# : Quick Start:

1. Review key terms related to the conceptual organization of data (Chapter 2)
2. Ensure that all required input data (Chapter 5) are stored in a single folder (Chapter 3) and named appropriately (Chapter 4).
3. Optional: Set up the preferences.txt file (Chapter 6), and save in Matlab working path. If no file is set up, the user will be prompted to enter input during program execution.
4. Run Jan\_RunKelbeCode.m with no input arguments (Chapter 7). Review algorithm details if desired (Chapter 8).
5. Intermediate data will be stored in intuitively named subfolders(Chapter 9); however the average user will only be interested in the final output data: registered PLY files for each input scan, which are collected and saved in a single output folder (Chapter 10).

# : Input Data Organization:

Input data conceptually organized as follows:

/Users/Kelbe/Desktop/Presentation1.pdf

Fig. 1: Data are organized hierarchically according to their Experiment, Site, and Plot.

* **<experiment>:** thematic descriptor identifying a unique study or experiment. An experiment is the organizational umbrella, which comprises one or more disjoint sites relevant to that study/experiment
* **<site>:** A smaller contiguous geographic area, e.g., 20x20m, in which 2 or more plots are set up. All plots within a site are registered to one another.
* **<plot>:** An individual location where a sample TLS scan was collected.

# : Input Data Location

Lidar data are stored in a single folder for each experiment, the path\_source

**<path\_source>** A string identifying the source path where the lidar data lives.

A default can be set in the preferences.txt file, or this can be left blank and the user will be requested to choose the appropriate folder during execution.

# : Input Data naming

Lidar data are named as follows:

**<experiment>-<site>-<plot>.<ext>**

For example,

TEST2-031-008.txt

TEST2-031-013.txt

TEST2-031-014.txt

…

# : Input Data Format

For each plot, the following lidar data files are required:

These are the standard outputs from RIT’s las\_processor. Refer to las\_processor documentation for details of individual file specifications. Low-hanging future work could generalize this to make it simpler for other systems to use by abstracting these requirements to the minimal set (e.g., only required inputs are x,y,z,I,r. The program then computes r,a,e automatically and generates .img and .hdr on the fly).

* **\*\_polar.txt**:
* **\*\_xyz.txt**
* **\*.hdr**
* **\*.img**
* **\*.txt**

# : Default Preferences file

A preferences file contains the input parameters for program execution. This file should be created by the user (see the included default file for example). These are optional parameters; if they are not set, the user will be prompted to select them during program execution.

* **<info\_experiment>** (required). See Chapter 2. Example: “TEST2” , “KIP3”, “HAVO”
* **<info\_sites>** (required). See Chapter 2. Example: “31”, or “0” if only one site per experiment
* **<info\_plots>** (required). See Chapter 2. An *array* of plots to register together, e.g., “1:25” will register plots 1 through 25; while “8,13,14” will only register plots 8,13,14.
* **<info\_suffix>** (required). A string specifying the date (MO-DAY) so that you can re-process the data (for example, with code modifications without wiping out your previous results. In general, keep this constant for your full set of analyses. Example acceptable value: “12-03”
* **<options\_skipseg>** (required)

Additionally, default paths can also be set in the preferences file. These are optional parameters; if they are not set, the user will be prompted to select them during program execution.

* **<path\_up>** (optional). See Chapter 9.
* **<path\_source>** (optional). See Chapter 3.

The preferences file will be loaded automatically if stored in the working directory. Or, the user will be prompted to locate it on the file system during program execution.

# : Program Execution

Run Jan\_RunKelbeCode.m with no input arguments. The preferences.txt file is loaded and the required inputs are parsed from this file.

# : Main Function Contents

Jan\_RunKelbeCode.m is a wrapper function that runs Dave Kelbe's code for

1. **stem detection,**
2. **manual refinement,**
3. **feature extraction,**
4. **pairwise registration, and**
5. **multiview registration.**

For more information, refer to the following publications:

1. **Stem Detection:** Kelbe, D., van Aardt, J., Romanczyk, P., Cawse-Nicholson, K., and van Leeuwen, M. (2015). Single-scan stem reconstruction using low-resolution terrestrial laser scanner data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 8 (7). doi: 10.1109/JSTARS. 2015.2416001.
2. **Manual stem refinement** (Hawaii addition): The user can choose to refine stem detection results for very difficult forest types, e.g., Hawaii, but manually adding or removing tree segments using a user interface
3. **Extract features**: Interface between [1] and [3], which extracts tie points (stem-terrain intersection) from each detected segment to serve as inputs to the registration module
4. **Pairwise registration**: *Determine pairwise pose estimates (R,t) between all pairs of scans.*  Kelbe, D., van Aardt, J., Romanczyk, P., and van Leeuwen, M. (2016). Marker-free registration of forest terrestrial laser scanner data pairs with embedded confidence metrics. IEEE Transactions on Geoscience and Remote Sensing 54 (7) DOI:10.1109/TGRS.2016.2539219.
5. **Graph-based registration**: *Analyze the ensemble of pairwise estimates to find the optimal global pose parameters, which link each node (scan) to the reference node* Kelbe, D., van Aardt, J., Romanczyk, P., and van Leeuwen, M. Multi-view, marker-free registration of forest terrestrial laser scanner data with embedded confidence metrics. IEEE Transactions on Geonscience and Remote Sensing (In Review).

# : Intermediate Output Data Organization

Interim data products are stored in the user-selected path,

**<path\_up>:** A string identifying the path to save the output data to.

A default can be set in the preferences.txt file, or this can be left blank and the user will be requested to choose the appropriate folder during execution.

Intermediate data will be stored in appropriate and intuitively organized subdirectories:

* **<path\_processed>**
  + **<info\_experiment>**
    - **<info\_suffix>**
      * **<info\_site>**
        + **<info\_plot>**

**eps**

**fig**

**mat**

**ply**

**png**

In general, these intermediate data products may not be useful for the casual user. These folder structures can be deleted after execution if processing is complete.

# : Final Output Data Organization

Final output data are the registered PLY files for each plot in <info\_plots>, with filenames as follows:

**<info\_experiment>-info\_site>-<info\_plot>-WCS.ply**

All output ply files are collected and saved into a single folder,

**<path\_processed>/<info\_suffix>-<info\_experiment>-<info\_site>-PLYOUT/**