

## An Alternative Vehicle Counting Tool Using the Kalman Filter within MATLAB

Daphne Espejel-García <sup>a\*</sup>, Luis Ricardo Ortíz-Anchondo <sup>a</sup>, Cornelio Alvarez-Herrera <sup>a</sup>, Alfonso Hernandez-López <sup>a</sup>, Vanessa Verónica Espejel-García <sup>a</sup>, Alejandro Villalobos-Aragón <sup>a</sup>

<sup>a</sup> Facultad de Ingeniería, Universidad Autónoma de Chihuahua, Circuito Universitario SN, Chihuahua, Chih., CP 31000, Mexico.

Received 16 October 2017; Accepted 25 December 2017

### Abstract

This study proposes an alternative and economical tool to estimate traffic densities, via video-image processing adapting the Kalman filter included in the Matlab code. Traffic information involves acquiring data for long periods of time at stationary points. Vehicle counting is vital in modern transport studies, and can be achieved by using different techniques, such as manual counts, use of pneumatic tubes, magnetic sensors, etc. In this research however, automatic vehicle detection was achieved using image processing, because it is an economical and sometimes even faster option. Commercial automatic vehicle detection and tracking programs/applications already exist, but their use is typically prohibitive due to their high cost. Large cities can obtain traffic recordings from surveillance cameras and process the information, but it is difficult for smaller towns without such infrastructure or even assigned budget. The proposed tool was developed taking into consideration these difficult situations, and it only requires users to have access to a fixed video camera placed at an elevated point (e.g. a pedestrian bridge or a light pole) and a computer with a powerful processor; the images are processed automatically through the Kalman filter code within Matlab. The Kalman filter predicts random signals, separates signals from random noise or detects signals with the presence of noise, minimizing the estimated error. It needs nevertheless some adjustments to focus it for vehicle counting. The proposed algorithm can thus be adapted to fit the users' necessities and even the camera's position. The use of this algorithm allows to obtain traffic data and may help small cities' decision makers dealing with present and future urban planning and the design or installment of transportation systems.

**Keywords:** Image Processing; Kalman Filter; Automated vehicle Detection; Traffic Densities Counting; MATLAB.

### 1. Introduction

An accurate estimation of vehicle densities is considered as a significant factor for managing traffic operations in urban zones. Traffic information includes vehicle counting, speed measurements, determination of peak hours, number of pedestrians, etc. [1]. Large cities usually do not have problems acquiring such information, because they have an adequate infrastructure to obtain it such as surveillance cameras throughout the city, speed check points and continuous vehicle counting. However, in smaller cities, without such infrastructure or even an assigned budget, traffic data should be attained in a manual manner or by adapting available technology.

Typically, vehicle counting is achieved by using different techniques, such as manual counts, pneumatic tubes, magnetic sensors, inductive loops, piezoelectric sensors, acoustic detectors, passive infrared, doppler and radar microwave sensors [2, 3]. However, sensor measurements exhibit estimation errors, and sometimes, present difficulties to install and/or remain fixed. For optimizing vehicle counting, automatic video-based tracking systems are

\* Corresponding author: [despejel@uach.mx](mailto:despejel@uach.mx)

 <http://dx.doi.org/10.28991/cej-030935>

➤ This is an open access article under the CC-BY license (<https://creativecommons.org/licenses/by/4.0/>).

© Authors retain all copyrights.

recommended [4]. The system detects and tracks vehicles while they pass through the camera, providing information such as vehicle count, average speed, or traffic congestions [5-7]. To achieve it, an algorithm must be created that segments and tracks the vehicle movements within the video [6, 8, 9].

Although automatic vehicle counting software already exist, and some of them use the Kalman filter [7, 10-14], their price is elevated, and in some cases, may be prohibitive. An alternative vehicle counting tool for low budget conditions is proposed, which involves video-image processing with the Kalman filter in Matlab. Users will only need a fixed camera on a high point (e.g. pedestrian bridge, light pole) and a computer with a powerful processor. The video is processed automatically through a Matlab code.

