Algorithm Simulation

Algorithm Simulation....1
Program Specification....2
Inputs:....2
Processes:....2
Outputs:....2
Design Choices and Rationale....3
Overview....3
Functional Programming....3
Class Overview....4
Structural Overview....5

Program Specification

Inputs:

- Configuration file path
- Input frame path(s)
- Output directory path
-
- Configuration data:
  - Number of steps per frame
  - Start value for each parameter
  - Stop value for each parameter
  - Default value for each parameter

Processes:

- Read and parse configuration file
- Read input frames (point clouds)
- Simulate each parameter and record results
- Convert results to .csv format

Outputs:

- .csv output file containing the following averages for each parameter:
  - (total points removed) / (total points)
  - (cone points removed) / (total points)
  - (ground points removed) / (total points)
  - (ground points removed) / (number of ground points)

Design Choices and Rationale

Overview

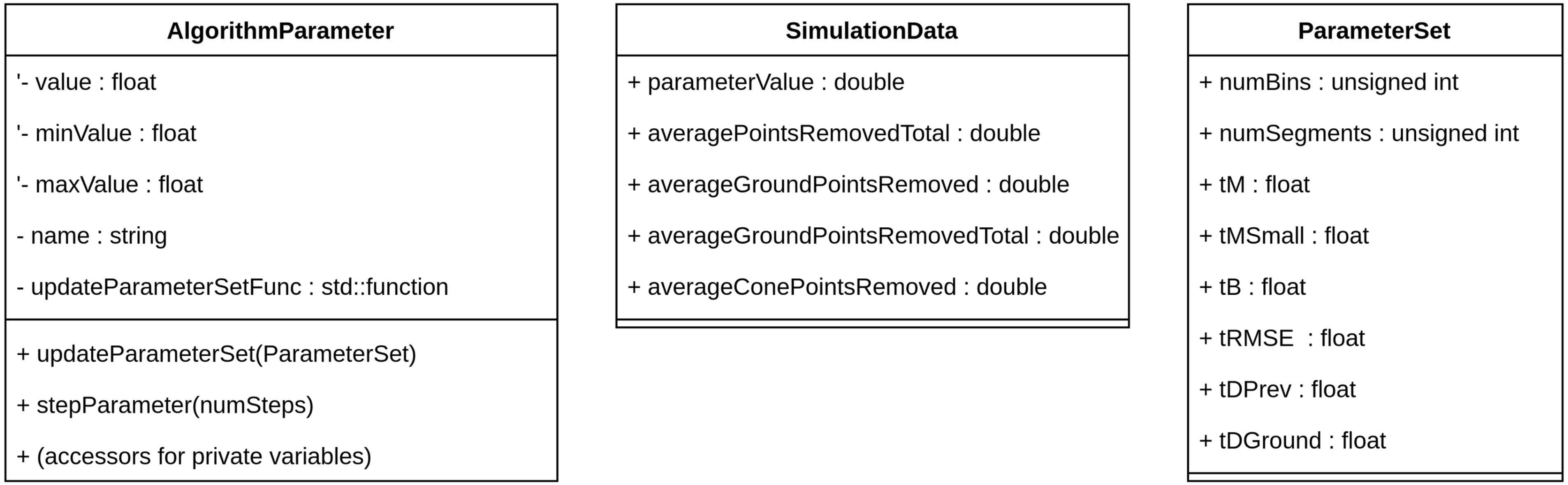
The “Algorithm Simulation” program is a data-collection program which runs the Ground Removal algorithm on point clouds and records its performance. The aim of the program is to collect adequate data about the behaviour of the Ground Removal algorithm and how its performance is affected by a range of parameter values. Ultimately, by running successive simulations on the ground removal algorithm, a parameter-set which best meets our needs can be discovered and used in the LiDAR pipeline.

Functional Programming

A functional design pattern is used in the implementation of the “Algorithm Simulation” program. This was done as the program can be described as a series of specific tasks which run sequentially and then terminate. It was therefore ideal to take a top-down approach in the program design so that each function can be implemented and tested independently. Method abstraction has also been implemented in the form of functional programming. Instead of making use of inheritance to allow the program to specify how each AlgorithmParameter alters a ParameterSet, which would require a large amount of boiler-plate code and headers, we use std::functions to provide the same polymorphism from one class without inheritance.

