Golf Bag Carrier Robot Computer Vision

Akshay Gupta[1] **and Nicholas Gloria**[2]

[1]Stanford University, Aeronautics and Astronautics, Stanford, CA, 94305, akshaygu@stanford.edu
[2]Stanford University, Aeronautics and Astronautics, Stanford, CA, 94305, ngloria@stanford.edu

Introduction

A new entrepreneurial venture at Stanford is aiming to launch an athletic robot product. The idea is to create the ideal golf caddy: an autonomous robot that will carry a golfer's bags and follow him around the course. This leaves the golfer to focus on playing the game and walking the course unhindered. For older golfers, this product will enable them to walk and enjoy the sport without dealing with a taxing effort of carrying around a golf bag.

This project focuses on the navigation aspects of the autonomous golf caddy. We plan to focus on two problems, classifying terrain seen by the visual sensors, and detecting and tracking objects within the image. Machine learning will allow the robot to use a simple camera to identify and track the target golfer to follow, identify untraversable terrain, and identify areas of the terrain that should not be traversed (e.g. putting green, tee box, sand traps).

While existing products are able to track golfers using bluetooth chips, it is often desirable for to not follow the golfer into areas such as the tee box, green, or bunker, or to avoid other obstacles along the way. Adding computer vision to the product will enhance the user experience.

Related Work

Computer vision techniques have been developed for the specific task of tracking people. The PFinder or person finder algorithm incorporates a priori information of the nature of the human body. Using Maximum A Priori (MAP) estimation, "blobs" or Gaussian shapes are fit to the torso, legs, hands, feet, and face of humans[1]. This is enhanced by dynamics modeling of the person and accounting of the camera rotation through image-to-image registration pre-processing.

Cross-country all-terrain navigation has proven challenging for efforts such as those of Manduchi et al.[2] Negative obstacles involving mud, snow, water, or ditches are especially difficult to detect. However, there has been a string interest in this ability, for example from the Department of Defense. Manduchi et al. use stereo color cameras and single-axis ladar for autonomous navigation. Classification based on the color cameras is utilized. While the robot in this course does not have access to ladar or stereo cameras, it can utilize color classification and take care to properly characterize negative obstacles.

Another approach is taken by Weiss et al. to augment autonomous navigation by indentifying the type of terrain currently being traversed[3]. A Support Vector Machine (SVN) is used with the accelerometer power density to classify the terrain. While the vehicle currently does not have an onboard accelerometer, this provides a potential future avenue for augmentation.

In this paper by Felzenszwalb et al. an important characteristic of the method is described is its ability to preserve detail in low-variability image regions while ignoring detail in high-variability regions. The algorithm classifies an image into global regions, drawing out boundaries for terrain and objects within the image. It is able to detect boundaries based on variability of pixels and its surrounding region while also ignoring erratic boundaries by ingnoring variability in a more global region and thus segments images really well.[4]

Another related paper was by Ng et al. that was able to avoid obstacles at high speed, with using just a monocular vision alone. They trained an algorithm using supervised data, of images labelled with depth details, and then used is to predict depth. This would be helpful in the current setup we have, which is essentially just one camera.[5]

Datasets and Features

The robot uses solely a stream of camera images for its operations. For the purposes of this project, images were collected from test operation of the actual robot vehicle with the camera to be used on the prototype. The prototype was manually piloted around, simulating operation in service to a volunteer test subject golfer. 30,000 high-resolution images were captured for use in this project.

Some supervised data was also provided for the training of the classification algorithms used. Images were hand-classified by color codes into the following categories.

- Traversable terrain [red]
- Pavement [yellow]

- Untraversable terrain (including sand traps, water hazards, obstacles, background) [purple]
- Putting green or tee box [green]
- Target golfer [blue]
- Sky/clouds [black]

These classifications were chosen with the intended algorithm in mind to have distinct features with the available pixel color and color gradient information. However, in the end, only traversable terrain area needs to be identified, which would consist of the "Traversable terrain" or "Pavement" classes. Thus, only the accuracy with respect to these two classes is directly valued.

An example of a supervised training example is shown in Figure 1. Only a handful of training examples were generated, as the goal was to show the viability of the algorithms as opposed to spend time developing an efficient way to hand-classify data. 11 images with 700,000 pixels (720 x 1080) each thus provided about 8.5 billion data points. However, for training of the actual product, a software must be developed to allow efficient hand-classification of examples, which would make the algorithm viable.

