Swarmathon Technical Report
University of Houston

Mary Burbage, Li Huang, Lillian Lin, An Nguyen, and Aaron T. Becker

Team Members
Mary Burbage
Li Huang
Lillian Lin
An Nguyen

Faculty Advisor
Aaron T. Becker
I certify that I have reviewed and approved this report.

***Abstract*—In the NASA Swarmathon competition, a team of Swarmie robots searches for resources and returns them to a home base. We began with the competition's base code and improved the mapping of detected April tag resources from the camera image to the competition arena. We saved this information so that the robots would know where uncollected tags could be found after dropping one off at the home. We also created a gridded search that helps the robot return to an area where it has previously detected resources and is more likely to find more. These improvements are offset by difficulties in detecting the tags in the images blurred by the robots' motion.**

I. Introduction

For the NASA Swarmathon competition, each team of Swarmie robots collects as many resources as possible within a limited time period and returns them to a home base. We approached this project with the idea that the Swarmie robots could be most effective when they have an accurate knowledge of their own positions and the positions of any resources they find. If they can locate resources with a good degree of accuracy when they cannot collect them, they can return to them quickly for later retrieval. This led us to concentrate our efforts on improving their self-localization and tag-mapping abilities. In addition, we worked on algorithms for improving the robots' goal location selection when exploring and searching for new resources.

II. Related Work

Work on coordinate frame transformations is discussed at length in Spong et al. [1]. This includes both translation from image coordinates to world coordinates and transformations among sets of coordinate frames that differ both in rotation and translation. The case of coordinate frames is covered in the specific case of an autonomous mobile robot in Siegwart and Nourbakhsh [2]. We implement these methods as we map the April tags detected in the camera image into their corresponding coordinates in the competition arena.

There has also been considerable work done in the area of search algorithms, an overview of which may be found in Choset [3]. We considered several types of search procedures and worked on a modified version of the ant colony optimization proposed by Dorigo et al. [4]. As we progressed, we found that we needed to focus our efforts on tag detection as any improvement in the coverage path of the arena is insignificant if the robot cannot detect the tags satisfactorily as it searches.

III. Methods

A. Camera Mapping

A primary component of our work on the Swarmathon project was improving the mapping of the April Tag resources as the Swarmie discovered them. To do this we mapped the tag positions in the camera image to positions relative to the robot and then mapped those relative positions to the world coordinates of the arena.

We began by establishing a coordinate frame $o^r x^r y^r z^r$ for the robot using the procedure in Spong et al. [1]. This coordinate frame is set with an origin o^r on the ground in line with the front edge of the robot and centered between the front wheels. Fig. 1 shows the orientation of the axes relative to the world coordinate frame $o^w x^w y^w z^w$. The z^r and z^w axes are aligned and vertical. The y^r axis is rotated from the x^w axis through a rotation of θ to align with $currentlocation.theta$ in the $Pose2D$ datatype.

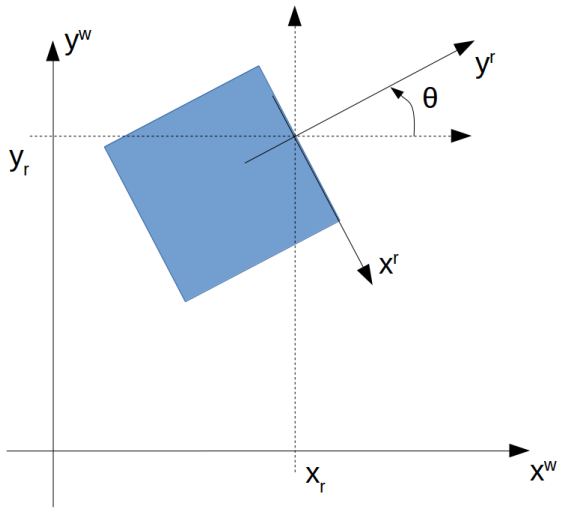

Fig. 1. The robot operates in a coordinate frame $o^r x^r y^r z^r$ within a larger world coordinate frame $o^w x^w y^w z^w$.

Once the coordinate frames are established, we can transform any point (x^r, y^r) in the robot's coordinate frame to its location (x^w, y^w) in the world coordinate frame through use of the transformation matrix H_r^w.

$$H_r^w = \begin{bmatrix} \sin(\theta) & \cos(\theta) & 0 & x_r \\ -\cos(\theta) & \sin(\theta) & 0 & y_r \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Having set the robot's coordinate frame, we needed to map the camera view into the robot's coordinates. To do this, we measured the field of view of the camera using a tape measure and sheet of paper as shown in Fig. 2. We used the on-screen image in Fig. 3 to find the limits of the camera's range. These limits are shown in Fig. 4. The robot's camera sits $z_c = 0.195\ m$ above the ground in the center of the robot. The nearest visible spot in the y^r direction is $0.205\ m$ away, and the farthest visible spot is $1.40\ m$ away. At $y^r = 0.205\ m$ the image extends $0.115\ m$ to either side of the center line. At $y^r = 1.27\ m$, which is the maximum distance at which the image has sufficient resolution to render the paper (and thereby an April tag) legible, the image extends $0.56\ m$ to either side of the center line.

