SCORPION LAB PROJECTS

This is an undergraduate research project (non-peer reviewed) completed in the scorpion lab in the Department of Biology at the University of Oklahoma.

A biological approach to indoor localization: Modifying an insect-inspired navigation technique to accurately locate a point.

Weslie Conner Howard [Honors Thesis]

Research supervisor: D. Gaffin; **Date:** May 2018

ABSTRACT: Object tracking technology is of great interest to our species. The *Global Positioning System* (GPS) works well outdoors but has limitations indoors. To date, the most efficient methods for indoor tracking are derived from WiFi networks, where the signal strength from radio signals is converted to concrete measurements with a building. Our goal was to improve real time localization indoors using a derivation of a technique called *Navigation by Scene Familiarity* developed from studies of ants. The idea is that the pixelated eyes of these insects convert scenes to matrices of stored information that can be used to retrace a path back to a goal. We adapted this approach by developing an algorithm that uses a series of geometric formulas to pinpoint the location of a panoramic test image by comparing a pixelated version of the test image to a grid of panoramic images taken at known positions. We used a grid of 900 images taken from a building hallway to test the accuracy of our method by varying both the sensor resolution (pixel density) and landscape image spacing. We found that test image accuracy does not vary much for sensor matrices greater than 10x10 but varies considerably based on image spacing. Location accuracy was about 10 cm for grid images spaced at 25.4 cm. Our method appears more accurate than WiFi localization techniques and may be useful in applications where accuracy is of utmost importance.

Keywords: localization, tracking, navigation

Introduction

Problems arising from current localization methods

The Global Positioning System (GPS) was developed by the U.S. Department of Defense in the early 1970s and is a satellite-based navigation system [6]. GPS is arguably the most popular method of navigation assistance and enjoys broad use by a variety of societal entities, including technology companies, militaries, and civilians. It uses an aerial satellite system to locate coordinates by microwave radio signals throughout the world. GPS technology can be found on smart phones, watches, and in most cars, to assist people with traveling or localization. However, GPS availability and popularity comes at great expense. For example, in fiscal years 2016 and 2017, the government spent $938 million and $847.4 million, respectively (U.S. Department of Defense 2015; 2016) and $1.1 billion was budgeted for GPS during the 2018 fiscal year (U.S. Department of Defense 2017).

The accuracy of GPS depends on numerous factors such as satellite availability, global location, atmospheric conditions, infrastructure blockage, and receiver quality (Standard Positioning Service Performance Analysis Report). Along with these factors, user technological quality also plays a role. GPS receiver equipment ranges from cheaper consumer-grade localization units to expensive military and professional-grade equipment [14, 5]. Civilians using a GPS localization system can expect an accuracy of approximately 5-10 meters (Standard Positioning Service Performance Analysis Report).

Even though GPS continues to be successful and beneficial, it is hampered in some places outdoors and it is of limited use indoors due to structural interference with satellite signals. As such, various localization methods have been developed and tested to improve precision both outdoors and indoors. Some of these methods include *Simultaneous Localization and Mapping* (SLAM; [12], *Speeded Up Robust Features* (SURF; [4]), and WiFi localization. Of these methods, WiFi localization seems to be the most accurate and popular, with a positioning approach similar to GPS. The most accurate method uses a single WiFi access point and time-of-flight calculations to detect the location of a client in an indoor structure [18]. The median reported error is 65 cm during line-of-sight measurements and 98 cm during non-line-of-sight measurements [18].

SLAM is a method of navigation that focuses on an agent creating a map of an unknown area while simultaneously using that map to navigate and locate its position within the map [17]. This method was pioneered by two groups of researchers interested in mobile indoor robotics and their relative positioning [15, 10]. Even though SLAM has been popular for indoor navigation for approximately two decades now, the algorithms are complicated and there are issues in outdoor and indoor environments that have too little or too much visual information [7].

SURF is a more recent technique that was developed in 2006 and has been applied to many navigational trials [4]. However, the algorithms needed for this method are complex and sometimes ambiguous. Also, the individual images used by SURF must be descriptive and distinct to allow the feature detection algorithms to work; in particular, the process occasionally has difficulties distinguishing images due to adjacent image borders and similarity between scenes [1].

Our goal is to improve indoor navigation and localization without using radio signals while also simplifying the methodology. Our approach is inspired by the *Navigation by Scene Familiarity Hypothesis* (NSFH) proposed for insect navigation [2]. It can be used in various situations and without the use of radio signaling technology such as GPS and WiFi. The method is parsimonious, does not require expensive or complicated technology, and requires little to no maintenance once established. Furthermore, our current accuracy exceeds reported WiFi localization methods.

