CLASSIFICATION AND INTEGRATION OF MASSIVE 3D POINTS CLOUDS IN A VIRTUAL REALITY (VR) ENVIRONMENT

Abderrazzaq Kharroubi¹, Rafika Hajji¹, Roland Billen², Florent Poux²

¹ Geomatics and Surveying Engineering school IAV Hassan 2
² Geomatics Unit, University of Liège
SUMMARY

1. CONTEXT AND PROBLEMATIC
2. WORKFLOW
3. POINT CLOUD PROCESSING
   1. Segmentation
   2. Classification
   3. Structuration
4. VR APPLICATION & PERFORMANCES
5. DISCUSSION & PERSPECTIVE
**CONTEXT**

**Acquisition tools**

**Massive point cloud**

**Processing algorithms**

**New visualisation modes**

**Computing power**

---

6th International Workshop, LowCost 3D - Sensors, Algorithms, Applications

INSA de Strasbourg, France, 02-03 December
PROBLEMATIC

1. Discrete sets of spatial data

2. Requirement for storage and processing capacities

3. Non-realistic aspect of point in VR
STEPS

Acquisition & preprocessing

Segmentation

Classification

Interface development

Implementation with Unity

Structuration in Potree’s format

VR application with Oculus

Performance test

Validation

C#
## SEGMENTATION

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of points</th>
<th>Attribute</th>
<th>Sensor</th>
<th>Size Go</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHÂTEAU_JEHAY</td>
<td>2,300,247,428</td>
<td>RGB, intensity</td>
<td>Leica P30</td>
<td>69,636</td>
</tr>
<tr>
<td>PCID10_RTWH_Exterior</td>
<td>312,710,687</td>
<td>RGB, intensity</td>
<td>TLS</td>
<td>4,907</td>
</tr>
<tr>
<td>PCID11_RTWH_CHAIR</td>
<td>259,101,028</td>
<td>RGB, intensity</td>
<td>TLS</td>
<td>4,807</td>
</tr>
<tr>
<td>PCID2_ULG_B5a</td>
<td>115,190,236</td>
<td>RGB, intensity</td>
<td>TLS</td>
<td>3,824</td>
</tr>
<tr>
<td>PCID8_NAAVIS_1</td>
<td>44,847,540</td>
<td>RGB</td>
<td>NavVIS</td>
<td>0,657</td>
</tr>
<tr>
<td>PCID6_REVO</td>
<td>53,800,194</td>
<td>Without</td>
<td>REVO</td>
<td>0,630</td>
</tr>
<tr>
<td>PCID9_NAAVIS_2</td>
<td>4,244,416</td>
<td>RGB</td>
<td>NavVIS</td>
<td>0,062</td>
</tr>
</tbody>
</table>
1. Manual segmentation

2. Label connected components

3. CANUPO
4. Histogram filtering

5. Cloth Simulation Filter

6. RANSAC
## Classification

<table>
<thead>
<tr>
<th>Indoor</th>
<th>Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Floor</td>
</tr>
<tr>
<td>1</td>
<td>Ceiling</td>
</tr>
<tr>
<td>2</td>
<td>Wall</td>
</tr>
<tr>
<td>3</td>
<td>Beam</td>
</tr>
<tr>
<td>4</td>
<td>Column</td>
</tr>
<tr>
<td>5</td>
<td>Window</td>
</tr>
<tr>
<td>6</td>
<td>Door</td>
</tr>
<tr>
<td>7</td>
<td>Table</td>
</tr>
<tr>
<td>11</td>
<td>Board</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>30</td>
<td>road_Sign</td>
</tr>
<tr>
<td>31</td>
<td>advertisingBoard</td>
</tr>
<tr>
<td>32</td>
<td>banc</td>
</tr>
<tr>
<td>33</td>
<td>bicycle</td>
</tr>
<tr>
<td>34</td>
<td>bicycleStation</td>
</tr>
<tr>
<td>35</td>
<td>Building_facades</td>
</tr>
<tr>
<td>36</td>
<td>busStation</td>
</tr>
<tr>
<td>37</td>
<td>car</td>
</tr>
<tr>
<td>41</td>
<td>Humains</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Format .LAS, ou bien .LAZ
Format specification .LAS version 1.2

Example after classification
Château de Jehay
PCID10_RTWH_Exterior
PCID11_RTWH_CHAIR
PCID2_ULG_B5a
PCID8_NAAVIS_1
PCID6_REVO
PCID9_NAAVIS_2

6th International Workshop, LowCost 3D - Sensors, Algorithms, Applications
INSA de Strasbourg, France, 02-03 December
• Hierarchical multi-resolution structure
• Input-format .txt, .ply, .las, .laz
• Output-format .bin, .las, .laz
The loading process uses three threads: the Unity main thread, a traversing thread, and a loading thread. In the main thread, visible GameObjects are updated once per image if any necessary changes have been detected in the traversing thread.

Game objects are created for Octree nodes that should be visible and do not yet have game objects, nodes that should no longer be visible have their game objects removed.

Determining which node the game objects should be created or deleted is the job of the traversing thread.

The loading thread is used to load point data from files,
Improved visual quality

Shaders en Cg

Square  Circle  Variable size

Interpolation

Cone  Paraboloid
Improved visual quality
VR APPLICATION

Easy interaction

1. Point size   2. Point type   3. Type of interpolation
4. Attribute type   5. The classes to visualize

3D immersion in VR

Oculus Headset   2 touch   2 Capteurs

6th International Workshop, LowCost 3D - Sensors, Algorithms, Applications
VR APPLICATION
The variation of the FPS number according to the number of points loaded, JEHAY with 2.3 billion.

### Environnement et tests

**Processor** | Intel® Core™ i6-6800K CPU @ 3.40GHz 3.40 GHz
---|---
**Graphic card** | NVIDIA GeForce GTX 1080
**RAM** | 48.0 Go
**Operating system** | Windows 10 Pro, 64 bits

**Settings to test**
- FPS
- Consumption in memory

**Settings that influence**
- Point size
- Number of loaded points
- Number of points cached
- Type of interpolation
Variation of the memory consumption (in Mo) according to the size of the point cloud loaded and the LRU cache.

Influence of the variation of the point size on the FPS, JEHAY with a budget of 2 Million, interpolation = paraboloid, LRU = 2M.
Development of semantic segmentation process to automate the enhancement of point cloud with class information.

Creation of a dual spatial and classification indexing to make possible querying and direct interaction with the point cloud in the VR environment.

Implementation of the Continuous Level of Detail (Schütz et al., 2019) rendering method.
Thank you for your attention