Automatic Traffic Scene Analysis with Belief Networks

by Shannon Imbo

Submission date: 13-Sep-2019 01:11PM (UTC+1200)

Submission ID: 1171763351

File name: Vision_Intelligence_Report_-_Shannon_Imbo_1303744.docx (522.44K)

Word count: 3594

Character count: 19502

Shannon Imbo - 1303744

Vision Intelligence Report

By Shannon Imbo

Artificial Intelligence was employed by Intelligent Vehicle Highway Systems (IVHS) to detect vehicles and decision making. It aids in improving traffic flow especially during peak hours, helps to identify accidents, and allows for better decision-making in autonomous vehicles. Contour tracking and affine motion models using Kalman filters were used primarily for the machine vision component, while the symbolic reasoning aspect used a dynamic belief network to create inferences regarding lane changing vehicles and accidents. The team managed to merge a low-level, vision-based surveillance system for the vision intelligence component of the model, with a high-level symbolic reasoner for the inference making component, to produce the automatic traffic scene analyser.

They successfully implemented a prototype system that integrates multiple vision-based traffic surveillance systems (Koller, Weber, & Malik, 1994). The prototype system is more efficient compared to the traditional loop detectors since the new system is less disruptive, cheaper to install, has greater range and provides more details about the traffic. In addition, dynamic belief networks have great human interpretability due to high-level and symbolic descriptions. Future work would've involved increasing reliability of the system's decision making, improving the surveillance to account for visual noise such as heavy rain, fog, and low-lighting, as well as including detection for sensor failures to continue tracking vehicles without any disturbances.

In terms of related work, IVHS uses three different major applications to develop its methods: Advanced Traveller Information Systems (ATIS), Advanced Traffic Manager Systems (ATMS), and Automated Vehicle Control Systems (AVCS) (ATMS together with ATIS, can be used to redirect traffic to avoid congestions and vehicle stalls. ATMS can be used to analyse busy intersections to identify the ones with high accident risks. AVCS would mainly be used to track nearby vehicle behaviours and assess the condition of incoming traffic lanes to navigate an automated vehicle on a highway (Niehaus & Stengel, 1991). Contour tracking (Kaas, Witkin, & Terzopoulos, 1988) and an affine motion model (Koller, Weber, & Malik, 1994) were of great help in analysing and retrieving vehicle trajectories through the many traffic scene pictures. The Kalman filter formalism served as the foundation for the system's major functionalities and helped develop its inferences even further. To further improve the system's ability to identify vehicles and track them, an adapted version of the moving object segmentation method (Karmann & Brandt, 1990) was implemented. Spline approximations for contouring (Kaas, Witkin, & Terzopoulos, 1988) allowed the model to have 12 different control points on each closed cubic spline to help approximate each vehicle and assign them unique shapes. The dHugin package was initially sought out by the team to try and update the belief network but was later discarded as it didn't provide any additional support to the system in terms of performance and it also lacked flexibility in terms of uncertainty for the dynamic network. Indexical nodes (Agre & Chapman, 1987) were added to dynamic networks to provide different states that a vehicle may be in according to an image (e.g. stalled, changing lanes, going forward, stopped, etc). Finally, traffic simulator software called SmartPath was used to reconstruct images procured by the Traffic Scene Analysis system and create three-dimensional computer-generated renderings.

Shannon Imbo - 1303744

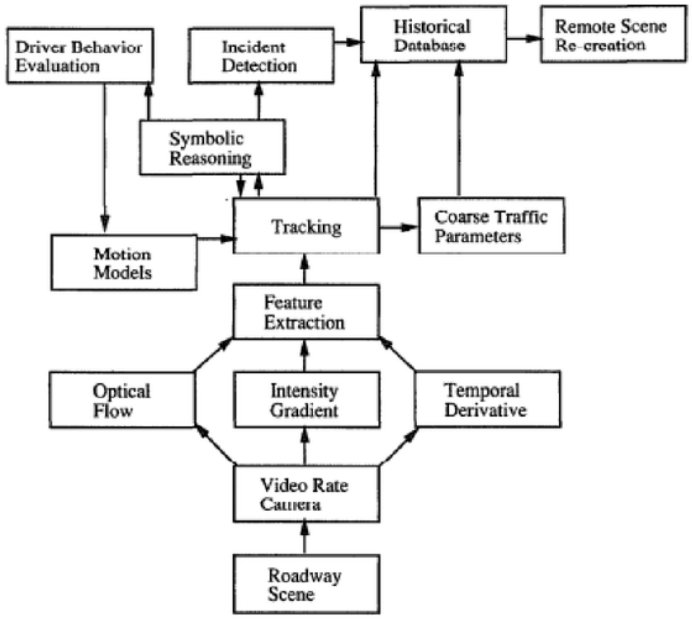

Figure 1: A complete diagram of the traffic surveillance system which includes the activities' flow. (Huang, et al., 1994)