Introduction to scene familiarity and previous work

Insect navigation has been extensively investigated because of the animals' ability to navigate swiftly and precisely despite their small brains. The NSFH suggests a method of navigation that is compatible with the simplicity of an insect's brain and does not require sequential memorization of landmarks along the route [2]. Instead, it is proposed that these insects use a measure of familiarity while navigating, comparing current pixelated views to those stored in memory. This method of navigation has been applied to a computer software algorithm that allows an autonomous agent to recapitulate a training path based on scene familiarity [8].

The key feature of the NSFH is that visual scenes are unique when processed through the complex pixelated eyes of navigating insects. More specifically, insects use the catchment areas created by the pixel-by-pixel differences between a scene and its adjacent scenes to guide them along their previously experienced route [13]. The summed absolute pixel-by-pixel differences (SAD) between a single image when compared to the surrounding images creates a funnel-like plot (Fig 1). The single image selected for comparison is at the lowest point of the funnel and the slopes enclosing the single image rise smoothly with distance from this image [16]. To navigate, the insect scans its surroundings and simply moves towards areas that reduce the discrepancy between the current view and the scenes stored in memory [2]. There is no need for odometrical measurements or a geomagnetic compass when using this technique and there is no cognitive map as proposed by some previous work [9, 11]. As proof of concept, a gantry robot traveled successfully along a previously experienced path using scene comparison and familiarity [3].

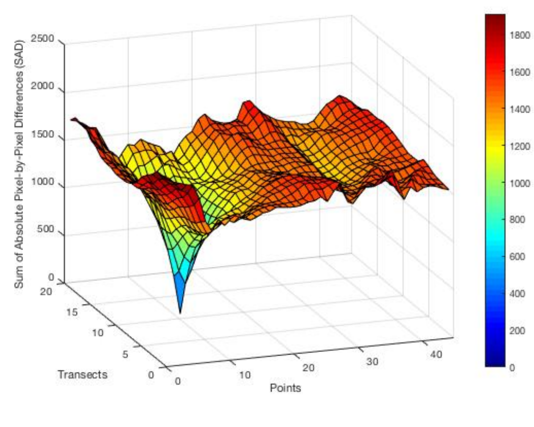

Figure 1: Catchment area plot. When a test image is selected and compared against all other images in the grid, a funnel shape catchment area forms over the focal test image. The similarity of an image to the focal image is indirectly related to the distance between them. The color blue indicates lover values and greater similarity, while red indicates the opposite.

In this study, we incorporate a second algorithm based loosely on the scene familiarity algorithm to accurately localize a point within a pre-defined grid of images. The tracking algorithm uses the SAD values of the agent's current view compared to a pre-localized grid of panoramic images and a series of geometrical formulas to localize the agent's position. We first used a dense, pre-existing landscape of hallway panoramic images to derive the most efficient and accurate image spacing and scene resolution for our algorithm. We then tested the system's ability to pinpoint several locations within a new image grid. The tracking algorithm has proven accurate to within 0.1 meters.

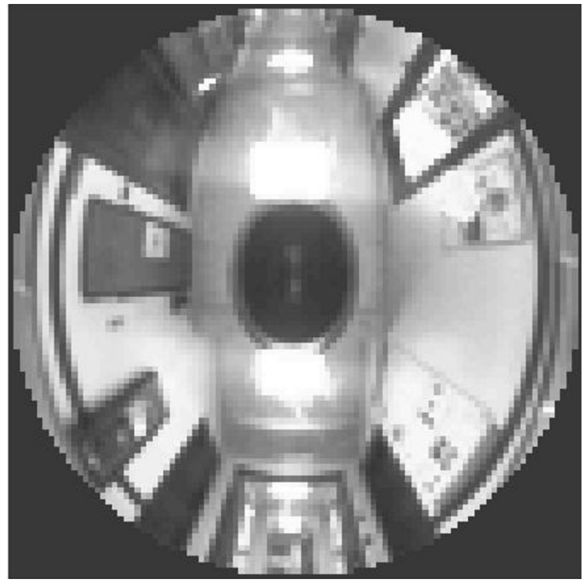

Figure 2: Circularized image. Panoramic images taken from Bloggie camera facing the ceiling are pixelated to a 100x100 grid and circularized in MATLAB

