https://doi.org/10.2352/ISSN.2470-1173.2020.16.AVM-299

Object Tracking Continuity through Track and Trace Method

Haney W. Williams
Department of Systems Engineering
Colorado State University
Fort Collins, Colorado, USA
Haney.W.Williams@gmail.com

Steven J. Simske, PhD
Department of Systems Engineering
Colorado State University
Fort Collins, Colorado, USA
Steve.Simske@colostate.edu

Abstract

The demand for object tracking (OT) applications has been increasing for the past few decades in many areas of interest: security, surveillance, intelligence gathering, and reconnaissance. Lately, newly-defined requirements for unmanned vehicles have enhanced the interest in OT. Advancements in machine learning, data analytics, and deep learning have facilitated the recognition and tracking of objects of interest; however, continuous tracking is currently a problem of interest to many research projects. This paper presents a system implementing a means to continuously track an object and predict its trajectory based on its previous pathway, even when the object is partially or fully concealed for a period of time. The system is composed of six main subsystems: Image Processing, Detection Algorithm, Image Subtractor, Image Tracking, Tracking Predictor, and the Feedback Analyzer. Combined, these systems allow for reasonable object continuity in the face of object concealment.

Keywords: *Continuous Tracking, Object Tracking, Image Subtraction, Semi Supervised Learning, Trajectory Prediction, Surveillance.*

1. Introduction

Object tracking is an active research area in computer vision thanks to the increasing demands in the Intelligence, Surveillance and Reconnaissance (ISR) applications and the Autonomous Vehicles Systems (AVS). The tasks of computer vision object tracking consist of: Image sensing, image enhancement, background extraction, object classification, tracking of the object of interest and feedback analyzer. To facilitate the development process, a visual sensing system is used; however, it is recommended to use a quadruple redundant system such that they complement each other. This quadruple redundant sensory system is composed of LiDAR, Visual Camera (RGB), and Thermal Camera, and RADAR sensor the performance of each system is shown on Figure 1. The Web-graph on Figure 2 [1] shows the combined performance of the four sensors and shows how they complement each other.

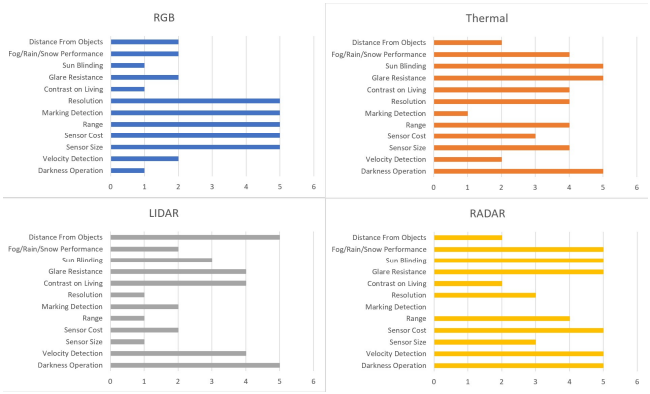


Figure 1 – Performance of the various sensors

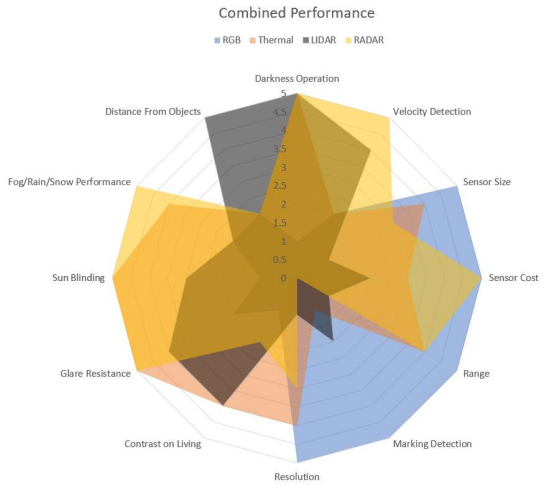


Figure 2 – Sensors Combined Performance

2. System Overview

The proposed system will span across three phases:

- Phase-1: A single static camera that observes the seen,
- Phase-2: Multi static cameras that slightly overlap their fields of view,
- Phase-3: A Moving camera, where the system controls the 6 degrees of freedom of the camera source assuming it fixed on rigid body as shown on Figure 3 below.

