

GENERATION OF REALISTIC VEHICLE TRAJECTORIES FROM VIDEO STREAMS

Internship Report
Submitted in Partial Fulfillment for the Award of
Master of Technology

By

Potluri Sai Rakshith
IMT2015032

To



International Institute of Information Technology Bangalore

June 2020

6/21/2020

Mail - IIITB Library - Outlook

Successful completion of internship from 16/12/2019 - 12/06/2020 – IMT2015032 – Potluri Sai Rakshith

Pathrudkar, Sagar <sagar.pathrudkar@siemens.com>

Fri 6/12/2020 9:47 AM

To: IIITB Library <iiitblibrary@iiitb.org>

Cc: IMT2015032 Potluri Sai Rakshith <Sai.Rakshith@iiitb.org>; IIITB_Placement <IIITB_Placement@iiitb.ac.in>

Dear Sir/Madam,

This is to certify that the internship report titled, 'GENERATION OF REALISTIC VEHICLE TRAJECTORIES FROM VIDEO STREAMS' submitted by 'Potluri Sai Rakshith' (IMT2015032) is a bonafide work carried out under my/our supervision at 'Siemens Corporate Technology' from 16th December 2019 to 12th June 2020, in partial fulfilment of the Master of Technology course of International Institute of Information Technology, Bangalore.

His performance & conduct during the internship was satisfactory.

Name: Sagar Pathrudkar

Designation: Research Engineer

Address: C503, Shriram Signiaa, Neeladri Road, Bangalore 560100

Date: 12th June 2020

Place: Bengaluru

Best regards,

Sagar Pathrudkar

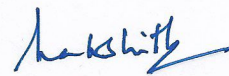
Undertaking by the Student

I, Potluri Sai Rakshith, hereby declare that the report of the internship program titled, **“GENERATION OF REALISTIC VEHICLE TRAJECTORIES FROM VIDEO STREAMS”** is prepared by me. I also confirm that, the report is only prepared for my academic requirement and not for any other purposes.

I also confirm that, the submitted softcopy has been reviewed and approved for submission by my supervisor.

Date: 22/06/2020

Place: HYDERABAD



Potluri Sai Rakshith
(IMT2015032)

ACKNOWLEDGMENT

I would like to express my sincere thanks and acknowledgement to Siemens Corporate Research and Technology India for providing me with an opportunity to do an internship.

I am especially thankful to **Sagar Pathrudkar, Ram Padhy, Saikat Mukherjee** whose guidance and support throughout the project was highly appreciated. I am also thankful to **Bristi Singh, Bony Mathew, Saadhana B, Indla Vijayasarithi** for their advice and help.

Last but not the least I would like to thank my parents and friends, without whom this would not have been possible.

Potluri Sai Rakshith

IMT2015032

CONTENTS

	Page No.
Acknowledgements	iv
Table of Contents	v
List of Figures	vi
Abstract	vii
1. INTRODUCTION	1
2. TRAJECTORY EXTRACTION	3
2.1 System Overview	3
2.2 Capturing Video and Frame Extraction	4
2.3 Camera Configuration	4
2.4 Detection and Tracking	5
2.5 Trajectory Projection and Extraction	6
3. TRAJECTORY GENERATION	7
3.1 LSTM-Based Trajectory Prediction	7
3.2 Convolutional GAN	8
3.3 Recurrent GAN	10
4. LIMITATIONS	11
5. CONCLUSIONS AND FUTURE WORK	12
GLOSSARY	13
BIBLIOGRAPHIC REFERENCES	14

LIST OF FIGURES

Figure Number	Title	Page No.
1.	Global Architecture of Trajectory Extraction Pipeline	3
2.	Optimization Function for Camera Calibration	5
3.	Mask-RCNN Detections	6
4.	Special Pooling in Social LSTM	7
5.	A Typical GAN Architecture	9

ABSTRACT

Urban environments are still a challenge for Autonomous Vehicles, due to strong interactions with other vehicles and pedestrians. Machine learning methods are increasingly explored to tackle these situations, but their performances are highly conditioned on the availability of vehicle trajectory datasets.

To collect extensive data on realistic driving behaviour for use in simulation, testing and training, we propose a framework that uses online public traffic cam video streams to extract vehicle trajectories and generate new trajectories conditioned on them.

