

BranchFallChangeDetection

Branch-level structural change detection using terrestrial/mobile LiDAR, ADTree QSM reconstruction, rTwig correction, and Python-based branch comparison.

A small data sample is provided in [/data/](#), containing the same tree scanned at two epochs (2022 and 2023).

Requirements

1. Python Environment

Python 3.10+

Using Conda:

```
conda env create -f environment.yml
conda activate branchChange
```

Or manual setup:

```
conda create -n branchChange python=3.11
conda activate branchChange
pip install numpy pandas scipy open3d scikit-learn matplotlib
```

2. ADTree

Download ADTree (v1.1.2 used in this workflow):

<https://github.com/hanliangzhang/AdTree>

Place [AdTree.exe](#) locally and update the path inside:

[/r/adTree.R](#)

3. R Dependencies

Install in R:

```
install.packages("rTwig")
install.packages("dplyr")
install.packages("R.utils")
install.packages("R6")
```

Input Requirements

File Naming Convention

Point clouds must be `.xyz` files named:

`arbre YYYY ID HH LATIN-SPECIES.xyz`

Example:

`arbre 2022 03 12.54 ACER-PLATANOIDES.xyz`

Where:

- YYYY = acquisition year
- ID = tree identifier (must match across years)
- LATIN-SPECIES = genus-species with dash separator

No headers. XYZ columns only. Units must be meters.

Folder Structure

Separate pre and post scans:

```
data/
├── 2022/    ← source (pre)
│   ├── arbre 2022 03 12.54 ACER-PLATANOIDES.xyz
├── 2023/    ← target (post)
│   ├── arbre 2023 03 12.88 ACER-PLATANOIDES.xyz
```

Tree IDs must correspond between years.

Workflow

Step 1 — Generate ADTree QSMs

In:

`r/adTree.R`

run :

`runADTreeDirectory()`

This generates raw QSM outputs from `.xyz` files. Make sure to change the directory inputs and outputs

Step 2 — Apply rTwig Corrections

In:

```
r/rTwigClass.R
```

run :

```
rTwigBatchCorrector$new()
```

and then

```
bc$run
```

This:

- Imports ADTree output
 - Applies `correct_radii()`
 - Runs `update_cylinders()`
 - Writes:
 - `cylinders_corrected.csv`
 - `branches_summary_corrected.csv`
-

Step 3 — Detect Branch-Level Change

In:

```
branchrunner.py
```

Run:

```
BranchesChange()
```

and then:

```
bc.run()
```

This:

- Matches pre vs post trees
 - Performs voxel change detection
 - Classifies branches (intact / fallen)
 - Writes:
 - `fallen_branches.csv`
 - `intact_branches.csv`
 - `tree_level.csv`
 - `failures.csv`
-

Credits and Citation

This package originates from:

Karl Montalban et al.

Exploring the impact of ice storm on urban forests and branch fall using mobile LiDAR

Urban Forestry & Urban Greening, 2025.