UCD School of Computer Science and Informatics
Scoil na Ríomheolaíochta agus na Faisnéisíochta UCD

Toward a new approach for massive LiDAR data processing

V-H. Cao[#1], K-X. Chu[#2], N-A. Le-Khac[#3], M-T. Kechadi[#3], D. Laefer[*3], L. Truong-Hong[*3]

[#]*School of Computer Science & Informatics, University College Dublin*
Belfield, Dublin 4, Ireland

[1]van-hung.cao@ucdconnect.ie

[2]chuxuankhoi@gmail.com

[*] *School Of Civil, Structural & Environment Engineering, University College Dublin*
Belfield, Dublin 4, Ireland

[3]{an.lekhac,tahar.kechadi,debra.laefer,linh.truonghong}@ucd.ie

Agenda

1. Introduction
2. Background
3. Comparative Study
   A. Comparity state-of-the-art approaches in HPC for LiDAR processing
   B. Software study
4. Toward a new approach for processing very large LiDAR data
   A. Loading data
   B. Building data structure

Introduction

- Laser scanning (also known as Light Detection And Ranging) has been widely applied in various applications
- Recently, an aerial laser scanning (ALS) has a scan rate of 1MHz and Full Waveform Digitizer (FWD) collection at up 120 kHz, where ALS data consist hundreds of millions of 3D point clouds associated with waveform data of laser pulses.
- The huge volumes and complexity of ALS data are to be great challenges for data processing as the limitation of the computing hardware.

Introduction (cont)

- In this paper, we firstly present a comparative study of software libraries and algorithms to optimise the processing of LiDAR data.
- We propose new method to improve this process with experiments on large LiDAR data.
- Finally, we discuss on a parallel solution of our approach where we integrate parallel/cloud computing in processing LiDAR data.

Background

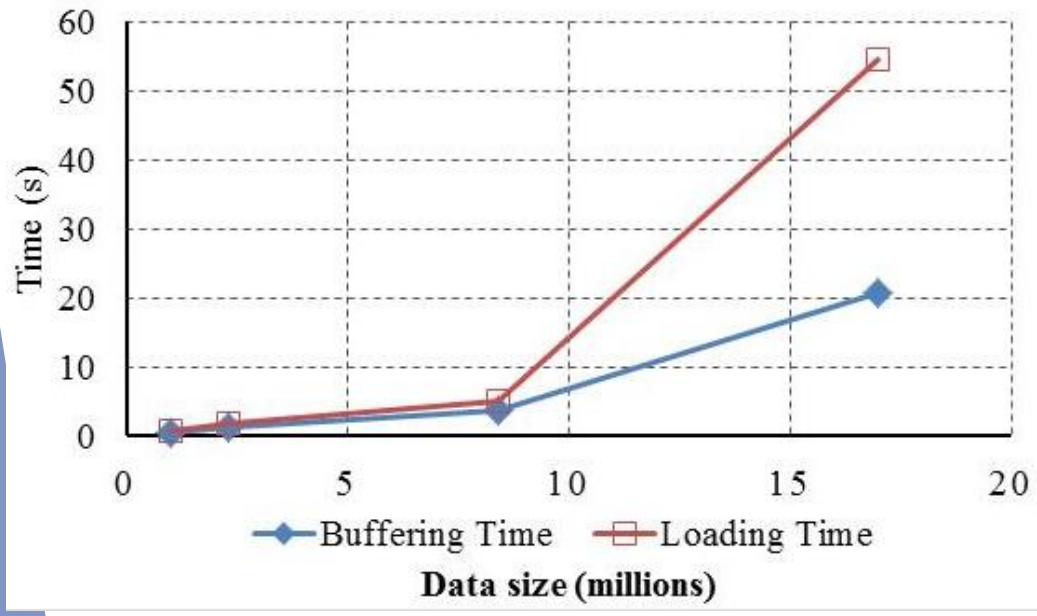
- Our long term goal is to develop efficient algorithm based on high performance computational (HPC) resources for classifying ALS data points into separate categories and for extracting the point cloud of separate objects.
- The irrespective classification process:
  - slope-based
  - cluster/segmentation-based
  - surface based
  - morphological filter
- The segmentation process:
  - model fitting-based methods
  - region growing-based methods
  - clustering feature based methods
- The nearest neighbour search plays out an important role in controlling the performance of the algorithm