This pipeline leverages recent advances in deep learning for object detection (using Mask-RCNN) to extract trajectories from the video stream to corresponding locations in a bird's eye view. We then use generative models such as GANs and LSTMs to generate new trajectories.

1. INTRODUCTION

Understanding vehicle motion is critical for self-driving cars and other autonomous moving platforms. Urban environments are still a challenge for autonomous vehicles with their inherent multi-modality and strong dynamic interactions with other vehicles and pedestrians. Recent advances in **Deep Neural Networks (DNNs)** have led to the development of data-driven autonomous cars that, using sensors like LiDAR, RGB cameras etc., that have been successful in addressing various shortcomings associated with autonomous driving. However, despite their superior performance, they often **demonstrate erroneous or unexpected corner-case behaviors** that can lead to potentially fatal collisions.

This calls for **extensive testing** of such models before they are put out in the world. Autonomous driving simulators offer the potential to rapidly test the data-driven algorithms. However, a significant concern with developing a simulation is how accurately it reflects the physical world. Currently, autonomous driving companies collect real-world traffic data through countless road trials and expensive sensor and telemetry equipment, a luxury possessed only by a few.

We use a computer-vision based framework for **accurate vehicle trajectories extraction** from single fixed traffic cameras. We then use tools from sequence prediction and generative adversarial networks: a recurrent sequence-to-sequence model to generate new realistic vehicle trajectories conditioned on the extracted trajectories. We leverage the fact that traffic cameras represent a cost-effective source of highly diverse vehicle trajectories and the subsequently generated trajectories are powerful enough to accurately model the underlying nature of vehicle motion.

With this work, we aim at developing a framework for realistic vehicle trajectories generation at large scale.

2. TRAJECTORY EXTRACTION

2.1 System Overview

The trajectory extraction pipeline is divided into multiple sub-components. Each component has specialized functionality and the output of one component is feeding into other consecutive components sequentially to finally get the extracted trajectories. Figure 1 gives a component-level architecture of the trajectory extraction pipeline.

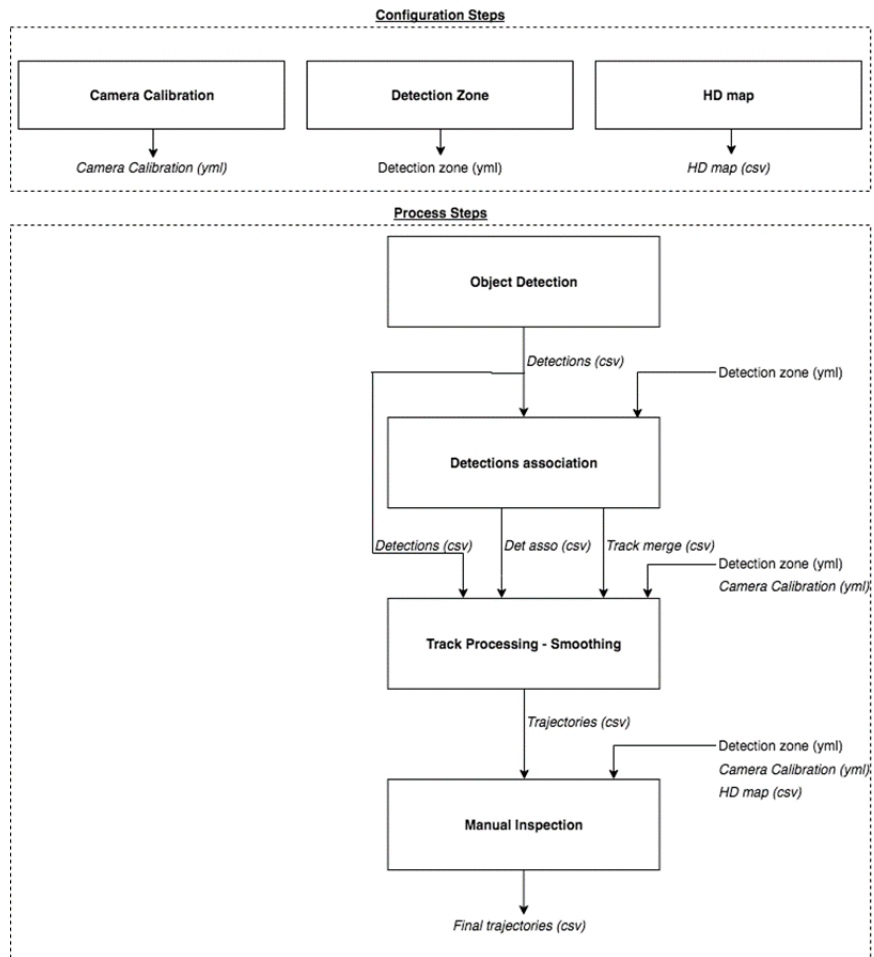


Figure 1. Global Architecture of Trajectory Extraction Pipeline [2]

