DOI: 10.1111/mice.12430

ORIGINAL ARTICLE

WILEY COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Measuring traffic lane-changing by converting video into space–time still images

Marcel Sala[1] | Francesc Soriguera[1] | Kevin Huillca[2] | Verónica Vilaplana[2]

[1]BIT – Barcelona Innovative Transportation, Civil and Environmental Engineering Department, UPC-BarcelonaTech, Barcelona, Spain

[2]Image Processing Group, Signal Theory and Communications Department, UPC-BarcelonaTech, Barcelona, Spain

Correspondence
Francesc Soriguera, BIT – Barcelona Innovative Transportation, Civil and Environmental Engineering Department, UPC-BarcelonaTech, Barcelona 08034, Spain.
Email: francesc.soriguera@upc.edu

Funding information
Spanish Ministry of Economy and Competitiveness, Grant/Award Number: TRA2016-79019-R/COOP; Ministerio de Economía y Competitividad

Abstract

Empirical data is needed in order to extend our knowledge of traffic behavior. Video recordings are used to enrich typical data from loop detectors. In this context, data extraction from videos becomes a challenging task. Setting automatic video processing systems is costly, complex, and the accuracy achieved is usually not enough to improve traffic flow models. In contrast "visual" data extraction by watching the recordings requires extensive human intervention. A semiautomatic video processing methodology to count lane-changing in freeways is proposed. The method allows counting lane changes faster than with the visual procedure without falling into the complexities and errors of full automation. The method is based on converting the video into a set of space–time still images, from where to visually count. This methodology has been tested at several freeway locations near Barcelona (Spain) with good results. A user-friendly implementation of the method is available on http://bit.ly/2yUi08M.

1 | INTRODUCTION

In recent years, lane-change activity has been identified as one of the main factors to trigger congestion in dense traffic, causing instabilities and the capacity drop (Coifman, Mishalani, Wang, & Krishnamurthy, 2006; Duret, Ahn, & Buisson, 2011; Laval, Cassidy, & Daganzo, 2007). Besides, the absence of lane changes has been associated with a smoothed and improved traffic performance (Cassidy, Jang, & Daganzo, 2010; Menendez & Daganzo, 2007; Qu & Wang, 2015). These findings are grounded on few empirical data. In fact, most of the current research on freeway lane-changing either use the laudable but limited and inaccurate Next Generation Simulation (NGSIM) data set (Coifman & Li, 2017; Federal Highway Administration, 2015), use theoretical lane-changing models (Coifman et al., 2006; Duret et al., 2011), or use data from ad hoc experiments (Sun & Elefteriadou, 2012; Sun & Kondyli, 2010). The main cause of the lack of freeway lane-changing data is that typical freeway sensors, such as loop detectors or license plate recognition devices, are unable to measure them. In such situations, video recordings appear as an alternative, while the image processing required for the data extraction constitutes a challenge.

While full automatization of the data extraction process from video recordings is technically feasible, in many cases it is not cost-effective and turns out to be insufficiently accurate. The limited amount of data to treat in research pilot tests does not justify the implementation of complex automatic image processing systems, which generally involve significant investments in hardware and software, as well as long learning periods. In addition, even the best automated tool requires extensive human intervention for obtaining the data accuracy required in most of the research applications (Coifman & Li, 2017). In this context, researchers usually

avoid automated methods and opt for the "visual" data extraction, which implies watching the recordings and manually annotating the relevant data. This is an extremely tedious and time-consuming task. Just imagine a situation where lane changes need to be counted in six different sections of a three-lane freeway for a total of 63 hr of video recordings. This was precisely the need that gave rise to the development of the fast, simple, and reliable semiautomatic video processing method presented in this paper.

The rest of the paper is structured as follows: Section 2 reviews different methods for video processing in order to extract traffic data; next, in Section 3 a detailed description of the proposed methodology for counting freeway lane changes is presented. Numerical assessment and guidelines on how to correctly use the method are also described. In Section 4, the method is applied to video recordings from two freeways accessing Barcelona (Spain). The reliability of the method is assessed, and a preliminary traffic analysis is presented showing the potential of this kind of data in understanding the real effects of lane-changing. Finally, in Section 5, some conclusions are outlined, highlighting the main contributions of the presented method. The paper is completed with three appendixes. Appendix A presents an analytical derivation of perspective deformation, which needs to be taken into account in the framing of the cameras. Appendix B presents the details behind the proposed Global Quality Index (GQI) for the recordings. Appendix C describes the full automatic video processing technique implemented in order to obtain a comparative scenario.

