

# Exploring the Performance of the iRobot Create for Object Relocation in Outer Space

**Hasani Burns and Chutima Boonthum-Denecke**

The Department of Computer Science, Hampton University, Hampton VA 23668, USA  
hasani.burn@gmail.com, chutima.boonthum@hamptonu.edu

## Abstract

This research explores the performance of the iRobot Create machine for optimizing object relocation in an outer space environment. It is an ultimate goal to have it become a symbol of innovation for robots that are sent into outer space. Functioning as a tool-bot, and an active assistant, this robot aims to assist in small duties and respond to commands. With its arm and color blob recognition capabilities, this robot has the potential to receive a request, register and associate it with existing objects in its line of sight, and maneuver the arm to act accordingly, grabbing the correct object and giving it to a worker or engineer. This poster and presentation explains current progress and implementation of the iRobot Create for this purpose.

## Introduction

The iRobot Create is a robot that is not only a flexible robot platform for students and educators (Touretzky, 2010), but is also a cheap and effective way to use robotics technology for the societal need. Robotics is the future and the design of robots on the hardware and software levels is essential in the acceleration of space exploration, excavation, and space operations. Robots are tools, and can be expertly programmed to carry out human-like tasks in environments that are not suitable or too dangerous for a person to be in. Combining iRobot Create with the Tekkotsu framework created at Carnegie Mellon University, allowed for us to examine the color recognition for objects using the Calliope (Figure 1), a modified version of the iRobot Create.

Tekkotsu (Tira-Thompson & Touretzky, 2011) in Japanese means “iron bones,” is a software package that aims to give the user a structure on which to develop robotics control and routine tasks by focusing on higher level programming. Tekkotsu uses the object-oriented,

template, and inheritance features of C++. Hence, the user does not need to worry about the low-level programming to control a robot's vision and movement.

AprilTags (Olson, 2011) augmented reality tags from APRIL Lab at the University of Michigan, a new visual fiducial system that uses a 2D bar code style tag allowing full 6 degree-of-freedom (DOF) localization from a single image (shown in Figure 2). Visual fiducial systems are used to improve human-robot interaction, allowing humans to signal commands by simply flashing an appropriate card to a robot.



Figure 1a: Calliope (Full-body)



Figure 1b: Calliope Gripper (2 DOF)

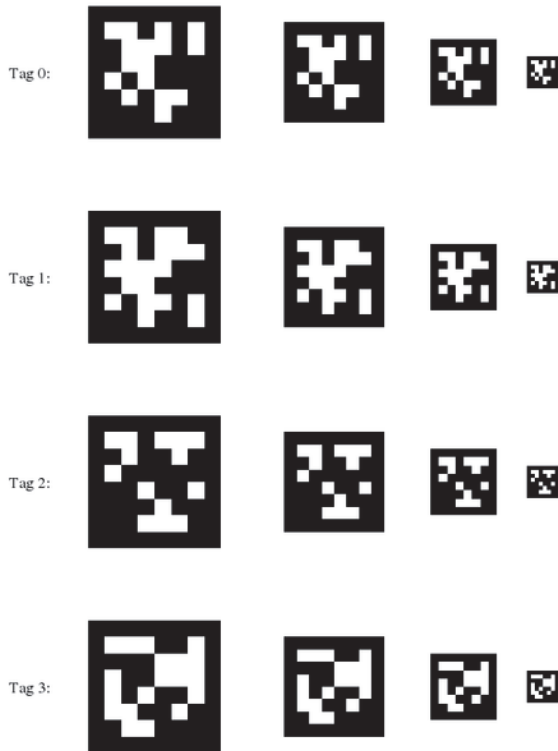


Figure 2: AprilTags

### Implementation

The Calliope, as shown in Figure 1, has an additional arm component with motors and controls that lets us extend the capability much farther than that of the traditional iRobot Create. To correctly plan out how this robot would be used, one first had to examine the environment which the robot would be carrying out its duties.

Zero-gravity environments call for precise movement, and especially for a robot that aims to grab tools and hand them to workers, it must be able to know when to let go of a certain tool, so as to not let it float off. The environment also calls for a robot that should not need to be controlled via operator, nor need any extra buttons pressed when out in the field. A robot such as this would be most effective acting autonomously, and that meant much emphasis on the robot's ability to sense commands, colors, shapes, tags, whatever it could use to get an understanding of what its next instruction should be.

The robot would begin in a ready or idle position, before proceeding to the next state (see Figure 3). It is inconclusive to which medium of command specifically would trigger the motor response from the robot, but the AprilTag system would work as well as a signaling

mechanism. From this command, the robot would begin its movement toward the source of which the tag signal came. As it makes its way to the target, the "eye" via the camera on the top of the robot will be scanning the path ahead for obstacles.

