***Python-based Tools for 3D Point Processing***

ESRI Database Services

|  |  |
| --- | --- |
| **Prepared by:** | Joe McGlinchy |
| **Last Modification Date:** | October 29, 2015 |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Plan Revision History** | | |
| **Effective Date** | **Description** | **Author** |
| 10/29/2015 | Initial version | Joe McGlinchy |
|  |  |  |
|  |  |  |

**Table of Contents**

[1.0 Purpose of this document 1](#_Toc433891007)

[2.0 Project Overview 1](#_Toc433891008)

[3.0 Strategic Objectives and outcome from R&D 5](#_Toc433891009)

[4.0 Resources and Schedule 6](#_Toc433891010)

[5.0 Risk Management 7](#_Toc433891011)

[6.0 Document and Source Control 7](#_Toc433891012)

[7.0 Quality Assurance and Testing 7](#_Toc433891013)

# 1.0 Purpose of this document

This R&D Project Plan Brief is prepared for special research and development projects. It summarizes the R&D project objectives, risks and desired outcomes. Any project done under the R&D project number **C96789** needs to use this project brief.

Note: This document is outside of ISO requirements and is not auditable.

# 2.0 Project Overview

For this project, we are proposing to create Python-based geoprocessing tools to process point data within building footprints in order to generate higher quality 3D models of those buildings. The tools we are proposing will accomplish the following tasks:

- Remove noise points within building footprint

- Generate mesh from remaining points

- Identify soft edges and planes in mesh

- Enrich point cloud by replacing soft edges with hard edges

- Enrich point cloud by replacing soft planes with hard planes

‘Higher’ quality 3D models, in this case, means higher quality than those shown in Figures 1-3. When a lidar system collects data over a building the general roof form is well represented but there tends to be noisey points along the surface of the roof. In the current processing workflow to create mutlipatch building models from lidar points, these noisey points contribute to a lower quality surface. The current process can be summarized as:

1. Clip lidar point cloud to building footprint
2. Convert points to point feature class
3. Process points to remove points on sides of buildings (noise)
4. Create TIN surface
5. Extrude footprints between TIN surface and ground

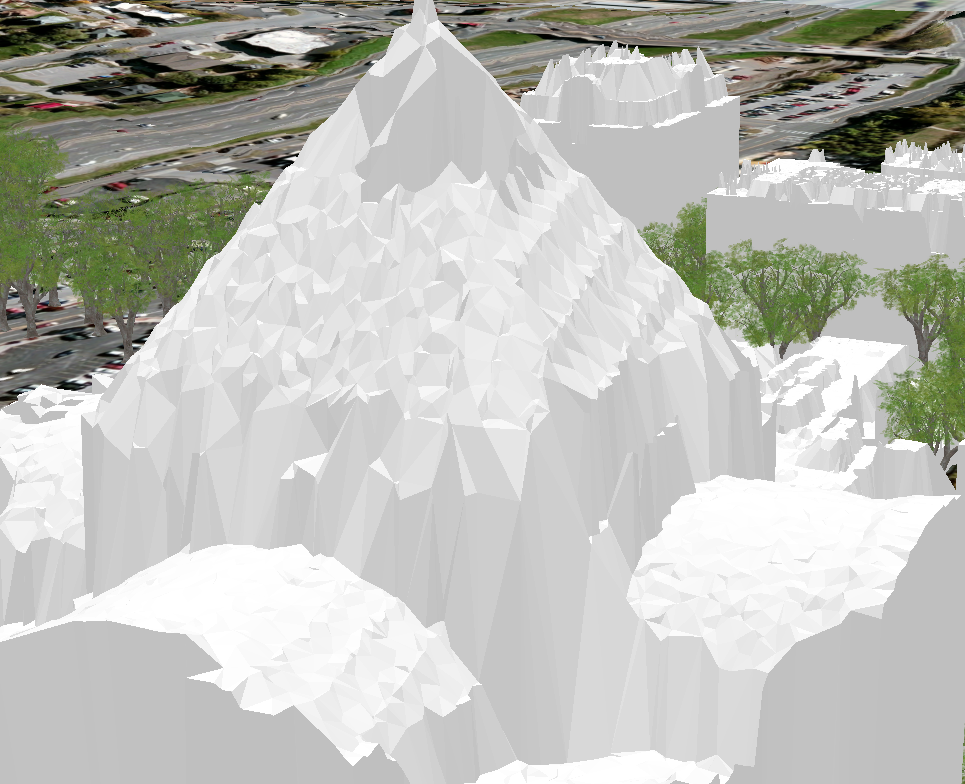


Figure Example of 'noise points' affecting quality of 3D Multipatch building model derived from lidar points.



