DRONE-BASED TRAFFIC FLOW ESTIMATION AND TRACKING USING COMPUTER VISION

A. de Bruin and M.J. Booysen

Department of E&E Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602 Tel: 021 808-4013; Email: mjbooysen@sun.ac.za

ABSTRACT

Traffic management has become increasingly important with growth in vehicle numbers unmatched by investment in infrastructure. A large part of management is measuring traffic flow. Video footage of traffic flow is normally manually checked to determine key traffic metrics, consuming many human hours. Moreover, installation and maintenance cost of recording equipment and supporting infrastructure is substantial, especially in the Sub-Saharan context. This paper proposes a novel solution to automate traffic flow estimation, using computer vision. The paper also introduces the notion of making the recording equipment mobile by using drone-based equipment, negating the need for fixed recording installations. The results demonstrate measurement accuracies of 100% down to 81% from ideal to worst case conditions, and successful implementation of drone control algorithms.

INTRODUCTION

According to the National Traffic Information System, there are currently around 11 million registered vehicles on South African roads (National department of transport, 2014). This number is increasing at an alarming rate, which requires that roads be upgraded continually. The study of traffic flow estimation is used to evaluate how well a particular road segment is accommodating traffic, as well as to determine the priority of road upgrades.

Current traffic monitoring techniques make use of intrusive static sensors in the form of inductive loop detectors, infrared (IR) detectors and radar guns (Thies et al., 2013). Visual monitoring is often done manually, with the operator watching hours of video footage while counting the cars as they pass through an area. Two of the significant problems associated with the above-mentioned techniques, is that they are both intrusive and time-consuming. Traffic cameras are mounted around most urban areas and are used primarily for security reasons. In the City of Cape Town alone, there are around 300 traffic cameras streaming live video directly to the Transport Management Centre (TMC) database. The cameras cover the majority of the roads throughout Cape Town, and would therefore provide unparalleled access to essential video data.

There are some areas throughout Cape Town that are not yet monitored by traffic cameras. The cameras and related infrastructure are expensive to install, and require many man hours to complete. A particularly attractive solution to this problem is to erect simple landing platforms that would allow an autonomous Unmanned Aerial Vehicle (UAV) to conduct fully autonomous traffic flow analyses.

The work explained in this paper makes use of pure computer vision techniques to automatically compute traffic metrics along road segments. This paper will focus primarily on uninterrupted flow in the form of freeways and national highways. A key objective was

to make the system as flexible as possible to maximise the capabilities of the estimation techniques. One of the main reasons for optimising flexibility originates from the novel concept of using both pole-mounted traffic camera footage, as well as footage obtained from other sources such as, but not limited to, the UAV's on-board camera.

The work discussed in this paper proposes a way of autonomously computing key traffic flow descriptors using pure computer vision techniques. The inclusion of an autonomous aircraft provides a novel means for obtaining essential video footage.

RELATED WORK

Various methods of intelligent traffic monitoring have been proposed throughout the years, most of which employ computer vision techniques to detect and track passing vehicles. This section will discuss similar works in literature and explain their contributions and shortfalls.

A method proposed by (Koller et al., 1994) uses traffic scene information to optimise traffic flow during busy periods, to identify stalled vehicles and accidents, and to aid the decision-making of an autonomous vehicle controller. The system employs a contour tracker and an affine motion model based on Kalman filters to extract vehicle trajectories over a sequence of traffic scene images (Koller et al., 1994). This system is slightly different to the one proposed in this paper as it does not focus on computing traffic flow metrics for performance analyses, but rather to aid in real-time decision-making for an autonomous vehicle controller.

A method proposed by (Muthukumar & Chintalacheruvu, 2012) makes use of the Harris-Stephen corner detector algorithm to efficiently detect vehicles in a video stream. The system was designed to detect and compute vehicle counts and speeds at arterial roadways and free-ways. The goal was to develop a system that would eliminate the need for calibration and have robustness against contrast variations. Similar to the aforementioned system, the system proposed by Muthukumar and Chintalacheruvu was designed primarily as an advanced warning and traffic control system.