2.2 Capturing Video and Frame Extraction

The proposed pipeline uses OpenCV as the interfacing framework to extract frames from video streams. As a result, we can process common video file types ranging from mp4 to avi. We extract a configurable number of frames from a given video, which are later used for vehicle detection and trajectory extraction.

2.3 Camera Configuration

The core of the pipeline is the calculation of the camera projective transform which used to project the detected trajectory from the camera's point of view to a geometric 2D Map. We use a pin-hole camera model to find the perspective transform function to convert camera view coordinate to its projection on a bird-view coordinate system. The pin-hole camera model is a simple model that describe the relationship between a 3D point in world coordinate frame is denoted by $M = [X, Y, Z]$, and its projection in pixels denoted $m = [u, v]$. The relationship between a 3D point and its image projection is given by the pin-hole camera model.

We use a calibration method to compute the intrinsic (focal length) and extrinsic (rotation and position) camera parameters. Using a satellite view of the area covered by the traffic camera, we identify four or more pairs of key point matches, between the traffic camera view and the satellite view ($m_i; M_i$), with (m_i pixel coordinate and M_i world coordinate of the i th pair, and used them to derive the projective transform matrix. [2]

Camera parameters such as rotation, translation and focal length of the traffic camera are found by minimizing the reprojection error:

$$\min_{\mathbf{R}_{cam}, \mathbf{T}_{cam}, f} \sum_i \|g(\mathbf{M}_i) - \mathbf{m}_i\|$$

Figure 2. Optimization Function for Camera Calibration [2]

Figure 2 provides the optimization function for camera calibration. This optimization provides us with a function that projects 3D world points onto the image plane according to the pin-hole camera model.

2.4 Detection and Tracking of Vehicles

The pipeline uses Mask-RCNN to detect the vehicles in the camera feed. Mask-RCNN provides detections with 2D Bounding Box, instance mask and instance type. The instance type and instance mask are used to get accurate vehicle positions. In this pipeline, the pre-configured camera model is used to compute the projection of 3D boxes on the image plane. For each class of vehicles (e.g. car, truck, bus) provided by the object detector, general 3D box parameters (length, width and height) are predefined. Then, assuming the vehicles are on the ground, their x, y position defined as the geometric centre of the vehicle on the ground can be estimated by maximizing the overlap between the 3D box projection on the image plane and the 2D instance mask provided by Mask-RCNN. [2]

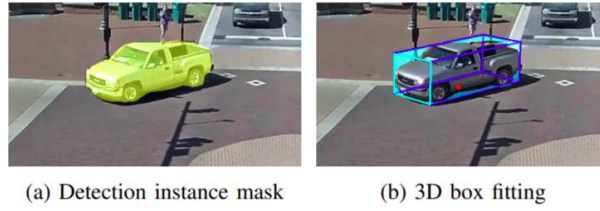


Figure 3. Mask-RCNN Detections [2]

As a tracking-by-detection method, detections in successive frames must be associated to form tracks. Using the instance masks provided by Mask-RCNN, a simple Intersection-over-Union tracker is applied which associates detections with high overlap in successive frames into tracks. Basic idea of IOU Tracker is to select the best bounding box from next frame, based on intersection over union of the detected bounding boxes. [2] Detections obtained from Mask-RCNN can be observed in Figure 3.

The Mask-RCNN model's performance can be improved by retraining the model on custom datasets. For example, to get better performance on Indian traffic camera videos, the model could be retrained on IDD Dataset which contains thousands of annotated vehicle classifications sampled from roads of Hyderabad and Bangalore.

2.5 Trajectory Projection and Extraction

Once, the camera model for a particular camera feed is configured, the vehicle extraction is automated. All the trajectories in a given video are processed and the trajectories are filtered. The extracted trajectories are then projected onto a local NED coordinate system with a preconfigured origin. The resulting data-set contains frame-to-frame tracked coordinates of vehicles in the local NED system. Optionally, we used mathematical models to convert the extracted trajectories to lat-lon coordinate system for further testing.