Figure 1. Example supervised training example. Images were hand-classified using image editing software and color codes.

Methods

A number of machine learning algorithms will be tested to develop the image classification required for the product. First, however, the classification objective must be clearly defined. This segmentation will then be used with a controller and path-planning algorithm to guide the caddy robot.

K-Means based Segmentation

The most basic algorithm conceived was to segment a set of images into a number of categories based on the color values using the K-Means algorithm. The segments generated can then be mapped to the desired segment groups. However, this algorithm is not robust to color variations by itself. But after running basic K-Means a filter can then be extracted based on the pixel values from clusters. This filter is then applied to the image, by comparing all pixels and there euclidean distance in the color space (RGB values) from the selected cluster's color value and then selecting based on a threshold. This results in filtering of traversable regions from the image.

Terrain Classification Using a Convolution and Softmax Regression

One algorithm explored for classifying terrain is a two-layer algorithm that uses a convolutional layer and a softmax regression. Classes are defined as explained in the Datasets and Features section.

Pre-Processing

For the purposes of making the algorithm computationally lightweight enough to run in real time on a limited computer, the images are drastically reduced in resolution from 720 x 1080 to 112 x 200.

Convolution

Segmentation can be vastly improved by looking at a small context around a pixel as opposed to merely the pixel itself. To this end, images were convoluted into a new data shape. The image would normally be $w \times h \times 3$. However, by including values within r pixels of the pixel of interest, the data is reshaped to $w \times h \times 3r^2$. This then provides the information of each pixel's context to a learning algorithm at the cost of increased data size. The algorithm now has access to gradients in color in addition to the color itself. This is particularly valuable for detecting egdes and borders, such as the border between a putting green and the fairway.

With the reduced image resolution of 112 x 200, a pixel convolution size of $r = 15$, each pixel can be influenced by pixels roughly 5% of the frame width away.

Softmax Regression

A simple softmax regression can be used to map between convoluted pixel information to classes. This must be trained with supervision, but this is achieved with the hand-classified learning examples. The softmax regression linear step maps a flattened state vector of dimension whr^2 to a set of class probabilities for k classes.

Results

Learning Algorithm Trials

We performed experimentation with two different classes of algorithms based on our classification goals. Bounding box generation for object detection and tracking K-means clustering for image segmentation tasks

Bounding Box Generation

Another algorithm that was employed was an off the shelf object detector from Tensorflow called Mobilenet SSD. We used a pretrained detector on our camera feed to obtain bounding boxes on the golfer, which was able to effectively track as we traversed around the field while detecting other obstacles such as trees, golf carts etc.

Figure 2. Bounding boxes generated using MobileNet SSD Single-Shot Multi-Box Detector to track golfer

K-Means based filter

We ran K-means clustering, and then selected one of the clusters of interest. The image was then run through this and all the pixels that lied in close vicinity of the cluster, were painted over with the same color. This was able to segment the images into regions of interest.

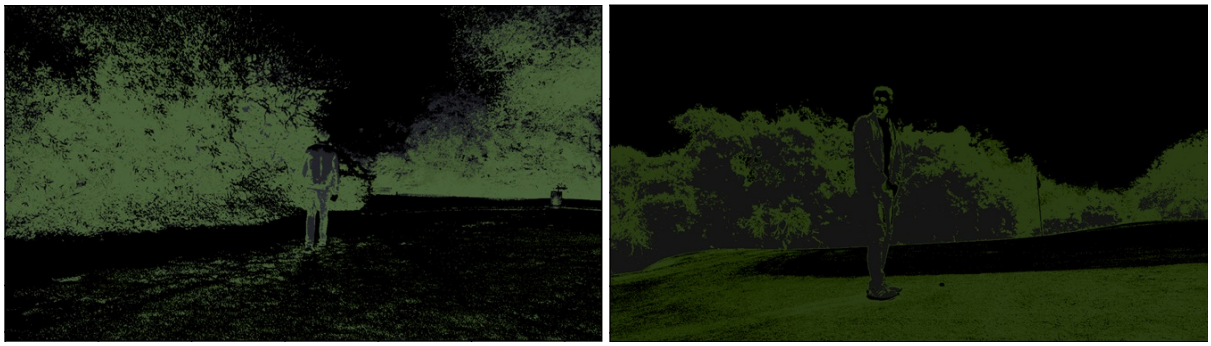

Figure 3. K-Means based pixel classification

Felzenszwalb segmentation

We used scikit to implement the Felzenszwalb segmentation algorithm. This generated boundaries in the image and segmeneted the image into major blobs based on the boundaries. These boundaries were effective in classifying out traversable and intraversable terrain. Adding pooling on top of this filter proved be an effective segmenting algorithm.