Using these values, we calculate the minimum and maximum angles between the camera and the visible world. We represent the vertical field of view with the angle ϕ.

$$\phi_{min} = \text{atan2}(0.195, 0.205) = 46\,\text{deg} \quad (2)$$

$$\phi_{max} = \text{atan2}(0.195, 1.40) = 82\,\text{deg} \quad (3)$$

Fig. 2. The robot camera's field of view is measured with a tape measure and a sheet of paper. The limits of the field of view are determined by the distance at which the paper is visible.

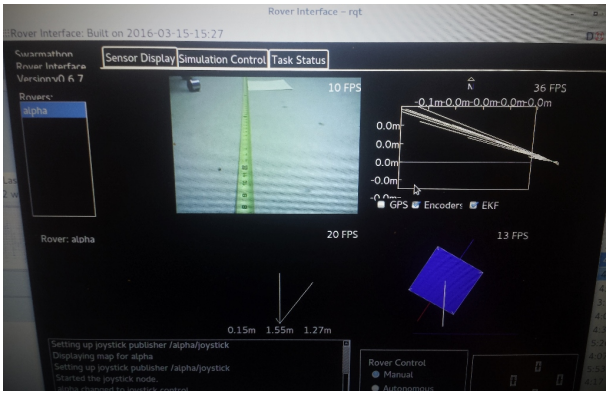

Fig. 3. The limits of the field of view are determined using Swarmathon simulation GUI.

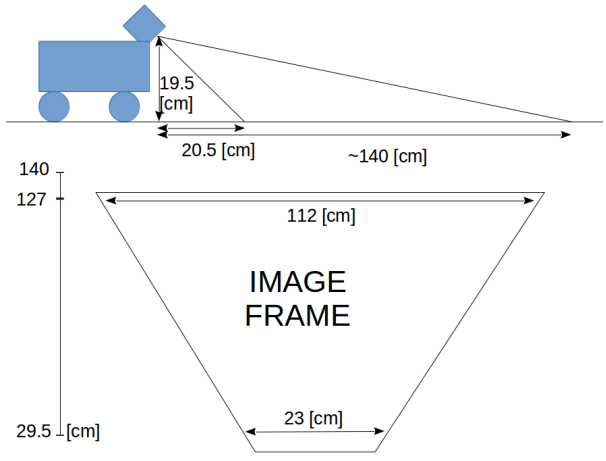

Fig. 4. The robot camera's field of view forms a trapezoid that is narrow near the robot but expands as the distance from the robot increases.

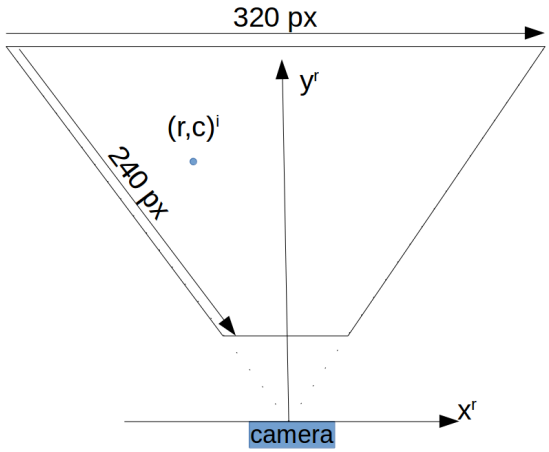

Fig. 5. The robot camera's field of view is measured with a tape measure and a sheet of paper. The limits of the field of view are determined by the distance at which information on the paper is legible.

These give a net vertical viewing angle of

$$\phi_{net} = \phi_{max} - \phi_{min} = 36\,\text{deg} \tag{4}$$

The horizontal field of view is represented with the angle ψ.

$$\psi_{min} = \text{atan2}(\sqrt{0.205^2 + 0.195^2}, 0.115) = 44\,\text{deg} \tag{5}$$

$$\psi_{max} = \text{atan2}(\sqrt{1.27^2 + 0.195^2}, 0.115) = 47\,\text{deg} \tag{6}$$

Since there is some discrepancy, we use an overall horizontal viewing angle of $\psi_{net} = \pm 45\,\text{deg}$.