The first low-level part of the traffic surveillance system to be discussed is its ability to extract features and tracking through vision-based intelligence. It is initially fed information through a series of different traffic images. Through vision-based intelligence, it detects and tracks vehicle shape and position while also associating these shapes between the images. It must achieve this task consistently through all the image sequences. Sensory noise and vehicles blocking other vehicles were the major issues for the system and so the Kalman filter was applied to handle vehicle motions by gathering data through repeated observations. The surveillance system is also capable of distinguishing moving vehicles from the background difference in pixel intensities in each new frame. They utilized a modified version of the Kalman Filter (Karmann & Brandt, 1990) to allow the model to adapt and identify the background according to the different weather and time of day conditions. The frame is updated per frame with this equation:

$$Bt + 1 = Bt + (a1(1 - Mt) + a2Mt)Dt$$

A quick summary of the equation is *Bt* is the background model at time *(t)*, *Dt* refers to the difference between the current frame and *Bt*. *Mt* represents a binary mask of moving objects within the present frame. Lastly, *a1* and *a2* measure the rate of change in *Bt* (Koller, Weber, & Malik, 1994).

The program identified cars as a blob-like shape with each vehicle having its own unique form. Using this method, it was able to differentiate individual vehicles in frames and to estimate their shapes as well as calculate vehicle trajectories. This was done through the extraction of closed contours of each individual vehicle, based on object motion, grey-value boundaries taken from spatial image gradients, and deriving the time of each image frame. When a "blob" passes a threshold test, it is then enclosed in a convex polygon and this is then used as the object's description by the system. A problem that occurred was that time-recursive shapes were inappropriate to use due to the potential of the number of vertices for a vehicle changing during an image sequence. This was solved by applying closed cubic splines with 12 control points to use as an approximation of the convex polygon and adding the control points retrieved by the Kalman Filter.

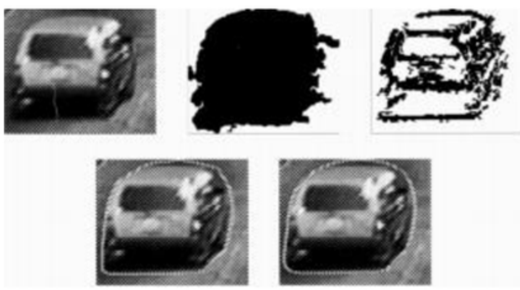

Figure 2: The first image is the original image of the car. The 2nd one then adds the mask and the 3rd represents the image's locations with acceptable spatial gradients along with its temporal derivatives. The 1st photo in the bottom has the first convex polygon enclosure applied to it and the last one utilizes the cubic spline approximation which smooths out the shape of the car for clearer identification (Huang, et al., 1994).

Affine transformation was used as vehicles only moved on the road and they tend to only move in a straight line in highways with occasional lane changing. The following equation was used to calculate the velocity:

$$u(x) = s(x - xm) + u0$$

S = scale parameter. If the value is 0 then there is no change in the scale. Positive and negative values of *s* indicate that the motion components are either away or towards the camera. *Xm* represents the centre of the moving image region while *u0* shows the displacement between two consecutive frames. The team applied a third Kalman filter to measure motion parameters since a linear matrix equation can be used to express the affine transformation stated above and further improve their calculations (Koller, Weber, & Malik, 1994).

Another issue that the team experienced were overlapping vehicles. It can distort the contours that have been extracted, making identification less accurate as well as affecting the vehicle's trajectories. To address occlusion reasoning, the team exercised the use of an algorithm designed to compensate for vehicles overlapping with each other in images. The algorithm was effective since the team knew the geometry of the images and it was simple to approximate the vehicle's trajectory as it is limited to the ground plane. A depth ordering was then applied to each vehicle, making it possible for vehicles to occlude with one another.