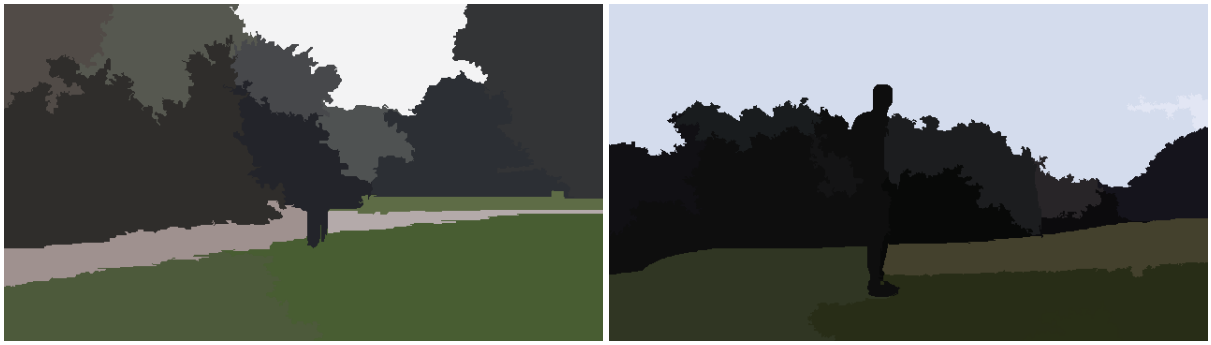

Figure 4. Felzenszwalb segmentation followed with cluster pooling

Convolution and Softmax Regression

After training the convolution and softmax regression algorithm on the handful of images in the training set, the classifier is tested on one of the many remaining images collected during field testing of the prototype. A couple of notable examples are shown below showing interesting behavior of the algorithm.

Figure 5 shows an image with the target standing in front of a putting green. The green is indeed successfully identified. Furthermore, red patches indicating traversable terrain are also identified. The sky is also properly identified. A clear indication that the gradient is being used is evident from the background classification, which picks up the borders of the green and sky, for better or for worse.

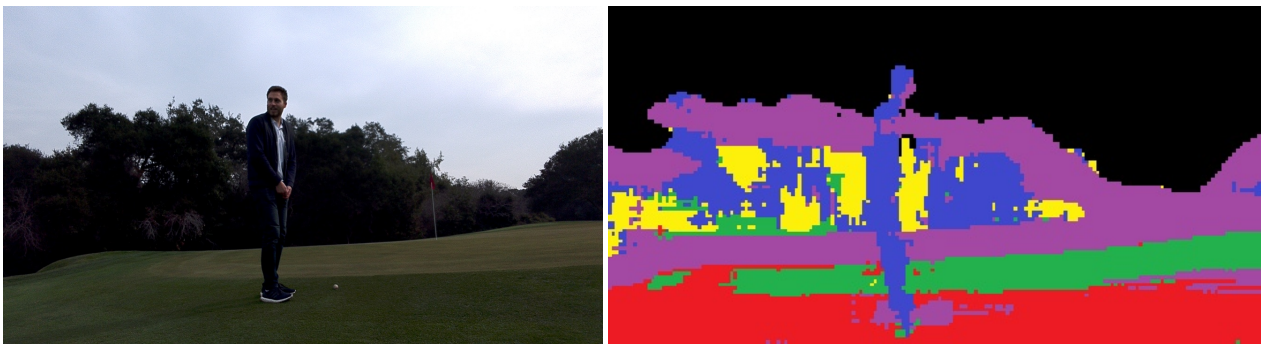

Figure 5. Test classification example. The classifier successfully detects the green, edges of the background, sky, and parts of the traversable terrain, but mistakenly adds a background layer surrounding the green.