2 | TRAFFIC VIDEO PROCESSING: A STATE OF THE ART

Automatic or semiautomatic methods for extracting traffic data from video recordings exist. In general, the more automation, the more demanding is the method in terms of the hardware and calibration required. Semiautomatic methods, where some human intervention is needed, are more flexible, can work in a wider range of configurations, and can be more precise than the fully automated ones. A selection of some methods of interest is presented next.

2.1 | Fully automated techniques

To the authors' best knowledge, the most automatic video detection tool used for traffic analysis is the NGSIM software (Federal Highway Administration, 2006, 2015). The NGSIM project (Next Generation SIMulator) included image recognition software able to automatically detect and follow the trajectory of every single vehicle in the study area. The software also estimates the size of the vehicle. The technical requirements to use this tool include the usage of special cameras and the rectification, stabilization, and georeferencing of the video recordings. This makes its usage only suitable for some specific cases where the equipment is already set up or when enough budget is available for treating large amounts of data. As if this type of requirements were not enough to limit the usage of fully automated techniques, precision is another limitation. Even the best automatic methods and tools cannot overcome the difficulties created by occlusions, projection distortions, shadows, vehicles with colors similar to pavement, etc. (Coifman & Li, 2017). Actually, the accuracy of reconstructed trajectories from video imaging techniques and their correction is a hot topic, analyzed for instance in Buisson, Villegas, and Rivoirard (2016), where a general (i.e., not site-specific) procedure is proposed. The NGSIM data are not an exception, and several works have proved that its trajectories database is plagued by errors. Moreover, measurement errors in the vehicles' lateral position are larger and more frequent, and these significantly affect the lane-changing count, as shown in Montanino and Punzo (2013a) and Punzo, Borzacchiello, and Ciuffo (2011). The errors in the NGSIM database were detected and filtered (Montanino & Punzo, 2015), and since the lateral position was difficult to correct, it was replaced by the lane assignment. Although the quality of the NGSIM-corrected database (available at Montanino & Punzo, 2013b) was greatly improved by the filtering procedure, significant errors in the lane assignment remain. This can be seen by realizing that, at some instances, two consecutive vehicles in the same lane change their relative order without changing lanes. Among other issues, this is analyzed in Coifman and Li (2017), where a deeper evaluation of the NGSIM data errors is performed. It is found that some of these errors are produced by motorbikes, as they often travel between the regular lanes, while others are due to vehicles assigned to the slow (i.e., rightmost) lane but actually traveling on the hard shoulder. Coifman and Li (2017) unveil other reasons for noise and errors in the NGSIM database, like the down-sampling of video recordings to 640 × 480 pixels, meaning that distant vehicles are only 12 × 12 pixels, too small for being robustly detected by automatic tracking tools. At the end of the day, they conclude that the lateral position reported in the raw NGSIM database should not be relied on heavily.

Apart from those in the NGSIM project, other approaches for automatic vehicle detection and tracking have also been developed. Background subtraction algorithms can identify moving objects if they are different enough from the still background (Stauffer & Grimson, 1999). The background changes with time, and the algorithm needs to detect and adapt to these variations. Methods differ in the type of background model and in the procedures used to update the model, which are generally based on machine learning techniques. Applications to traffic data range from the complex behavioral analysis, as in Kumar, Ranganath, Weimin, and Sengupta (2005), where recorded pedestrian-vehicle and vehicle-check

