

# Multi-Agent Control and Computer Vision

## Annotation to Create Training Sets For Perception System

Kenneth Jones and Michael Wolf  
*East Los Angeles College and  
California Institute of Technology, Jet Propulsion Laboratory*  
kennethhjones1511@yahoo.com  
(Dated: August 28, 2015)

Described within this paper is the process of annotating images to form training sets to be used with the Surface Autonomous Visual Analysis and Tracking (SAVANt) perception system. To provide a full understanding of the process and the need for these training sets, a description of computer vision will be provided to give context. Different image sets were given varying annotation priority based on the objects they contained, sea state, and position of the sun. Objects of interest were either semi-submerged, floating, or boats that came into clear view of the Unmanned Surface Vehicle (USV) during the run. Image sets containing more of these objects, rougher sea states, or sun glare were given a higher priority when deciding a set to annotate. Once a set had been selected, it was opened in an annotation tool designed with MATLAB. The annotation tool was a graphical user interface which was constantly kept up to date using GitHub. New versions would be pushed to Git and then pulled by the various annotators. Usage of the annotation tool will be described in depth as well as the approach used to annotate the images. The process of tracking and detecting objects on the water and methods used will also be noted.

### I. INTRODUCTION

Prior to going in depth into the need for annotated data sets a basic understanding of computer vision must be acquired. A computer vision algorithm has the ability to acquire and process an image in order to determine an action to take. This description allows the process of computer vision to be broken down into its three major components, acquisition, processing, and deciding. In order to make this definition more tangible, imagine this scenario. A person is driving down a street and a stop sign comes into view. The driver, having passed Drivers Ed. and received a license, knows that the vehicle must come to a complete halt at the stop sign. The driver then begins to apply pressure to the brakes in order to bring the vehicle to a full stop. This list of actions is identical to what is performed by a computer vision algorithm. The event of light bouncing off of the stop sign and entering the driver's eyes is the acquisition component, because at that instance the image has entered the driver's brain to be understood. At this point, the driver realizes that she or he is looking at a stop sign, due to being told repeatedly during his childhood what a stop sign looked like; this is the processing component of computer vision. Lastly, from having passed Drivers Ed. the driver knows to halt the vehicle at the stop sign, thusly applying the brake in order to do so; this is the deciding component. From this description it can be said that the annotated data sets are equivalent to the driver's childhood, in the context that it was the time in which he or she learned to identify the stop sign. Just as a child would recognize a stop sign the more they are exposed to one, the larger the annotated data set, the more the computer vision

algorithm will be able to recognize the objects within them and distinguish between similar looking objects.

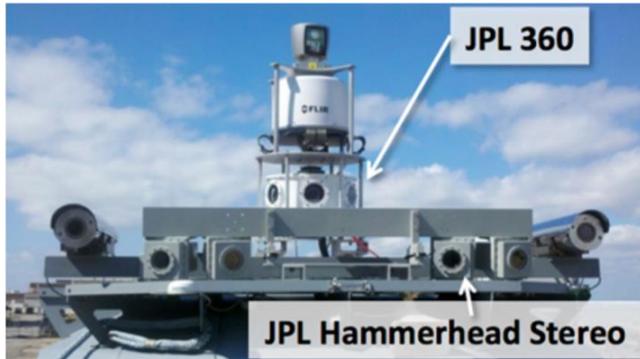


**FIGURE 1:** Example of computer vision on the right and annotated data on the right

### II. PROJECT GOAL

By supplying the SAVANt perception system with training sets it will improve at detecting objects on the water, such as various types of boats, semi-submerged hazards, and buoys. Once the USV has a sufficiently large training set, it will be able to react accordingly to what it sees in images, allowing it to be used in a variety of situations. A few situations that come to mind would be due to military application. By equipping this system to a naval vessel, the navy would be capable of having an automated fleet. There are many benefits to this capability such as twenty four hour surveillance of shipyards by patrol boats, the escorting of high value manned vessels, and the patrolling of seas with harsh environments. Other than the military this technology can be used in other sectors, such as the shipping industry. Every year the shipping industry loses millions of dollars due to piracy. Piracy is the act of holding hostage the crew of a ship or its

contents in order to ransom them off. By automating the ship, there would be no crew to hold hostage and no way for pirates to move the ship in order to hold its cargo hostage. Thus eliminating the possibility of piracy, cutting down cost with the removal of a crew, and improving the efficiency of the shipping industry since it will be possible to ship twenty four seven.



