Reading a PTX file into an itkMesh

Release 0.00

David Doria

September 1, 2009

Rensselaer Polytechnic Institute, Troy NY

Abstract

This document presents a set of classes to read a PTX file (a common file format produced by LiDAR scanners) into an itkMesh object. We propose these classes as addition to the Insight Toolkit ITK `www.itk.org`.

Contents

A LiDAR scanner acquires points that compose a 2D surface embedded in 3D. The Leica HDS3000 Scanner Acquires points in a grid, from bottom to top, left to right, as shown in 1. Using the fact that the order the points were acquired in is known, we can create a mesh from these points using the grid 4-connectivity.

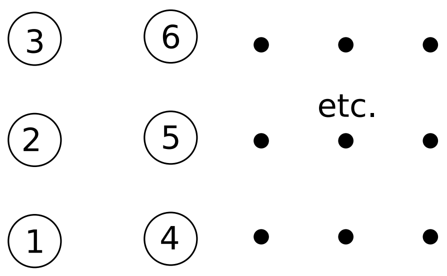

Figure 1: Order of point acquisition

1 PTX File Format

The PTX file format contains implicit information about the structure of the points. The header specifies the grid size and scanner orientation, and the rest of the file contains the point coordinates, their reflection intensity, and their color when mapped to the digital image that is taken at the same time as the scan.

```
Number of points per column
Number of points per row
X Y Z (Scanner location)
R3 R3 R3
R3 R3 R3
R3 R3 R3 (3x3 rotation matrix - typically identity)
R4 R4 R4 R4
R4 R4 R4 R4
R4 R4 R4 R4
R4 R4 R4 R4 (4x4 rotation matrix - typically identity)
X Y Z Intensity R G B (the coordinate of the point, the intensity of the reflection, and the c
X Y Z Intensity R G B
X Y Z Intensity R G B
....
```

An example PTX file is:

```
2
3
0 0 0
1 0 0
0 1 0
0 0 1
1 0 0 0
0 1 0 0
0 0 1 0
```

```
0 0 0 1
-6.123001 -8.470993 -1.720901 0.411292 67 59 63
-6.225357 -8.612289 -1.720413 0.451453 90 101 101
-6.320847 -8.744064 -1.717575 0.418433 72 69 74
-6.417191 -8.876999 -1.715958 0.414588 58 55 63
-6.521957 -9.021622 -1.713974 0.419165 55 51 58
-6.629074 -9.169510 -1.711929 0.413428 53 53 59
```

2 Storing the Points in an itkMesh

2.1 itkPTXPointInfo

We will use the MeshTraits idea of itkMesh to associate non-coordinate data with each mesh node. We created a simple class to store the data associated with each point.

```
class PTXPointInfo
{
public:
typedef itk::RGBPixel<unsigned char> ColorType; //note: the ptx format uses unsigned chars fo

ColorType Color;
double Intensity;

};
```

2.2 itkPTXMeshReader

This class simply has one function; Read(filename). The Read function is responsible for opening the ptx file (always in ascii mode), parsing the lines, and storing the point coordinates, intensity, and color information in an itkMesh and returning a pointer to the mesh. The Read function also creates the mesh connectivity based on the grid structure of the PTX file.

Creating Mesh Connectivity

The links are created from the current point to the point below it, as well as from the current point to the point left of it. This is done only if the point indeed has a neighbor in the corresponding direction (points that lie on the edge of the grid). An example of the order that links are created is shown in 2.

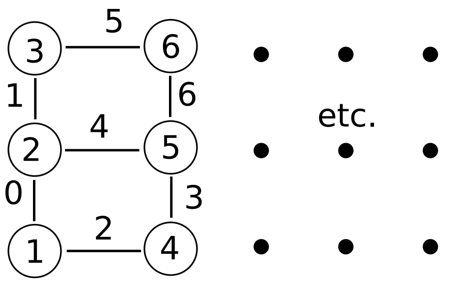

Figure 2: Order of Link Creation

3 Example Usage

The usage is very straight forward - simply create a PTXMeshReader object and then call its Read function with the path of the PTX file you wish to read.

```
std::string Filename = "test.ptx";

//create a mesh object that will be filled by the Read() function
itk::PTXMeshReader MeshReader;

//Read the PTX file and store it in mesh
itk::PTXMeshReader::MeshType::Pointer mesh = MeshReader.Read(strInputFilename);
```