The image frame is 320 pixels wide and 240 pixels tall. With the viewing angles ϕ_{net} and ψ_{net}, this leads to a vertical resolution of

$$res_v = \frac{\phi_{net}}{r_{max}} = \frac{36\,\text{deg}}{240\ pixels} = 0.15\frac{\text{deg}}{pixel} \tag{7}$$

and a horizontal resolution of c_{max}.

$$res_h = \frac{\psi_{net}}{0.5 * c_{max}} = \frac{45\,\text{deg}}{320 pixels} = 0.14\frac{\text{deg}}{pixel} \tag{8}$$

This image field is shown in Fig. 5.

For any object located at the pixel coordinates (r, c) in the image, we calculate the position (x^r, y^r) of the object with respect to the robot's coordinate frame.

$$y^r = z_c \tan((r_{max} - r)res_v + \phi_{min}) \tag{9}$$

$$x^r = y^r \tan\left(\left|c - \frac{c_{max}}{2}\right| * res_h\right) \text{sign}\left(c - \frac{c_{max}}{2}\right) \tag{10}$$

Substituting values in 9 and 10, we have

$$y^r = 0.195 \tan\left((240 - r) * 0.15 + 46\right) \tag{11}$$

$$x^r = y^r \tan\left(|c - 160| * 0.14\right) * \text{sign}(c - 160) \tag{12}$$

both of which are measured in meters. Once the tag location in the robot's frame has been calculated, it can be transformed into the world frame using 1 and saved for future use.

B. April Tag Library

A second important component of our approach was taking better advantage of the capabilities included in the April tag library. The default image detection code returned only the first image detected by the camera, but it is useful to know the locations of tags that cannot be immediately returned to the home base. Instead of handling only the first tag detected in the camera image, our methods returned the whole array of tags detected and the coordinates at which they were detected in the image frame. Applying the transformations discussed above, we reported the locations of all the April tags that had been found into the robot's mobility node.

The homography matrix returned by the April tag library allows us to compute the locations of the tags directly. The methods presented in the camera mapping discussion provide a camera projection matrix to translate the images to real-world locations as in Olson [5].

C. Magnetometer calibration

As we worked with the physical robots, we found inaccuracies in their localization which we traced to the magnetometer calibration. To calibrate the sensor, raw magnetometer readings in both the x and y directions are sampled as the robot rotates in a complete circle. The mid-point between the minimum and maximum readings for each axis is saved as an offset for that axis. This mitigates the hard iron distortion generated by the combination of electronic and metal components in the robot.

D. Weighted Search Areas

We also created a node-weighted planar search algorithm based in part on the ant colony optimization scheme in Dorigo et al. [4]. This algorithm is an improvement over the random walk search of the base Swarmathon code in the case that the resources are distributed in higher concentrations in some areas rather than uniformly distributed. When the resource density is constant across the arena, the random walk model is at least equally efficient because there is no benefit to the higher weighting of areas that have been visited over the areas that have not. Since a real-life resource collection problem is more likely to find resources grouped together, we concentrated on that problem.

Our node-weighting algorithm detects the boundary of the map then sets up grid divisions inside the world coordinate frame. The whole map is divided into quadrants of equal size, and each quadrant is further divided into four smaller areas of equal size. Each area continues to be subdivided in the same fashion until the smallest quadrant measures no more than $0.8\ m$ and is labeled as a node. Once the grid is established, each square is assigned a weight. As targets are detected in a square, the weight of that square and the neighboring squares increases. When a robot reaches a desired region but finds that the targets have already been collected, the node's weight decreases. With each move, the robot tries to locate the highest weight to set as its next goal location. The system of grid division reduces on-board computation so that the robot does not have to scan the whole map to find the next goal location. In conjunction with this, each robot is assigned a unique identifier so that different robots can carry out different tasks or work in different areas of the arena.

Fig. 6. The UH and UH-Clear Lake teams of Swarmies prepare to compete for April tags scattered around an arena sized for the preliminary competition.

IV. Experiments

We ran trials of both virtual and physical competitions. For virtual competitions, we ran thirty-minute trials for both the three-robot and six-robot teams. We ran simulations with both the default random search and our node-weighted search to verify that our method did in fact improve upon the original code given with the challenge. Simulations included both the clustered and power law distributions.

Our biggest test was a small competition hosted by the UH-Clear Lake team. We created an arena in an outdoor basketball court as shown in Fig. 6. Each team provided four $2.4\ m$ wall sections, which could produce a complete $4.8\ m$ square arena, and which could also be used for a full-size competition arena by shifting them to provide barriers in any place when a robot approached the edge of the arena.