Once the positions and velocities of all vehicles have been calculated, a high-level symbolic description was then applied to both the vehicles and the traffic scene itself. This was done by applying multiple dynamic belief networks on each vehicle along with fast rollup. The concept was originally procured from J. Pearl, who, in 1988, used nodes to represent random discrete variables and arcs to display casual connections between each variable in a directed acyclic graph. Conditional probability was applied to each node's possible state, which is also affected by their parent nodes, and each probability was turned into a table to provide a mimic for uncertain events. Posterior probability distributions can be computed and retrieved for each subset of nodes. The belief network was used to compute and make inferences under uncertainty. The topology of the network was expressed through arcs that showed evidence of relationships between variables. By making the relationships conditionally independent, the overall topology's size was significantly reduced as there were less probabilities that needed to be defined compared to a full joint probability distribution. The dynamic belief network was useful when applied to vehicles as the variables for each object change over time. Network structures replicated observations that were segmented into different 'time slices. Nodes were connected to other nodes current and previous time slices. Slices are updated every time a new slice is created but the old slice is then 'rolled-up' into the new slice to recalculate the probability table.

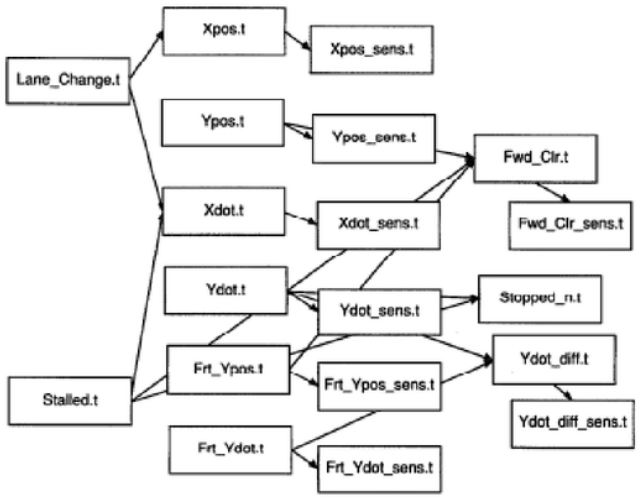

Figure 3: A portion of the initial belief network for one vehicle (Huang, et al., 1994).

The table represents the potential state of the vehicle and its probabilities. To elaborate the table further, Xpos_sens.t corresponds to a sensor that determines the vehicle's position on the X-axis which translates to its left-right positioning on the image's time slice. Another example is Lange_change.t which is a high-level event on which it can take on three different states whether it is going straight, moving to the left lane, or moving to the right lane.

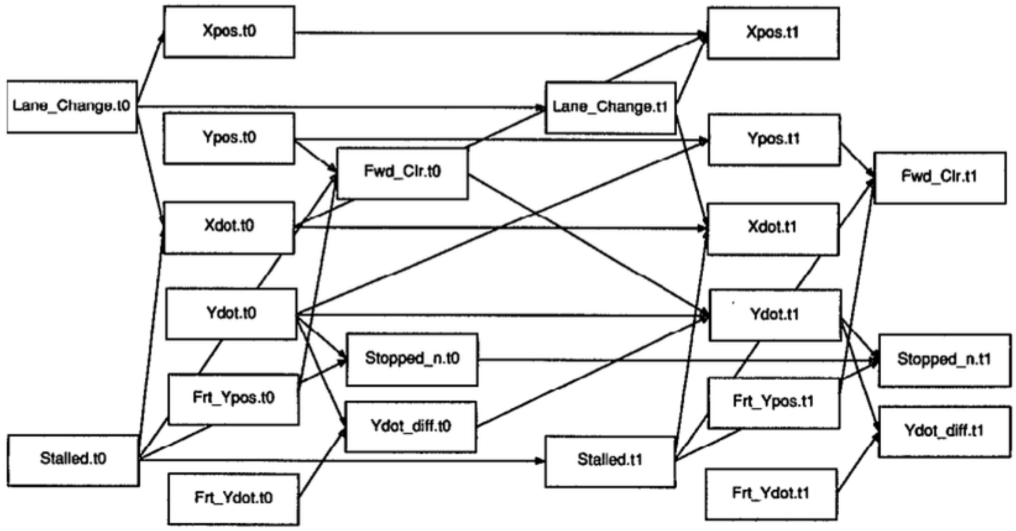

Figure 4: A belief network for one single vehicle with another time slice added into it. *T0* indicates the first posterior time slice while *T1* indicates the one after (Huang, et al., 1994).