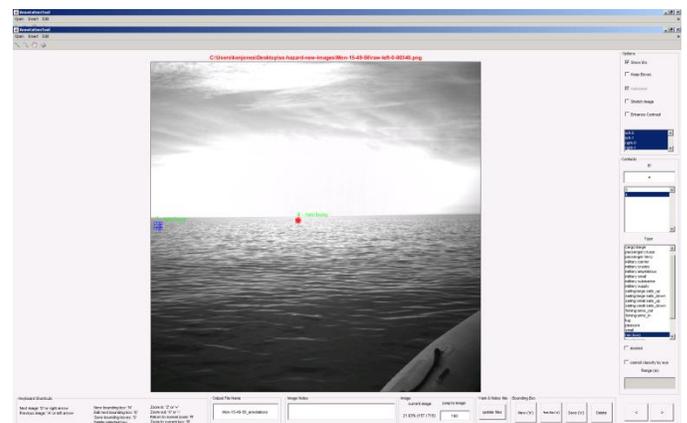
**FIGURE 2:** Hammerhead Stereo and 360 Camera set mounted on a maritime vessel for testing.

### III. PRIORITIZING IMAGE SETS

Before the annotating can begin a set of images must first be selected. There are always more image sets that need to be annotated than the amount of annotators. Due to this, annotators have to be selective in regards to which image sets they attempt to annotate. The vast amount of image sets, combined with the limited amount of time and annotators created a priority system that ranks image sets depending on the clarity and variety objects, the sea state, and the sun position within the images. At the time, there was a lack of annotated boats, so image sets that contained a variety of boats that were not too far off in the distance were given priority. An ideal image set would be the USV out in open water approaching another naval vessel to the point that the other boat was fully distinguishable in the images taken with the USV's camera set. This process would then be repeated allowing for the same naval vessel to capture it at multiple angles, increasing the likely hood that the algorithm would be able to detect the ship from any angle. This process of bringing an object into clear view from multiple angles is ideal for an object on open water. Of course this was not always the case if the sea state or the sun angle of the image set interfered. These image sets were still of high interest for that very reason, because it exposed the computer vision algorithm to multiple scenarios in which an object may not be fully visible. Since these data sets were desirable image sets with calmer seas, less visible objects, and ideal sun angles were normally of a lower priority, than images sets with the opposite traits.



**FIGURE 3:** An image being examined in the annotation tool. Image is an example of a rough sea state.



**FIGURE 4:** An image in the annotation tool. Image is an example of sun glare in an image set.

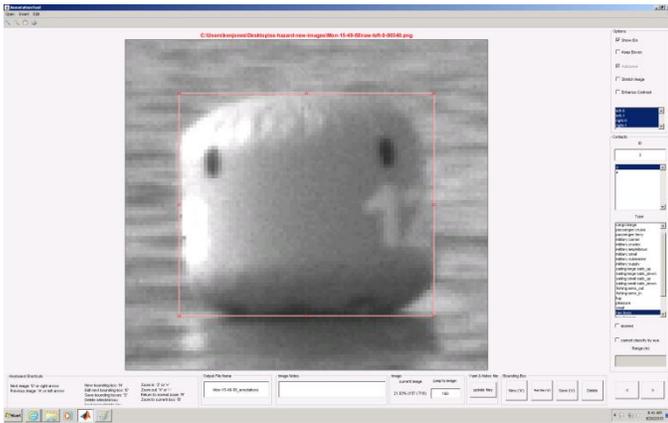
### IV. ANNOTATION TOOL

The annotation tool will be discussed in three parts, how images were annotated, the features of the annotation tool, and how it was maintained and accessed.

