

Technology Review

Michael Rodriguez
Oregon State University
Group 67 - CS 444 - Fall 2017



Abstract

For this senior design project we will be partnering with the OSU Robotics department to help train an autonomous robot recognize regular objects. We will try to accomplish this by using ROS (Robot Operating System), trying different image tagging techniques, and improving data gathering.

1 INTRODUCTION

This project is formally called, "Deep Learning for Object Recognition on a Mobile Robot" and my role involves improving data gathering methods and using robust image tagging techniques. The project as a whole however will use neural networks to help train the robot to recognize regular objects and classify them into the proper class. In addition to the neural networks, the robot will also utilize online learning techniques which will help the robot build knowledge on pre-existing knowledge that it may already contain from looking at an item in the past. This is a place where we will be majorly invested in because the robotics department has yet to implement this type of learning on the robot. My role in this project is more tilted towards improving the techniques to gather data for the robot so that the learning can be more efficient and effective than it currently is. Therefore, in this technology review I will be looking at different technologies that are available to help me choose the best operating system for the robot, test out different image tagging techniques and research different data gathering methods.

2 OPERATING SYSTEM FOR ROBOT ACTUATION (OVERVIEW)

An operating system for a robot basically serves as the brain of the robot. The operating system provides the tools necessary to make the robot do an unimaginable number of different things. One of those things that it can help us out with is object recognition and classification. In this section, I will be discussing the potential operating systems that our PR2 robot could be running.

2.1 ROS

ROS (Robot Operating System) is the most commonly known and used operating system for robots. The main idea behind the creation of ROS was mainly to offer standardized functionalities that perform hardware abstraction and helping those in robotics development from having to continue rebuilding the wheel on their own. ROS is maintained by a company called Willow Garage which develops software and hardware for their robots which conveniently happens to include the PR2. Everything that is produced by Willow Garage is open source and has BSD licensing. Being open source is one of the big reasons that ROS has evolved so quickly and become so prominent in the field of robotics research and development. ROS provides resources that are organized into a hierarchical structure on disk. Two key concepts exist that stand out more than the others and that is the package and the stack. The package is a directory that has external libraries, APIs data, configuration files, nodes and an xml configuration file. The stack is a collection of packages that offer functionalities like navigation, positioning, mapping, and others [1].

2.2 Microsoft Robotics Developer Studio

The Microsoft Robotics Developer Studio is a Windows-based environment used to help create robotic applications. Some of the perks that RDS boasts is a lightweight REST-style, service-oriented runtime, a set of visual authoring and simulation tools, tutorials and sample code to help developers get started. RDS makes programming simple by allowing asynchronous input from the multiple sensors on the robot and output to actuators and motors. This would help our robot improve the speed on how fast it trains itself on an object by expediting the process. RDS also has a DSS Service Oriented Architecture that helps make it simple to interact with the robot through a web browser or windows-based application [2]. By being able to interact and respond to our robot through such an interface, it would benefit the feedback compartment of our project where the robot periodically checks in with us to confirm that it is targeting the right object.

2.3 NAOqi

NAOqi is the software that runs and controls the robot. The NAOqi framework which is used to program NAO solves basic robotic needs such as parallelism, resources, events, and synchronization. This framework is versatile in that it allows development in Windows, Linux, and Mac OS. NAOqi is similar to ROS in that its API functions for both C++ and Python and its modules are transparent between both languages as well [3]. Although this operating system seems to have the same major components that ROS has, it doesn't have nearly as much open source content and documentation. Additionally, NAOqi doesn't have a visual interface that we can interact with to verify that our robot is focusing on the correct object.

2.4 Conclusion

The operating system we decided on to use on the robot is ROS. This choice was a no brainer in that the PR2 robot we will be working with was actually developed by Willow Creek, the same guys that started and maintain ROS. In addition to being related to the same company, Oregon State University is the primary hosting site for ROS and every roboticist in the lab uses it. This will make it easier for us to reach out to faculty in the robotics department for help since we will be speaking the same robot language as them. Our client Professor Smart is also fond of ROS, so we ultimately did not have a say in what operating system we wanted to use.

3 IMAGE TAGGING - OBJECT SELECTION WITH INTERFACE

Image tagging involves using a marker system that involves a set of patterns that our robot can detect and familiarize itself with. The purpose of having a marker system is that it helps the robot orient itself in the environment that it presides in. For our testing purposes, we want the robot to pin the marker on its coordinate map of the room generated from its sensors. This specific spot on the map will serve as the testing center where we will be placing objects for the robot to train on.

3.1 Vision Markers

Vision Markers in computer vision are typically either squared or circular. The images contained within these two shapes just has to be something that the robot's camera is able to recognize. The size of the marker is not as super important, what is important is that you are able to maximize the distance from which the robot can successfully detect the marker. The shapes have individual advantages that make it better or worse depending on the goal at hand. Squares can be accessed uniformly using homography and unique vertices to orient the camera in a unique way. This allows for square tags to carry a larger symbolic data payload when used. Circular tags on the other hand can be accessed from any camera angle by computing from the whole contour surrounding the marker. This makes circular tags have better location and pose accuracy [4]. For our project, we want to be able to do a full 360 rotation around the object from varying angles so having circular tags and their ability to be recognized from all angles is a huge upside.