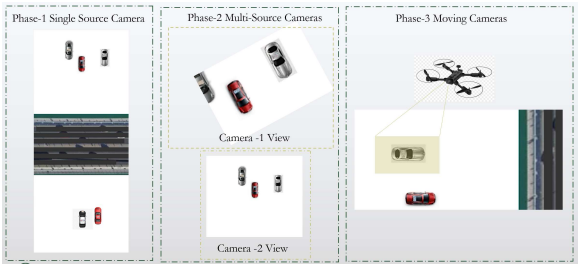


Figure 3: Three Design Phases

The modules are divided as the tasks mentioned above to facilitate the enhancement of any of the subsystems independently.

- The image sensing requirements dictates the camera technology to be used. The camera technology is not limited to only the resolution and the field of view of the camera but also the frequency range for example the frequency requirements could be outside of the visual range such as in the microwave and infrared range in the military application and in the x-rays or higher in the medical application.
- The image enhancement also referred to as the image processing module is responsible for noise removal, sharpening, deblurring, and normalizing the image.
- The background extraction can be achieved with background subtraction; however, a more elaborate subsystem needs to be in place to account for the dynamic scene changes for instance, changing in illumination, shadow casting. In static the cameras, the system shall account for subtle changes as part of the background such as flying flag or tree branches moving; however, moving camera, on design phase-3 the system shall account for moving background.
- The object classifier is responsible to detect and recognize the object of interest with a machine learning algorithm. These algorithms can be chosen from many different algorithms such as Convolutional Neural Networks (CNN), Support Vector Machines (SVM), or statistical based models such as the likelihood ratio.
- The tracking subsystem is responsible for predicting the next location of the Object Of Interest (OOI) based on the previous trajectory.
- The feedback analyzer assigns the figure of merit to the system.
- The camera controller decides which camera to turn on to keep OOI in view for phase-2 and controls the vehicle in phase-3.
- Lastly, the camera-correlator performs the affine transformation between the various cameras field of view in Phase-2.

Figure 4 below shows the system overview of system for Phase-1 which is discussed in this paper. To better control the scenario / testing the scene was synthesized. This paper discusses the related work, the theory of each subsystem, then follows with the results.

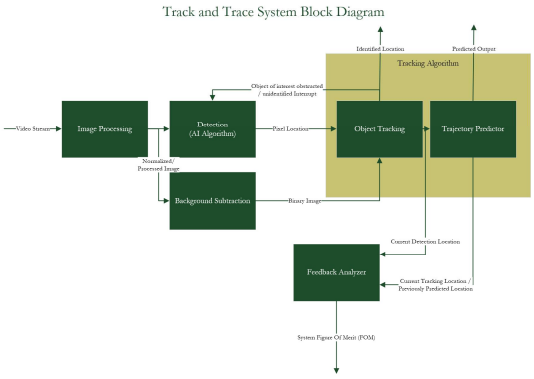


Figure 4: System Block Diagram

2.1. Image Processor Subsystem

The input to the image/video processing subsystem is a video steam. The input is separated into image frames, where each frame is normalized, histogram equalized, deblurred and sharpened. Then each frame is assigned to a red, green and blue channel as well as the hue, saturation and intensity channels. This allows for a custom usage of each channel in the detection and the tracking subsystem. Thus, the output of this subsystem is the processed RGB and HSI subframes channels as well as an image enhanced grayscale. Figure 5 below shows the block diagram of the image processor subsystem.

