Application of Deep Learning Methods on Autonomous Tracking of a Moving Body

Elena Pradhan
Application of Deep Learning Methods on Autonomous Tracking of a Moving Body

by

Elena Pradhan

An Honors Capstone

submitted in partial fulfillment of the requirements

for the Honors Diploma

to

The Honors College

of

The University of Alabama in Huntsville

05/01/2019

Honors Capstone Director: Dr. Chang-kwon Kang

Assistant Professor of Mechanical and Aerospace Engineering Department

Student

Date

05/01/2019

Director

Date

5/1/2019

Department Chair

Date

5/9/19

Honors/College Dean

Date
Honors Thesis Copyright Permission

This form must be signed by the student and submitted as a bound part of the thesis.

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Elena Pradhan

Student Name (printed)

[Signature]

Student Signature

05/01/2019

Date
Application of Deep Learning Methods on Autonomous Tracking of a Moving Body

Elena Pradhan
Department of Mechanical and Aerospace Engineering

Abstract - Determination of autonomous, real-time position of a moving object can provide useful data for the development of control and guidance algorithms of autonomous aerial, underwater, and terrestrial vehicles. Recent advancement in vision-based tracking methods and machine learning approaches have enabled autonomous, real-time object motion tracking approaches that are cost effective. This paper assesses the application of vision based localization algorithms and machine learning recognition algorithms to generate trajectories from analysis of video recordings of moving objects. The object recognition is a two-part process. Firstly, Simple Learning Iterative Clustering is used to produce super pixels at each individual frame of the video, followed by the use of regional convolutional neural network for object detection to recognize the super pixel that contains the brightly colored markers. In addition, the feasibility to use this method as well as its potential application is discussed.

I. Introduction

In the advancement of aerial vehicles, a higher level of autonomy is desirable for landing, surveillance, refueling, obstacle detection, accurate and fast localization. Localization is the determination of position, frame of reference and orientation of an object. The combination of Global Positioning System (GPS) and Inertial Navigation System (INS) is currently the most popular multi-sensory fusion method used for localization [1]. Beard et al. [1] were able to prove the use of GPS and INS to provide altitude measurements sufficiently accurate for automatic control of an unmanned aerial vehicle (UAV). However, the reliability of a localization solution obtained using a GPS/IMU is highly dependent on satellite visibility and minimal drift. A GPS tends to require at least three satellites to be visible and is stable over a long time. An IMU is prone to drift but is stable over a short term [2]. Sensors such as a barometer can complement short-time stability of Inertial Measurement Unit (IMU) and location specificity of GPS. In such systems, the GPS receiver provides three-dimensional velocity and position. An IMU provides three-dimensional linear accelerations and angular rates and the barometer provides the altitude of the vehicle indirectly. The navigation processing unit performs real-time navigational computation with these measurements. Vision-based sensors and computer vision algorithms offer a promising solution to the problems posed by the existing technology. Color and hue information from a forward-looking camera can be used to segment the skyline and accurately provide altitude estimation as well predict roll and pitch angles of an aircraft [3]. The application of a vision-aided nonlinear model predictive controller on visual information acquired real time has been proven feasible for localization [4]. Olivares-Mendez, Kannan and Voos show different approaches for the use of high speed processing power to generate real time locations. These approaches use real-time high speed analyses on visual data to provide prediction on positioning and orientation. The recent advancements in artificial intelligence (AI) provide a new approach to autonomy in vision-based systems. The use of AI provides a solution that would eliminate the need for human intervention and human error. Neural networks for computer vision would make all further decision making using the localization
Among existing methods of image segmentation, grouping perceptually meaningful atomic regions into superpixels is useful to decrease computational complexity in the subsequent processing. Mean-shift clustering [5], Quick-shift, Normalized cuts [6], k-means, and SLIC [7], have been considered as candidates for super pixel generation shown in Table 1. The criteria used for the comparison between these algorithms include speed for segmentation, computation complexity and accuracy of segmentation. Segmentation speed refers to time in speed required by these algorithms to complete segmentation. Computation complexity shows the relationship between times taken for segmentation to the number of pixels. O(N) complexity shows the linear relationship between time taken and number of pixel N. Segmentation accuracy refers to the algorithm’s ability to segment clear boundaries based on class of the object.