One of the main inhibiting factors of using computer vision for traffic detection is the fact that these visual-based systems do not perform well under low-light conditions. A system proposed by (Kannegulla et al., 2013) makes use of thermal imaging cameras and pure computer vision techniques to detect and track vehicles under extreme illumination conditions. The combination of thermal imaging technology and highly optimised computer vision techniques, allowed for the development of a system that would measure traffic density extremely accurately. As was the case with the previous two systems, the system proposed by Kannegulle et al. is not focused on generating traffic flow descriptors for future road planning, but simply to optimise the immediate flow of traffic.

Table 1 lists some of the features associated with the above-mentioned systems.

Table 1: List of features for each of the aforementioned systems.

Koller et al.	Chintalacheruvu et al.	Kannegulle et al.
• Identify stalled vehicles and accidents.	• Compute vehicle count. • Compute vehicle speed.	• Designed primarily for optimising traffic flow.

• Used as an autonomous vehicle controller. • Contour tracking and affine motion model based on Kalman filters.	• No calibration required. • Performs well under low-light conditions. • Highly optimised for real-time performance.	• Makes use of thermal imaging cameras. • Performs extremely well under low-light conditions. • Highly accurate computation of traffic density.

SYSTEM DESIGN

The autonomous drone-based traffic flow estimation system can effectively be separated into two parts. The first part consists of the computer vision system used to detect and calculate vehicle velocities for calculation of key traffic metrics. The second part involves the design of an autonomous target tracking and landing system for the UAV.

Figure 1 shows the hardware components and their corresponding methods of communication. The ground station communicates with a GSM modem via a USB-Serial connection. Commands are sent from the ground station to the modem as simple AT strings. The modem interprets these strings, and prepares the IP packets to be sent over the mobile network. Data is transmitted to a cloud-hosted database via mobile network, where it is interpreted by Trintel's SMART platform, and displayed graphically on an online dashboard.
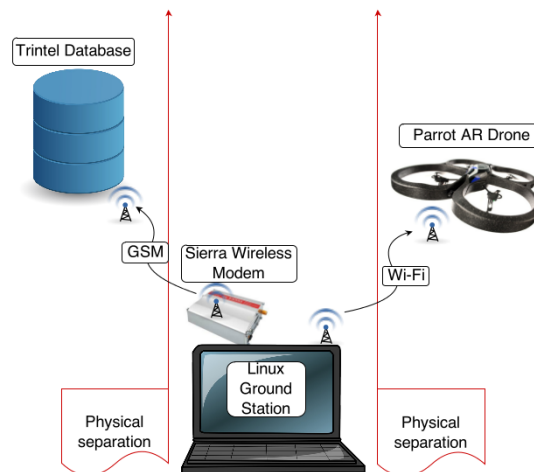

Figure 1: Hardware integration

The ground station communicates with the Parrot AR drone via its Wi-Fi module as shown in figure 1. The SDK network library handles the network interfacing between the ground station and the drone. Reference angles and angular velocity references are sent to the drone as fractions of the maximum set-point values.

The design phase is separated into two subsystem designs. The first detailed design is concerned with the traffic flow estimation process, and has a specific focus on the supporting computer vision techniques. The second subsystem design focuses on the control system and additional computer vision techniques used in the automation of the drone's flight control.

Traffic flow estimation

The main traffic flow algorithm is required to automatically detect the number of vehicles that pass through a given area, as well as to determine their relative velocities. Once the

vehicles are detected and their velocities estimated, they are then classified according to relative size (motorbikes, cars and trucks).

A particularly challenging aspect was to design a system that relied entirely on visual references. The idea was to design and implement a non-intrusive system that makes use of existing traffic cameras placed around a city. It is important to note that traffic cameras need not be the only source of video feed. As mentioned earlier in this paper, the idea is to eventually incorporate an unmanned aircraft into the system that can autonomously fly to remote locations which might not currently have an established traffic camera network. The system is required to be extremely flexible in order to accommodate a variety of different video sources, and therefore relies heavily on highly adaptive computer vision techniques to compute all traffic metrics.

Road profile creation

Every road location is unique in the way in which the static traffic cameras are placed. This causes a potential problem, especially when computing relative vehicle velocities as well as classifications based on relative vehicle sizes. An elegant and particularly robust solution was developed to deal with this problem. The idea was to create and save road profiles that would store all location-specific information. Due to the static nature of the pole-mounted traffic cameras, road profiles would only need to be generated once for each location. If the drone is to be used for traffic analysis, a location profile would have to be generated each time it lands to accommodate for orientation-specific parameters.