During the competition, we discovered that the midday sun's glare over-saturated the camera images of April tags, often making them completely illegible. We experimented with placing polarized lenses (sunglasses) in front of the camera, which improved the image as shown in Fig. 7. While this cannot be applied in the competition, it provided interesting information.

Even with the glare, some of the robots were able to detect tags, but they either could not navigate to the home circle or could not detect the home tags once they were in the home circle. Between the two issues, no tags could be reported so there was no score for the competition.

Due to the detection problems, we only performed one half-hour run in the outdoor competition and moved inside to run since the chances of recognizing the tags would be much higher without the glare of the sun. We used a small arena as shown in Fig. 8. Each team ran two trials, but after detecting tags, the robots rarely brought them back home; they either failed to return home or failed to recognize the home tag again.

Additional testing with the physical robots was performed in a square area approximately $12\ m \times 12\ m$ in an atrium. We tried to keep the robots away from the edges of the space as much as possible in order to reduce magnetic interference.

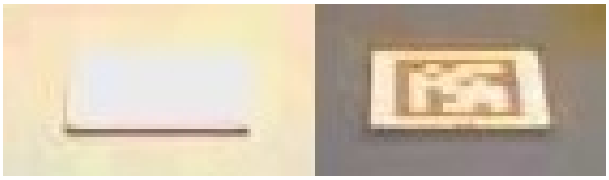

Fig. 7. The robot camera's image becomes oversaturated on a sunny day. A washed out April tag image (left) can be read when a polarized lens is placed in front of the camera (right).

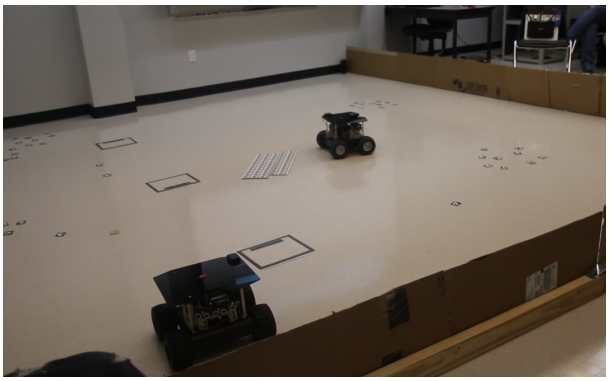

Fig. 8. The UH team of Swarmies searches for scattered and clustered April tags in a small indoor arena.

V. Results

Virtual simulation results showed a great improvement in the resource collection rates for the node-weighted search algorithm over the original code. In three-robot, thirty-minute runs we averaged 60.7 tags retrieved and 61.7 tags detected for a four cluster tag distribution and 62.6 tags retrieved and 63.3 tags detected for a power law distribution. This compares favorably with the 7.2 clustered tags and 10.0 power law distribution tags retrieved in the average of our base code trials. This shows a six- to eight-fold increase. Unfortunately, the difficulty we had detecting tags with the physical robot meant that we could not verify these results with the Swarmies.

During physical testing we found that even after calibrating the magnetometer, the Swarmie still has an error of approximately $0.5\ m$ to $1\ m$ when it infers its position. This is an improvement on our results before calibration. Part of the error may be due to the indoor testing site we used, which has electrical wires in the walls around the perimeter that may introduce significant error into the heading calculations and thereby into the coordinate calculations.

Our test runs also showed that the robot's motion blurred the camera image to a point at which it cannot reliably recognize the April tags as it moves. Reducing the robot's linear speed when in the translation state to as little as 10% PWM showed only incremental improvement.

VI. Conclusion and Future Work

After running our tests with the Swarmie, we found that image detection was in fact an important area in which to focus our work. We were not able to solve the problem of quickly and reliably identifying tags, including the home tag, but we were able to improve the robot's knowledge of where uncollected tags were located when we did identify tags in the camera image. Through a start-up calibration sequence, we were also able to improve the Swarmie's knowledge of its own location.

Further work is required on tag detection. In future we will implement a motion deblurring algorithm that takes the robot's motion into account to generate a deblurring mask. This will allow us to use a higher translation velocity for the robot. We may also implement blob detection to change speeds when a potential April tag is detected. A faster robot would slow down when the camera image includes something that may be a tag. With this we would also implement the search methods that we were unable to fully test and implement this year.

Acknowledgments

The authors would like to thank the Swarmathon team at the University of Houston - Clear Lake for hosting a practice competition.

References

[1] Mark W. Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*. John Wiley & Sons, Hoboken (N.J.), 2006. ISBN 0-471-64990-2.

[2] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA, 2004. ISBN 026219502X.

[3] Howie Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, October 2001.

[4] Marco Dorigo, Mauro Birattari, and Thomas Stützle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.

[5] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, May 2011.