3. TRAJECTORY GENERATION

We use various generative and sequence-to-sequence models to generate new trajectories conditioned on the behaviour of our extracted trajectories.

3.1 LSTM-Based Trajectory Prediction

Long Short-Term Memory (LSTM) networks have been shown to successfully learn and generalize the properties of isolated sequences like handwriting and speech. Inspired by such work, this pipeline uses a LSTM based model for our trajectory prediction problem as well. The vanilla LSTM is agnostic to the behaviour of other sequences. This limitation is addressed by connecting neighbouring LSTMs through a social pooling strategy. Individual agents adjust their paths by implicitly reasoning about the motion of neighbouring agents. These neighbours in-turn are influenced by others in their immediate surroundings and could alter their behaviour over time. The hidden states of an LSTM to capture these time varying motion-properties. This specific version of LSTM is called Social LSTM. [1]

Given the initial states of a trajectory, a trained LSTM model allows us to generate trajectories conditioned on the previously extracted trajectories.

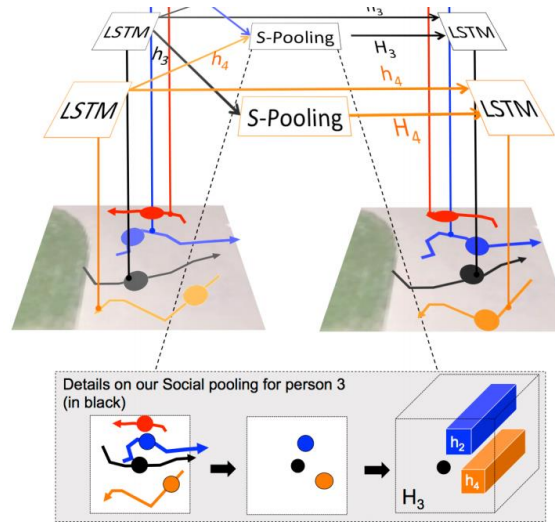


Figure 4 Special Pooling in Social LSTM [1]

Figure 4 depicts the social pooling layer using Social LSTM.

3.2 Convolutional GAN

Generative modelling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset. Convolutional GANs have predominantly been used in computer vision and image processing. Developing a GAN for generating images requires both a discriminator convolutional neural network model for classifying whether a given image is real or generated and a generator model that uses inverse convolutional layers to transform an input to a full two-dimensional image of pixel values.[5]

Convolutional Neural Networks (CNN) excel at processing images because using kernels and non-linear activations help the model in understanding intrinsic spatial arrangements of pixels in images.

The same concept can be applied for trajectory generation. By feeding in the entire trajectory as a pattern of different positions, helps the model understand the inherent pattern of trajectories. This can be used to generate new trajectories conditioned on the spatial pattern of existing trajectories. A typical GAN Architecture can be seen in Figure 5 with separate Generator and Discriminator components.

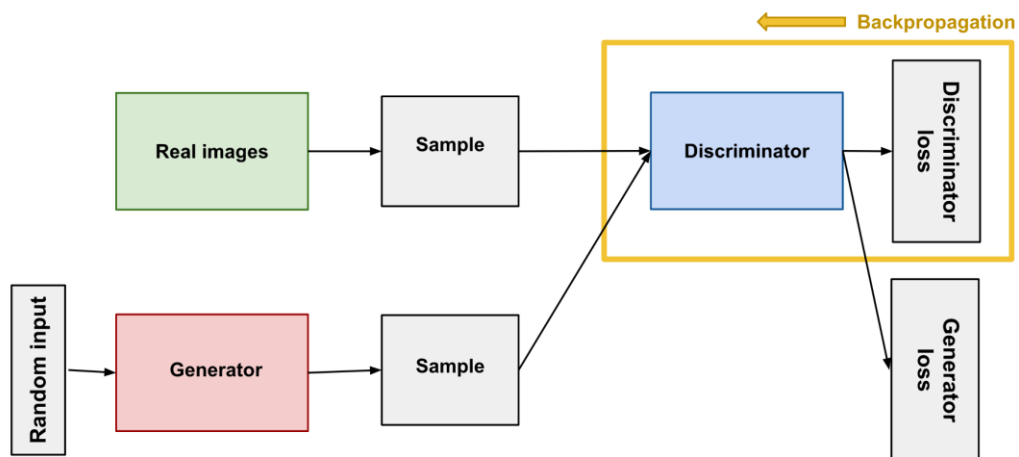


Figure 5. A Typical GAN Architecture [4]