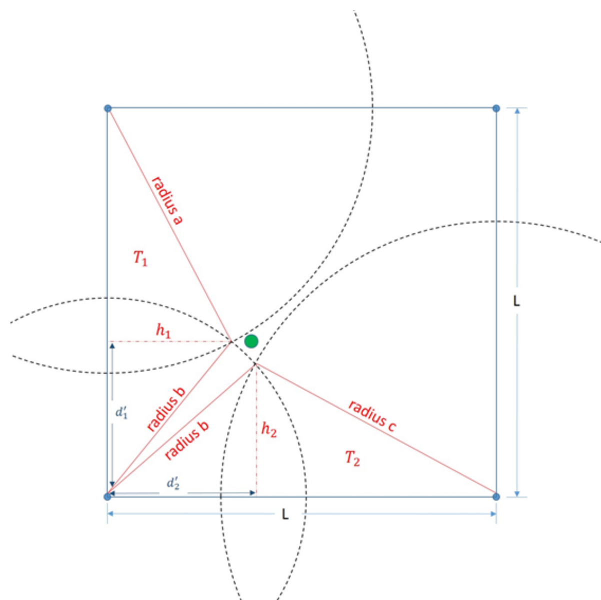

Figure 3: Determining test image location using SAD analysis. The four surrounding blue points represent the landscape points. The green point in the middle represents the test point that is being located by the algorighm. (L = distance between landscape points; T = triangle; h = height; d' = distance between closest landscape point and vertex of triangle)

Materials and Methods

Description of the tracking algorithm

In the trials discussed below, we declare the images that make up the pre-defined grid as landscape images, which serve as the reference points for the tracking algorithm and we declare the images that we desire to be located as test images.

To develop the tracking algorithm, we used a pre-existing, high density scan of 900 images acquired from a hallway on the first floor of Richards Hall on the campus of the University of Oklahoma [8]. The 900 images were equally spaced at 12.7 cm, taken at a height of 128 cm with an upward facing panoramic lens, circularized and pixelated to a 100x100 pixel resolution (Fig 2), and saved in a MATLAB structured array. These individually plotted images formed a 20x45 coordinate grid that served as landscape points.

To initiate the algorithm, the SAD between a test image and each image in the landscape image grid are calculated and the three images in the landscape with the lowest values are selected (which presumably are the ones nearest to the test image). Next, three hypothetical circles with radii representing the SAD of the selected images are drawn around the respective image points (Fig 3). The radii of these circles form two triangles adjacent to each other, with the lowest SAD value forming a side that is shared between the triangles with identical lengths. The shared vertex of the triangles meets approximately at the desired test location. At this point, we calculate the average pixel value (APV) between three landscape points that surround the test point. The APV is used to relate and convert the given pixel value to distance by dividing the APV by the distance between points (in cm) to give a pixel/centimeter (pix/cm) ratio. The pixel value of the three lowest points (the three points closest in proximity) are divided by the pix/cm ratio which allows use of these values directly in the tracking algorithm.

The geometrical equations in the algorithm begin with separately finding the area of both triangles using Heron's perimeter formula, perimeter $(s) = \frac{(a+b+c)}{2}$, and then using Heron's area formula, Area $(A) = \sqrt{s(s-a)(s-b)(s-c)}$. Next, the areas of each triangle are used to find the height of the triangles by rearranging the height formula, height $(h) = \frac{2A}{base}$. Using the height of the individual triangles and the radius of the image with lowest pixel value, the distance between the closest point and the vertices of both triangles is found with Pythagorean's theorem, distance $(d) = \sqrt{base^2 - height^2}$. The distance of one triangle will always be in the same direction as the height of the opposing triangle, and vice versa. Next, the coordinates of the test point are approximated using midpoint formulas, $\frac{x1+x2}{2}$, $\frac{y1+y2}{2}$, because its location will be an equal distance between the two triangle vertices. Depending on the position of the test point, in regard to the three landscape points, we can multiply the x-direction value and/or the y-direction value by negative one to dictate the correct orientation.