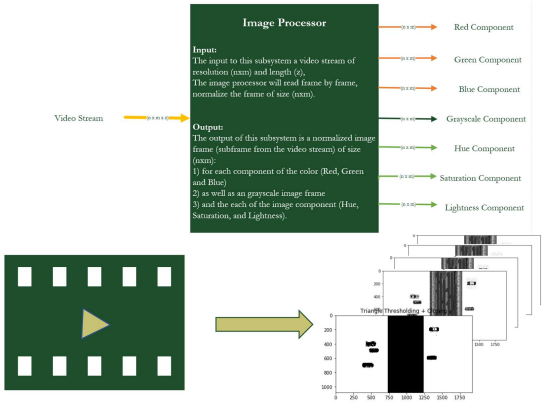


Figure 5: Image Processor Subsystem Block Diagram

2.2. Background Subtraction Subsystem

[2] The Background Subtraction subsystem takes the output of the prior subsystem which consists of the RGB and the HSI subframes and performs edge detection, and mathematical morphology functions and thresholding also known as opening, closing, erosion and dilation on each subframe to extract the object in the scene. This subsystem also performs segmentation such as watershed function to separate multiple overlapping objects. The result is then subtracted from the base image to extract the moving objects from the fixed ones. At a later stage, another function will be added to remove subtle movements that are based on repetitive temporal-spatial characteristics such as flying flag, or tree branches moving to better perform in the outdoor environment. Figure 6 below shows the output of this subsystem which is a frame that consists of only the moving objects.

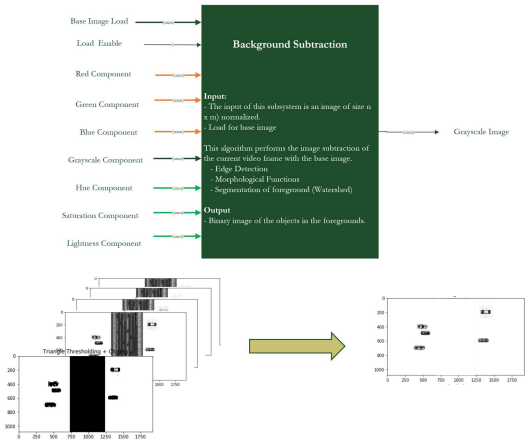


Figure 6 – Background Subtraction Subsystem Block Diagram

2.3. Detection Subsystem

The input to the detection subsystem is the output of the image processor subsystem which consists of the RGB and the HSI subframes. This subsystem is responsible to locate the Object Of Interest (OOI) in the scene. Later, this block utilizes a variety of custom pre-trained learning algorithms that the user can select; however, up to this point a Convolutional Neural Network (CNN) was developed and is presented in this paper. This subsystem outputs the centroid of the OOI. The centroid is calculated based on the binary image of the OOI shape; thus, it is morphology-based calculation of the centroid. This subsystem only executes at the beginning when the system gets powered up and locates the OOI or when an interrupt occurs. The interrupt occurs if the object tracking subsystem fails to locate the OOI due to obfuscation of the OOI in the scene. Figure 7 below shows the block diagram of the detection subsystem.

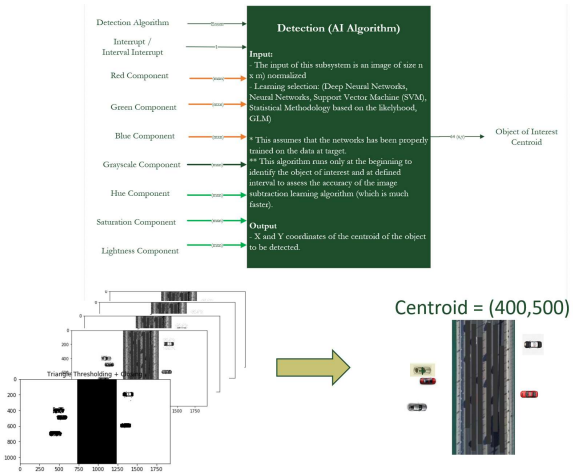


Figure 7: Detection Subsystem Block Diagram

2.4. Object Tracking Subsystem

The input of the object tracking subsystem consists of the OOI centroid that was previously calculated by the detection module, the OOI valid bit, and the grayscale image output from the background subtraction module. The subsystem consists of multiple tracking algorithm that are native to openCV. Some of these algorithms are Boosting, Multiple Instance Learning (MIL), Kernelized Correlation Filter (KCF), Tracking and Learning Detection (TLD), CNN tracker (GOTURN), Minimum Output Sum of Squared Error (MOSSE) and Discriminative Correlation Filter also known as DCF-CSR. All these algorithms and their performance will be compared and contrasted in a

