April Tag Detection:Calculating distance use ROS Transform package

Kirubel Tadesse, G. Matthew Fricke, Joshua P. Hecker, and Dr. Melanie E. Moses

Motivation: NASA Swarmathon involves university and community college students to participate in virtual and/or physical robotics competition. Through this program students are expected to learn ROS (Robotic Operating System), develop an autonomous foraging algorithm, and assemble a Swarmie rover. By doing so, students not only learn about new robotic technology, but also have the opportunity to add to future NASA space exploration research. The inspiration came from the biological behaviors that ants exhibit during the foraging process. It is a key component that has inspired the iAnt development team to create a search algorithm that models ant foraging behavior. Dr. Melanie Moses and her students at the University of New Mexico implemented search algorithms into the iAnt robots. The research resulted in a partnership with NASA to further advance the search algorithms to be used for space expeditions to Mars.

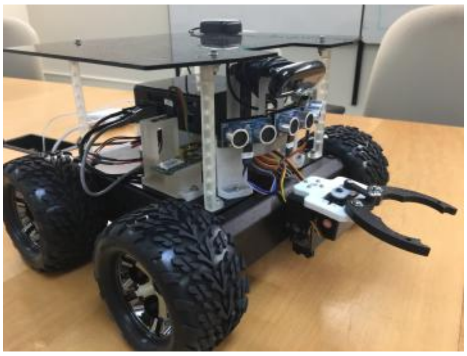

Figure 1: NASA Swarmathon rover image

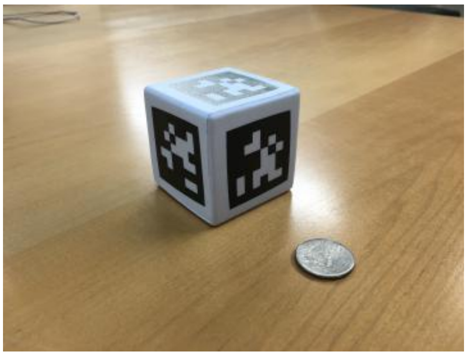

Figure 2: The April tag cube

Methods and Results: This research focuses on determining the distance between a tag and a rover. Based on the incoming camera image, the rover navigates to the location of april tag cube, which is an ideal food for the Swarmie. When the tag is within the rover camera view area, the rover calculates the distance between the tag and its current location. The most effective way to accomplish this is using ROS TF (Transform) to broadcast and listen to the transform between the incoming tag coordinates and *base_link* coordinate frame. By using the transform frame published by the apriltag_detection package, the coordinate points of the april tag is obtained. The broadcaster node then subscribes to this *tag_detection* topic and broadcasts the transformation from the *head_camera* to *the base_link*. Once the transform tree

is properly constructed, the transform listener node calls *TransformPose()*, a built in function with *TransformListener*, and publishes a *geometry_msgs/PoseStamped x, y, and z* values for the rover.

Algorithm: Initially, we tested a *similar triangle method* to calculate the distance using the focal length. Focal length is obtained by experimentally measuring a known distance - between the camera and the april tag - and multiplying it with the ratio of pixel width to the actual width of the object. After obtaining the focal length, the distance is calculated by using the formula

$$D = (WxF)/P \tag{1}$$

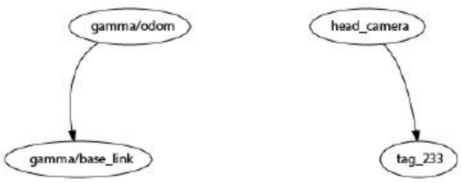

Figure 3: Transform broadcaster by the EKF/odom and April tag library

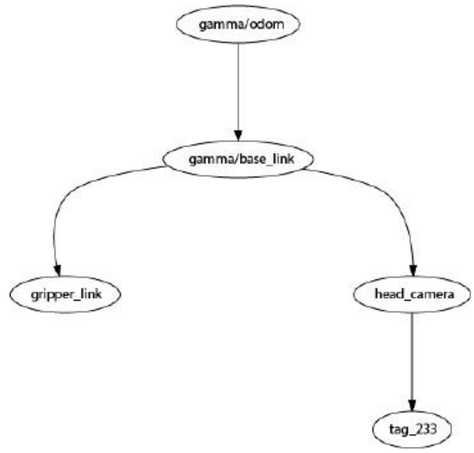

Figure 4: Full Transform tree after the writing of the broadcaster and listener node

W is the actual width, F is the focal length, and P is the pixel width on the camera image. Based on the measurement we took, the result was exact as tag was about 0.5m close to the camera. As we go farther from the camera, this method was inaccurate.

conclusion: Based on *similar triangle method*, the degree in which the rover camera is angled is not taken to consideration. It assumes the camera view to be a square; in reality, the camera view is trapezoid which makes it challenging to obtain the exact value of the distance. Therefore, we found ROS transform to be an effective way to solve this problem.

Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory J.E. Rawlins. Searching in the plane. *INFORMATION AND COMPUTATION*, 106(2):234–252, 1991.

Joshua P. Hecker and Melanie E. Moses. Beyond pheromones: Evolving robust, adaptable, and scalable ant-inspired robot swarms. *Swarm Intelligence*, 9(1):47–70, Feb 2015.