Overview of the algorithm

1. Calculate the area of each triangle (Heron's Formula):

$$perimeter(s) = \frac{(a+b+c)}{2}. \tag{1}$$

$$Area(A) = \sqrt{s(s-a)(s-b)(s-c)}. \tag{2}$$

2. Calculate height by rearranging triangle height formula: height

$$(h) = \frac{2A}{base}. \tag{3}$$

3. Calculate distance between lowest pixelated point and vertex (Pythagorean's theorem) distance:

$$(d') = \sqrt{base^2 - height^2}. \tag{4}$$

4. Calculate midpoint for x-direction and y-direction using distance and height (midpoint formulas):

$$\frac{x1+x2}{2}, \frac{y1+y2}{2}. \tag{5}$$

Optimizing the components of the algorithm

We used the landscape image dataset to further examine optimal spacing between landscape points along with optimal pixel resolution. To measure optimal spacing between landscape points, we reduced the number of landscape points from a 20x45 grid to a 10x23 grid of 230 images by removing the outer edge of landscape points in MATLAB. We then systematically removed points within the grid to increase the distance between landscape points from 12.7 cm to 25.4 cm. We repeated this process to create a 7x15 grid of 105 images spaced at 38.1 cm, a 5x12 grid of 60 images spaced at 50.8 cm, a 4x8 grid of 32 images spaced at 76.2 cm, and a 3x6 grid spaced at 101.6 cm. Removing landscape points from the initial 900-image grid allowed us to designate the eliminated images as test images, while testing the spacing of our landscape simultaneously. For the different grid sizes, we selected the same nine test images in various locations within the grid to analyze the accuracy of the algorithm depending on test image location (Fig 4).

We also tested resolution by changing the pixel density of the circularized panoramic images in each grid. The tracking algorithm was tested for accuracy using the five grids and tested at resolutions varying from 10x10 pixels to 100x100 pixels. After the photos were circularized and pixelated in MATLAB, an inner circle of the image was cropped out to eliminate the visually inferior quality of the ceiling from the images, allowing the NSFH algorithm to focus on the visually rich walls of the hallway. Each image retained approximately 62% of its original image after editing (Fig 5).

Error calculations

The pre-recorded location of a test image and the output from the tracking algorithm were both presented as (x, y) values. The localization error was calculated by first finding the absolute difference between the pre-recorded location of the test image and the localization output from the tracking algorithm. Next, the (x, y) error values were inserted into the equation, $\sqrt{x^2 + y^2}$, to determine the localization error for the tracking algorithm. This form of localization error was used with the purpose of calculating the upper boundary of the error, to conclude that the location of the test image according to the tracking algorithm could not exceed the calculated error.

Examples of animations showing the image comparison process for 4x8, 10x23, and 20x45 landscapes can be downloaded here: 4x8 grid animation; 10x23 grid animation; 20x43 grid animation.

Results and Discussion

The results show that the tracking algorithm is more dependent on landscape image-spacing than pixel resolution (Fig 6). The algorithm's accuracy is indirectly correlated with landscape grid spacing. The 10x23 grid (25.4 cm spacing between points) produced an average localization error of 9.28 ± 1.84 cm for pixel resolution 20x20 to 100x100; the 20x20 resolution was