post interactions are classified into sets of predefined behavioral scenarios, to the simple use of cameras as automatic vehicle counting devices (Bhaskar & Yong, 2014). Actually, the idea of using cameras as virtual loop detectors is quite popular in the literature (Avery, Wang, & Zhang, 2007; Uke & Thool, 2013), in this last case allowing the selection of the part of the camera coverage that will constitute the detection zone. The monitoring of traffic intersections is another traditional application of cameras and automatic imaging methods, as many loop detectors would be necessary instead. Take as examples Cheung and Kamath (2005), where trajectories are identified at a complex intersection, or Zheng, Wang, Nihan, and Hallenbeck (2006), who use the recordings to detect oversaturation at a traffic signal junction. Other works use movement detection techniques to convert cameras into automatic vehicle classification schemes (Morris & Trivedi, 2006a, 2006b), or into incident detection systems (Mak & Fan, 2006), although other incident detection approaches based only on advanced processing of loop detector data exist (Adeli & Karim, 2005; Ghosh-Dastidar & Adeli, 2003; Karim & Adeli, 2003). It is also noticeable the increasing research attention placed on the application of these techniques to the tracking of cyclists and pedestrians, for whom traditional detectors are unable to provide even the most basic measurements. A real-time implementation to track pedestrians and estimate their speed from video recordings and using a background subtraction approach is presented in Masoud and Papanikolopoulos (2001). If the off-line estimation suffices, the approach in Malinovskiy, Zheng, and Wang (2009) is able to work precisely with grayscale low-resolution videos.

Movement detection techniques, like background subtraction, work reasonably well when vehicles move at fast and constant speeds (i.e., in free-flowing traffic). In contrast, their reliability is compromised when they travel at different and varying speeds, and even more when they stop, as it happens in congested traffic. Edge or contrast detection methods (Canny, 1986) are an alternative, which might work better in these situations, as long as the angle of vision toward the vehicle does not change much through the camera coverage. Shadows and lighting transitions also represent a problem in these types of methods, due to the different appearance of vehicles under different lighting conditions. As examples, Alireza, Karami-Mollaie, and Baleghi (2012) use color distributions of objects for the real-time tracking of vehicles and Shaoqing, Zhengguang, Jun, and Chen (2009) use edge detection to automatically classify vehicles. In general, these methods are able to detect and track vehicles reasonably well in the proximity of the camera, but errors appear in the vehicle tracking at the furthest part of the image.

Occlusions are another important source of error for all automatic imaging techniques for vehicle tracking. Occlusions appear when a vehicle is hidden, for instance behind a truck, a tree, or a panel of traffic information. In dense traffic, with a significant number of occlusions, errors arise in the automated trajectory estimation. In order to alleviate this problem, Coifman, Beymer, McLauchlan, and Malik (1998) propose a feature-based tracking system. Tracking particular features instead of the entire vehicle makes the system more robust to partial occlusion. Another approach is proposed in Nguyen and Smeulders (2004), where the tracking is based in template matching. Templates are temporally smoothed appearance features of vehicles, which are more resistant to partial occlusions and abrupt lighting changes. Other works propose the usage of aerial videos in order to avoid occlusions (Angel, Hickman, Mirchandani, & Chandnani, 2003; Zheng et al., 2013). The zenithal view not only eliminates occlusions, but also reduces perspective distortions of the size and shape of vehicles when they move closer to, or farther from the camera. In spite of the fact that these two properties facilitate vehicle tracking from aerial videos, the image vibration due to the recording from helicopters or drones, is a challenging technical problem of this approach. The stabilization of the recordings needs to be addressed, for instance using the technique proposed in Knoppers, Van Lint, and Hoogendoorn (2012).

In conclusion, to the authors' knowledge, there is no research work that addresses specifically the automatic detection of traffic lane-changing activity from video recordings. And although lane-changing can be seen as a side result of the tracking of vehicles in their lateral movement, the typical precision of trajectory estimation methods from video recordings (i.e., ±5%) (Kumar et al., 2005) is similar to the lane width, especially at the furthest part of the image, meaning that the lane-change counting from tracking methods is not reliable enough.

2.2 | Semiautomatic video processing

Semiautomatic processing includes all the methods that cannot completely eliminate the human visual intervention but can limit it to very specific tasks. This may lead to an efficiency improvement with respect to the completely visual and manual processing, and to a higher accuracy with respect to full automation. The methodology developed by Patire (2010) and Patire and Cassidy (2011) for a specific traffic analysis on the Tomei expressway accessing Tokyo in Japan, falls under this category. This methodology requires using several cameras placed nearby (e.g., 100 m apart) and without in-between on/off ramps. The method first converts the video to standstill images, called epochs. An epoch is the image resulting from one pixel line of the video accumulated through time; this line is called "the scanline" (see Figure 1). This implies that, among all the pixels of the video scene, only those on the scanline are used. In Patire (2010) and Patire and Cassidy (2011), the epochs are constructed from scanlines set perpendicular to