#### A. The Annotation of Images

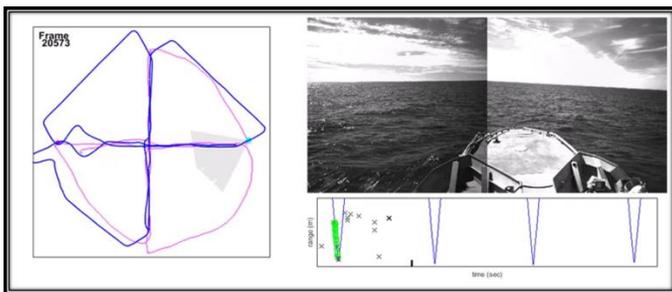
Once an image set was decided upon, the annotation tool was launched. In order to use the annotation tool, MATLAB had to be running on the computer since the tool was a MATLAB script. The image set of interest was opened and loaded into the annotation tool. At this point it is up to the annotator to distinguish the objects in an image. With every image it is important to look through it carefully in order to pick up elusive objects. If in a later image an object suddenly appears at the center without coming into frame first, then most likely it was in prior images but it had been missed. This means that the object needs to be back traced through the images to the point in which it comes into frame. This can

potentially be a very detrimental process, if multiple objects have already been annotated. If that is the first object to show up in the images then reversing through the images in order to annotate the object does not take up much time.



**FIGURE 5:** Object bounded by the blue box of the image found in **FIGURE 4**

If there are multiple objects that had been annotated prior to discovering the object, not only must the first frame of contact be found but the contact ID's for all the previous annotated objects may need to be relabeled to reflect the order in which they came into frame. If the contact ID's are not reoriented, it not only makes the annotators job harder, but can potentially confuse the computer vision algorithm if the annotator labels a new object out of order. By having the contact ID's numerical order reflect the order in which they are seen within the image set, when the annotator cycles through the image set to check his or her work the objects can be tracked just with their contact ID. Also, the numerical ordering helps convey the time in which the object appeared in the image set. It also helps to record the ID assigned to an object along with a brief description of where it is in respect to the USV. This helps identify an object that has already been labeled but appears multiple times in the same image set after passing out of frame. An example would be when the USV makes a U-turn or is on a clover path and an object that has already been seen is viewed once again, that object should be given the contact ID it has already been assigned.



**FIGURE 6:** Example of Clover Leaf Pattern

Other than keep track of contact ID's the process of annotating an object found on the water's surface or in the distance is straight forward. An object identified in an image is surrounded by a bounding box. The box's perimeters are as close to the object's sides as possible. This is to ensure that only the object is annotated and not the surrounding background; the more background that is included in a bounding box, the less effective the annotation. Another important aspect to object annotation is the assignment of an object description. Within the annotation tool, there is a list of titles that can be given to an annotation box in order to describe the object within such as navy buoy, military small, and passenger cruise. These were only three possible titles within a list of approximately 20 different descriptions that could be assigned. The title given to an object was at the discretion of the annotator, so research had to be done to roughly understand what each of the descriptive titles looked like. Adding more depth to describing the bounded object, there was also a check box to designate if the object was docked or not and a check box for an object is indistinguishable at its current distance. Once all the objects in an image had been annotated with their proper contact ID's and titles, the data was saved in a text file, and the next image was looked at. Once an image set was completed, the YAML file was created and then uploaded to server along with the annotated images. The YAML file contained data on where objects were located in an image, in a form recognized by the vision algorithm.

## B. Annotation Tool Features

The annotation tool has many features that streamline the process of annotation. These features will be discussed in depth to explain their function and how they affected the process. The features to be discussed are the ability to toggle ID's on and off, stretch an image, enhance the contrast, keep the bounding boxes from the previous image, and shortcut keys. The ability to toggle the Contact ID's on and off improves the process of tracking an object through multiple slides and bounding objects that are bunched together. When Contact ID's are on, an object can be tracked through multiple images and when coming to a completely new image, the contact ID's of the previous image will be fresh in the annotator's head. The ability to turn them off is useful when bounding objects that are bunched together or a larger object that is behind a smaller one. Sometimes when annotating in these situations the contact ID's can get in the way of the bounding box. Stretching an image is also a fairly useful feature, because it will stretch objects on the water making them easier to detect. This puts less strain on the annotator when trying to work quickly, since the goal is to detect objects as quickly as possible and work one's way through an image set. Enhancing the contrast has limited applications and can be either a gift or a curse depending on the lighting of the image set being annotated. If left on, lightly colored objects on the water may be indistinguishable from the background, which

