The Problem

In order to detect vibrations on a spiderweb using a Laser Doppler Vibrometer, scan points must be selected. Scanning specific areas vice the whole web saves time. In the interest of further saving time, a script could be created to label scan points in the place of a researcher.

Premise

The results of image recognition training involving intersections of spiderwebs were mostly negative. The latest script for the image recognition approach, Release Candidate 1.2, was extremely variable in its success. As shown below, it was able to correctly identify some groups of points but miss others entirely.

Deciding the Method

The selected language is MATLAB, because it is a common language between members of the Oregon State University LRAM group. This allows for future development with relative ease.

Possible Solutions

Within MATLAB lay two options:
- Use image recognition to train the script to identify intersections.
- Repurpose the FAST, Minimum Eigenvalue, or Harris-Stephens algorithms for feature selection to search for intersections rather than corners.

Optimal Conditions

Goals for the script are as follows:
- Detect most points within an image
- Output the points to a format that the vibrometry software can read
- Operate within a period of time less than what a researcher would have to devote to labeling points on the image

How It Works

The current version, entitled Release Candidate 1.6.3, has three components to how it selects the points.

The first part uses the Harris-Stephens algorithm to detect edges within a contrast-increased grayscale version of the original image. The Harris-Stephens algorithm was chosen because it consistently identifies a large amount of points when compared to the other two algorithms that MATLAB offers.

These numerous points within a certain radius of each other are then separated into groups using MATLAB’s use of the Kd-tree algorithm. The center of these groups becomes a new point for reassessment. This process is repeated until a singularity is reached and further repetition has no impact on the points that are selected.

The selection of new points from the second step is then put through subtractive clustering with a subset of the “strongest” original points that the Harris-Stephens algorithm presented.

Settings

There are several options that can be altered in order to make the script run better. These are the base distances at which groups of points are detected, the rates at which those distances scale with the image size, and the percentage of the original points that is kept for including in the subtractive clustering.

Current Features

- Detects most points (60-97% depending on the image, settings, and region of interest)
- Outputs selected points to a delimited text file
- Identifies, normalizes, and finalizes more than a thousand points in under a second. All images tested run in under a minute.

Status Moving Forward

By the goals established at the beginning of the process, the work on the project is complete. The code can potentially be further improved in the future to reduce the number of false positives acquired. Doing this could involve the manual removal of all points selected within a designated area.