Using Machine Vision to Follow a Moving Agent

By Zara Louis, Derek Hudson and Angelo Cioffi

Abstract

In this project, we address the problem of person detection when faced with distractions. This project involved programming the robot to follow a predetermined target through a light crowd. This program also allows the robot to reestablish its focus on the target if it loses track of him or her. We used the robot's Kinect to record its surroundings and track the target. The program utilizes color histograms to determine the location of the target and keep track of it even as the image and surroundings change. The robot also moves to follow the target while keeping an appropriate distance behind it. If the target stops moving, the robot stops moving as well so that it never gets too close. This project runs on ROS and utilizes code from OpenCV and the Building Wide Intelligence project.

Introduction

As robots become ubiquitous in daily life, it is more of a necessity for them to be able to identify people around them. In foreign environments, localization can be a difficult and time consuming process. Much like a person following a guide in a foreign city, a robot can follow an individual to the desired destinations and make note of the locations without having to self-localize. This would be less time consuming for the robot and potentially the programmer, and would ensure that the robot is more capable of adapting to its environment. Unlike trying to track a moving individual through street cameras, a robot has to move as well; its perspective is much more limited and can be easily obstructed by other individuals. Therefore, it is imperative that a robot be able to track and follow a person quickly and efficiently, before they become lost in a crowd.

We address this problem by creating a program that tracks a person through a crowd using their color histogram. The color histogram allows for fairly accurate tracking. Furthermore, our program allows the robot to follow at a safe distance without ever going too close to the target by programming it to stop moving when it gets to a certain distance from the target. The rest of our paper details some background work that we have done on this subject as well as past projects with material related to this specific task. It then describes the technical methods we used to program the robots and the errors we faced with these methods. Finally, it gives information about the experiments we ran and the conclusions we drew at the end of this experiment.

Background

There have been various scientific journal articles written about subjects such as person detection and machine vision, which we looked to for inspiration in this paper. The first article that proved to be useful in our project was published in the IEEE International Conference on Computer Vision journal. It was entitled “Density-aware person detection and tracking in crowds” by Mikel Rodriguez, Ivan Laptev and Josef Sivic. This article address person detection and tracking in crowds. It looks at video footage of large crowds and tracks individual people while ignoring the rest. This is similar to our goal of tracking a single person through a light crowd. This specific program accomplishes this task through estimations of crowd density using estimations of linear regression. Combining this with person detectors allows the system to localize individuals without having any knowledge about the structure of the surrounding crowd. In this way, the program can efficiently track a single person while filtering out the extraneous members of the crowd.

Another highly relevant article was from the proceedings of the 2006 IEEE/RSJ. Written by Takashi Yoshimi, Manabu Nishiyama, Takafumi Sonoura and various others, “Development of a Person Following Robot with Vision Based Target Detection” describes the creation of a robot that can hopefully one day serve as a personal assistant. This robot can avoid obstacles while still following the target. It avoids obstacles using ultrasonic sensors and follows the person using the color and texture of their clothes.This algorithm ignores the background distractions and more or less keeps its target on the robot. Instead of using cameras, this robot uses a stereo vision system. The job of the robot is to find the specified person and follow him/her person at their pace while avoiding obstacles by using ultrasonic sensors. The robot also has to redetermine the person's location it they loses him/her. The robot keeps the target in vision and checks for obstacles at the same time. To ensure that the robot is following at the correct distance, it constantly measures distance/direction and controls speed to ensure they are at the correct distance. When an obstacle is detected, the robot keep its vision on the person it is following but moves its body so that it goes around the obstacle. It does this by calculating the obstacle avoidance vectors. Clearly, this robot has very similar aims to this project, but using far more advanced technologies and more complex algorithms. However, many of the concepts can still be applied to the work being done in our project.

We have completed related work in the past with ideas very similar to this one. For example, we used the ROS turtlesim program to model real robots. In one such project, we made the turtle move towards certain, predetermined points. We also did a few projects in which we programmed the turtle to follow other agents in the grid, while disregarding the actions of certain 'distractors' (other agents). Finally, we experimented with machine vision by using the Kinect to locate a certain color within its surroundings and turn towards the object possessing this color. We will be using this ability to locate a certain color in this project so that we can track the certain target human being based on his or her color histogram while ignoring other distracting factors.

Technical Approach

We used a two-pronged approach to detect people and keep track of them as they move through a light crowd. The first of these approaches detects people: the HOG Person Detection software. HOG stands for Histogram of Oriented Graphics and is a method for compressing an image of potentially tens of millions of values down to a vector of a thousand or so values that can more readily be worked with. This list of values is called the HOG Descriptor. The person detection that comes with OpenCV is based on HOG and works by taking many subimages within the image and checking whether each subimage contains a person. The subimage is determined to contain a person by comparing its HOG Descriptor with a particular HOG Descriptor. The bounding box for the subimage is then added to a vector of rectangles and, after all potential subimages are checked, the resulting vector of rectangles is given back to the program. The particular HOG Descriptor mentioned above was gradually trained to match as many people in many different poses by a machine learning algorithm specialized for the task named Support Vector Machine.

