**Instructions for generating a lidar hyperoctree search database**

Mike Zimmerman, January 19, 2017[[1]](#footnote-1)

The hyperoctree code is presently tuned for lidar datasets containing 3D positions on the body surface and times of measurement, i.e. many (x, y, z, t) tuples. The code produces “browse” and “search” directories for populating the respective panels on the LidarHyperTreeSearchPanel. It could be extended to higher-dimensional datasets but performance has not been tested in such a scenario.

Here are the steps required to integrate a new dataset into SBMT via the tree-search mechanism:

1. Implement the abstract classes FSHyperTreeGenerator, RawLidarFile, and FSHyperPointWithFileTag for your chosen dataset.

(*For instance, the NlrFSHyperTreeGenerator provides a concrete implementation of the openFile method, which returns a new NlrRawLidarFile, which, in turn, produces instances of NlrFSLidarPoint. This effectively tells FSHyperTreeGenerator what kind of file to load its data from, a way to generate LidarPoints in memory from raw files on disk, and a way to tag each point with the file it originally came from. See also the OlaFSHyperTreeGenerator, OlaRawLidarFile, and OlaFSHyperPoint classes.*)

1. Calculate the bounds of the dataset.

Add your instrument to the LidarInstrument enum. This requires knowledge of the x, y, z, and t bounds of your dataset.

1. Export the following classes to runnable jars:
   1. Your implementation of

1. One day before the Trumpian nightmare begins in earnest. [↑](#footnote-ref-1)