later paper. A compressed version of the OOI will be used to expedite the process. In this paper, this subsystem is not developed yet. The output of this subsystem consists of the binary centroid of the OOI and a validity bit that indicates that the object has been found by one of the algorithms stated above. The main difference of this module and the detection module is that this module is dependent on a temporal knowledge based on the multiple frames; thus, it performs faster than the detection module. Figure 8 below shows the block diagram of this subsystem.

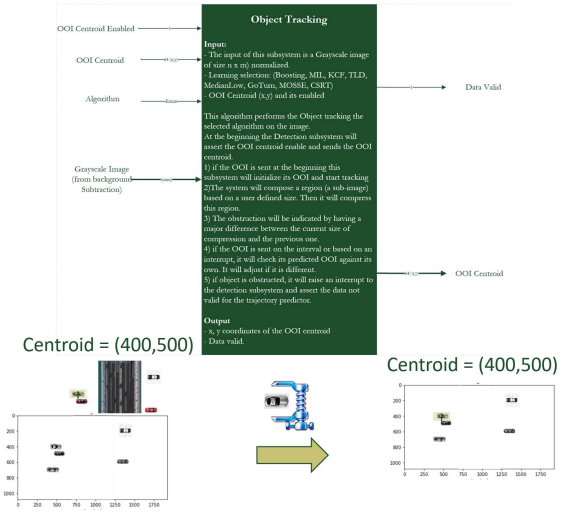


Figure 8 – Object Tracking Block Diagram

2.5. Trajectory Predictor Subsystem

The inputs to this module are the centroid of the OOI, the validity bit, and a user defined number that dictates the amount of time to extrapolate the trajectory path. This subsystem stores the discrete centroid location of all open hypotheses then performs a cubic spline interpolation to extrapolates the prediction to the amount of time requested by the user. This interpolation curve describes the characteristics and the behavior of the OOI which assist in building a model for the OOI. Figure 9 below shows the block diagram of the trajectory predictor subsystem.

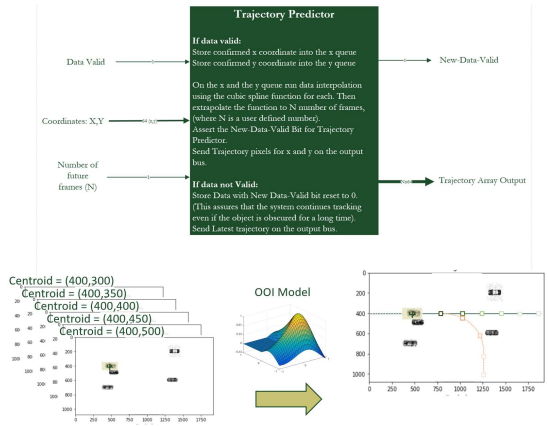


Figure 9: Trajectory Predictor Block Diagram

2.6. Feedback Analyzer Subsystem

The input to this subsystem is the coordinates of centroid and the predicted trajectory calculated from the trajectory predictor module. In this subsystem, the accuracy of the overall system gets accessed by comparing the trajectory to the detected module, then a figure of merit gets assigned to each coordinate as Figure 10 shows below in the block diagram.