## 2. Background

The developed vehicle counting tool was tested in an express way, called Periferico de la Juventud (PDJ), considered the most important traffic road of Chihuahua City, which is the capital of the state of Chihuahua, located in northern Mexico (Fig. 1). In 2010, the city had a population of 809,232 inhabitants, with a ratio of one vehicle for every two inhabitants [15], and nowadays, approximately 500,000 vehicles are registered in the city [16]. Chihuahua City is considered a metropolitan zone, and its road infrastructure occupies about the 42% of urban area, and contains a dozen main roads, with traffic jam issues during peak hours [17]. PDJ is located west of Chihuahua City; its length is ca. 20 km and has a N-S direction (Fig. 1). According to Chihuahua city's planning department, PDJ was not properly designed, resulting in a series of bottlenecks appearing due to incorrect location of in/out ramps [17]. PDJ does not have traffic lights, and communicates both residential and business zones along its way.

Currently, Chihuahua City faces public transportation problems [18]; the number of automobiles has augmented, requiring a larger circulation space within the city [17]. Therefore, it is vital to obtain vehicle counting data to be able to simulate traffic scenarios, and thus to propose answers integrating displacements of pedestrians, drivers, cyclists, and public transportation users.

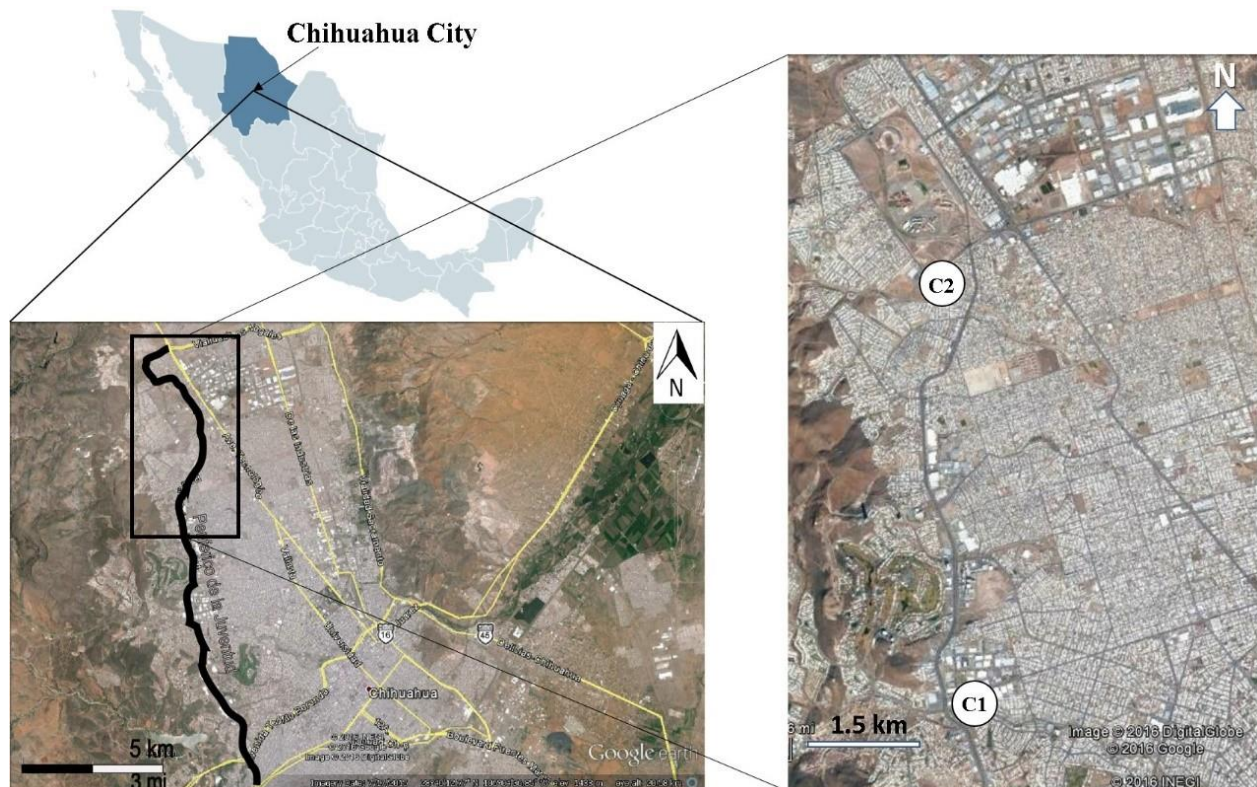


Figure 1. Location map of the case study, where C1 and C2 are the two points of video and manual data acquisition [19].