Another test case is shown in figure 6, depicting the golfer walking towards a cart path. A large swath of traversable terrain is successfully identified, although the cart path and a strip of grass along it are improperly identified. Once again, the target is also classified. Areas of the background are inconsistently classified into sky and a false background layer is created between the cart path and fairway with a width of approximately the convolution width.

In the context of the actual product, the goal of the classifier is solely to identify areas of traversable terrain and cart paths to guide the cart. Whether the background is successfully classified into sky or trees is irrelevant. Thus, the softmax regression is moderately successful. The algorithm's viability has been demonstrated, although further training examples are required to demonstrate high accuracy.

Conclusion

The computer vision algorithms explored with the project show promise in meeting the requirements of the autonomous golf caddy robot. Tracking the target person has been successfully demonstrated with use of the Mobilenet SSD Single-Shot Multi-Box Detector algorithm, utilizing a neural network. Classifying the terrain has proved to be more difficult due to the subtle differences in terrain, such as the difference between fairway and green. For this purpose, a number of approaches have been explored. Felzenszwalb edge detection proved to provide useful functionality by pooling classifications within detected edges. A simple unsupervised K-means segmentation algorithm also demonstrated limited performance in identifying traversable terrain. However, the convolution and softmax regression algorithm proved to be the most promising in identifying

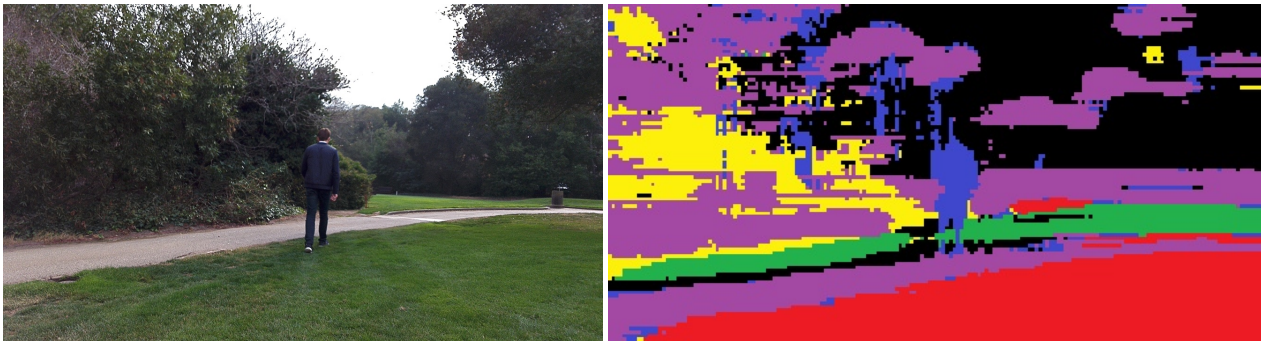

Figure 6. Test classification example. The classifier successfully detects the near traversable terrain and target. Areas of the background and sky are misclassified, as well as the cart path.

the relevant classes with robustness, while still being computationally lightweight enough to run in real time with a limited processor. However, a more efficient means to create supervised data and provide more training examples must be improvised in order to expand the accuracy and robustness of the classifier.

Contributions

Both authors collaborated to collect data in the field with the product. Akshay Gupta focused on the human tracking algorithm and implementing the bounding box tracking. Furthermore, he investigated K-means segmentation and Felzenszwalb edge detection and associated pooling as approaches to classification. Nicholas Gloria focused on the classification problem, developing the two-layer convolution and softmax regression algorithm and devising the means of generating supervised data.

Code

The repository containing all code for the project can be found at the following link. https://drive.google.com/open?id=1t3cRiDiwy8AAfzsRdrktjciCVB-Yk-Tb

References

1. Wren, C. R., Azarbayejani, A., Darrell, T. & Pentland, A. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**, 780–785 (1997).
2. Manduchi, R., Castano, A., Talukder, A. & Matthies, L. Obstacle detection and terrain classification for autonomous off-road navigation. *Auton. robots* **18**, 81–102 (2005).
3. Weiss, C., Frohlich, H. & Zell, A. Vibration-based terrain classification using support vector machines. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 4429–4434 (IEEE, 2006).
4. Felzenszwalb, P. F. & Huttenlocher, D. P. Efficient graph-based image segmentation. *Int. journal computer vision* **59**, 167–181 (2004).
5. Michels, J., Saxena, A. & Ng, A. Y. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, 593–600 (ACM, 2005).