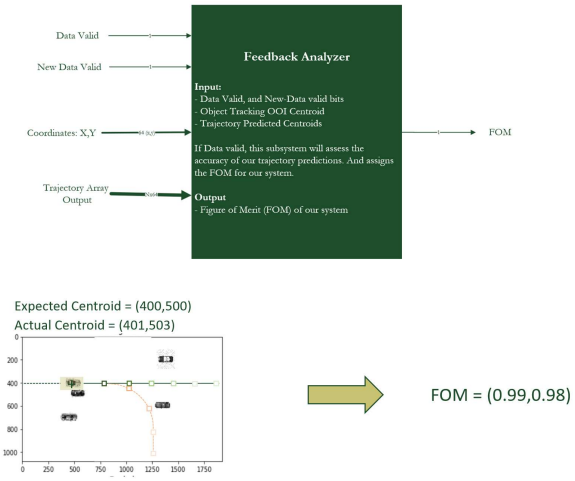


Figure 10- Feedback Analyzer Block Diagram

3. Implementation Approach

Five different movies were synthesized using Adobe Animate. The reason that the synthesized data is used instead of a real scenario is that there is a better control on the development and testing scenarios. The first set of four movies show a top view of different cars that drive around the scene at different zoom levels and different locations on the stage.

These four movies will be used for the CNN training and testing phase as Figure 11-left shows below. The fifth movie shows all cars again top view driving across the scene which is used to develop this proposed system. One of the cars acts as the Object of Interest whereas the others are present to better mimic a real environment. This is shown on Figure 11-right below.

Figure 11- (Left) Example Training Movie, (Right) Full Scene for Development

4. Convolutional Neural Network

This section discusses the architecture used, the results obtained from that model, and lastly, the future enhancements to improve the artificial intelligence model.

4.1. Architecture of the CNN

Several CNNs were developed with various filter sizes and various numbers of convolutional and Max-pooling layers. The CNN discussed below and shown on Figure 12 gave the best results. The CNN is composed of 3 pairs of convolutional and max-pooling layers. The first Convolutional Layer is composed of 32 filter of size 3x3 kernel whereas the second and third convolutional layers are composed of 32 filters of size 6x6 kernels. The purpose of the convolutional layer is to extract the high-level features such as the edges, color, and gradient orientation. All max-pooling layers are of size 2x2x32. The purpose of the max-pooling layer is to reduce the spatial size to decrease the computational power required to process the data. It also serves as a noise suppressant of the unwanted signals in the image. Lastly, the output is flattened then it is fed to fully connected layer to learn all the extracted features of the convolutional layers and max pooling.

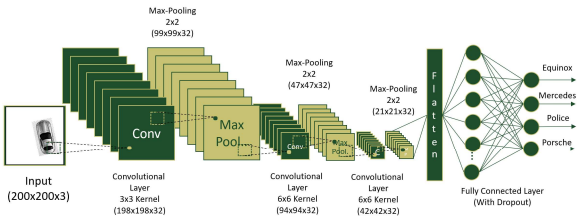


Figure 12: Convolutional Neural Network Architecture

4.2. Training Results

Using the previously discussed network, the training achieved 99.89% accuracy whereas the validation achieved 91.58% accuracy as shown on Figure 13-(left)(middle) below. Some of the cars performed more poorly than the others, as shown on Figure 13-(right). This is due to the quality of the car images that were used.

EPOCH#	Training		Validation	
	Loss	Accuracy	Loss	Accuracy
1	1.3712	0.3085	1.2274	0.463
2	0.8881	0.6201	0.9086	0.5906
3	0.4807	0.7917	0.5659	0.7219
4	0.2587	0.8823	0.4127	0.7895
5	0.1585	0.9355	0.2139	0.8099
6	0.0887	0.9683	0.25	0.8189
7	0.0479	0.9871	0.6294	0.8546
8	0.0151	0.9968	0.6915	0.8622
9	0.0247	0.9957	0.1835	0.8839
10	0.0081	0.9989	0.1775	0.9158

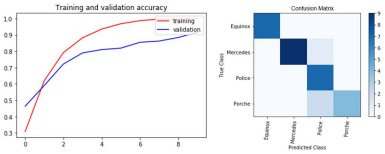


Figure 13: (Left) Results from the training and Validation, (Middle) The ROC curve of the training vs the validation, (Right) The Confusion Matrix of all the outputs.

4.3. Future Enhancement

To improve the results, a higher confidence synthesis will be developed and reported in future articles. Autodesk Maya will be used to develop the synthesized scene and objects: this will allow the 3D development of the objects as well as the camera placement.