To understand how the belief network interacts with time slice, *Ypos.t1* indicates a vehicle's forward position on the highway and it depends highly on *Ypos.t0* velocity to determine its own. The belief network can also exhibit how vehicles react to their neighbouring vehicles and adhere to road rules. For example, *Ydot.t1*'s state (a range of speed of the vehicle such as 21-30 km/hr, 31-40 km/hr, and so on) will be determined according to the state of *Ydot.t0*, *Fwd_Clr.t0* (the space in front of the vehicle), and *Ydot_diff.t0* (the difference of speed between the vehicle and the vehicle in front of it). If the vehicle in front of it is going much slower than its own velocity, then it's much slower than its own velocity, then its *Ydot.t1* value will be lower than *Ydot.t0*.

As each individual vehicle have their own belief network, it is easy to imagine that it can get quite overwhelming for the system. Position and velocity from a vehicle will affect another vehicle and vice versa. To make all probability distributions consistent throughout the entire system would require a large network with interchanging connections between each vehicle's subnetwork. It would also prove too expensive for computational power as each network would need constant modifying and recompiling at different time slices. The team decided to assign unique dynamic belief networks for each vehicle instead of creating a consistent global scale network as an initial solution to the problem. To support the existing solution, indexical nodes were utilized to calculate a vehicle's variables and how it may be affected by nearby vehicles. Sensor data was used to pre-process node states according to vehicles in front of each other. This resulted in an effective alternative to the probability distributions as it achieved a locally consistent high-level description for each vehicle.

Each node has variable semantics incorporated into them. This means that nodes have different states in different time slices and this is achieved by modifying a node's conditional probability table in each different time slice. An example would be that, *Stopped_n.t* node represents how long the vehicle has been stopped for *n* time slices. The node then acts as a counter and if *n* reaches, say 50 time slices, then it is then given positive probability of the vehicle being stalled (*Stalled.t)*.

The HUGIN (Andersen, Olesen, Jensen, & Jensen, 1989) system was originally meant for standard dynamic beliefs but it was vital for the traffic surveillance system to roll the network forward by modifying, making it more dynamic. The team managed to add the ability to add new time slices to the network and include old information from previous time slices to cleanly delete old time slices. Their initial approach was to create new networks for each new time slice but resulted in poor performance. An alternate approach that the team attempted was using two precompiled networks with each having two slices which allowed the system to alternate between the two networks. When new sensor information was presented by a new time slice, the system utilized the oldest time slice for the whole model by using sequences of matrix multiplication. Then, the new probability tables were stored in the first slice of every other network, but the new sensor information was added to the second slice of the current network, and its influence was then transmitted to gain new posterior probabilities. The new network was then used for the next time slice until the first network started the rollup procedure. This method prohibits alternation of the dynamic belief network structure, but it improves performance drastically by avoiding recompilation of the network for every time slice.

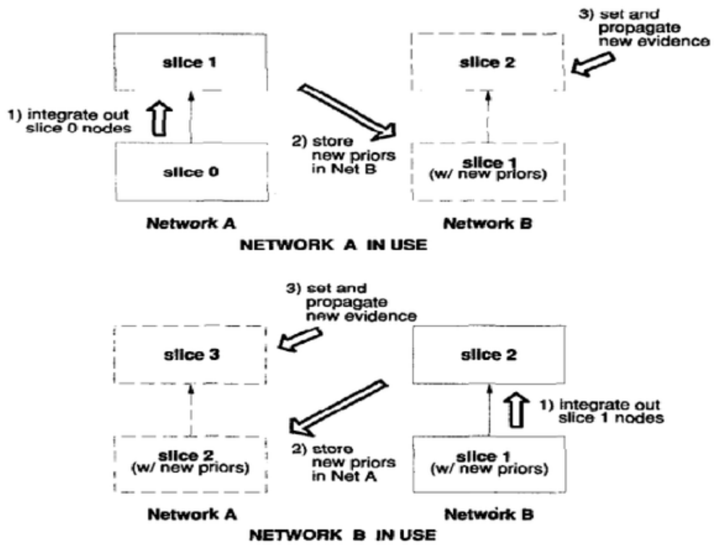

Figure 5: Flow chart of dynamic network rolling forward (Huang, et al., 1994).