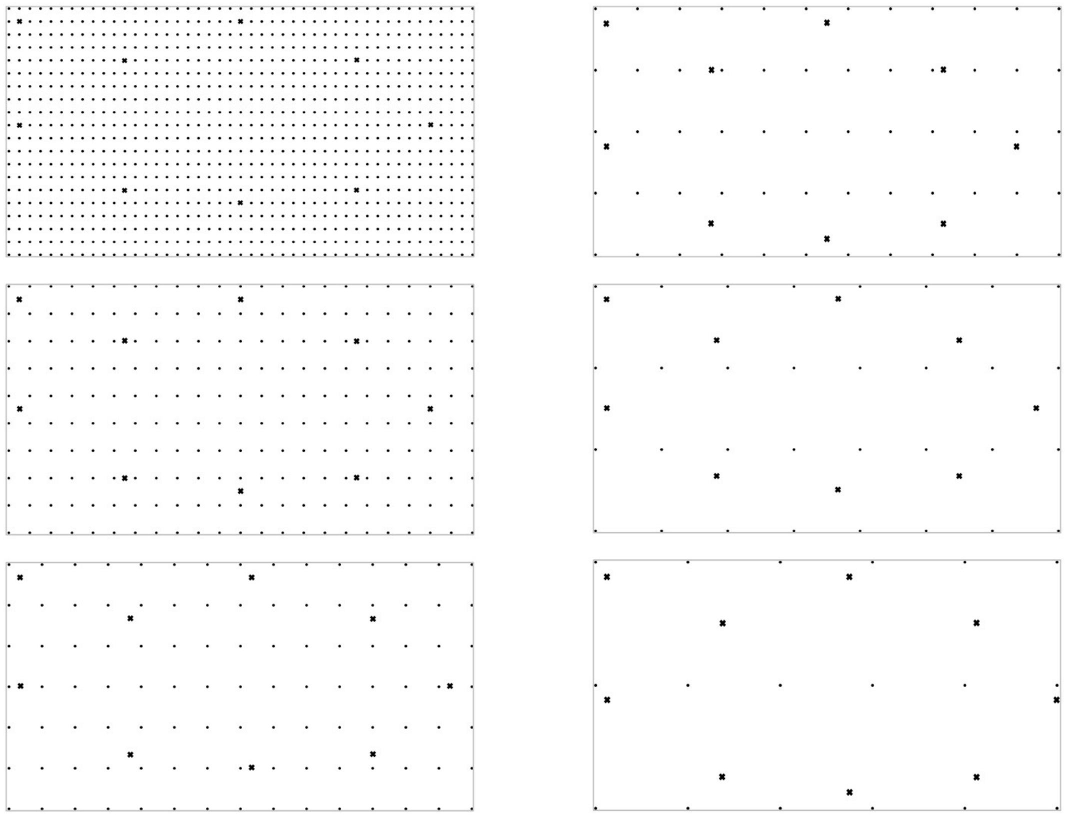

Figure 4: Test image locations within 900-point image landscape. Points were removed from the grid to create various spacings between the landscape images. The test image locations, which are standardized for all grids, are marked with "X".

the most accurate at 9.055 $\pm$ 1.76 cm. The accuracy decreased to over 100 cm for the 3x6 grid (101.6 cm between points). In general, within each grid spacing, there was no significant difference in accuracy among pixel resolutions 20x20 to 100x100, but the 20x20 resolution tended to be slightly more accurate than others. However, the 10x10 resolution accuracy for the 230-image grid (10x23) and the 105-image grid (7x15) were approximately 2.4 times greater than the average localization accuracy of other resolutions. The remaining three grids (5x12, 4x8, 3x6) with fewer landscape images did not produce a significant decrease in localization accuracy at the 10x10 resolution. See the appendix for the results of test images at various resolutions with a new landscape grid taken at a different camera elevation.

The tracking algorithm successfully located the nine test images within the 900-image grid with a localization error of 9.06 $\pm$ 1.76 cm (10x23 grid, 20x20 resolution). The 20x20 pixel resolution produced the best accuracy overall, but there were no significant differences between the pixel resolutions for resolutions greater than 10x10. The tracking algorithm is more accurate than current localization via WiFi [18].

A plausible reason for the inaccuracy at grids with fewer images and increased spacing could be due to the poor catchment areas that form during the image comparison process. We noticed that the catchment areas that formed when using the 230-image, 10x23 grid were much more defined than the catchment areas that formed when using the 32-image, 4x8 grid (Fig 7). We hypothesize that the quality of the catchment areas plays an important role in our tracking algorithm as it does in insect navigation schemes proposed in other studies [19].

The accuracy of this indoor localization method suggests that the tracking algorithm could be used in various applications that are difficult for other forms of indoor localization. For example, it might be useful for accurately delivering medications in a crowded hospital where patients' beds are in close proximity of each other. In a less life-affecting situation, it might be possible to remove chain referees, line judges, and ball spotters from football games. These officials

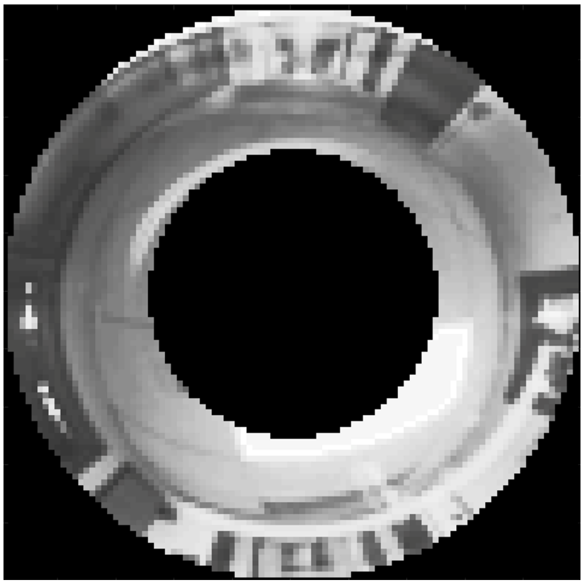

Figure 5: Circularized, pixelated, panoramic image with cropped center.

use an archaic method to attempt to spot the exact position of the football at the end of each play. This process is subject to human error opinion and is hotly contested by coaches and sports pundits. The tracking algorithm could eliminate human error by precisely locating the football on the field after a play. A small micro-cinema camera would be placed on the football and designated as the test image. Landscape images could be marked throughout the football field prior to competition and would be taken at ground level. After the play, the players on the field would gather in their respective huddles near the sidelines to allow for the tracking algorithm to localize the position of the football.