SLIC is chosen for the computational complexity that is linear to the number of pixels in the image, O(N), while offering accuracy comparable to aforementioned algorithms. SLIC is a relatively new clustering algorithm, which is an optimized adaptation of the pre-existing k-means clustering algorithm. K-means algorithm computes the distance between each cluster to every pixel in the image, whereas SLIC limits computation to 2S×2S size around the cluster. This optimizes the process by reducing redundant computations. SLIC uses the five-dimensional [labxy], where l is the numerical value for lightness, a represents the value for green-red component, b for the blue red component of the image, and xy represents spatial position of the pixel in the image. This is after normalization for the scale of the image accounts for both color similarity and proximity while clustering pixels into a superpixel.
Table 1: Comparison between super pixel generation methods

<table>
<thead>
<tr>
<th>Superpixel Algorithms</th>
<th>Superpixel Algorithms Segmentation Speed for a 320×240 Image (s)</th>
<th>Complexity</th>
<th>Segmentation Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph-shift</td>
<td>1.08</td>
<td>O(N logN)</td>
<td>74.6%</td>
</tr>
<tr>
<td>Quick-Shift</td>
<td>4.66</td>
<td>O(dN2)</td>
<td>75.1%</td>
</tr>
<tr>
<td>Normalized Cuts</td>
<td>178.15</td>
<td>O(N3/2)</td>
<td>75.9%</td>
</tr>
<tr>
<td>SLIC</td>
<td>0.36</td>
<td>O(N)</td>
<td>76.9%</td>
</tr>
</tbody>
</table>

2. Regional-Convolution Neural Network for object detection

Convolutional Neural Network (CNN), R-CNN and You Only Look Once (YOLO) were considered for this application as they are commonly applied deep learning algorithms for object detection. A CNN works by passing an image (an array of pixels) through a series of convolution layers, which includes filters. These filters extract features from the image and periodic pooling layers. Pooling layers decrease the spatial size of the image and reduce necessary convolution layers. The output is then passed through a fully connected layer, which flattens the matrix to a fully connected layer and classifies the image with a certain level of confidence. A softmax classifier is used to calculate the loss function. The loss function back propagates into the neurons to adjust the bias and minimize the loss function. A CNN algorithm is only able to classify one object per image. Therefore, it was not deemed optimal as most applications have larger volume of marker and multiple kinds of marker that need localization.

R-CNN utilizes CNN. Furthermore, R-CNN goes beyond classification of the object class and isolates the position of the object on the image. R-CNN identifies Regions of Interest (ROI) then extracts CNN classification from each ROI shown in Figure 1. The accuracy of a deep learning algorithm relies on the quantity and quality of training datasets. YOLO, on the other hand, splits a picture into S×S bounding box and calculates the probability of an object being present in the box. Although an order of magnitude faster than most object detection algorithms, YOLO has difficulties separating small objects. This makes YOLO less desirable for this application.

Figure 1: RCNN Architecture
III. Experimental Setup

This section discusses the necessary equipment, setup, data acquisition methods, pre-processing performed on the image, and the source and development of the algorithms for replication of this experiment.

A. Equipment and Setup

The sensors of choice for this experiment were cameras (iPad cameras) as they are information rich, lightweight and low-price vision-based sensors. The cameras operate at 25 frames per second (fps). The algorithm is expected to encounter moving frames as the vehicle attempts to detect a marker and make decisions based on the marker. However, to reduce time and cost of the experiment associated with constructing and flying a UAV each time to collect data, the camera frame is kept fixed while the marker is displaced by attaching multiple markers to the leg of a test participant. In order to ensure that the marker is always visible and three dimensional position is always obtained, 3 cameras are placed around the moving marker, orthogonal to each other shown in Figure 2. In addition to simulating a moving frame expected to be encountered by the UAV, testing is conducted at different lighting levels to collect diverse data for training the deep learning algorithm.

![Figure 2. Setup with three orthogonal cameras for 3-D positioning](image)

B. Data-Acquisition

The application of this approach to computer vision for navigation and detection requires real time image processing capabilities. This will involve the use of powerful and fast data acquisition software such as Data Acquisition Toolbox by MATLAB. However, to train the deep learning algorithm, prerecorded videos were used in post processing. Individual frames are extracted from the video using a cv2 library in python. This approach provided a large range of datasets per each test run.

C. SLIC

Python’s skimage library [9] provides all traditional clustering algorithms including SLIC. This algorithm is used in combination with the data acquisition algorithm. Hence, as each frame is extracted from the video, the frame goes through hue adjustment which is then converted to an array of super pixels. The result of this algorithm is an image where the markers are separated into a superpixel different from the surrounding superpixels ready to be fed into the object detection algorithm as shown in Figure 3.
5. Object Detection

The open source machine learning library developed by the Google Brain Team within Google’s AI [10], Tensor Flow, is used for the Object Detection. Tensorflow Object Detection Application Programming Interface (API) is a tested R-CNN open source framework. Using Common Object in Context (COCO) API, a large image dataset meant for training object detection and segmentation algorithms, Tensorflow Object Detection API is able to train accurate machine learning algorithms. As an example of a test image trained using COCO API on a platform called jupyter notebook is shown in Figure 4. The dataset can be replaced with a custom dataset containing images of objects to be recognized/localized [8]. In combination with the algorithm for SLIC, this is a computationally efficient method to recognize the marker and further localize it.
**IV. Loss Function**