Comparative study

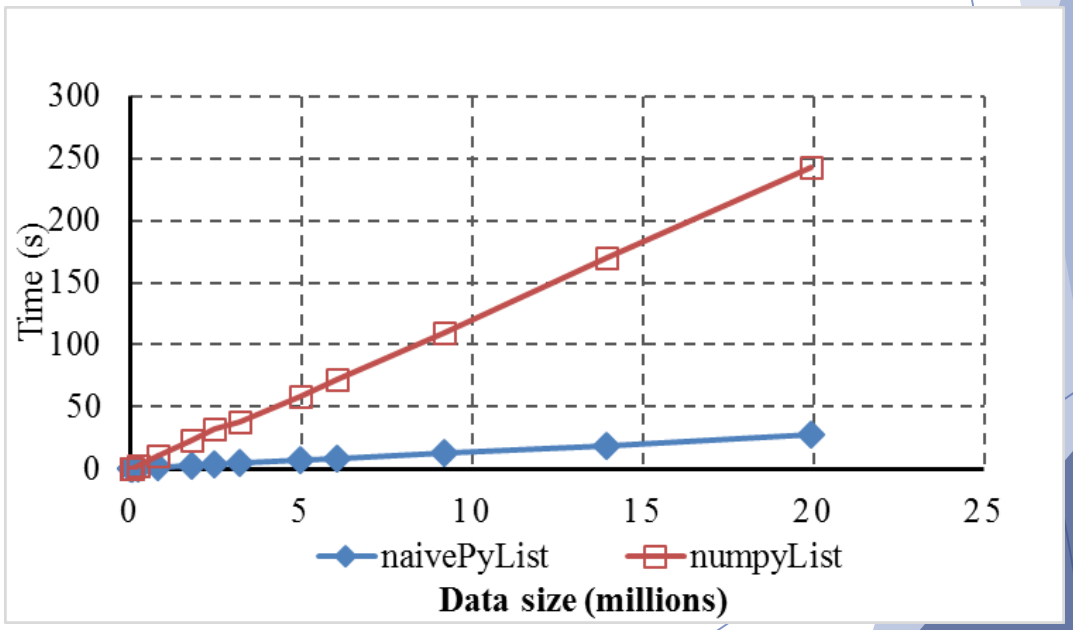
- In this part, we will take a look at some current approaches in HPC for LiDAR data processing
- In particular, the HPC-based paradigms in this part comprise
  - Hardware systems
    - Multi-core CPU architecture
    - Graphic processing units (GPUs), and general-purpose computing on graphics processing units (GPGPUs).
  - PC Cluster
  - Field programmable gate array (FPGAs)
  - Cloud computing environments