The tracking algorithm could also be used to assist with marketing techniques. Many superstores use marketing algorithms based on same-trip purchases to place items in their stores. Placing a camera on a shopping cart that records the location of a shopper in the store during their trip could help marketing companies with the placement of products to further increase profits.

In the future, we hope to conduct human trials using the NSFH algorithm to navigate while simultaneously tracking the location of the individual venturing indoors. During the creation of the image grids, it came to our attention that mapping the grid with equally spaced landscape images is time consuming and tedious. In the trial, each point was individually placed and measured using a tape measure before the images were taken. We imagine a drone could be

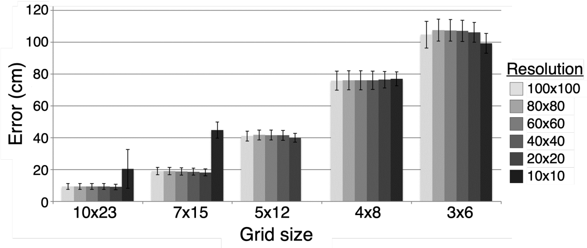

Figure 6: Accuracy of the tracking algorithm at various grid sizes and resolutions. We compared the estimate of the tracking algorithm to the location of nine known images. The plots show the mean ($\pm$ SD) accuracy of the algorithm for the five grid spacings and six photo resolutions. The 5x12 grid did not produce results within MATLAB due to unknown reasons.

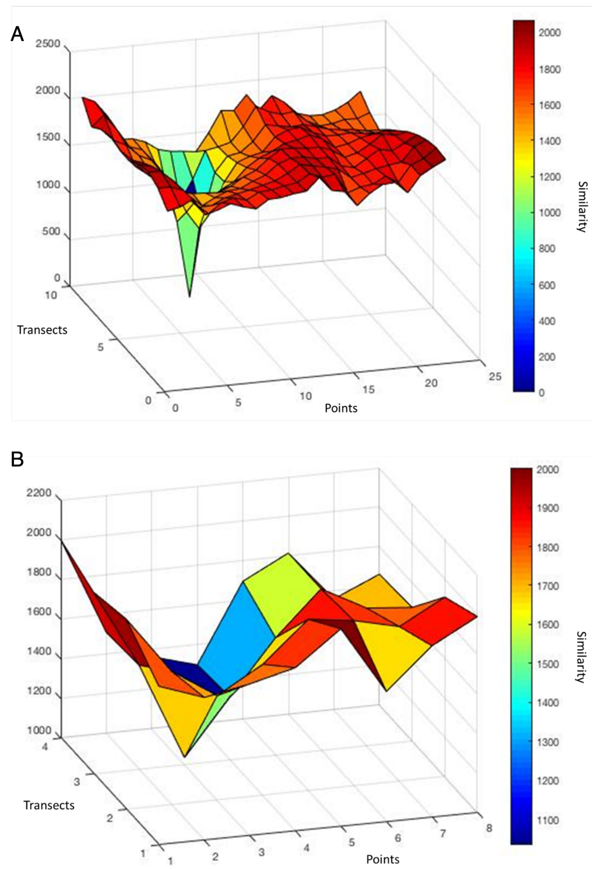

Figure 7: Comparison of catchment areas for different grid sizes. (a) The catchment area relative to a test image in the 230-image grid spaced at 25.4 cm is well-resolved compared to the catchment area (b) for the same test image in the 32-image grid spaced at 76.2 cm.

programmed to capture images at specific intervals to both increase the speed and accuracy of the landscape images.

Finally, additional modifications could improve the performance of the algorithm. For example, using color instead of grayscale images would provide more visual information and might improve the tracking precision. Also, it would be interesting to focus the camera on the ceiling where the visual information is more stable compared to the walls and hallway, where changing artwork, moving people, and other objects can alter the visual landscape. In line with this, simple ceiling tiles of random patterns could be developed to diversify the information and reduce the chance of aliasing. Alternatively or in addition, a camera could be directed to the floor below the agent; the wealth of pattern variations inherent in terrazzo flooring could be quite useful in this regard.

