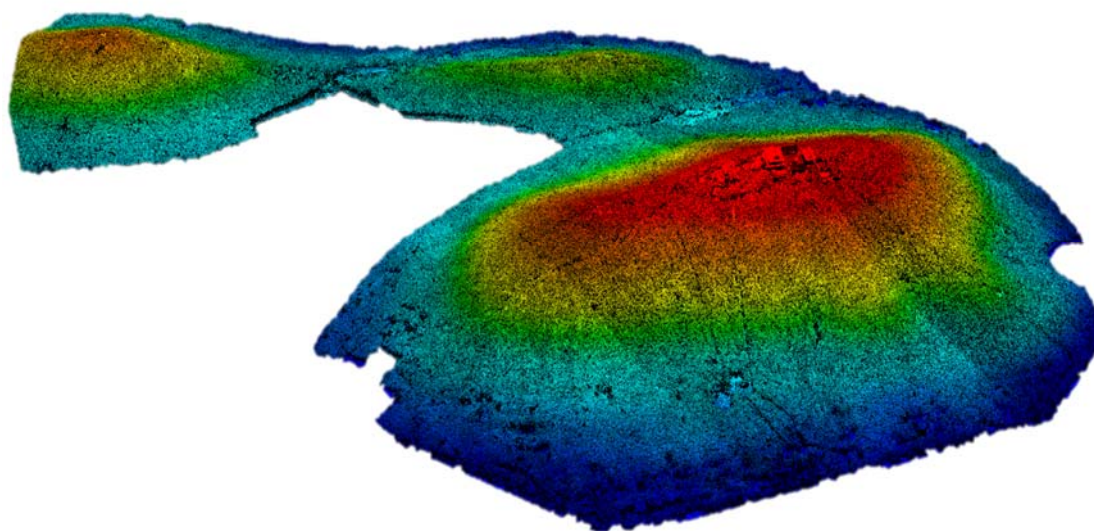*School of Geography, Beijing Normal University*

Airborne LiDAR Data Filtering Algorithm

Based on Cloth Simulation

CSF_CC

Wuming Zhang, Jianbo Qi, Peng Wan, Hongtao Wang, Donghui Xi,

Guangjian Yan, Xihan Mu

State Key Laboratory of Remote Sensing Science, Beijing Key Laboratory of Environmental Remote Sensing and Digital City, School of Geography, Beijing Normal University, Beijing 100875, China.

June, 2016

1 Introduction

Separating point clouds into ground and non-ground measurements is an essential step to generate digital terrain models (DTMs) from airborne LiDAR (light detection and ranging) data. Many filtering algorithms have been developed. However, even state-of-the-art filtering algorithms need to set up a number of complicated parameters carefully to achieve high accuracy.

For the purpose of reducing the parameters users to set, and promoting the filtering algorithms, we present a new filtering method which only needs a few easy-to-set integer and Boolean parameters. This method is based on cloth simulation which is a 3D computer graphics algorithm and is used for simulating cloth within a computer program. In this proposed approach, a LiDAR point cloud is inverted, and then a rigid cloth is used to cover the inverted surface. By analyzing the interactions between the cloth nodes and the corresponding LiDAR points, the locations of the cloth nodes can be determined to generate an approximation of the ground surface. Finally, the ground points can be extracted from the LiDAR point cloud by comparing the original LiDAR points and the generated surface. This filtering algorithm could be called cloth simulation filtering, CSF.

To promote the application of the CSF, the researchers developed a graphic user interface based the Cloud Compare (http://www.cloudcompare.org/). The GUI was developed for the Microsoft's Windows platform. The source code and executable program are available in the CSF_CC_Filelist.

If you use Cloth Simulation Filter (CSF) in your work, please cite the following paper, which can be downloaded from researchgate(https://www.researchgate.net/profile/Wuming_Zhang2). Thank you.

Zhang W, Qi J, Wan P, Wang H, Xie D, Wang X, Yan G. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sensing*. 2016; 8(6):501.