## 3. Methodology

### 3.1. The First Test Case: Ideal Dam Break Over Movable Bed

Currently, image processing is considered as a suitable tool to improve image quality or to look for information amongst them. Analyzing images in real time involves several processes, such as the study of iconographies' flux captured by a camera, and the system control based on the results [20]. One of the major problems to be considered, is working with image sequences is degradation due to presence of noise, which appears while acquiring the images [11,

21-23]. It is imperative then to remove the noise and process the images in real time [24] by applying a filter, such as Kalman, which can separate random signals from noise [25].

The Kalman filter is an algorithm designed to function as an estimator, that minimizes the estimated error [26-28]. The filter predicts random signals, separates signals from random noise or detects signals with the presence of noise. The filter uses a sequence of measurements detected over time, and produces more accurate variable estimates [25]. Nowadays, the Kalman filter is widely used in navigation and guidance technology [29], vehicle control (air- and spacecraft) [30], signal processing and econometrics [29]. Traffic engineering has embraced this method in image processing.

For traffic data, the Kalman filter was first used to predict vehicle volumes that circulated at a certain point and in short term [31, 32]. Subsequently, vehicle tracking algorithms have been created using the Kalman filter to obtain improved estimates of traffic densities and to facilitate the data acquisition [4, 7, 10, 33-35]. There are also new algorithms for vehicle detection that combine the Kalman filter with other methods, such as the hybrid median filter [13], but require a more complex programming language.

### 3.2. Data Acquisition

The processed images were recorded by a fixed camera positioned in two pedestrian bridges (points C1 and C2, Figure 1) located in the middle (C1) and in the northern part (C2) of PDJ, facing north. Along with the video images procurement, manual vehicle counting was completed to validate the automated counting tool.

The camera used was a 9.2 megapixels video HD handycam, installed in a tripod over two different pedestrian bridges. The algorithm was performed in Matlab 2013b, on a laptop with 1 TB Rom, 12 GB Ram and icore 7 processor under Windows 10 OS. The speed of processing images was 4 fps.

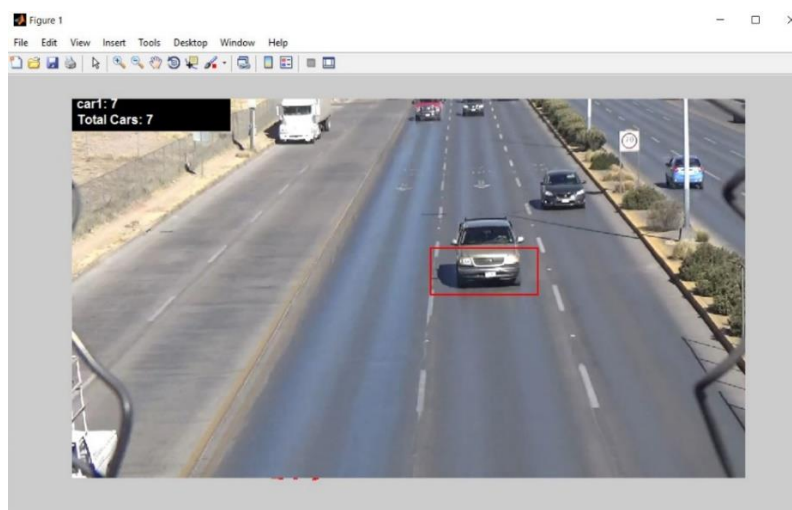
### 3.3. Video Analysis for Vehicle Counting

Matlab was the selected software to process the images for its accessibility and for containing the Kalman filter function. The Kalman filter code for Matlab already exists, but it needs modifications to focus it for vehicle counting. To generate the video analysis, several adjustments were applied, which consisted in: (a) configuring the vehicle detection and tracking by adding a corner box with the total count, (b) delimiting geometrically and manually the lanes to analyze, and (c) eliminating the noise that increases the total sum, such as shades and other objects (e.g. pedestrians, sidewalk edges).