Acknowledgments

I would like to thank Dr. Doug Gaffin, Brad Brayfield, and the Scorpion Lab members, for all their help and encouragement. Thank you to the University of Oklahoma Biology Department and George Martin for assistance on my trials. A special thanks to the Honors College for the *Undergraduate Research Opportunities Program* grant in support of this project. Lastly, thanks to Dr. Phil Gibson, Dr. Doug Gaffin, and Brad Brayfield, for their participation on my *Honors Thesis Defense Committee*.

References

[1] Abeles, P. Resolving Implementation Ambiguity and Improving SURF, Mar. 2012. arXiv:1202.0492 [cs].

[2] Baddeley, B., Graham, P., Husbands, P., and Philippides, A. A model of ant route navigation driven by scene familiarity. *PLOS Computational Biology 8*, 1 (Jan. 2012), e1002336.

[3] Baddeley, B., Graham, P., Philippides, A., and Husbands, P. Holistic visual encoding of ant-like routes: navigation without waypoints. *Adaptive Behavior 19*, 1 (Feb. 2011), 3–15.

[4] Bay, H., Tuytelaars, T., and Van Gool, L. SURF: Speeded Up Robust Features. In *Computer Vision – ECCV 2006* (Berlin, Heidelberg, 2006), A. Leonardis, H. Bischof, and A. Pinz, Eds., Lecture Notes in Computer Science, Springer, pp. 404–417.

[5] Bondavalli, A., Ceccarelli, A., Gogaj, F., Seminatore, A., and Vadursi, M. Experimental assessment of low-cost GPS-based localization in railway worksite-like scenarios. *Measurement 46*, 1 (Jan. 2013), 456–466.

[6] El-Rabbany, A. *Introduction to GPS.*

[7] Fuentes-Pacheco, J., Ruiz-Ascencio, J., and Rendón-Mancha, J. M. Visual simultaneous localization and mapping: a survey. *Artif Intell Rev 43*, 1 (Jan. 2015), 55–81.

[8] Gaffin, D. D., and Brayfield, B. P. Autonomous visual navigation of an indoor environment using a parsimonious, insect inspired familiarity algorithm. *PLOS ONE 11*, 4 (Apr. 2016), e0153706.

[9] Gould, J. L. The locale map of honey bees: Do insects have cognitive maps?

[10] Leonard, J. J., and Durrant-Whyte, H. F. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation 7*, 3 (June 1991), 376–382.

[11] Menzel, R., Greggers, U., Smith, A., Berger, S., Brandt, R., Brunke, S., Bundrock, G., Hülse, S., Plümpe, T., Schaupp, F., Schüttler, E., Stach, S., Stindt, J., Stollhoff, N., and Watzl, S. Honey bees navigate according to a map-like spatial memory. *Proceedings of the National Academy of Sciences 102*, 8 (Feb. 2005), 3040–3045. Publisher: Proceedings of the National Academy of Sciences.

[12] Milford, M., and Wyeth, G. Persistent Navigation and Mapping using a Biologically Inspired SLAM System. *The International Journal of Robotics Research 29*, 9 (Aug. 2010), 1131–1153. Publisher: SAGE Publications Ltd STM.

[13] Philippides, A., Baddeley, B., Cheng, K., and Graham, P. How might ants use panoramic views for route navigation? *Journal of Experimental Biology 214*, 3 (Feb. 2011), 445–451.

[14] Schrader, D. K., Min, B.-C., Matson, E. T., and Eric Dietz, J. Real-time averaging of position data from multiple GPS receivers. *Measurement 90* (Aug. 2016), 329–337.

[15] Smith, R. C., and Cheeseman, P. On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research 5*, 4 (Dec. 1986), 56–68. Publisher: SAGE Publications Ltd STM.

[16] Stürzl, W., and Zeil, J. Depth, contrast and view-based homing in outdoor scenes. *Biol Cybern 96*, 5 (May 2007), 519–531.

[17] Thrun, S. Simultaneous Localization and Mapping. In *Robotics and Cognitive Approaches to Spatial Mapping*, M. E. Jefferies and W.-K. Yeap, Eds., Springer Tracts in Advanced Robotics. Springer, Berlin, Heidelberg, 2008, pp. 13–41.

[18] Vasisht, D., Kumar, S., and Katabi, D. {Decimeter-Level} Localization with a Single {WiFi} Access Point. pp. 165–178.

[19] Zeil, J., Hofmann, M. I., and Chahl, J. S. Catchment areas of panoramic snapshots in outdoor scenes. *J. Opt. Soc. Am. A, JOSAA 20*, 3 (Mar. 2003), 450–469. Publisher: Optica Publishing Group.