BibTex

2 Algorithm Details

Our method is based on the simulation of a simple physical process. Imagine a piece of cloth is placed above a terrain, and then this cloth drops because of gravity. Assuming that the cloth is soft enough to stick to the surface, the final shape of the cloth is the DSM (digital surface model). However, if the terrain is firstly turned upside down and the cloth is defined with rigidness, then the final shape of the cloth is the DTM. To simulate this physical process, we employ a technique that is called cloth simulation (Weil, 1986). Based on this technique, we developed our cloth simulation filtering (CSF) algorithm to extract ground points from LiDAR points. The overview of the proposed algorithm is illustrated in Fig. 1. First, the original point cloud is turned upside down, and

then a cloth drops to the inverted surface from above. By analyzing the interactions between the nodes of the cloth and the corresponding LiDAR points, the final shape of the cloth can be determined and used as a base to classify the original points into ground and non-ground parts.

2.1 Fundamental of the cloth simulation

During cloth simulation, the cloth can be modeled as a grid that consists of particles with mass and interconnections called a Mass-Spring Model (Provot, 1995). Fig. 2 shows the structure of the grid model. A particle on the node of the grid has no size but is assigned with a constant mass.
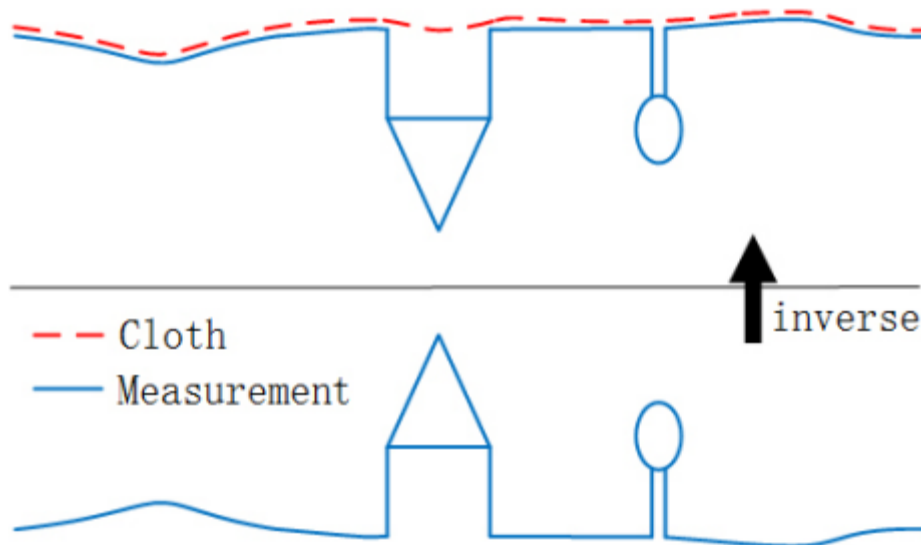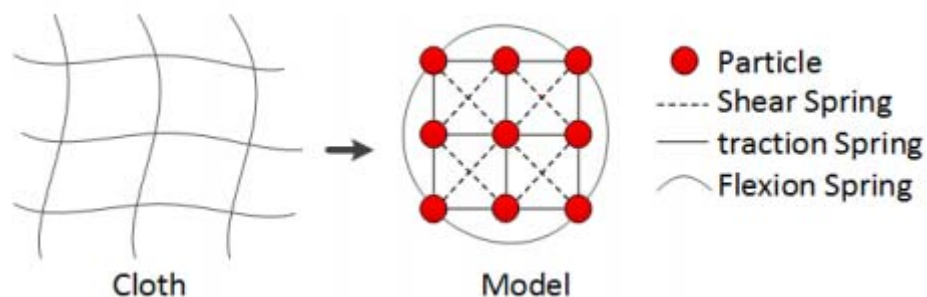

Fig. 1. Overview of the cloth simulation algorithm

Fig. 2. Mass-Spring Model of cloth

2.2 Implementation of CSF

First, we calculate the displacement of each particle only from gravity, i.e., solve equation (1) with internal forces equal to zero. Then, the explicit integration form of this equation is

$$X(t + \Delta t) = 2X(t) - X(t - \Delta t) + \frac{G}{m}\Delta t^2 \qquad (2)$$

where m is the mass of the particle (usually, m is set to 1) and Δt is the time step. Given the time

step and initial position, the current position can be calculated directly because G is a constant.