will cause problems later on in the image set due to an unannotated object that suddenly appears in frame. The ability to enhance the contrast works perfectly when looking for an object within a dark background. Arguably the most useful, the ability to keep the bounding boxes from the previous image greatly increased productivity. Initially, the annotator would annotate an image, bounding all the objects found within and then move onto the next image, having to redraw all the same bounding boxes and relabel them with the correct information that was used in the previous image. Once the feature to retain the bounding boxes from the previous image was completed, no more relabeling needed to be performed. What makes this feature so powerful though is that it doesn't just randomly place the bounding boxes in the new image, but uses a horizon assisted pattern tracking algorithm to figure out where the previous bound box may have been. If the horizon stays similar to the previous image, generally the algorithm will be able to detect the pattern and place the bounding box directly over the object, reducing the time it takes to annotate an image even further. If not, repositioning the boxes isn't a major detriment, but it is still a massive improvement from trying to annotate without the feature at all. It is important to understand that the keep the bounding boxes feature will bring over any bounding box from the previous image that does not exist in the new image. This will bring the boxes into a new image as well as bring them into an image that has been already annotated. Due this it is always important to check if the feature is active or not when checking work. Lastly there are the shortcut keys that have been programmed into the annotation tool. These keys make it possible to use the cursor solely for drawing bounding boxes around objects while, fully utilizing the annotation tool using just the keyboard, increasing the efficiency of the annotator. The shortcut keys allow for switching between bounded objects, cycling through images and saving files, among other useful commands.

### **C. Maintaining the Annotation Tool**

Throughout the use of the annotation tool any bugs found or suggested improvements were recorded and were either fixed or implemented. This was done through the use of a GitHub library that was the repository for the annotation tools code. Unfortunately, due to a lack of experience with MATLAB GUIs and only a brief exposure to the annotation tool I was unable to contribute much towards the debugging of the code. Despite being a novice, having some background with coding and MATLAB proved useful when analyzing some components of the code. An instance of this would be when a new batch of images had just been received and they contained a boat that was to be annotated with a specific name

that did not currently exist in the title list. Initially it was assumed that adding the new name to the list would be simple and could be managed by making a quick alteration to the code, yet it turned out to be otherwise. This was because rather create the list values within the code, the titles were hard coded into the component when it was created in the GUI builder. Knowing where the list had been generated, the new title was added and then pushed to the repository. Knowing where to look for the list's code would not have been possible without a history in coding. After this alteration the new code was pushed to the GitHub repository where it could be accessed by all the other annotators.

## **V. COCLUSION**

With the task of annotating image sets there was not an ultimate goal other than try to exhaust all the image sets that needed to be annotated. This actually seemed impossible considering how image sets kept arriving from various testing runs or old images were rediscovered. Even though this task seems remedial it played a vital role in the effectiveness of the overall system and I think that there is a certain level of respect it deserves as a result of that. Since the annotation of images plays such a low level role, it allowed for annotators to learn more about the system they were affecting and how it worked as one unit. I was able to learn more about computer vision than I thought I would and am now better at writing code, since I have experienced some of the pitfalls of working through another person's code.

## **VI. ACKNOWLEDGEMENTS**

I would like to thank my mentor Michael Wolf and co-mentors Christopher Kanan and Christopher Assad for the time they put in to answering all my questions and assistance with this internship. I would also like to thank Professor McCudden making the CURE program possible for students within the LACCD District. I cannot stress enough how great this program is. I would also like to thank Professor Ramirez from ELAC for introducing me to the many programs at JPL and to Roslyn Soto. I would like to thank Roslyn and Eddie for all the hard work they put in at the JPL education office and that hope that they continue to inspire many individuals attending community college. Finally I would like to thank all the other interns I met and then friends that I made over this summer. I would especially like to thank my fellow intern Jose Padilla for all of his hard work and his insightful contributions to this project.