The other major key is the color histogram, which determines which person to follow in a light crowd. A color histogram is basically a vector where each position acts as a bin and contains the frequency of shades of a color within the ranges of that bin. This can be useful because comparing histograms is a relatively simple when compared with other, more complicated varieties of image recognition. Because this method works solely on color, a person tracked when in a particular pose will remain tracked as long as their pose did not significantly influence their color histogram. Color histograms can be compared a variety of ways, one of which is the intersection, which takes the sum of the minimum value for each position in the histograms, and correlation, which attempts to find how well the curves fit each other. We used the correlation method in this project. The rectangle with the best-fitting histogram would then be used to move the robot. If the selected person’s rectangle was too far to the left or right of the centerline of the image, the robot would turn left or right. If the area of the rectangle associated with the selected person was shrinking, the robot would move forward, and if the area was growing, the robot would backtrack.

Experiments and Demonstrations

As our in-class demonstrations show, we were overall successful in creating a program for the robots that would allow it to follow a predetermined target from a reasonable distance, despite other agents being in the screen. We did not conduct formal experiments, but instead tested the robot in various situations to see how it reacted. We first tested the robot's ability to detect and track a single person. The robot was extremely successful in this task. The person detection algorithm, although slow, was overall quite accurate. We then test its ability to track a single person when another person enters the robot's field of vision. The robot was sometimes unable to recognize the second agent as a person, but it was overall effective at determining who to follow and staying connected with them even when there were other agents to distract it. Finally, we tested the robot's ability to follow an agent after losing sight of them due to an obstruction from another person. The robot was able to successfully reestablish its connection with the person it was tracking and continue accurately following them.

Although our experiment was successful overall, there were a couple things we could have improved. The first is the accuracy of the robot's person detection. There are times when the robot does not detect a person, despite them being in the camera's vision. While the program was effective when it came to detecting and focusing on its target, the HOG detection sometimes missed other humans in the Kinect's screen. The second issue with the resulting program is the slight lag. This is primarily due to the code being used in the person detection. The inefficiency of the code slows down the robot's movements and makes the program far less accurate. This is what causes the robots movements to be somewhat slow and less fluid than they could be. Some of the issues with this experiment

could be due to the technology being utilized. While other experiments and projects use expensive cameras and sonar technology, the robots being used in this project use the Kinect camera. This slightly less advanced technology could have contributed to some of the issues being seen in our resulting program. The Kinect has a smaller field of vision and produces a far less detailed final image. It also lags more than other cameras would.

Conclusion and Future Work

In conclusion, the HOG person detection works well but lacks the speed to make this program practical and run well in real time. Furthermore, the color histogram correlation comparison technique is effective for tracking a target from screen to screen. We believe that, overall, our project was successful, but still could be worked on more.

One major problem we faced when completing the code for this project was attempting to use a 3-D matrix for the histogram instead of 3, one dimensional matrices. This likely reduced the overall accuracy of the histogram, but could have improved performance. It also allowed us to normalize histogram matrices, which meant we could compare images without worrying about relative size. While this may have reduced overall accuracy, it allowed us to complete the project efficiently while still getting correct results.

Our most important future task would be to improve the overall functioning of this program. While we were happy with the results of this experiment considering the time constraints, there were definitely areas where we could improve the functioning of the robot. For example, we could improve the robot so that it moves faster towards its target instead of constantly stopping and re-adjusting. We could also improve the person detection software so that it is more efficient and runs more fluidly. The improved program would be more effective at identifying humans and would allow the robot to react faster. Simply improving this person detection program would make using the robot more accurate.

Upon making the code more accurate and efficient, we would want to apply this program to other types of robots and technologies. One important application would be to drones. Creating drones that can effectively detect a target and track it through a crowd and various different settings would be very useful for the future. There are many other applications for person tracking and following, such as robotic security guards and when analyzing video footage. We believe that there are a lot of extremely useful applications of this program.

References

- OpenCV Person Detection Tutorial: http://www.magicandlove.com/blog/2011/08/26/people-detection-in-opencv-again/
- Explanation on how the HOG Person Detector Works: http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/
- More in-depth explanation of how HOG works: http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf
- OpenCV HIstogram Calculation Tutorial: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_calculation/histogram_calculation.html
- OpenCV Histogram Comparison Tutorial: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_comparison/histogram_comparison.html
- OpenCV Tutorial on getting matrix of specified rectangle: http://docs.opencv.org/2.4/doc/tutorials/core/mat_the_basic_image_container/mat_the_basic_image_container.html
- BWI file on how to move the robot and see its current location: https://github.com/utexas-bwi/segbot/blob/master/segbot_navigation/src/move_base_interruptable_server.cpp
- Journal article about person detection in a crowd: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6126526&tag=1
- Journal article about robots using machine vision to follow a target: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4059266