For circumstances where cities lack surveillance videos or have budget issues, videos can be obtained placing a fixed camera on a high point, such as a pedestrian bridge or any light or electricity pole. For analysis purpose and a better performance in Matlab, images should be in black and white, and the videos must be converted to .AVI or .MP4 format.

### 3.4. Vehicle Detection in MATLAB

Detection and vehicle tracking within Matlab, using the Kalman filter, is a practical method to evaluate the traffic density at a given point (Figure 2).



**Figure 2. Vehicle counting in Matlab. Camera was stationary over a pedestrian bridge (C2) at Periferico de la Juventud.**

To make the code work, a flow chart (Figure 3) and the following steps should be followed:

- 1) To accelerate image processing and to obtain the vehicle total count quicker, it is recommended to cut the video every four images.
- 2) Matlab's "Built-in" function "ginput()" allows the user to move and click the mouse to get coordinates in 2D, in this case, to delimitate the lane width that defines the counting area.
- 3) The large shades or the headlights are usually counted as a vehicle and cause error. The shades' RGB range tends to be low (the closest to [0,0,0], the darker the image). Depending of the time the video was taken, the shades' colors are matched with the asphalt color using a cycle, skipping the code block tolerance that added vehicles to the counting. It is necessary to analyze all the frame and to know its size, by assigning a size "X" to a variable "nz" to create the cycle.
- 4) Once the area is delimited and the shades eliminated, the Kalman estimation is performed. The program displays a rectangle that identifies the vehicle movement and size, reporting it as one count. A tolerance range must be established for the result. With a variable flag, the program decides if the movement detected is a vehicle and if it has been already counted. Pixels in the background are not altered by the vehicle detection.
- 5) A corner box can be added to be displaying the vehicle counting.

The variables employed in the Kalman filter algorithm are declared as the following matrixes:

- R – Noise measurement matrix
- H - Transform from measure to state matrix
- Q - System noise matrix
- P - The status covariance matrix
- A - State transform matrix
- X - Current frame.

From this point the Kalman algorithm is applied to reduce the possible noise present in the coordinates enclosing the rectangular space where the detected vehicle is located. This is basically the prediction and correction stages, inherent to the Kalman filter. During the prediction phase, an *a priori* estimate of a given state at a time k is given, and an error covariance associated with the *a priori* estimate. In the correction phase, a measurement update needs to be calculated, as well as the Kalman gain, *a posteriori* estimation, and its error covariance associated with *a posteriori* estimation.

The initial variable values were first, as an input the actual frame with the background already subtracted, so it could be used in the correction phase and to be able obtain the updated value of the matrix  $X_k$ , as well as with the value of the Kalman gain (the closer it is to 0, the reliability of the prediction is greater) an updated value is obtained. And each time the cycle progresses, the pixels containing the input change, and therefore, the prediction variable.

