

Package ‘RMCC’

July 21, 2025

Type Package

Title Airborne LiDAR Filtering Method Based on Multiscale Curvature

Version 0.1.1

Description Multiscale Curvature Classification of ground returns in 3-D LiDAR point clouds, designed for forested environments. 'RMCC' is a porting to R of the 'MCC-lidar' method by Evans and Hudak (2007) <[doi:10.1109/TGRS.2006.890412](https://doi.org/10.1109/TGRS.2006.890412)>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

LinkingTo Rcpp, BH

Imports Rcpp

Depends R (>= 2.10)

Suggests tinytest

NeedsCompilation yes

Author Jean-Romain Roussel [aut, cre],
Evans Jeffrey S [aut],
Hudak Andrew T [aut],
Green Code LLC [cph] (Author of the mcc code under US federally funded contract),
Jarno Elonen [cph] (Author of the included thin plate spline library),
Thomas Becker [cph] (Author of the included any_iterator library)

Maintainer Jean-Romain Roussel <jean-romain.rousseau@ulaval.ca>

Repository CRAN

Date/Publication 2024-07-04 06:30:02 UTC

Contents

MCC	2
rmcc_cloud	3
Index	4

MCC	<i>Airborne LiDAR filtering method based on Multiscale Curvature Classification</i>
-----	---

Description

Airborne LiDAR filtering method of ground points based on Multiscale Curvature Classification (Evans and Hudak, 2017; see references). This function is an R wrapper around the library written by the original authors of the algorithm.

Usage

```
MCC(cloud, s = 1.5, t = 0.3)
```

Arguments

<code>cloud</code>	data.frame with 3 columns containing the coordinates of the point cloud.
<code>s</code>	numeric. Scale parameter. The optimal scale parameter is a function of the scale of the objects (e.g., trees) on the ground, and the sampling interval (post spacing) of the lidar data.
<code>t</code>	numeric. Curvature threshold

Details

There are two parameters that the user must define to run MCC, the scale (`s`) parameter and the curvature threshold (`t`). The optimal scale parameter is a function of 1) the scale of the objects (e.g., trees) on the ground, and 2) the sampling interval (post spacing) of the lidar data. Current lidar sensors are capable of collecting high density data (e.g., 8 pulses/m²) that translate to a spatial sampling frequency (post spacing) of 0.35 m/pulse ($1/\sqrt{8 \text{ pulses/m}^2} = 0.35 \text{ m/pulse}$), which is small relative to the size of mature trees and will oversample larger trees (i.e., nominally multiple returns/tree). Sparser lidar data (e.g., 0.25 pulses/m²) translate to a post spacing of 2.0 m/pulse ($1/\sqrt{0.25 \text{ pulses/m}^2} = 2.0 \text{ m/pulse}$), which would undersample trees and fail to sample some smaller trees (i.e., nominally <1 return/tree).

Therefore, a bit of trial and error is warranted to determine the best scale and curvature parameters to use. Select a las tile containing a good variety of canopy and terrain conditions, as it's likely the parameters that work best there will be applicable to the rest of your project area tiles. As a starting point: if the scale (post spacing) of the lidar survey is 1.5 m, then try 1.5. Try varying it up or down by 0.5 m increments to see if it produces a more desirable digital terrain model (DTM) interpolated from the classified ground returns in the output file. Use units that match the units of the lidar data. For example, if your lidar data were delivered in units of feet with a post spacing of 3 ft, set the scale parameter to 3, then try varying it up or down by 1 ft increments and compare the resulting interpolated DTMs. If the trees are large, then it's likely that a scale parameter of 1 m (3 ft) will produce a cleaner DTM than a scale parameter of 0.3 m (1 ft), even if the pulse density is 0.3 m (1 ft). As for the curvature threshold, a good starting value to try might be 0.3 (if data are in meters; 1 if data are in feet), and then try varying this up or down by 0.1 m increments (if data are in meters; 0.3 if data are in feet).

Value

An integer vector containing the ids of the points that belong on the ground.

References

Evans, Jeffrey S.; Hudak, Andrew T. 2007. A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments. IEEE Transactions on Geoscience and Remote Sensing. 45(4): 1029-1038.

Examples

```
data(rmcc_cloud)
head(rmcc_cloud)

id_ground = MCC(rmcc_cloud, s = 3.5, t = 0.3)
```

rmcc_cloud	<i>Airborne LiDAR point cloud</i>
------------	-----------------------------------

Description

A dataset containing a small point cloud acquired with airborne LiDAR.

Usage

```
rmcc_cloud
```

Format

A data frame with 28668 rows and 3 variables:

X x coordinates

Y y coordinates

Z z coordinates

Index

* **datasets**

rmcc_cloud, [3](#)

MCC, [2](#)

rmcc_cloud, [3](#)