The best way to analyse how successfully the system performed was to apply it to real-world scenarios. The team was able to test their system with real-world applications by providing it with 270 different images in a sequence which contained a four-lane highway. Their results displayed most of the abilities mentioned in the methods section of this report and can be seen in the figures below.

Shannon Imbo - 1303744

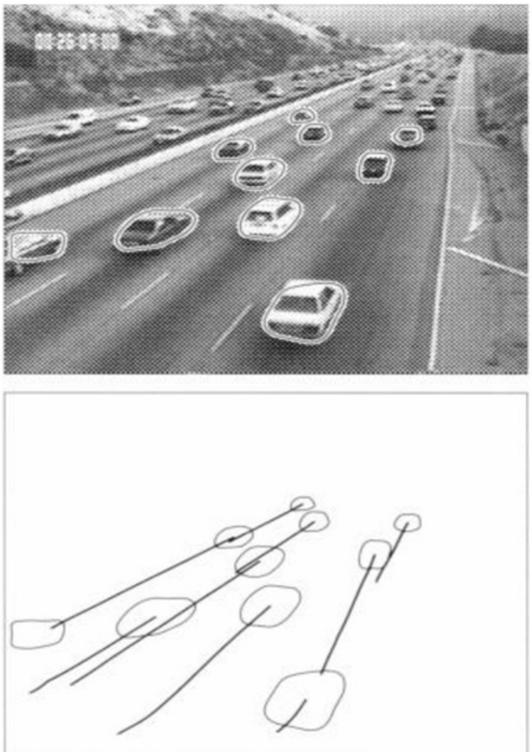

Figure 6: Images showing overlaid contour estimates with convex polygons and their contour estimates with trajectories (Huang, et al., 1994).

As per figure 6, the above image shows how the system was able to successfully apply a convex polygon contour on the vehicles to approximate their shapes. The system did very well in distinguishing the vehicles from the background using the Kalman filter-based adaptive background model (Karmann & Brandt, 1990). Though the shape is not perfect, what matters most is that the system is able to individually identify each vehicle. On the bottom image of figure 6, it displays a simpler image but it easier to interpret for humans and it also displays how well the system calculates the trajectories of the vehicles. The team also mentions that the bottom image found in figure 6 started from frame #0. The line is not perfectly straight which exhibits that the model was correcting itself as it observes and evaluates the vehicle's trajectory frame by frame. This was achieved by the combination of the Kalman Filter and the Affine Transformation algorithms by first applying contours to the vehicle, then applying the transformation algorithm to assess the trajectory. The occlusion reasoning algorithm is also at work in the background of the scene although it is not clearly shown in the images. Evidence of it working can be seen in the lower image of figure 6. The trajectory of the vehicles wouldn't be so accurately displayed if occlusion reasoning wasn't at work as vehicles move at different speeds. Due to some cars being slower than others, faster vehicles must've passed the slower ones, hence blocking or occluding the slower vehicles from the camera. The occlusion algorithm would've calculated the vehicle's depth to continue tracing it behind another vehicle. The overall result from the algorithms displayed excellent outcomes as portrayed in figure 6. The system operated on a Sun SparcStation 10 which is an outdated workstation computer and would've limited the computational abilities of the system. If the same system was performed on a more modern computer, then it should be able to render more images and be able to apply belief networks to more vehicles.