The AprilTags can be recognized also through use of the segmenting camera, as shown in Figures 4. From the raw image, it can differentiate between the tag image, as shown by the small square outlining the segment camera view, and the rest of the world. With the Tekkotsu framework, the position of the tag from the camera in space with the X, Y, and Z coordinates can be extracted.

Using Map Builder nodes with the default tag family, the robot can get a distance from the AprilTag from the center of it (see Figure 5). This allows for precision when measuring how much the robot must travel in order to reach the destination. Future plans include perfecting this – detecting AprilTag and its distance, along with the mapping functionality to maneuver around any obstruction, avoiding contact and damage, and then finally able to return to the appropriate path or home-base. Once the robot reaches the target destination, it will halt at that position and take a second set of image and use its segmented camera view to distinguish the colors. Finally, after receiving the tool-specific command, the Calliope would search for the color corresponding to that tool, grab it, and then hand it up and forward.

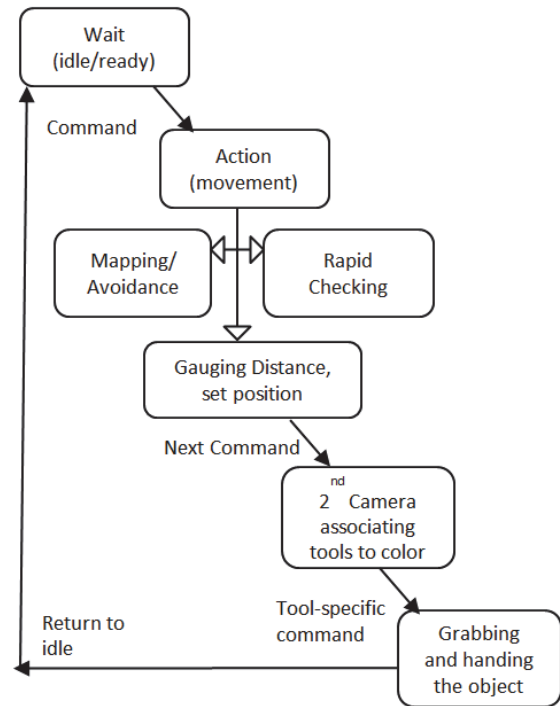


Figure 3: State Transition Concept Diagram

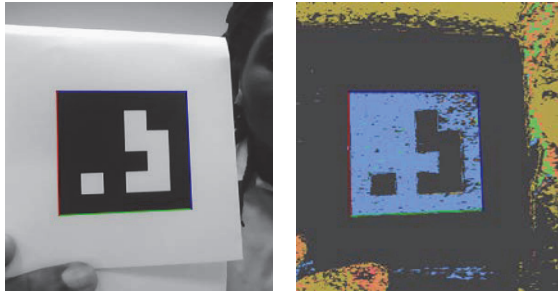


Figure 4: AprilTags (Raw and Segmenting Camera)

It takes a huge effort to use Tekkotsu’s MapBuilder components to activate and allow the Calliope to be able to “see.” It is even more important to be able to recognize an object in terms of the possibility for color association for tools, especially small objects where AprilTags may be invisible or hard to detect. Utilizing color image segmentation, color classes are assigned to each pixel, and calibrated so that when the Calliope sees that specific tone of color, it can record it properly.

The MapBuilder is created in the state machine through Visual Routine State node, and is told in this case to look for various blobs of colors in an image. A specific set of target colors can be listed; hence, segmented image will show only these targeted colors.

The image in Figure 5 shows the color segmentation working alongside the raw image capture. Even when not in the best light, the camera is able to capture and distinguish between the red, green, and blue colors. Manipulating the code based on the color and object detected allows the robot to be able to react a certain way, when a certain color is seen and recognized. This is done by adding new nodes that act as a function call to make the robot move its body, arm, or register the data in a structure such as an array.

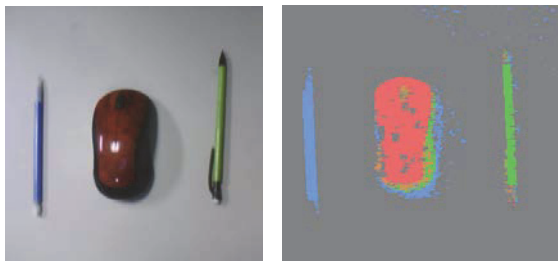


Figure 5: Raw Objects and Color Segments

## Kinematics

The arm’s kinematics allow it to move in the X, Y, and Z direction and can be manually controlled through the Tekkotsu interface, and also programmed to a set of poses, set by number and axis position.

The positions are loaded via a command in another motion node, and can be fluidly switched between each other to properly animate. They can be set to have a time interval run between them, as with any other function in the framework. Figure 6a and 6b show the 3D models of Calliope and how the arm function works. The arm movement will be based on the object location.