In an attempt to make the road profile creation process a more user-friendly experience, an interactive, self-learning method was designed. The method involved a three-stage creation process with the first stage being fully autonomous, and the last two requiring some basic user input. Once a user has input the necessary parameters, the system stores all location-specific data in a uniquely identifiable DAT file.

Background modelling

The key challenge to realising the system is successfully identify and track objects in a video stream. The Background Subtraction (BS) technique, for use in computer vision, is designed to successfully differentiate a moving object from its corresponding static background scene. The system discussed in this paper makes use of the BS technique for the detection and tracking of passing vehicles.

In order to conduct background subtraction, it is necessary to obtain a model of the static background scene. Background modelling consists of two primary phases - phase one is responsible for background initialisation while phase two is aimed at updating the background model. A good approximation of the static background scene is obtained by making use of a running frame average technique. Figure 2 shows the result of the running average algorithm when approximating the background scene. The approximated background is a good initialisation point for the background modelling process.

Once the background scene has been approximated, each individual background pixel is then modelled using a Mixture of Gaussian distributions (MoG modelling technique).
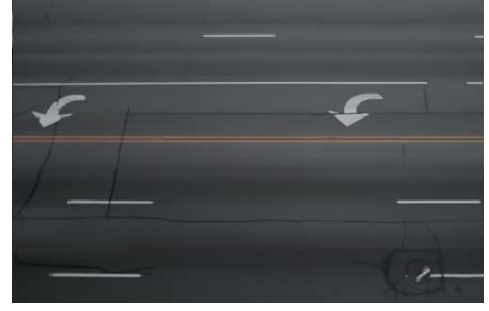
Figure 2: Running average technique. The left-most image shows the vehicles moving across the road segment, while the right-most image shows the result of

A mixture of N Gaussian distributions is then generated to model each individual background pixel. Background pixels are characterised based on their persistence and variance. Equation 1 represents the Gaussian Mixture Model (GMM) equation:

$$p(x) = \sum_{k=1}^{K} \pi_k \, \eta(x|\mu_k, \Sigma_k) \tag{1}$$

Where the multivariate Gaussian distribution is given by

$$\eta(x|\mu_k, \Sigma_k) = \frac{1}{|2\pi\Sigma_k|^{1/2}} \exp[\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)] \tag{2}$$

As the illumination environment changes throughout the operational life of the model, updates are required to ensure model accuracy. The background model is updated with each successive frame so as to accommodate for the various illumination effects. The learning rate parameter specifies the rate at which the model is updated - this parameter was optimised by means of empirical investigation.

Shadow removal

The use of the BS algorithm does not provide a complete solution with regards to object detection. A particular disadvantage of using the MoG technique, is that the object shadows tend to be classified as part of the foreground. The reason for this is that shadows share the same movement patterns as the objects that create them. Shadows also tend to exhibit similar pixel intensity characteristics as the corresponding foreground objects (Lovell et al., 2012). When two vehicles are in close proximity to one another, their corresponding shadows make them appear as a single object - leading to reduced tracking and counting accuracy.

The shadow detection technique used in this system is based on the *chromaticity* characteristics of shadows. Chromaticity is a measure of colour that is independent of intensity (Lovell et al., 2012). The idea behind the method of chromaticity is to detect shadows based on their pixel characteristics. There are three primary characteristics that distinguish shadow pixels from their non-shadow counterparts. It is known that the intensity of a shadow pixel (V in HSV) is lower than that of the corresponding background pixel (Lovell et al., 2012). Furthermore, it known that a shadow cast on the background does not change the pixel hue (H in HSV), and that a shadow pixel often exhibits lower saturation (S in HSV) characteristics (Lovell et al., 2012). A pixel *p* is therefore considered a shadow pixel based on the above-mentioned criterion. Once the frame coordinates of the shadow pixels have been identified, the corresponding pixels are subsequently

removed from the foreground mask (result of the background subtraction) before being put through a bilateral filter to minimise noise.