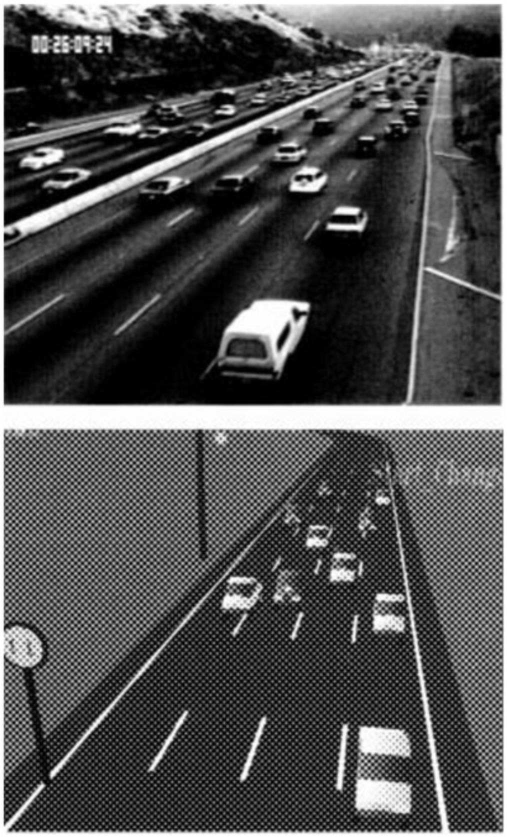

Figure 7: The below image shows the SmartPath being used to refabricate the original traffic scene above.

Future work will contain artificial knowledge about driver traits such as lane-changing and braking behaviour. Road geometry and weather conditions will also be anticipated by the system to augment inferences. It'll also visually identify vehicle lights such as the tail lights emitting red to signify braking, hazard lights flashing yellow to identify potential stalled vehicles, and signal lights to improve vehicle trajectories. Due to the limitations of the technology present during 1994, the system would have also been very restricted to its available processing power. Though unmentioned in the paper, assumptions could be made about how technological breakthroughs in computer hardware would've been eventually available for the Traffic Analysis System to operate on. This would've enhanced its original boundaries on the dynamic belief networks mentioned above.

Since the symbolic reasoner can already detect signal lights, the vision system was scheduled to be upgraded to detect these activities and include vehicle shadows for better vehicle detection. Improvements could be further achieved by integrating both the visual and the reasoning components of the system in terms of decision making and better inferences. For instance, a vehicle signalling to the right would mean that its expected trajectory would be biased towards the right direction. Currently, the system does not assume anything in terms of vehicle signals or driver behaviours as it needs to wait for input from the traffic scenes and calculates the output based on these details. Another feature to be added is an ability to continue tracking a vehicle if the video surveillance loses "sight" of the vehicle due to heavy rain or other interferences. This can be achieved by predicting its most likely position and detecting its neighbouring vehicles' behaviours. Simple sensors will be used to identify any other sensors failing, allowing the remaining working sensors to continue tracking the vehicle uninterrupted. Finally, the team intends to perform more vigorous tasks for the system such as providing it more extensive video sequences and changing the hardware that they originally made the system with.

Shannon Imbo - 1303744

The prime issue with system is that image quality had to be as clear as possible and with minimal noise and clutter. This would mean that weather conditions had to be bright and sunny for the system to accurately identify and track vehicles. Lighting conditions also played a big role as the system heavily relied on this to differentiate vehicles from backgrounds in different frames. In terms of belief networks, it simply reacted to how it perceived vehicles, but a more effective way would be to combine reactions with predictions to improve the accuracy in tracking and identification. Real-world scenarios are never ideal as there are a lot of variables usually occurring and uncertainties are always present. Video surveillance issues could range from the weather changing, clouds altering the lighting, or even birds blocking the camera lens. The dynamic component for making inferences could also suffer from bad driver behaviours such as failing to indicate, vehicles being in between lanes (driving on the line), and not following road regulations such as speeding. These issues could be mitigated by creating a database of driver behaviour history to help the system make decisions based on previous actions that the system has observed.

To conclude, the artificial intelligence performed very well in a controlled environment, but it must also be further trained to interact with noises and disturbances. The model can be better trained by using algorithms such as Naïve Bayes, K-nearest neighbour, decision trees, and even combining multiple algorithms (meta-learning) to be compared to an observation where the results are known. The AI's observation (training model) can be compared to a dataset where the correct predictions are already known to continuously test the model's ability to correctly identify vehicles. Classification can be used to help identify vehicles of similarity in terms of shape (such as SUVs and sedans) and clustering can even be used to link driver behaviours to improve decision making.

References

Agre, P., & Chapman, D. (1987). Pengi: An Implementation of a Theory of Activity. *Proceedings of the Sixth National Conference on Artificial Intelligence*.

Andersen, S., Olesen, K., Jensen, F., & Jensen, F. (1989). HUGIN - a Shell for Building Bayesian Belief Universes for Expert Systems. *Proceedings of the Tenth International Joint Conference*.