Comparative study

Python language

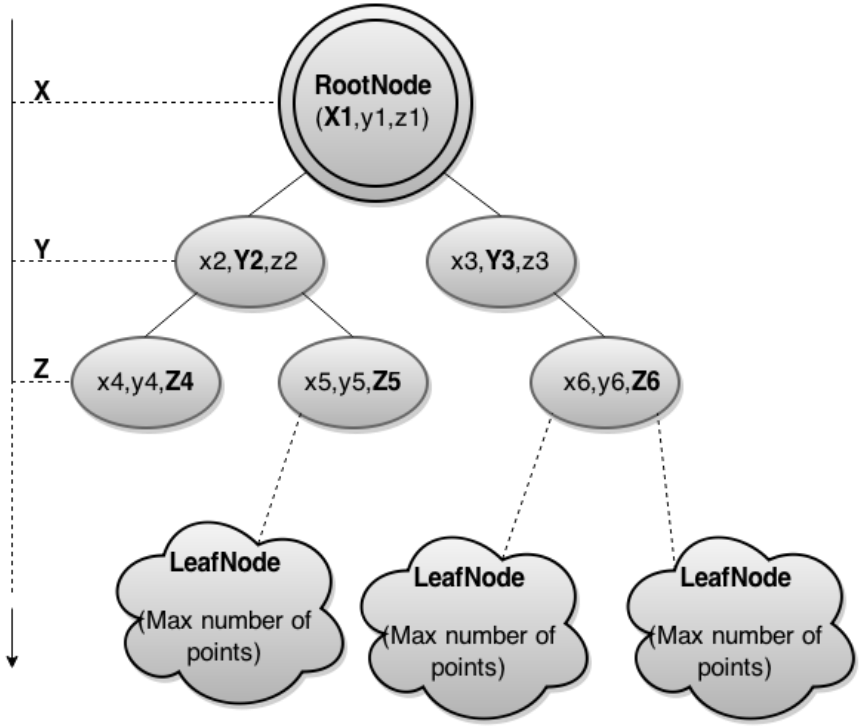


Libraries



Toward a new approach for processing very large LiDAR data

- Loading data
  - Vectorising
  - Multi-threading
- Application of data structure



Toward a new approach for processing very large LiDAR data

Algorithm 1 Optimal kdContruct

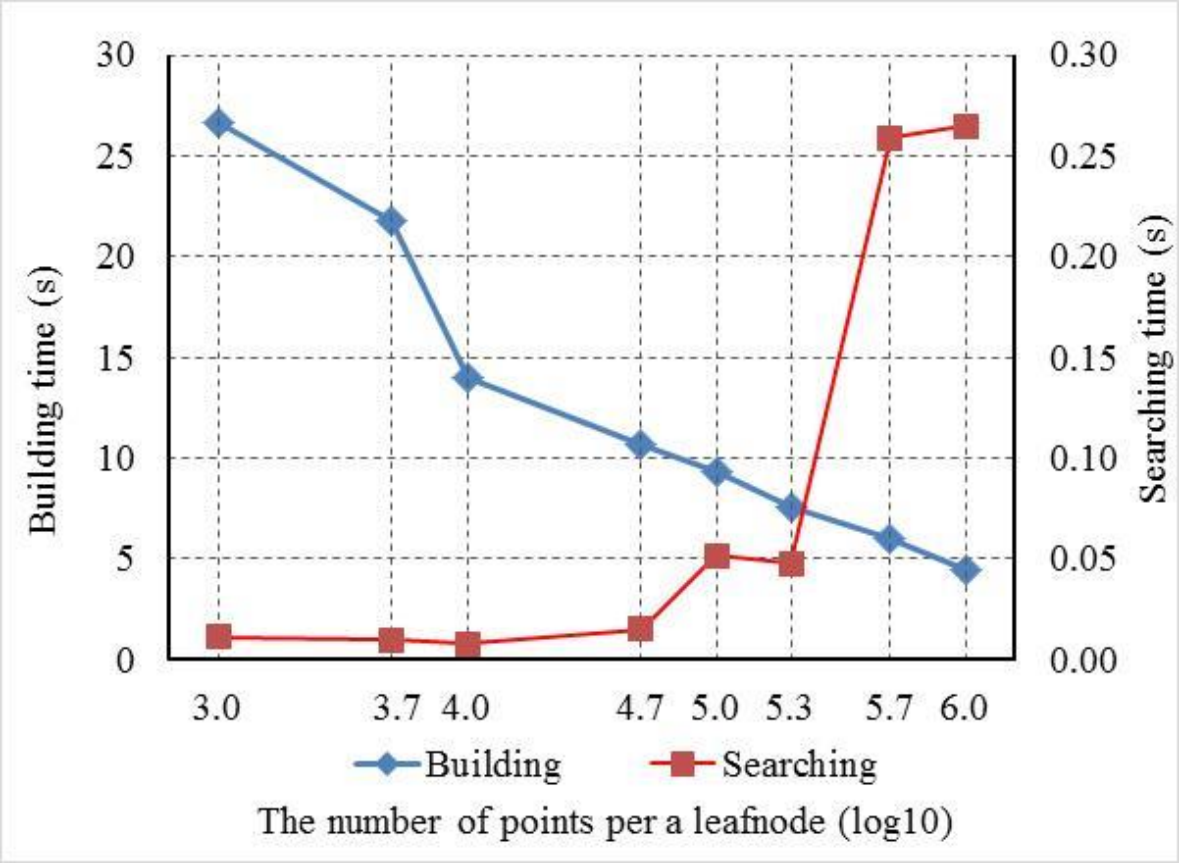
1: **procedure** kdConstruct (*trainingSet*)
2: **if** *trainingSet.size*() <= Leafsize **then**
3: **return** *kdtree.leafnode* // Returns a kdTree
4: **else**
5: (*s, val*) ← chooseSplit(*trainingSet*) // *s* is splitting
dimension, chooseSplit function based on sliding midpoint rudes
6: *trainLeft* ← {*x* ∈ *trainingSet* : *xs* < *val* }
7: *trainRight* ← {*x* ∈ *trainingSet* : *xs* ≥ *val* }
8: *kdLeft* ← kdConstruct(*trainLeft*)
9: *kdRight* ← kdConstruct(*trainRight*)
10: **return** *kdtree*(*s, val, kdLeft, kdRight*)
11: **end procedure**