Figure Another example of 'noise points' affecting quality of 3D Multipatch building model derived from lidar points.

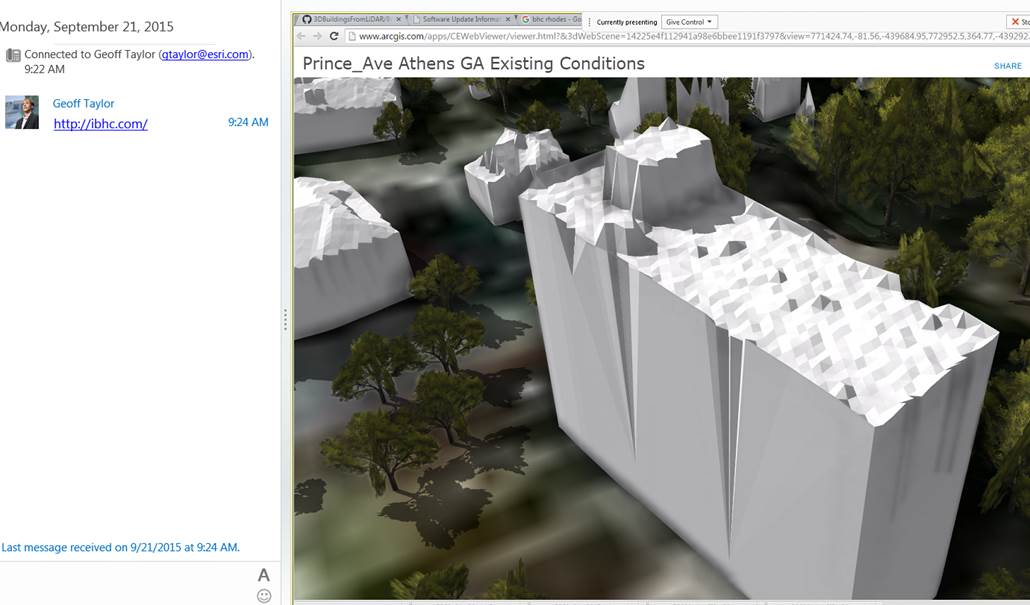


Figure Example of 'noise points affecting quality of 3D Multipatch building model derived from lidar points.

Step 3 in the current workflow helps remove points that occur along the sides of buildings. This is currently accomplished with COTS ArcGIS tools in the Python framework, using the 3D Analyst extension. This step most notably helps remove some of the noise points along the *sides* of the building. However, as shown in Figure 4, the processing does not remove all noise points along the sides of the building.

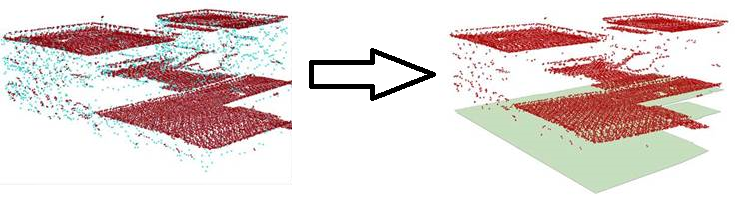


Figure Current noise point removal based on 3D proximity.

The results in Figure 4 indicate that we need additional 3D point processing algorithms to remove the remaining noise points. An implementation of Point Cloud Library’s StatisticalOutlierRemoval filter is available in the CloudCompare 3D point cloud and mesh processing software (open source, [www.cloudcompare.org](http://www.cloudcompare.org) ). An example of using this filter in CloudCompare on a relevant dataset is shown in Figure 5.