Vehicle speed detection

Traffic flow estimation theory does not only depend on the number of vehicles passing through a specific road location, but on the relative velocities of the vehicles as well. Vehicle velocities are usually obtained using radar guns, inductive loops and IR counters (National Research Council, 2010). However, these methods are seen as intrusive, as additional hardware needs to be incorporated into the existing road structure. A particularly attractive alternative is to use the existing camera infrastructure to automatically compute relative vehicle velocities.

Velocity is a measure of object displacement per unit time. In order to determine the velocity of passing vehicles, it is necessary to first obtain the displacement (distance) of the vehicles between consecutive frames. Once the displacement of the pixels corresponding to the individual vehicles is known, the frame rate of the video can be used to compute the relative vehicle velocities. Equation 3 represents the velocity equation:

$$v = \frac{pixel\ displacement\ between\ consecutive\ frames}{time\ delay\ between\ frames\ (1/frame\ rate)} \qquad (3)$$

Optical flow tracking provides a way of determining pixel displacement between consecutive frames. Optical flow operates under two primary assumptions. The first assumption is based on the fact that the pixel intensities of an object should remain constant between consecutive frames (OpenCV, 2011). The second assumption is that neighbouring pixels will have a similar motion to that of the pixel under observation (OpenCV, 2011).

Any optical flow algorithm involves complex mathematical calculations conducted on a large number of individual pixels. Therefore, the implementation of a highly optimised algorithm is necessary to ensure real-time performance. The OpenCV platform includes an optical flow tracking method based specifically on the method proposed by (Lucas & Kanade, 1981). The method requires unique feature points on the objects in order to track pixels accurately. According to (Shi & Tomasi, 1994), corners of an object are good features to track and are therefore used in the optical flow tracking process. Figure 3 shows the optical flow vectors superimposed onto the moving vehicle.
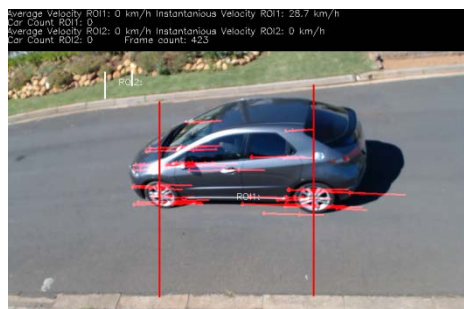

Figure 3: Optical flow vectors

Traffic flow metric computations

Autonomous traffic flow estimation is recognised as the fundamental core of this system. Determining the total vehicle count and respective vehicle velocities in the previous sections was a necessary step in computing traffic flow metrics. It was decided that the following metrics would be useful in describing uninterrupted traffic flow data: Time Mean Speed (TMS), Volume, Flow Rate, Density, Peak Hour Factor (PHF) and Level of service (LOS).

Once the system is able to identify the number of vehicles moving through a particular road segment (background subtraction) and their corresponding velocities (optical flow tracking), the above-mentioned traffic flow metrics are then autonomously generated based on predetermined equations described by the Highway Capacity Manual (National Research Council, 2010).

<u>Autonomous aircraft</u>

The autonomous aircraft was included to provide a novel form of autonomy for future traffic analyses. The idea is that the drone will eventually fly to pre-determined destinations using a GPS navigation system. Once the drone is within visual range of the landing platform, a unique identifier in the form of a checkerboard pattern will be used as a reference for the visual target tracking system. When the control system has stabilised the drone in front of the target, an autonomous landing system will land the drone on the platform below. The drone's front-facing camera (FFC) can then be used as a mobile substitute for the static pole-mounted traffic cameras.

Target tracking

In order to detect whether a checkerboard shape is currently in the frame, each frame is converted to a greyscale image to maximise the distinction between the black and white checker squares. The frame is then put through a binary threshold function before a pattern recognition algorithm is used to identify the location of the checkerboard. Figure 4 shows the drone On Screen Display (OSD) once the target has been identified. An algorithm, running on the ground station, determines the translation and rotation of the checkerboard in 3D space. This information is then used by a feedback PID control system to automate the drone's flight and ultimately stabilise it at a set distance from the target position.
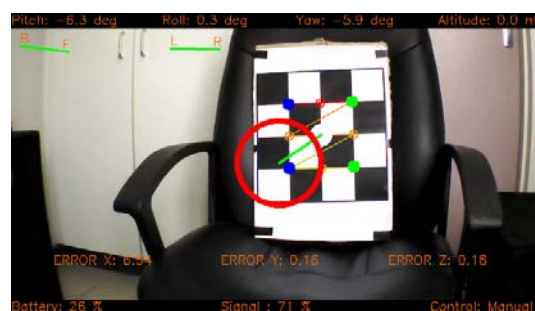