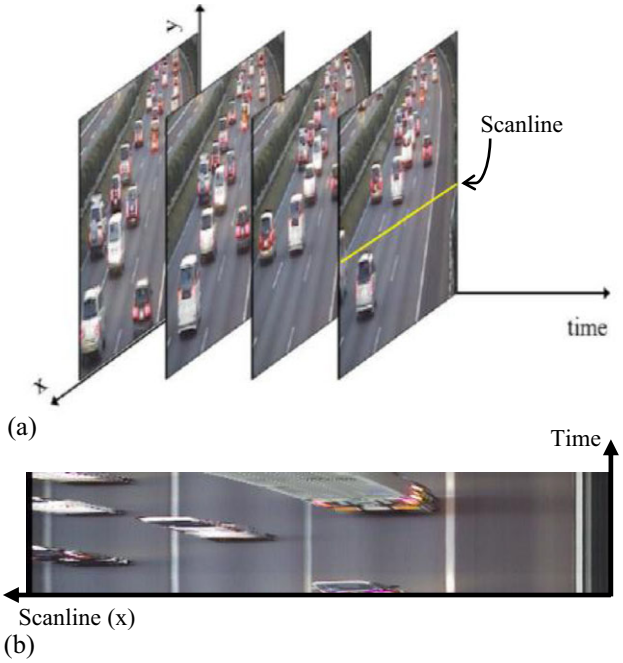


FIGURE 1 From video to epoch. Original methodology as in Patire (2010). (a) Video frames through time. (b) Epoch
Note. Traffic direction follows the positive direction of the "y" axis.

the traffic stream, resembling virtual traffic detectors, where vehicles appear over the pavement colored background, although suffering some perspective deformation. With few manual clicks the software semiautomatically recognizes the vehicles in successive cameras and a rough estimation of vehicles' trajectory is achieved, making possible to count lane changes to the precision given by the density of the cameras used. The methodology proposed in this paper for measuring the lane-changing activity will be based on this idea of the epoch.

2.3 | Visual video processing

This is the raw option, and it is considered as the baseline reference for comparison. It implies watching the entire video length while measuring the traffic variable of interest. In case of lane-changing, authors' experience confirms that they can only be counted reliably by playing the video at a maximum of double speed, and that the entire video needs to be played for every pair of lanes. Some options exist to enhance the completely manual procedure. For instance, counts could be automatically saved when the user "clicks" an (x, y) coordinate of the image, similar to what is done in Campbell (2012) and Campbell and Skabardonis (2013), where authors aim to classify drivers' reactions when facing a stop sign (i.e., stopping, yielding, or skipping the signal). These options eliminate the need of note taking and can speed up the visual video processing, although the entire video still needs to be played. Those who have undergone this visual activity know that this is an incredibly tedious task.

3 | A SEMIAUTOMATIC VIDEO PROCESSING METHOD FOR MEASURING LANE-CHANGING ACTIVITY

A semiautomatic video processing method for measuring freeway lane-changing activity is proposed in this paper. The methodology is based on the idea of the epoch, as described in Patire (2010). Basically, the method reduces the three-dimensional video (x, y, t), where (x, y) are the screen coordinates and (t) is the time, to a two-dimensional subset (s, t) called epoch, where $s_{(x,y)}$ is the parametric curve describing the scanline. In other words, an epoch is a standstill image constructed by accumulating the pixels contained in a scanline for each video frame through time (see Figure 1). However, in the proposed method the scanline is no longer a straight line perpendicular to the traffic stream, as it is in Patire (2010), but a line following the lane division (see Figure 2a). With this new definition of the scanline, any vehicle crossing the line (i.e., a lane change) will appear in the epoch as a "stain" over the pavement background (see Figure 2b,c). Despite being affected by perspective distortion, as the vehicle keeps moving forward during the lane-changing maneuver, the shape of the vehicle can still be identified in the epoch.

Then, it is only needed to identify lane changes from the epoch. This does not only provide the count of lane changes, but also their precise location and time. Three methods are proposed in the paper to identify lane changes from the epoch:

1. *Epoch*: Simply by visual inspection of the epoch.
2. *GUI*: Visually from the epoch with the help of a graphical user interface (GUI).
3. *Automatic*: Using automatic image recognition.