5. Conclusion

This paper discussed an object tracking system that is developed in three phases. The first phase will track an Object Of Interest (OOI) from a stationary camera, the second phase system will track the OOI across a network of camera, and the third phase the system will track the OOI using a moving camera. This paper discussed the first phase architecture and decomposed it into its subsystems. It later discussed the architecture and the results of the CNN that was used in the detection subsystem. In later papers, the background subtraction, the trajectory tracking, and the feedback analyzer will be discussed in greater detail.

6. References

[1] M. Walters, "Sensors Technologies for Autonomous Vehicles," in *Electronic Imaging*, Burlingame, CA, USA, 2020.

[2] L. Maddalena and A. Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance

Applications," *IEEE,* vol. 1057, no. 7149, pp. 1169-1177, 2008.

[3] J. W. Davis and A. F. Bobick, "The Representation and Recognition of Human Movement Using Temporal Templates," *1063-6919/97 IEEE,* pp. 928-934, 1997.

[4] S.-C. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *SPIE 5308, Visual Communications and Image Processing* , San Jose, CA, , 2004.

[5] L. Maddalena and A. Petrosino, "A self-organizing approach to detection of moving patterns for real-time applications," in *2nd Int. Symp. Brain, Vision and Artificial Intellicence*, Springer, Berlin, Heidelberg, 2007.

[6] L. Maddelena and A. Petrosino, "A self-organizing approach to detection of moving patterns for real-time applications," 2007. [Online]. Available: http://www.na.icar.cnr.it/~maddalena.l/HSV-SO/HSV-SO2007.html.

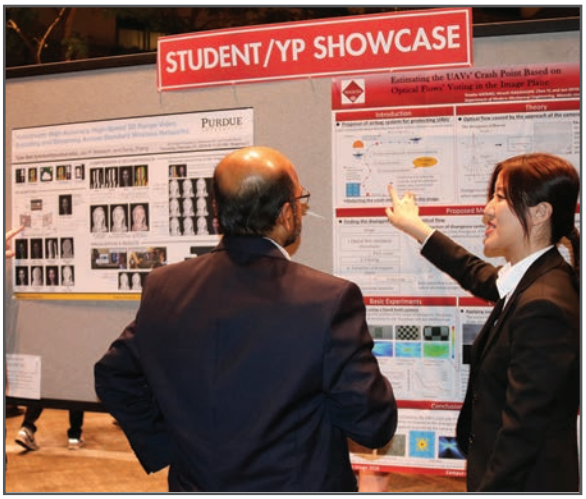
JOIN US AT THE NEXT EI!

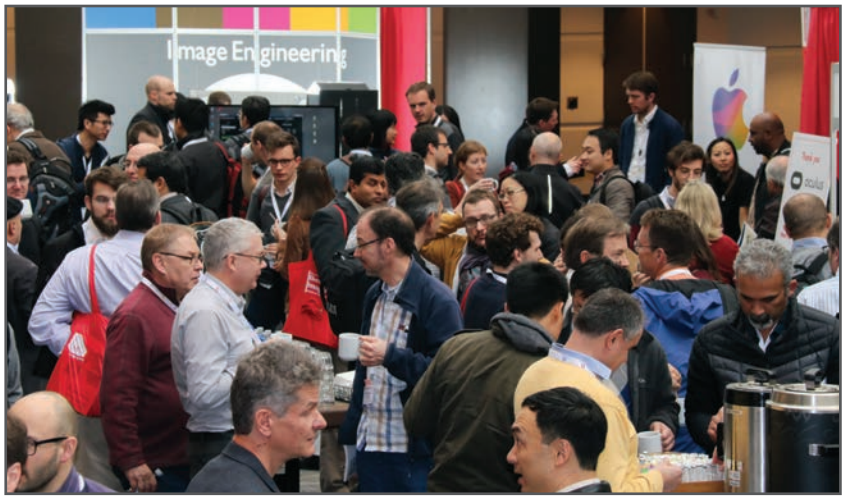
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org