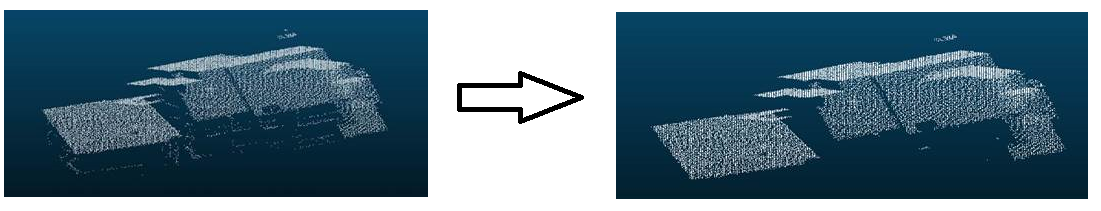


Figure Statistical Outlier Removal result.

Once the noise points on the sides of the buildings have been removed, quality issues along the salient edges of the 3D model still exist. Most can be identified as ‘soft edges’, and these are the final remaining quality issues we would like to fix through tools developed as a result of this R&D proposal.

An example of an edge that can appear noisy, or ‘soft’, is shown in Figure 6. The identification of this edge and subsequent production of a ‘hard’ line through it are discussed in [1]. One of the components of this proposal is to explore the implementation of the algorithm that is presented in [1]. This 3D line will be densified and converted to points in order to enrich the point cloud that goes into generating the final mesh, as shown on the right in Figure 6.

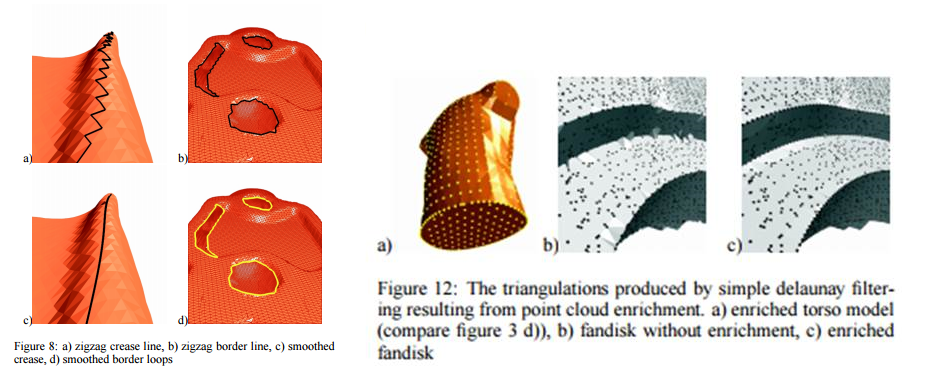


Figure These figures show the processing ideas in [1] that will be implemented as part of this R&D project.

Finally, the algorithm in [1] will also allow us to identify planar surfaces in the point cloud. This will assist us in generating more crisp 3D models of buildings in point clouds generated from image pairs (stereo techniques) or collections (photogrammetry techniques). It is clear from Figure 7 that point clouds generated from lidar provide a cleaner surface from which we can generate an accurate model (left), and additional processing may be needed for point clouds generated from imagery (right). An example building that is present in a 3D mesh generated from a photogrammetric point cloud is shown in Figure 8.

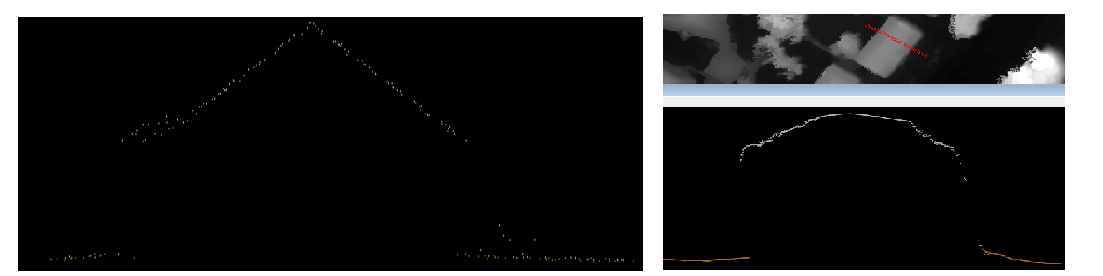


Figure (Left) Profile view of gabled roof as sampled by lidar system. (Right) Profile of gabled roof present in point cloud generated from passive imagery techniques.



Figure Example 'soft' mesh generated from photogrammetric point cloud.