To constrain the displacement of particles in the void areas of the inverted surface, we consider the internal forces at the second step after the particles have been moved by gravity. Because of internal forces, particles will try to stay in the grid and return to the initial position. This procedure is illustrated in Fig. 3.
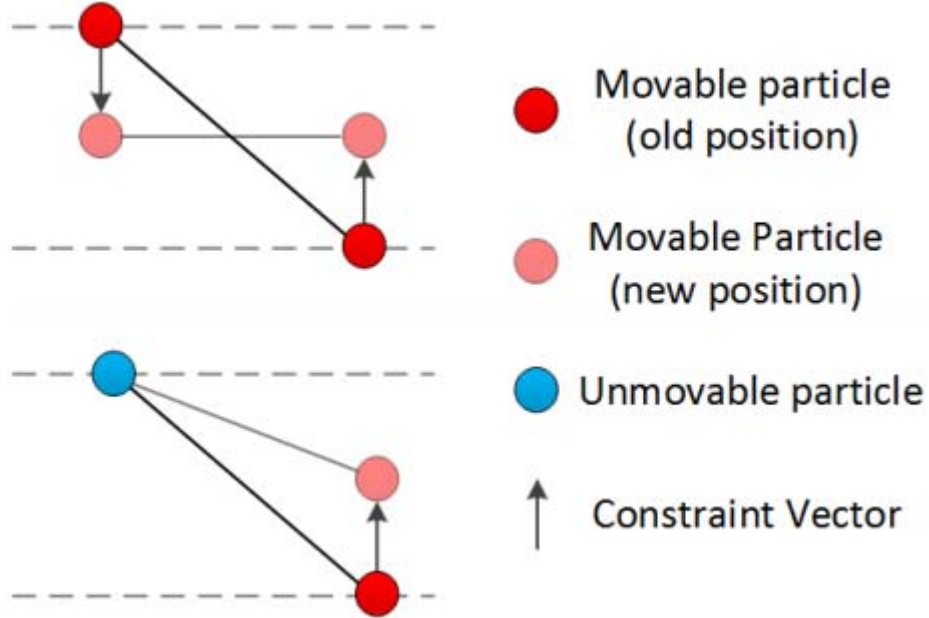

Fig. 3. Constraint between particles

The procedure of the proposed filtering algorithm is presented as follows:
(1) Inverting the original LIDAR point cloud.
(2) Initiating cloth grid. Determining numbers of particles according to the user defined grid resolution (GR). The initial position of cloth is usually set above the highest point.
(3) Projecting all the LIDAR points and grid particles to a horizontal plane and finding the nearest LIDAR point for each grid particle in this plane. Then recording the height value (h) of the nearest LIDAR point.
(4) For each grid particle, comparing the height difference between cloth particle and its nearest LIDAR point. If the height of particle is equal to or smaller than h, then this particle is set as unmovable.
(5) For each grid particle, calculating the position affected by gravity if this particle is movable; and calculating the displacement of each particle affected by internal forces.
(6) Repeating (4)-(5) until the maximum height variation (M HV) all particles less than a threshold or the program exceeds the maximum number of iterations which is specified by the user.
(7) Computing the cloud to cloud distance between the grid particles and LIDAR point cloud. Hence, for each LIDAR point, a distance value is obtained (d).
(8) Differentiating ground from non-ground points. For each LIDAR points, if the distance value (d) is smaller than hcc, then this point is classified as BE, otherwise it is classified as OBJ.