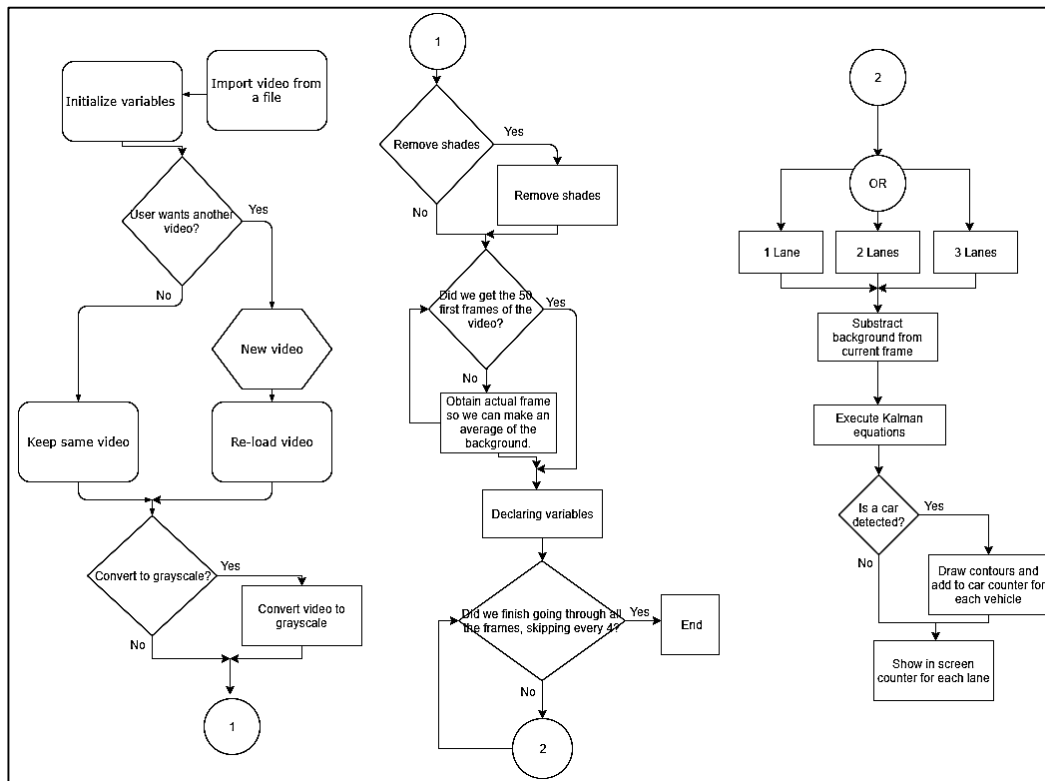


Figure 3. Proposed procedure of applying the Kalman filter to images

## 4. Results

The Matlab code effectiveness was measured in distinct video recordings, at different city points. In Table 1, the Matlab analyses (automated) for one lane are presented along with the manual counting, to measure the efficiency of the suggested code. C1 and C2 are recording videos from two different points, facing the same direction. The best performance was attained while using time periods of 90 seconds, avoiding the program cycling. By using a more powerful processor, analyzed time periods can be increased. The difference in the total count is attributed to the variable speed in the vehicles. The slowest detected movement sometimes is counted as two vehicles. Errors can also be accounted for extreme weather conditions (heavy rain, snow, hail) that may change the asphalt color, thus the shade does not match with it.

**Table 1. Difference in the number of tracked vehicles in Matlab (automated) and in a manual manner.**

Video time (s)	C1 Automated	C1 Manual	Difference	C2 Automated	C2 Manual	Difference
0 - 90	20	21	1	30	28	2
91 - 180	22	20	2	28	27	1
181 - 270	28	27	1	25	25	0
270 - 360	18	18	0	28	26	2
361 - 450	21	23	2	29	29	0
451 - 540	23	21	2	31	30	1
541 - 630	25	25	0	33	32	1
631 - 720	24	23	1	27	27	0
721 - 810	17	15	2	27	26	1
811 - 900	20	20	0	26	25	1
Total	218	213	11	284	275	9
Error%			2.3%			3.2%

As observed in Table 1, there is a tendency of counting one or two more vehicles in some video segments, presenting an error of 2 – 3%, which can be discarded, unless a more accurate study is needed. If this is the case, the manual count is recommended. The presented program can be improved to add speed detection.

## 5. Conclusions

The result of adjusting Kalman filter function within Matlab to create a vehicle counting tool, can be considered as an economical alternative for obtaining traffic data. The algorithm can be adapted to fit the users' necessities and/or to the camera's position. At this stage, it is recommended to process videos with a length smaller than 90 seconds to avoid the failing of the processor. Likewise, it is suggested to analyze one, or maximum three lanes at a time to optimize the program performance, which can also be valuable to obtain the traffic occupancy rate. This code is useful at peak hours, when a manual count is programmed, because only one person is needed to complete it. In this case, Matlab was the selected software to achieve the goals in this paper, but the Kalman filter can be used to process images in any other platform of your choice.

## 6. Acknowledgements

The authors express their gratitude to the Facultad de Ingenieria from the Universidad Autonoma de Chihuahua, and to the National Science and Technology Council of Mexico (CONACYT) for all their support. Authors are also thankful to anonymous reviewers for their valuable comments.