Appendix

We created a new 3x10 grid of images equally spaced at 0.885 m. We derived the 0.885-meter spacing by first placing the two outside transects at 30 cm in from the walls of the first-floor Richards Hall hallway to avoid conflicts with the laboratory cart. Dividing the width between these transects in half resulted in three transects spaced equally at 0.885 m. We then used this value to equally space the 10 points on each transect. After creation of the 30-point grid, 10 test points were placed within the landscape. The test points were placed in a specific manner to allow for different scenarios and various distances from landscape images. For example, some test points were placed directly in the center of four landscape points while other test points were placed relatively close to a certain landscape point. We then measured the distance between each of these test points and the four closest points. The landscape points were marked on the ground with a small strip of blue tape and a black mark, while the test points were marked with orange tape and a black mark. A Sony MHS-PM5K Bloggie HD video camera with an upward facing panoramic lens was attached to the top of a 1.93-meter wooden board that was clamped vertically to a laboratory cart to take photos at each marked location. Overall, the camera was 1.98 meters from the ground which allowed for humans to walk beneath without interfering with the image. A plumb bob was attached directly below the camera on the opposite, ground facing end of the board to ensure that the images were directly above the marked locations. Measurements were taken to compare the tracking algorithm's output against our hand-measured location values. The results of the these trials are shown in (Fig 8).

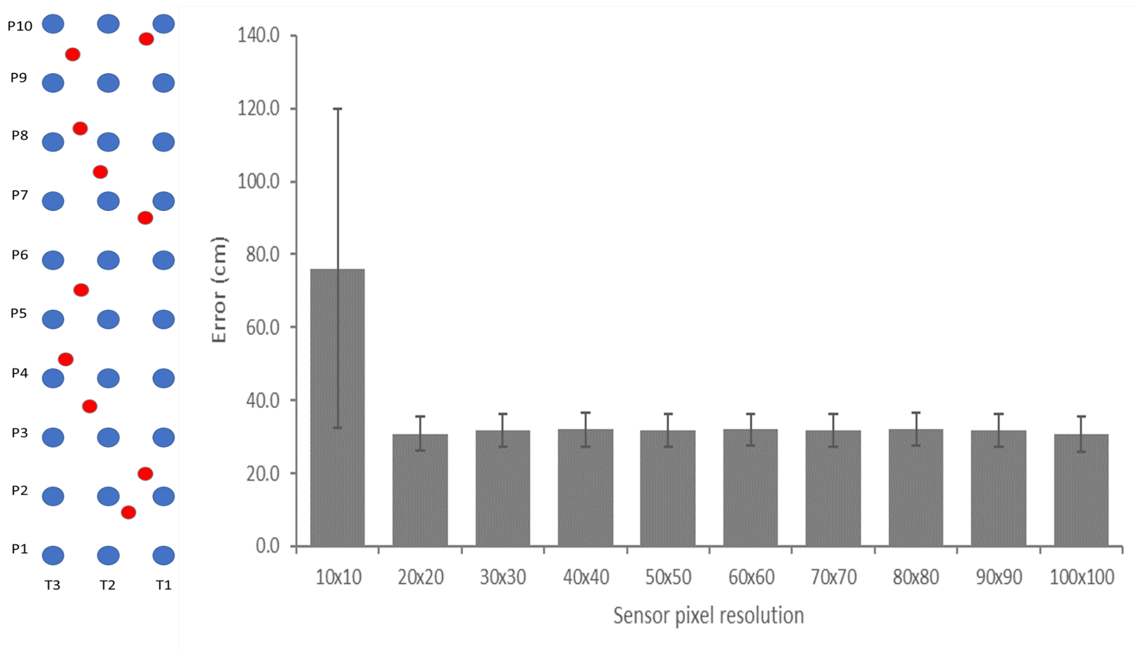

Figure 8: Test of new test landscape images with test points. Left: The circularized panoramic images, represented by blue circles, were arranged in a 3x10 grid landscape and equally spaced at 0.885 meters apart. The smaller red circles represent the positions of the test images. Right: Accuracy of the tracking algorithm at various pixel resolutions on the new landscape grid. The average localization error for the 10 points for pixel resolutions 20x20 – 100x100 was 31.6 $\pm$ 4.6 cm (SE). The accuracy for the 10x10 pixel resolution was approximately 2.4 times greater than the average localization accuracy and the standard error was about 10 times greater. The 20x20 resolution was the most accurate with a localization accuracy of 30.9 $\pm$ 4.7 cm