For steep slopes, this algorithm may yield relatively large errors because the simulated cloth is above the steep slopes and does not fit with the ground measurements very well due to the internal constraints among particles, which is illustrated in Fig. 4. Some ground measurements around steep slopes are mistakenly classified as OBJ. This problem can be solved by a post-processing method that smoothes the margins of steep slopes.
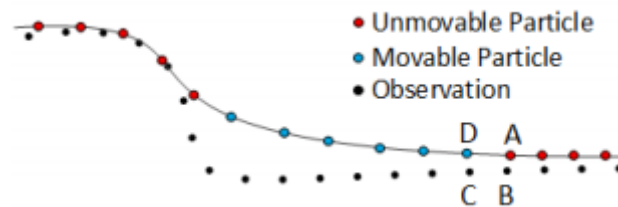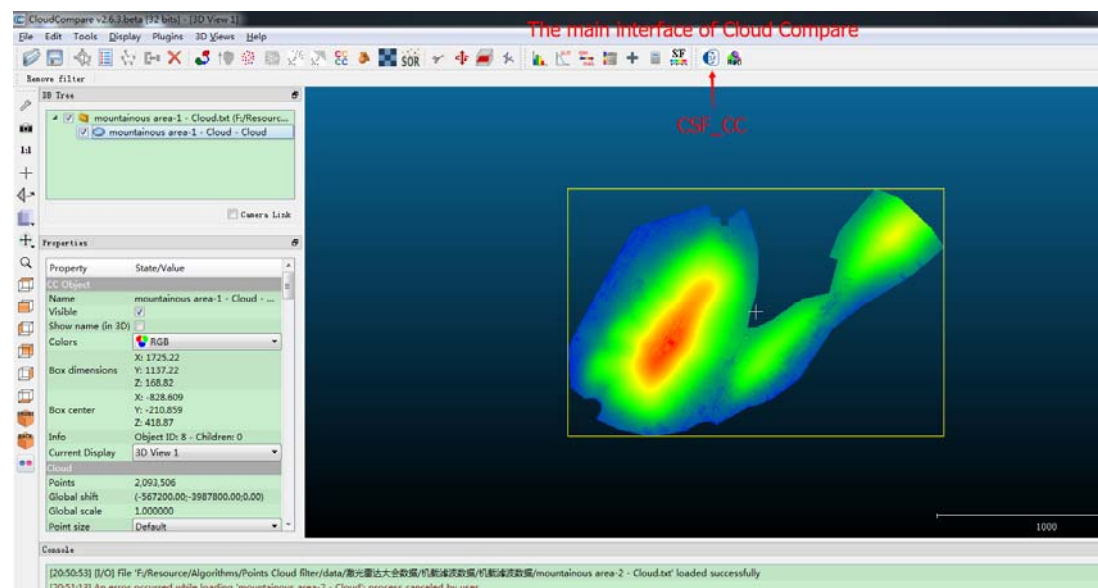

Fig. 4. Post-processing of the steep slope.

3 GUI Details

Graphic User Interface

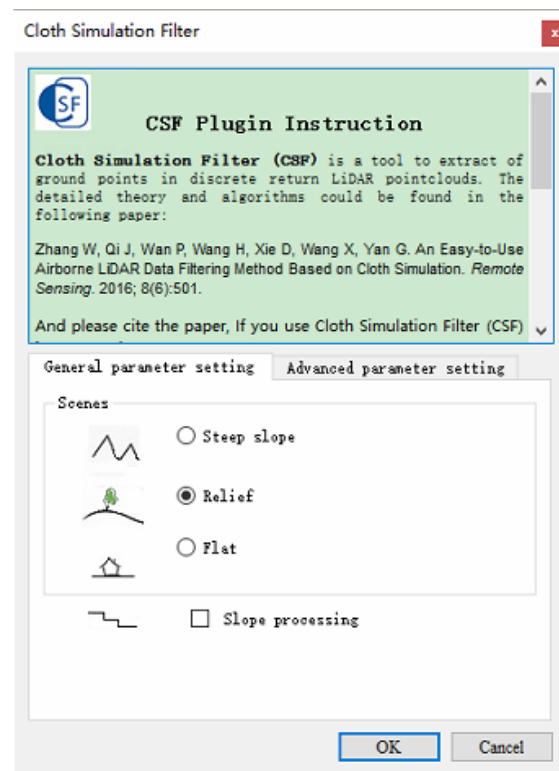Here is a quick overview of the main user interface:

The detailed information of Cloud Compare refers to the user manual:

http://www.cloudcompare.org/doc/qCC/CloudCompare%20v2.6.1%20-%20User%20manual.pdf