Algorithm 2 Brute-force

1: c ← first(P) // generate a first candidate solution for P
2: **while** c < > Λ do
3: **if** valid(P,c) **then** output(P, c) // check whether candidate c is a solution for P then return output c
4: c ← next(P,c) // generate the next candidate for P after the current one c
5: **end while**

Toward a new approach for processing very large LiDAR data

- Dataset: approximately 18 million points.
- Testing platform: Intel Core i7-3517U 1.9 GHz Processor, 2 GB DDR3 RAM, 256 GB Solid State Drive, Windows 8.1.

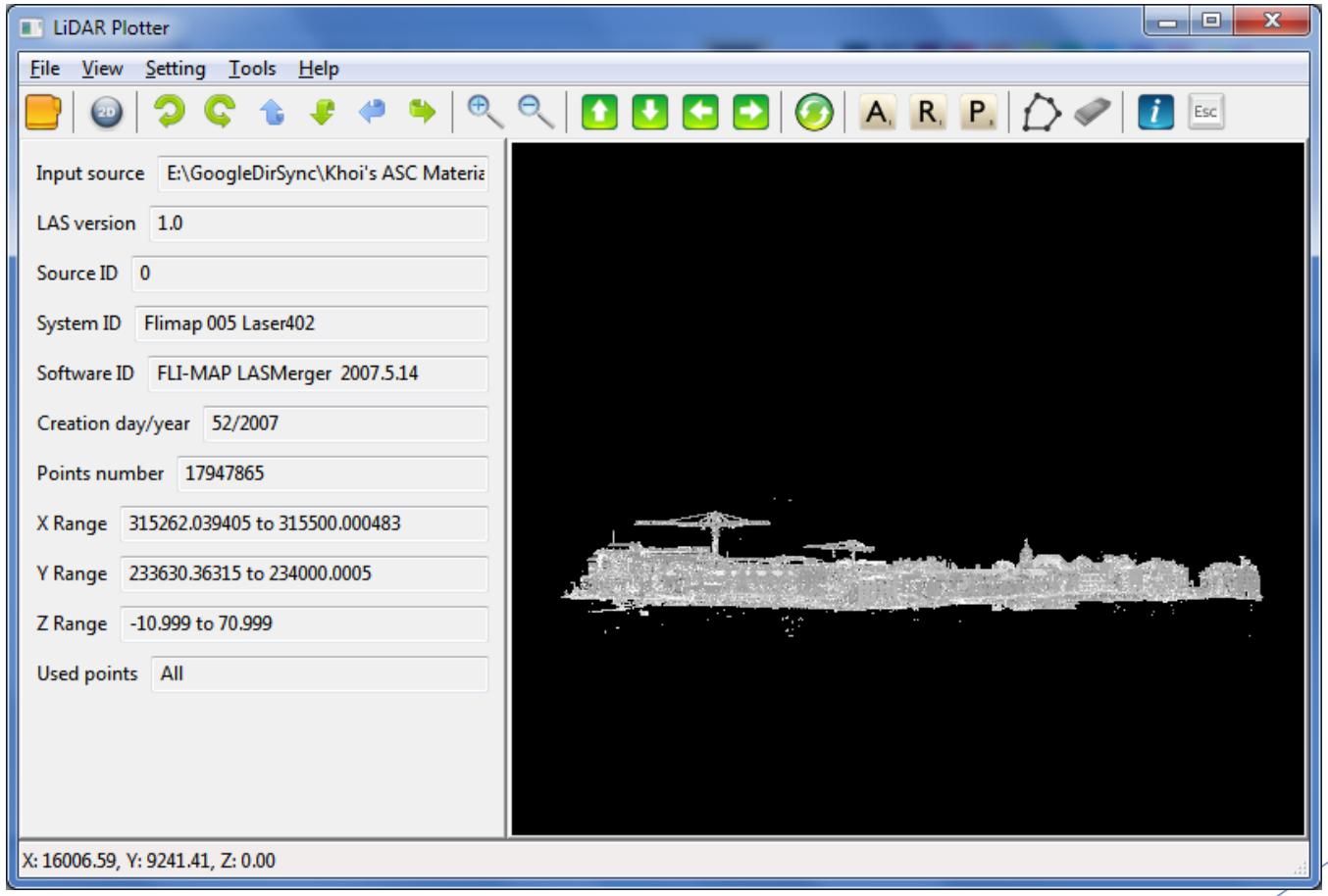


Toward a new approach for processing very large LiDAR data

Algorithm 3 Par-Optimal kdContruct

1: **procedure** kdConstruct (*trainingSet*, *TpNode*)
2: **if** *trainingSet.size*() <= Leafsize **then**
3: **return** *kdtree.leafnode* // Returns a kdTree
4: **else**
5: (*s*, *val*) ← chooseSplit(*trainingSet*) // *s* is splitting dimension, chooseSplit function based on sliding midpoint rudes
6: *trainLeft* ← {*x* ∈ *trainingSet* : *xs* < *val* }
7: *trainRight* ← {*x* ∈ *trainingSet* : *xs* ≥ *val* }
8: *kdLeft* ← kdConstruct(*trainLeft*, *TpNode.left*)
9: *kdRight* ← kdConstruct(*trainRight*, *TpNote.right*)
10: **return** *kdtree*(*s*, *val*, *kdLeft*, *kdRight*)
11: **end procedure**

Toward a new approach for processing very large LiDAR data



Conclusion and Future work

- In this paper, we conduct comparative studies of existing libraries and methods to determine the key issues that affect the performance of LiDAR processing.
- We also propose a new strategy for ALS data processing.
- We describe moreover the ability of improving the performance of our approach by integrating parallel computing based on an efficient network topology *TreeP*.
- Experimental results of parallel approach for both *kd-tree* construction and brute-force searching with very large size of LiDAR data are also being produced.
- These results will allow us to test and evaluate the robustness of our approach.


danke
謝謝
ngiyabonga
teşekkür ederim
спасибо
dank je
thank you
gracias
tapadh leat
hvala
mauruuru
bedankt
dziękuję
mochchakkeram
go raibh maith agat
obrigado
sagolun
sukriya
kop khun krap
arigatō
takk
dakujem
grazie
мерси
terima kasih
감사합니다
ευχαριστώ
merci