## 7. References

- [1] Garber, N.J. & Hoel, L.A. (2005). Traffic and highway engineering. Cengage Learning. New York, Thomson ed., 1170 p. ISBN 10: 0495082503.
- [2] Zheng, P., & Mike, M. (2012). An investigation on the manual traffic count accuracy. Procedia-Social and Behavioral Sciences, 43, 226-231. DOI: 10.1016/j.sbspro.2012.04.095.
- [3] Findley, D. J., Cunningham, C. M., & Hummer, J. E. (2011). Comparison of mobile and manual data collection for roadway

- components. *Transportation Research Part C: Emerging Technologies*, 19(3), 521-540. DOI: 10.1016/j.trc.2010.08.002.
- [4] Jung, Y. K., & Ho, Y. S. (1999). Traffic parameter extraction using video-based vehicle tracking. In *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on* (pp. 764-769). IEEE. DOI: 10.1109/ITSC.1999.821157
- [5] Park, M.W., Palinginis, E., Brilakis, I., Laval, J., Hunter, M., & Guensler, R. (2014). Unsupervised framework traffic counting: speed estimation based on camera network data. In *Computing Civil and Building Engineering*, 1594-1601. DOI: 10.1061/9780784413616.198
- [6] Bhaskar, P.K. & Yong, S.P. (2014). Image processing based vehicle detection and tracking method. In *Computer and Information Sciences (ICCOINS), 2014 International Conference*, IEEE, 1-5. DOI: 10.1109/ICCOINS.2014.6868357
- [7] Moridani, A.K., Fakhrmoosavy, S.H., & Moridani, M.K. (2015). Vehicle detection and tracking in roadway traffic analysis using Kalman filter and features. *International Journal of Imaging and Robotics*, 15(2), 45-52, ISSN 2231-525X.
- [8] Lee, K.W., & Kim, J.W. (1999). An automated abstraction and search for surveillance video based on spatio-temporal events of objects. *ISO/TEC JTC1/SC29/WG11MPEG99/p615*.
- [9] Almagambetov, A., Velipasalar, S., & Casares, M. (2015). Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras. *IEEE Transactions on Industrial Electronics*, 62(6), 3732-3741. DOI: 10.1109/TIE.2015.2400420
- [10] Bas, E., Tekalp, A. M., & Salman, F. S. (2007, June). Automatic vehicle counting from video for traffic flow analysis. In *Intelligent Vehicles Symposium, 2007 IEEE* (pp. 392-397). IEEE. DOI: 10.1109/IVS.2007.4290146
- [11] Zhou, J., Gao, D. & Zhang D. (2007). Moving vehicle detection for automatic traffic monitoring. *IEEE Transactions on Vehicular Technology*, 56(1), p. 51-59. DOI: 10.1109/TVT.2006.883735.
- [12] Suryatali, A. & Dharmadhikari, V. B. (2015). Computer vision based vehicle detection for toll collection system using embedded Linux. In *Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference IEEE*, 1-7. DOI: 10.1109/ICCPCT.2015.7159412
- [13] Vishnu, V.M., Rajalakshmi, M. & Nedunchezian, R. (2017). Intelligent traffic video surveillance and accident detection system with dynamic traffic signal control. *Cluster Computing*, 1-13. DOI: 10.1007/s10586-017-0974-5
- [14] Sun, X., Ding, J., Dalla Chiara, G., Cheah, L., & Cheung, N. M. (2017). A generic framework for monitoring local freight traffic movements using computer vision-based techniques. In *Models and Technologies for Intelligent Transportation Systems (MT ITS), 2017 5th IEEE International Conference*, 63-68. DOI: 10.1109/MTITS.2017.8005592
- [15] Censo de población y vivienda México 2010. INEGI (Instituto Nacional de Estadística y Geografía) [www.inegi.org.mx](http://www.inegi.org.mx).
- [16] Rodríguez, H. Exceso de parque vehicular ¿Por qué hay tantos automóviles en la ciudad? *La Crónica Chihuahua*, January 2016. [www.cronicadechihuahua.com/capital/2016/Enero/exceso-de-parque-vehicular/](http://www.cronicadechihuahua.com/capital/2016/Enero/exceso-de-parque-vehicular/)
- [17] Plan de Desarrollo Urbano del Centro de Población Chihuahua, Tercera Actualización. IMPLAN, H. Ayuntamiento de Chihuahua (2009) [www.implanchihuahua.gob.mx/pdu2040/psmus/diagnostic\\_urbano.pdf](http://www.implanchihuahua.gob.mx/pdu2040/psmus/diagnostic_urbano.pdf)
- [18] Quezada, M. and Barrientos, H. Se queda Chihuahua sin transporte público. *El Diario de Chihuahua*, July 2016. [http://diario.mx/Estado/2016-07-02\\_1290468f/se-queda-chihuahua-sin-transporte-publico/](http://diario.mx/Estado/2016-07-02_1290468f/se-queda-chihuahua-sin-transporte-publico/)
- [19] Source: "Chihuahua City" 28°40'12.77" N and 106°04'36.85" W. Google Earth. July 17, 2015. June 10, 2016.
- [20] Laplante, P. A., & Neill, C. J. (2003). A class of Kalman filters for real-time image processing. In *SPIE* (Vol. 5012, pp. 23-29). DOI: 10.1117/12.477499
- [21] Schneider, F., Easterbrook, S. M., Callahan, J. R., & Holzmann, G. J. (1998, April). Validating requirements for fault tolerant systems using model checking. In *Requirements Engineering, 1998. Proceedings. 1998 Third International Conference on* (pp. 4-13). IEEE. DOI: 10.1109/ICRE.1998.667803
- [22] Biemond, J., Rieske, J., & Gerbrands, J. (1983). A fast Kalman filter for images degraded by both blur and noise. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(5), 1248-1256. DOI: 10.1109/TASSP.1983.1164186
- [23] Kanhere, N.K. & Birchfield, S.T. (2008). Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. *IEEE Transactions on Intelligent Transportation System*, 9(1), 148-160. DOI: 10.1109/TITS.2007.911357
- [24] Piovoso, M., & Laplante, P. A. (2003). Kalman filter recipes for real-time image processing. *Real-Time Imaging*, 9(6), 433-439. DOI: 10.1016/j.rti.2003.09.005
- [25] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1), 35-45.