An overview of the CSF_CC interface

Set up of parameters

The parameters which need to be set by user can be divided into *General parameter* and *Advanced parameter*. The general parameter means that it must be set each time the program runs. The advance parameter means that it can be set according to the users need.
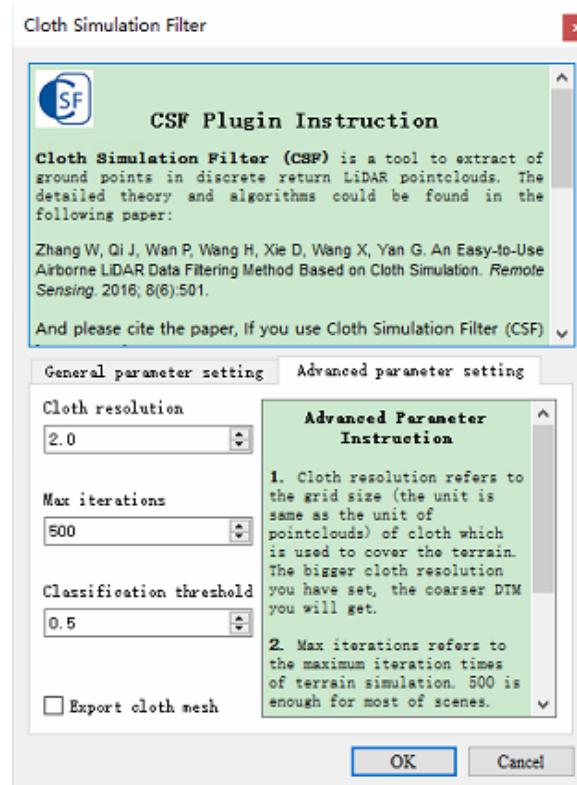
General parameters

※ Scenes

Three options are under this parameter : Mountain area with dense vegetation, complex scenes, and flat terrain with high-rise buildings. This parameter help users to set scenes type of the point clouds. When you set up this parameter, the rigidness will be determined actually. Please refers to the article: An Easy-to-Use Airborne LIDAR Data Filtering Method Based on Cloth Simulation.

※ Slope post processing for disconnected terrain

For steep slopes, this algorithm may yield relatively large errors because the simulated cloth is above the steep slopes and does not fit with the ground measurements very well due to the internal constraints among particles. This problem can be solved by selecting this option. If there are no steep slopes in your scenes, just neglect it.

Advanced parameters

※ Cloth resolution

Cloth resolution refers to the grid size of cloth which is use to cover the terrain. The bigger cloth resolution you have set, the coarser DTM you will get.

※ Max iterations

Max iterations refers to the maximum iteration times of terrain simulation. 500 is enough for most of scenes.

※ Classification threshold

Classification threshold refers to a threshold to classify the original point cloud into ground and non-ground parts based on the distances between original point cloud and the simulated terrain. 0.5 is adapted to most of scenes.

BibTex

ABSTRACT = {Separating point clouds into ground and non-ground measurements is an essential step to generate digital terrain models (DTMs) from airborne LiDAR (light detection and ranging) data. However, most filtering algorithms need to carefully set up a number of complicated parameters to achieve high accuracy. In this paper, we present a new filtering method which only needs a few easy-to-set integer and Boolean parameters. Within the proposed approach, a LiDAR point cloud is inverted, and then a rigid cloth is used to cover the inverted surface. By analyzing the interactions between the cloth nodes and the corresponding LiDAR points, the locations of the cloth nodes can be determined to generate an approximation of the ground surface. Finally, the ground points can be extracted from the LiDAR point cloud by comparing the original LiDAR points and the generated surface. Benchmark datasets provided by ISPRS (International Society for Photogrammetry and Remote Sensing) working Group III/3 are used to validate the proposed filtering method, and the experimental results yield an average total error of 4.58%, which is comparable with most of the state-of-the-art filtering algorithms. The proposed easy-to-use filtering method may help the users without much experience to use LiDAR data and related technology in their own applications more easily.},