The following sections describe the main difficulties in the application of the method and how to deal with them.

3.1 | Constructing the epoch

The first step in order to construct the epoch (i.e., the space–time still image) from a video recording is to obtain the string of pixels composing the scanline. Recall that the scanline is defined as an $s_{(x,y)}$ parametric curve approximately following the lane division, where (x, y) represent the coordinate axis of the video frame. Considering these axes, the video frame can be seen as a matrix of pixels, whose dimensions are the video resolution, and where each pixel is identified by the

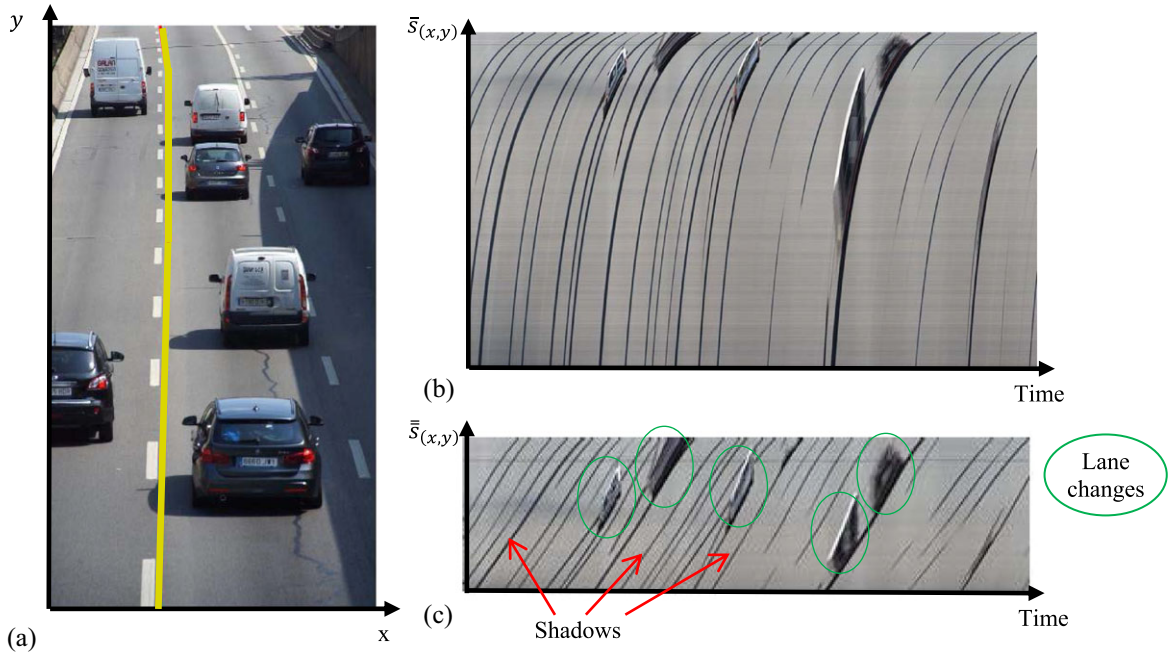


FIGURE 2 New scanline definition and epoch construction. (a) Video screenshot with the new scanline; (b) Epoch; (c) Same epoch as in (b) with perspective correction

Note. $\bar{s}_{(x,y)}$ refers to the arc-length of the parametric curve $s_{(x,y)}$, which defines the scanline; $\bar{\bar{s}}_{(x,y)}$ refers to the same concept but when the perspective correction is applied; see Appendix A for details.

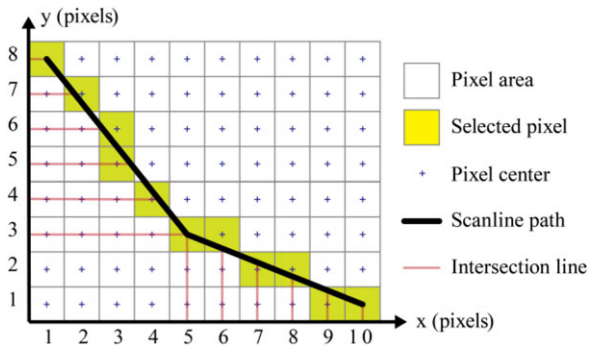


FIGURE 3 Example of the scanline pixel selection method

(x, y) coordinates of its center. Thus, a methodology is necessary to determine those pixels crossed by the scanline.

