

## **ICU: An Introduction To Object Identification and Mapping**

Brian MacAllister and John Santore

Department of Mathematics and Computer Science

Bridgewater State College

Bridgewater, MA

bmacallister@bridgew.edu, jsantore@bridgew.edu

For both hobbyists and novice students alike, vision and mapping are popular topics in the field of Robotics. However, from a beginner's perspective, it can be quite daunting to incorporate both into a project without having some practical experience or background knowledge on either subject.

Likewise, a significant amount of time will be wasted if unworkable solutions are chosen before more practical ones. It may be difficult for the beginner student to work on such a project, using a system whose components were not previously setup to work together. With these factors and the limited amount of time a beginner student may be allocated to work on such a project, it is important for the student to come up with a quick and workable design. Thus, one needs to find an ideal project: one that affords a novice student a plausible chance of completion while providing necessary direction.

A project that incorporates both vision and mapping is a scavenger hunt in which a robot must search through a room for a finite set of objects and display a map of where they are located. One such project was recently completed by a single undergraduate student during the course of a semester. During this time the student needed to develop a scavenger hunt program that could perform the previously mentioned scavenger hunt tasks of object identification and mapping with a robotic mobile platform and an attached color-blob tracking camera sensor module. The platform used was an "Acroname" "Garcia" robot and the camera module was a "Cmucam2" vision sensor. The objects assigned to the student were four brightly colored objects of varying shapes and sizes. However, the finished project was designed to accommodate more objects. The software, under the appellation of

"ICU" was designed under the premise that the software would be used and improved by future students.

Information used to help identify each colored object is separated and stored into instances of a 'part' class, with each part instance representing a physical part of a colored object. For example, one of the objects used was a multi-colored ring toss, which had five differently colored rods. Each part is assigned an RGB color range that is unique to the surrounding environment, and to other parts of different color. Parts of similar color, such as blue, are assigned the same color range. Each part stored has a minimum and maximum set of dimensions that are set based on dimensions returned by the camera at an ideal range, in prior testing.

The process of finding and identifying objects is composed of searching through all the stored color ranges until a blob of a size larger than a few pixels is found. This is followed by a simple set of actions consisting of rotating to the left or right while moving toward the blob. Once the front sensors return a distance that is less than or equal to the previously mentioned ideal range, ICU will attempt to identify the object. Identification is handled by attempting to match each of the stored objects parts to the dimensions returned by the camera. A priority is given to objects with the most parts. When ICU successfully matches an object, its color ranges are temporarily made unavailable until an identification attempt is finished. This is done to ensure that no two or more objects of similar color and size are conceivably matched from the same object. While looking for objects, ICU commands the Garcia to move in a straight path while rotating and backing up from obstacles that it senses through its surrounding infrared sensors.

Localization is handled by keeping a record of the robot's angle and coordinates within a two dimensional plane. Distances and rotations are calculated by taking the distance retrieved by the Garcia controller and multiplying it by a constant to increase accuracy, as previous findings showed a mostly

constant difference between distance readings returned by the Garcia controller and the actual distances moved. During a scavenger hunt run, points are saved containing the coordinate values of objects found. These objects can be both target objects and obstacles. Obstacles, such as walls, are found whenever one of the surrounding infrared sensors gets a reading less than its maximum range. Coordinates for all objects are calculated through simple vector addition. The coordinates for the robot are also updated in the same manner. Rather than checking for obstacles periodically, ICU checks for them each time the robot has moved a few centimeters. A point representing a target object is saved the first time the object is found. When all target objects are found, or the end user asks to end the program, ICU takes all the points that are created and displays an ASCII character map within a command terminal. The map's scale is determined by how far the robot has traveled from its starting position.

ICU was later tested by the student's instructor in several trials. The environment consisted of an open area in the back of a class room with surrounding foam core walls, some obstacles, and the target colored objects. During the trials, ICU was able to find and locate most of the target objects and display a map of where they were located. However, there were a few discrepancies. Sometimes the robot could not identify an object because it was not centered on the object. Other times it could never find the object because the object never came into view of the camera. Also the map that the robot produced could not draw the objects to scale since they were each assigned a single point in the map's grid. Although, the finished product may not be as polished as most systems used in real world applications, the scavenger hunt project and student's solution provides a novice with invaluable experience in both vision and mapping and the problems they present.