3.3 Recurrent GAN

Recurrent Neural Networks (RNN) are a rich class of dynamic models which extend feedforward networks for sequence generation in diverse domains like speech recognition, machine translation and image captioning [5]. They leverage the sequential nature of data, which proves to be crucial in trajectory generation. Using RNNs instead of CNNs address certain shortcomings in Convolutional GANs. Instead of looking at the trajectory as a whole, Recurrent GANs try to understand the sequential nature of trajectories and generate new trajectories by fitting a model to the underlying distribution of existing trajectories.

We use a modified version of RGAN called Social GANs [3]. Social GAN is a RNN Encoder-Decoder generator and a RNN based encoder discriminator with the following two novelties:

(i) a variety loss which encourages the generative network of our GAN to spread its distribution and cover the space of possible paths while being consistent with the observed inputs.

(ii) a new pooling mechanism that learns a “global” pooling vector which encodes the subtle cues for all agents involved in a scene.

This can be used to generate new trajectories conditioned on the sequential distribution of existing trajectories. [3]

4. LIMITATIONS

- Although the pipeline is robust enough to extract trajectories at a large scale, there are a few limitations.
- The camera configuration requires considerable amount of manual tuning. The key point matches between camera view and the 2D satellite view must be spread out and accurate to calculate a good approximation of the perspective transform matrix.
- Not robust to occlusions and congestion as the detection-to-track association relies on visual information. This limits the application of this pipeline to well-lit camera feed, preferably where the camera is looking down on the vehicles.
- Tracks can jump from one target object to another (identity switch), due to erroneous detections taking two distinct objects as one single object or when the vehicles move at high velocities.
- When the extracted trajectories are converted back to lat-lon coordinates from a local NED system, there is an expected shift in the coordinates from the ground-truth. But, the shape and orientation of the trajectory is still retained.
- Trajectory modelling and generation is a data intensive process. Generative and Sequence-to-Sequence models are extremely data hungry and need humongous amounts of data for good results.

5. CONCLUSIONS AND FUTURE WORK

In this project we proposed a pipeline which extracts vehicle trajectories from raw camera feed and these extracted trajectories can subsequently be used to generate realistic vehicle trajectories.

Future work can include automating certain manual components such as configuration of the camera model. The tracking of vehicles between frames is done using IOU tracker, which can be improved upon and can address certain limitations such as sparse trajectories and identity switches. Trajectory generation can be improved upon by adopting multi-agent paradigms. Vanilla GANs consider each individual trajectory agnostic to other vehicles in the same frame.

Incorporating such dynamic constraints into model will lead to better and more naturalistic results. All the proposed generative models are off-the-loop methods but these can be changed by using simulator-in-the-loop methods such as GAIL and PS-GAIL [6]. Such methods allow multi-agent constraints while training the model. This would lead to considerable improvements in generated trajectories.

GLOSSARY

CNN: Convolutional Neural Network

GAIL: Generative Adversarial Imitation Learning

GAN: Generative Adversarial Network

IOU: Intersection-Over-Union

LSTM: Long Short-Term Memory Network

NED: North-East-Down Coordinate System

PS-GAIL: Parameter Sharing Generative Adversarial Imitation Learning

RGAN: Recurrent Generative Adversarial Network

BIBLIOGRAPHIC REFERENCES

- [1] Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

- [2] Clausse, S. Benslimane and A. de La Fortelle, "Large-Scale extraction of accurate vehicle trajectories for driving behavior learning," *2019 IEEE Intelligent Vehicles Symposium*

- [3] Gupta, J. Johnson, L. Fei-Fei, S. Savarese and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks,"

- [4] Google, "Generative Adversial Networks", Available: <https://developers.google.com/machine-learning/gan/> [Accessed June 18 2020]

- [5] Jason Brownlee, "Machine Learning Mastery", Available: <https://machinelearningmastery.com/> [Accessed June 18 2020]

- [6] Raunak P. Bhattacharyya, Derek J. Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, Mykel J. Kochenderfer, "Multi-Agent Imitation Learning for Driving Simulation"

[7] Xinhe Ren , David Wang , Michael Laskey, Ken Goldberg , “Learning Traffic Behaviors for Simulation via Extraction of Vehicle Trajectories from Online Video Streams “