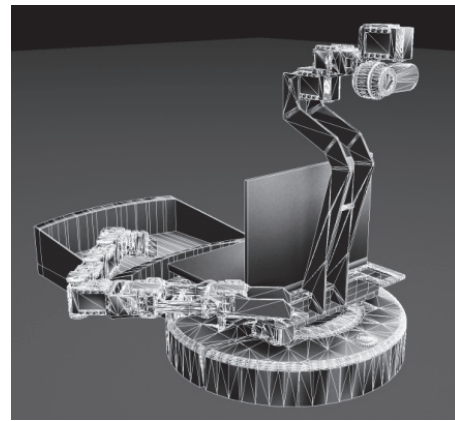


Figure 6a: 3D Model of Calliope

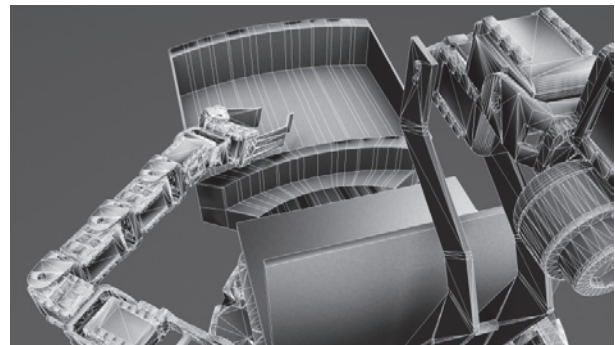


Figure 6b: 3D Model of Arm Function

## Discussion and Future Work

The Calliope robot is now able to recognize AprilTag and color blobs such that it can react based on what it sees. Both AprilTag and color Blobs have pros and cons, so depending on the situation one may be chosen over another. For example, in an unknown and limited space, color Blobs be may better used to signal the robot. It

would benefit to have a single program that switches between the two signals.

The Calliope's color recognition can be improved when color segmentation is properly defined. It is unfortunate that light and shade play a major role in color recognition. In some case, orange or pink can be seen as red; while green as blue. Therefore, color calibration should be done prior to any deployment. Unfortunately, it is unlikely to calibrate colors due to uncertainty on lighting.

### What can be improved?

Because Calliope only has a 2D-Arm (Figure 1), which means it can only pick up a standing object (i.e. with height that Calliope 2D-Arm can grab). However, it will not be able to pick up any object that lies down in a tray or on the floor. Therefore, a toolkit tray must be in vertical arrangement. If Calliope has a 5D-Arm (Figure 7, a prototype), it can be maneuvered to pick up both standing-up and laying-down objects.

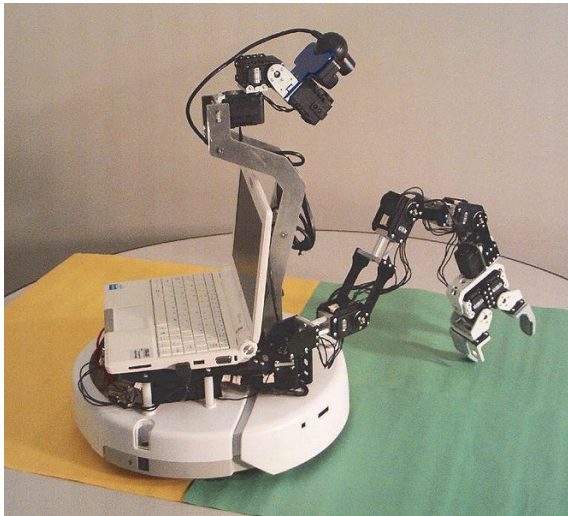


Figure 7: Calliope with 5D-Arm<sup>1</sup>

The robot should be able to navigate through known and unknown area. Mapping and localization should be integrated with a robot navigation system such that its position is known in the map, including origin and destination.

**Drawback:** Calliope (i.e. iRobot Create) is not an ideal robot for a space mission that requires a robot to navigate on a planet surface, such as Mars. Calliope can move on a smooth surface; therefore, its body has to be modified to support less ideal surface. Larger or stronger wheels and more body-height from the ground are potential solutions.

<sup>1</sup> <http://wiki.tekkotsu.org/index.php/File:Calliope-armv4.jpg>

**Potential applications:** While an outer space mission is unlikely, Calliope with 5D-Arm can be used in other area such as a mechanic working underneath a car and needing different tools. Instead of getting out and up for them, the mechanic can tell the robot what tool to retrieve.

### Acknowledgements

This research was supported in part by the National Science Foundation (CNS-1042466), NASA Virginia Space Grant Consortium (VSGC) and various corporate partners. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, NASA-VSGC, or our corporate partners.

### References

- Olson, E. (2011). AprilTag: A robust and flexible visual fiducial system. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 3000-3407, IEEE.
- Tira-Thompson, E.J., & Touretzky, D.S. (2011). The Tekkotsu robotics development framework. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 6086-6089.
- Touretzky, D.S. (2010). Preparing computer science students for the robotics revolution. *Communications of the ACM*, 53(8): 27-29. New York, NY.