# 3.0 Strategic Objectives and outcome from R&D

The arcpy site package in ArcGIS 10.3.x, and other versions prior and moving forward, allow for numpy, scipy, and other advanced analysis packaged to be used in the ArcGIS framework. The desired outcome of this project is to create a python toolbox that will allow for the advanced processing of 3D point clouds at a very local scale, e.g., points within a building footprint, to facilitate more accurate and visually appealing 3D models. If successful, we will demonstrate these tools to relevant parties within Database Services and also to the relevant Sales and Marketing team members.

The total 3D mapping and 3D modeling market is expected to grow from USD 1.90 Billion in 2015 to USD 16.99 Billion by 2020, at an estimated Compound Annual Growth Rate (CAGR) of 55.0% from 2015 to 2020. North America, the first mover in this market, is expected to remain as the major market till 2020 [2] (see <https://esri.my.salesforce.com/0D57000002sapca> ). Industries that use 3D which could benefit from these tools being available include, but are not limited to:

* Commercial (e.g., Real Estate)
* Local Government
* Federal Government (e.g., DoD)
* Safety and Security (e.g., first responders)
* Entertainment
* Non-profits
* Infrastructure / Telecommunications
* Architecture, Engineering, Construction (AEC)

# 4.0 Resources and Schedule

The budget for this project shall be split among the relevant tasks:

- Remove noise points within building footprint (28 hours)

- Generate mesh from remaining points and evaluate (16 hours)

- Identify soft edges and planes in mesh (28 hours)

- Enrich point cloud by replacing soft edges with hard edges (28 hours)

- Enrich point cloud by replacing soft planes with hard planes (28 hours)

Thus, the hours budget for this project will total 128 hours. Development will be shared between Joe McGlinchy and others in DBS who are interested in 3D processing with relevant Python experience (To Be Determined). We will also consult with Geoff Taylor and others in his group (Emerging Markets) who have familiarity with these markets and the desired output data products.

Since the platform for the tools outlined in this proposal are based in Python, we will leverage the Arcpy site package as much as possible. However, some specific open, 3rd party Python site packages will also be investigated and leveraged when needed. These include, but are not limited to:

* **PyHull**, a Python wrapper to Qhull (<http://www.qhull.org/> ) for the computation of the convex hull, Delaunay triangulation and Voronoi diagram. It is written as a Python C extension, with both high-level and low-level interfaces to qhull [3].
* **NearPy**, a simple yet modular little framework for Approximate Neaarest Neighbor (ANN) search. It uses the also very popular Python frameworks numpy and scipy, which provide functionalities for scientific computing [4].
* Computational Geometry Algorithms Library (**CGAL**), which provides wrappers for the following packages (most relevant to this project in **bold**) [5-6]:
  + Few Kernel primitives (details here)
  + **2D Triangulations**
  + **3D Triangulations**
  + 2D Alpha Shapes
  + 2D Convex Hulls and Extreme Points
  + dD Spatial Searching (only for 2D and 3D cases)
  + 3D Fast Intersection and Distance Computation (AABB Tree)
  + 3D Polyhedral Surfaces
  + 2D Conforming Triangulations and Meshes
  + **3D Surface Mesh Generation**
  + **3D Mesh Generation**
  + **2D and Surface Function Interpolation**
  + 2D Voronoi Diagram Adaptor
  + Halfedge Data Structures
  + **Point Set Processing**
  + Intersecting Sequences of dD Iso-oriented Boxes

# 5.0 Risk Management

While I believe that we should be able to create the Python tools that implement these algorithms, there is the possibility that these tools could be slow. We have not tested these Python packages so we are unsure how the tools will perform.

# 6.0 Document and Source Control

This document will be stored in the [DBS Research Lab Sharepoint document library](http://pswebsp/dbservices/DBSRDLab/Shared%20Documents/Forms/AllItems.aspx). Code will be stored in a private repository in the ArcGIS GitHub organization visible (read permissions) to all of Esri.

# 7.0 Quality Assurance and Testing

Developer testing is sufficient for this R & D project.

Links

[1] <http://graphics.stanford.edu/courses/cs164-10-spring/Handouts/papers_gumhold.pdf>

[2] <http://bit.ly/138Eu1W>

[3] <https://pypi.python.org/pypi/pyhull>

[4] <http://nearpy.io/>

[5] <http://doc.cgal.org/latest/Point_set_processing_3/index.html#Chapter_Point_Set_Processing>

[6] <https://github.com/CGAL/cgal-swig-bindings/wiki/Package_wrappers_available>