D, K., J, W., T, H., J, M., O, O., Rao, B., & Russell, S. (1994). Towards robust automatic traffic scene analysis in real-time. *Proceedings of 12th International Conference on Pattern Recognition.* Jerusalem: IEEE.

Gupte, S., Masoud, O., Martin, R., & Papanikolopoulos, N. (2002). Detection and classification of vehicles. In *IEEE Transactions on Intelligent Transportation Systems* (pp. 37-47). IEEE. Retrieved from https://ieeexplore.ieee.org/Xplore/home.jsp: https://ieeexplore.ieee.org/abstract/document/994794

Huang, T., Koller, D., Malik, J., Ogasawara, G., Rao, B., Russel, S., & Weber, J. (1994). Automatic Symbolic Traffic Scene Analysis Using Belief Networks. Berkeley, California, USA.

Kaas, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active Contour Models. In *International Journal of Computer Vision 1* (pp. 321-331).

Karmann, K.-P., & Brandt, A. v. (1990). Time-Varying Image Processing and Moving Object Recognition. *Proceedings of the 5th International Workshop*. Amsterdam.

Kilger, M. (1992). A shadow handler in a video-based real-time traffic monitoring system. *Proceedings IEEE Workshop on Applications of Computer Vision.* Palm Springs: IEEE.

Kjaerulff, U. (1993). User's Guide to dHUGIN. Aalborg University: Institute of Electronic Systems.

Koller, D., Weber, J., & Malik, J. (1994). Robus Multiple Car Tracking with Occlusion Reasoning. In *Motion Segmentation and Tracking* (pp. Computer Vision - ECCV '94). Berlin: Springer.

Niehaus, A., & Stengel, R. F. (1991). Rule-Based Guidance for Vehicle Highway Driving in the Presence of Uncertainty. *American Control Conference*.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* California: Morgan Kaufmann Publishers.

Automatic Traffic Scene Analysis with Belief Networks

ORIGINALITY REPORT

15%	7%	13%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	S. Russell. "Towards robust automatic traffic scene analysis in real-time", Proceedings of 12th International Conference on Pattern Recognition ICPR-94, 1994 Publication	5%
2	Submitted to AUT University Student Paper	1%
3	polen.itu.edu.tr Internet Source	1%
4	www.coursehero.com Internet Source	1%
5	info.cs.uab.edu Internet Source	1%
6	www.macs.hw.ac.uk Internet Source	1%
7	archive.org Internet Source	1%
8	people.eecs.berkeley.edu Internet Source	1%

9	S. D. Mayunga. "A semi-automated approach for extracting buildings from QuickBird imagery applied to informal settlement mapping", International Journal of Remote Sensing, 5/2007 Publication	1%
10	STEVEN E. SHLADOVER. "Review of the State of Development of Advanced Vehicle Control Systems (AVCS)", Vehicle System Dynamics, 1995 Publication	1%
11	S. Russell. "Towards robust automatic traffic scene analysis in real-time", Proceedings of 1994 33rd IEEE Conference on Decision and Control CDC-94, 1994 Publication	<1%
12	José Javier López. "3-D Audio with Video Tracking for Multimedia Environments", Journal of New Music Research, 9/1/2001 Publication	<1%
13	calhoun.nps.edu Internet Source	<1%
14	Jose Manuel, Sergio Luis, Manuel Vargas, Federico Barrero. "Chapter 4 Computer Vision Techniques for Background Modeling in Urban Traffic Monitoring", InTech, 2010 Publication	<1%

15	resource.isvr.soton.ac.uk Internet Source	<1%
16	Submitted to University of Newcastle upon Tyne Student Paper	<1%
17	web.satd.uma.es Internet Source	<1%
18	www2.ceegs.ohio-state.edu Internet Source	<1%
19	D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, S. Russell. "Towards robust automatic traffic scene analysis in real-time", Proceedings of 12th International Conference on Pattern Recognition, 1994 Publication	<1%
20	Uffe Kjærulff. "dHugin: a computational system for dynamic time-sliced Bayesian networks", International Journal of Forecasting, 1995 Publication	<1%
21	Marie-Pierre Dubuisson. "Contour extraction of moving objects in complex outdoor scenes", International Journal of Computer Vision, 01/1995 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		