Figure 4: On Screen Display (OSD)

Feedback PID control system

The Ziegler-Nichols tuning method was selected as the primary PID tuning methodology. A particular advantage of using this technique, is that it does not require a mathematical model of the plant. Instead, the technique is carried out with Hardware In the Loop (HIL)

investigations. This allows for the parameters to be tuned according to the actual plant dynamics, thereby contributing to the design of a more effective practical controller.

RESULTS

The aim of this section is to discuss and reflect upon the results observed throughout the system testing procedure. As with the detailed design section, this section once again deals with the two distinct subsystems individually. The individual subsystems were tested independently before the final integration and testing was completed.

<u>Traffic flow estimation</u>

System testing and results analysis is an essential part of determining the efficacy of the methods used in the final system. The performance of the computer vision techniques were tested using four carefully selected video sources. The test videos were chosen to test the system under various degrees of tracking difficulty.

Test videos 1 and 2 were chosen as the baseline comparison tests. Test video 3 was chosen due to the lower overall illumination caused by bad weather conditions. Test video 4 was chosen due to the position of the illumination source during the analysis hour.

In each case, the actual vehicle count is compared to that of the measured vehicle count; with and without shadow removal. Table 2 shows the accuracy of the vehicle counting algorithm before and after the shadow removal technique is implemented.

Table 2: Vehicle counting results

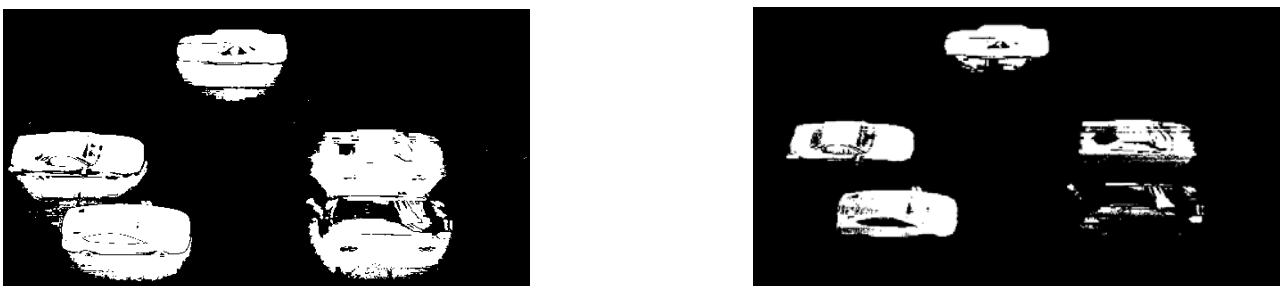| Test	Actual count	With shadows		Shadows removed	
		Count	*Accuracy*	Count	*Accuracy*
Video 1	36	48	67%	36	100%
Video 2	81	53	65%	79	98%
Video 3	29	20	69%	25	86%
Video 4	42	25	60%	50	81%

Figure 5: Shadow removal result. Left-most image shows the foreground mask before shadow removal. Right-most image shows the result of the shadow removal algorithm.

The results from table 2 conclude that shadow removal improves counting accuracy by at least 15 percentage points. The most obvious reason for this increase is attributed to the way the system interprets shadow characteristics. Without the shadow removal functionality, separate vehicles in close proximity to one another are sometimes counted as a single vehicle. In other cases, a shadow is seen as a completely separate moving entity, leading to a single vehicle being counted twice.

In order to test the accuracy of the velocity computations, the actual vehicle velocities were compared to that of the system measured velocities. Two test vehicles were driven past a pole-mounted camera at speeds of 10, 20, 30 and 40km/h (according to speedometer readings). Table 3 shows the vehicle speed estimation results.

Table 3: Vehicle speed results: Test vehicle 1

Run	Direction	Estimated Speed (km/h)	Speedometer Reading (km/h)	Difference (km/h)	Accuracy
1	Right to Left	11.4	10	-1.4	86%
2	Left to Right	21.5	20	-1.5	93%
3	Left to Right	28.6	30	+1.4	95%
4	Right to Left	41.8	40	-1.8	96%

Drone control

The ability of the drone to track a target and minimise the error signal would give an indication of the tracking system performance. The most important indicator would be determined by the efficacy of the landing algorithm to successfully land the drone on an 80x80 cm platform.