The loss function is a function of difference between the true value and estimated value of a parameter. In the context of machine learning, determination of the loss function is the method used to evaluate how well an algorithm models a dataset. Therefore, a minimal loss function is characteristic of a well-trained algorithm. For Object Detection deep learning algorithms, the loss function is modeled by the cross entropy method, also known as the log loss. An ideal object detection algorithm will have a log loss of 0. For the Tensorflow Object Detection API, the loss function is to be calculated for an increasing number of datasets. The goal of this object detection instance is to minimize the loss function to a value less than 1 so as to make the detection reliable. A comparison between loss function trends is made between methodology with and without pre-processing.

\[
\text{Log Loss} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i) \log(1 - p_i),
\]

where \( p_i \) is the model’s probability that any observation is in the expected class, \( y_i \) is the binary indicator of whether the class label is correct for the observation made.

**V. Results and Discussion**

A small set of 14 images were used to train the existing object detection algorithm in tensor flow. Loss function iteration of 700 steps is conducted to detect a class named marker. The algorithm is being preliminarily tested where the confidence of the algorithm is between 20-32\% (Figure 5). Rigorous training with larger numbers of loss function iteration is necessary to increase this confidence level.

![Figure 5. Test instance with 22% confidence](image)

However, the goal to minimize the loss function below 1 and prove significance is achieved and a plateau in the loss function value for both methodology is observed. In comparison (Figure 6), the loss function value with pre-processing plateaued at a lower range of 0.5-0.6 while the method without pre-
processing stayed at 0.8. This is useful if there are large number of training instances. For example, with UAV images at various terrains, the training time and sample size can be minimized by simply adding a pre-processing step. Preprocessing will optimize and makes training the algorithm more effective overall.

![Loss function over time](image)

**Figure 6. Loss Function Comparison graph**

### B. Accuracy

Motion capture technology is capable of tracking markers and generating extremely accurate trajectories of their motion. Motion capture cameras and simple IP cameras can be used in parallel to compare the trajectories generated by a motion capture system vs. the deep learning marker tracking algorithm using simple IP cameras. This system will allow one to quantify the accuracy of the results produced by the deep learning technique.

### C. Feasibility

Preliminary training and testing using the object detection algorithm on a machine with 2.3GB of RAM and 3GHz speed suggests that this method is feasible for real time applications if the algorithm is used alongside SLIC and pre-processed to decrease computational complexity. Further analyses on exact computational speed of the computational algorithm. This is required to determine whether the cost of the high speed processor and storage for large volumes of data is a good tradeoff for the ability to localize without being fully reliant on GPS or motion capture technology.

### VI. Concluding Remarks

From the results obtained, when the algorithm is refined and adapted with upgraded hardware features, this technique of using deep learning to autonomously track markers could be an adequate replacement for expensive contemporary equipment such as motion capture technology. Apart from decreasing the cost of indoor experiments regarding biomechanics and UAV controls, this technique provides promise in accurate localization outdoors when combined with other sensors. This is especially useful in
remote operations involving dangerous terrain or outer space. The subsequent steps to implement this technique involves extensive training of the existing algorithm with some datasets in different backgrounds, lighting, different fields of view, as well as development and integration of a spatial localization algorithm for the generation of a trajectory. Extensive training will minimize the loss function and increase the confidence of detection. In addition to minimizing the loss function, use of a variety of backgrounds will make the technique applicable to aerial surveillance. Since the technique was tested in the simplest possible setup for the purpose of proof of concept, the hardware upgrade is necessary for the technique to be implemented in a real time application. This will require a powerful data acquisition system, a high speed network, increased processing power, as well as moving vehicles, depending on the application. In order to increase the level of autonomy in an aerial vehicle, this determined hardware will have to be mounted on-board the vehicle. This will require another feasibility study where the gain in processing speed is compared to its implication on mass and cost budgets of an aerial vehicle. The ultimate goal of this technique is to be able to localize objects for autonomous navigation, and to conduct surveillance with a UAV. A secondary application could be to generate the trajectory to track movement of the human body, a biomechanics application.

Therefore, the final algorithm should either combine GPS/IMU measurement to calculate relative motion and trajectory or use a fixed reference point of co-ordinate (X0, Y0, Z0) in the camera frame. The inclusion of this ability to generate trajectory will expand the application of this algorithm.

VII. Acknowledgement

The author would like to thank Dr. Chang-kwon Kang in the Department of Mechanical and Aerospace Engineering and Dr. Shannon L. Mathis in the Department of Kinesiology at University of Alabama in Huntsville (UAH) for their advisory role in this project and the UAH Honors College for funding this project.
VIII. References