Class Overview

Header	Namespace	Name	Description
parameter_set.hpp	algorithm_simulation	ParameterSet	A record structure representing each element of the ground removal that is considered a “parameter”. All parameters should be variable and used in the ground removal process. This means that a ParameterSet should directly translate to one specific execution of the ground removal algorithm.
algorithm_parameter.hpp	algorithm_simulation	AlgorithmParameter	A class representing a specific parameter used in the algorithm. The parameter holds a certain value as well as minimum and maximum bounds. The class provides functionality to step through the parameter. Each class instance must also provide a function which describes how a ParameterSet object shall be updated with the value of the AlgorithmParameter instance.
simulation_data.hpp	algorithm_simulation	SimulationData	A record structure which holds all of the data recorded after a parameter has been simulated with a specific value. Each SimulationData class is directly translated to one row in a .csv file.



AlgorithmParameter

'- value : float

'- minValue : float

'- maxValue : float

- name : string

- updateParameterSetFunc : std::function

+ updateParameterSet(ParameterSet)

+ stepParameter(numSteps)

+ (accessors for private variables)

SimulationData

+ parameterValue : double

+ averagePointsRemovedTotal : double

+ averageGroundPointsRemoved : double

+ averageGroundPointsRemovedTotal : double

+ averageConePointsRemoved : double

ParameterSet

+ numBins : unsigned int

+ numSegments : unsigned int

+ tM : float

+ tMSmall : float

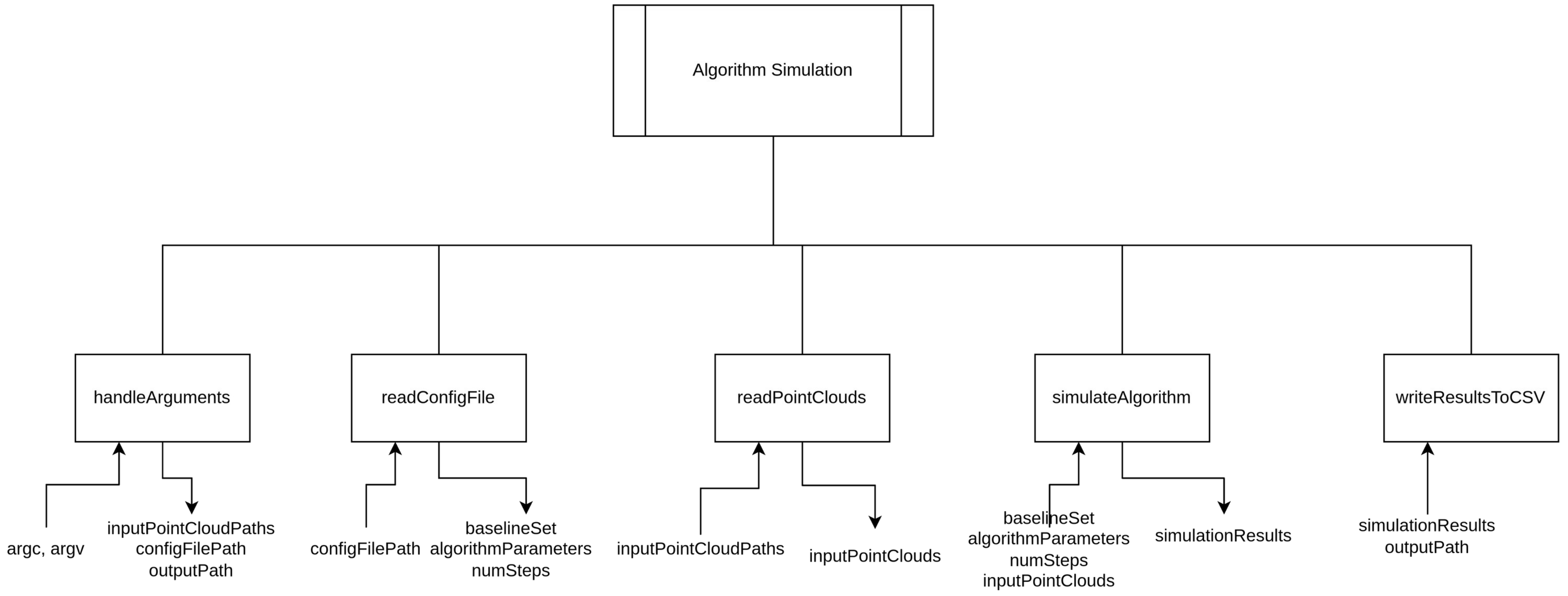
+ tB : float

+ tRMSE : float

+ tDPrev : float

+ tDGround : float

Structural Overview


Algorithm Simulation
handleArguments
readConfigFile
readPointClouds
simulateAlgorithm
writeResultsToCSV
argc, argv
inputPointCloudPaths
configFilePath
outputPath
configFilePath
baselineSet
algorithmParameters
numSteps
inputPointCloudPaths
inputPointClouds
baselineSet
algorithmParameters
numSteps
inputPointClouds
simulationResults
simulationResults
outputPath