In order to obtain a quantitative measure of the landing accuracy, 34 test landings were conducted. After each landing the relative distance from the centre of the platform to the hull of the drone was measured. This measurement would give an indication of the landing accuracy, and would facilitate the calculation of a successful landing probability. Figure 6 shows a scatter plot of the aircraft position after each successive landing.
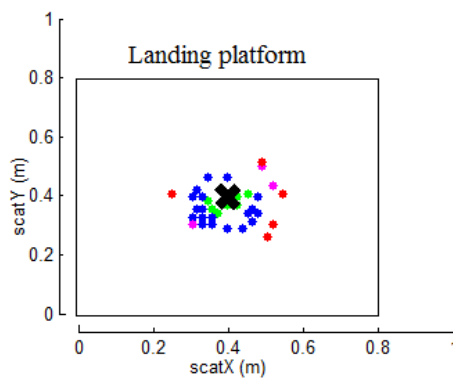

Figure 6: Landing coordinate scatter plot.

Tests were conducted in semi-ideal conditions, where minimal external disturbances were experienced. Prop wash from the aircraft did, however, result in some air turbulence. It is apparent, however, that the control system was able to deal with these disturbances and ultimately stabilise the aircraft above the landing platform. The results depicted in figure 6 show that out of the 34 test runs conducted, 100% were successful landings.

It is impossible to guarantee that the drone will land on the centre of the platform each and every time. To deal with this limitation, the traffic tracking algorithm was designed to be as flexible and as adaptable as possible so that the position of the drone was not of concern. A background model is generated based on the current position of the camera feed, which inherently minimises the limitations placed on the position of the source.

CONCLUSION

This paper addressed two key challenges in the field of traffic flow estimation – laborious manual vehicle counting and the need for multiple and fixed recording infrastructure. The former challenge was addressed by automating vehicle detection and automatic calculation of traffic flow metrics using computer vision techniques. The latter challenge was addressed by introducing drone-based recording equipment that uses pole-mounted landing platforms, making it especially useful in for use in remote and fiscally challenged areas. The results demonstrate that the solutions work with high accuracy, with detection ranging from 81% to 100%, and 100% landing accuracy for the drone.

REFERENCES

A demonstration video of the complete system is available online at: http://goo.gl/jT7lke

Gopal, M., 2010. *Digital control and state variable methods*. Mc Graw Hill.

Kannegulla, A., Reddy, A.S., Sudhir, K.V.R.S. & Singh, S., 2013. Thermal Imaging system for Precise Traffic Control and Surveillance. *International Journal of Scientific & Engineering Research*, 4(11), pp.464-67.

Koller, D. et al., 1994. Towards robust automatic traffic scene analysis in real-time. In *Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference*., 1994.

Lovell, C., Sanin, A., Sanderson, C. & Brain, C., 2012. Shadow Detection: A Survey and Comparative Evaluation of Recent Methods. *Pattern Recognition*, 45(4), pp.1684--1695.

Lucas, B. & Kanade, T., 1981. Proceedings DARPA Image Understanding Workshop., 1981.

Muthukumar, N. & Chintalacheruvu, V., 2012. Video Based Vehicle Detection and its Application in Intelligent Transportation Systems. *Journal of Transportation Technologies*, 2(4), pp.305-14.

National department of transport, 2014. *Live vehicle population*. [Online] Available at: http://www.enatis.com/index.php/downloads/doc_download/575-livevehpopulationclassprov20140331 [Accessed 15 September 2014].

National Research Council, 2010. *HCM 2010 : highway capacity manual*. Washington, D.C.: Transportation Research Board.

OpenCV, 2011. *Optical flow*. [Online] Available at: http://docs.opencv.org/trunk/doc/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html [Accessed 12 September 2014].

Shi, J. & Tomasi, C., 1994. Good features to track. In *IEEE Computer Society Conference*., 1994.

Thies, W. et al., 2013. Accurate speed and density measurement for road traffic in india. In *Proceedings of ACM*., 2013. ACM.