3.2 RFID

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags that are near or on objects. RFID works great with robots because they are a low-priced sensor that eliminates the problem of robots having difficulty visually identifying tags. It doesn't matter if it is light or dark out, whether the object is right side

up, sideways, or upside down, or even if its in a different room. The robot can be equipped with long range UHF RFID antennas that allow it to roam a room and see where the strongest signal from the RFID tags is coming from /citeRFIDtag. The robot is essentially playing the game hot or cold with the tag until it is able to locate it. The method of using RFID tags would be the most accurate and have the best long-range detection capability if implemented correctly.

3.3 Chemical Markings

Chemical markings can be used as a technique for storing temporary information in an environment. These markings eventually fade away once their purpose has been fulfilled for a certain amount of time. Similar to how dogs and cats mark their territories with pheromone markings, robots can use the same technique to send/receive a message. These short-term markings can be referred to as short-lived navigational markers (SLNMs). These SLNMs can be detected with sensors that our robot possesses and interpret its meaning by us telling it what exactly it should be associating with that specific chemical [5]. This technology isnt super practical for our testing purposes and is a little outdated. In most cases we want our marker to be permanent for the testing environment so that it can always use it as a frame of reference.

3.4 Conclusion

For the image tagging technique we decided on using the RFID tagging method. This method not only saves us the valuable computational resources needed to identify your average visual marker, but also allows us to utilize our robots sensors to their full potential. RFID tagging is capable of increasing our maximum distance from the robot to the tag while being more accurate and efficient in finding said tag.

4 DATA GATHERING

Data gathering is one of the most pivotal parts of this project. Without data, the robot has nothing to run through its neural network to classify objects. Gathering a training set of images large enough for the robot to train on is typically expensive on computational resources and time. In this section, I am going to explore different data gathering techniques to see which one suits our needs best.

4.1 Traditional Approach

The traditional data gathering approach involves first taking pictures of the object in its natural environment. Next, the robot either searches the environment for the object or it uses object instances gathered prior to the test to place the object in the environment in a natural way. The next step involves hand-tagging the pictures marking bounding boxes or tight outlines in each picture [6]. This last step repeats as the robot goes 360 degrees around the object gathering raw picture data for the object. This method is not the most accurate because the quality of the dataset relies heavily on the human input required to get the pictures. A similar data gathering approach is currently being used in the OSU robotics lab, which means there is plenty of room for improvement.

4.2 Synthetic Approach

A synthetic approach implemented by a group of students at Stanford University involves taking pictures of the object in a green screen setting. The next step is to build a new larger training dataset by perturbing foreground, background,

and shadow of the images taken using a probabilistic model. This allows for the true distribution of an object class to be modeled just as good as if it were to have been real data [6]. This approach saves a large amount of man hours that would have otherwise been needed for hand-labeling the pictures for the dataset. Our current lab resources do not offer the green screen technology needed to implement such a solution but it is something that we can bring to the attention of our client. The robot currently takes roughly about over an hour to gather its dataset to train on for one object so decreasing the amount of time needed to create a dataset would be fantastic.

4.3 Flashlight Approach

A data gathering approach that I believe will help us gather better quality pictures in our training set is called the flashlight approach. One of the biggest obstacles that we face when trying to classify an object is the amount of light exposure that it receives and having that affect the quality of the images we gather. To make the testing more consistent across all objects, I suggest that the robot hold a flashlight up to the object that it is capturing images on. This will allow the robot to get a much clearer image of the object and keep it consistent across all tests. This approach might produce a bit of glare depending on the material of the object but since the robot currently relies on human interaction with the hand-labeling, we can selectively choose to not include sections with glare in it. This last approach is a bit of a stretch but is something worth trying on at least one test to determine if it is a significant enough increase on the quality of the training set.

4.4 Conclusion

The data gathering method that we decided to go with is the synthetic approach. The main advantage of this approach is that the results you get from it are nearly the same as if you were to have gathered real life pictures of the object in optimal conditions. Only difference of course being the process is being done synthetically and saves our team a huge amount of time in the lab for when we want to gather new training sets on new or old objects.

REFERENCES

- [1] V. Mazzari, "Ros robot operating system," <https://www.generationrobots.com/blog/en/2016/03/ros-robot-operating-system-2/>, 03 2016, (Accessed on Nov 12).
- [2] Microsoft, "Welcome to robotics developer studio," <https://msdn.microsoft.com/en-us/library/bb648760.aspx>, (Accessed on Nov 12).
- [3] "What is naoqi framework," <http://doc.aldebaran.com/1-14/dev/naoqi/index.html>, (Accessed on Nov 12).
- [4] J. K. A. Pagani and D. Stricker, "Detection and identification techniques for markers used in computer vision," <http://vesta.informatik.rwth-aachen.de/opus/volltexte/2011/3095/pdf/6.pdf>, 01 2010, (Accessed on Nov 13).
- [5] R. A. Russell, "Odour detection by mobile robots," <https://books.google.com/books?id=O2FTjO9X2xQC&pg=PA117&lpg=PA117&dq=Marking+techniques+for+robots&source=bl&ots=uODndBxYdz&sig=jbNGH4OUaFGM2tnXnsL6dVSCTTM&hl=en&sa=X&ved=0ahUKEwjNwPHs-r3XAhVMwmMKHTIPBN0Q6AEIRTAH#v=onepage&q=Marking%20techniques%20for%20robots&f=false>, 07 1999, (Accessed on Nov 13).
- [6] B. Sapp, A. Saxena, and A. Y. Ng, "A fast data collection and augmentation procedure for object recognition," <http://ai.stanford.edu/~asaxena/robotdatacollection/stairdatacollection.pdf>, 2008, (Accessed on Nov 13).