- [26] Bishop, G., & Welch, G. (2001). An introduction to the Kalman filter. Proc of SIGGRAPH, Course, 8(27599-23175), 41.
- [27] Grewal, M. S., & Andrews, A. P. (2001) Kalman filtering: theory and practice using MATLAB, John Wiley, NY. ISBN: 9780471266389, DOI: 10.1002/0471266388
- [28] Welch, G. & Bishop, C. (2006). An introduction to the Kalman filter. UNC-Chapel Hill, TR 95-041.
- [29] Zarchan, P. & Musoff, H. (2000). Fundamentals of Kalman Filtering: A practical approach. American Institute of Aeronautics and Astronautics, Inc. ISBN 978-56347-455-2.
- [30] Kiencke, U., & Nielsen, L. (2000). Automotive control systems: for engine, driveline, and vehicle. Springer-Verlag Berlin Heidelberg, 512 pp., ISBN 978-3-540-26484-2, DOI: 10.1007/b137654
- [31] Szeto, M.W., & Gazis, D.C. (1972). Application of Kalman filtering to the surveillance and control of traffic systems. Transportation Science, 6(4), 419-439, DOI: 10.1287/trsc.6.4.419
- [32] Okutani, I., & Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through Kalman filtering theory. Transportation Research Part B: Methodological, 18(1), 1-11, DOI: 10.1016/0191-2615(84)90002-X
- [33] Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., & Russell, S. (1994, October). Towards robust automatic traffic scene analysis in real-time. In Pattern Recognition, Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on (Vol. 1, pp. 126-131). IEEE. DOI: 10.1109/CDC.1994.411746
- [34] Gazis, D., & Liu, C. (2003). Kalman filtering estimation of traffic counts for two network links in tandem. Transportation Research Part B: Methodological, 37(8), 737-745, DOI: 10.1016/S0191-2615(02)00059-0
- [35] Hsieh, J.W., Yu, S.H., Chen, Y.S., & Hu, W.F. (2006). Automatic traffic surveillance system for vehicle tracking and classification. IEEE Transactions on Intelligent Transportation Systems, 7(2), 175-187, DOI: 10.1109/TITS.2006.874722.