The method consists of, first, dividing the scanline in pieces according to the dominant dimension, x or y. This is to split the portions of the scanline where $\partial s/dx > \partial s/dy$, from those where this is the other way around. The process is particularly easy if the scanline is defined as a piecewise linear function (as in Figure 3). The coordinates in the dominant direction of the extremes of each portion define the number of pixels assigned to it. Then, it is only needed to determine the orthogonal component of these pixels. This is achieved by computing the intersection of the scanline with the coordinate of the pixel in the dominant direction. This intersection falls within one pixel, which is selected as part of the scanline.

To clarify these concepts, Figure 3 shows a simple example. Imagine a video with a resolution of 10×8 pixels, and a piecewise scanline defined by the following three points: [(1,8); (5,3); (10,1)]. The scanline can be divided in two portions (i.e., the two piecewise segments). In the segment [(1,8); (5,3)], y is dominant, so pixels between $y = 8$ and $y = 3$, including them, are selected (i.e., six pixels). Then, the x coordinate of the pixels is determined by the intersection between the scanline and the y coordinates (i.e., in red in Figure 3). The process can be repeated with the segment [(5,3); (10,1)], where, in this case, x is dominant.

3.2 | Perspective and distortion

This new definition of the scanline implies perspective and distortion problems. Recordings of a traffic stream made from a finite height are affected by perspective deformation. The parts of the road closer to the recording point appear larger than those further apart. Perspective deformation affects the epoch, as long as all the locations on the scanline do not exhibit the same distance to the recording point. The field of view (FOV) is an optical property defined as the angle of vision at which the camera is recording. Large FOVs and low height recording points imply larger perspective deformation and should be avoided. Appendix A provides an analytical derivation of the perspective deformation effects and guidelines in the setup of the camera to minimize them. Also, a methodology to correct the perspective deformation

is described in Appendix A. With such correction, all the pixels in the epoch represent the same fixed real-world distance.

Second, distortion appears due to the limitations in the camera internal optics. The edges of the FOV appear larger than they are in reality due to this optical distortion. A numerical assessment of the distortion effects, assuming simplified pinhole optics, is also presented in Appendix A. Results show that for small FOVs, which are the ones recommended for this application, the deformations due to distortion are small and can be neglected.

3.3 | Occlusions and shadows

An additional problem of the proposed method is that not only lane changes appear as elements differentiated from the pavement in the epochs. Vehicles travelling on the adjacent lanes and their shadows may "occlude" the scanline, even when no lane-changing maneuver is taking place. The occlusion of the scanline also happens when a vehicle starts to change lane and aborts the maneuver for some reason before finishing.

While the first time one sees an epoch it is rather difficult to distinguish lane changes from occlusions, shadows, and aborted maneuvers, after some practice the analyst gets familiar with the method and eventually learns to distinguish them easily in most cases. Occlusions and shadows can be differentiated from lane-changing maneuvers in the epoch because they appear as long shapes covering a significant part of the scanline. The shape and position of the occlusion varies depending on the camera framing and the scanline definition, but always shows the same part of the vehicle and the vehicular shape cannot be identified. In contrast, lane changes appear as clear vehicular shapes, which cross the scanline in a much limited space–time region of the epoch. It is even possible to distinguish the headlights and the windshield after some practice. The dark, long, and thin lines in Figure 2b,c illustrate the shadow effect, and the large patch of white in Figure 4b corresponds to a vehicular occlusion. Figure 4 not only illustrates the problem of occlusions, but also further clarifies the method of obtaining lane changes from space–time still images. Figure 4a shows six frames of a video sequence of 12 s. One lane change takes place between $t = 8$ and $t = 12$ (i.e., a black car). This can be seen in the video frames. The crossing of the scanline is reflected on the epoch (Figure 4b). Also, a large white van starts occluding the scanline at $t = 2$ and until $t = 8$. Again, this can be observed in the epoch and tracked back in the video frames.

Aborted lane-changing maneuvers can also be identified from epochs. Figure 5 shows an epoch where a light truck starts a lane change and immediately after returns to the original lane. This is detected in the epoch by looking at the trajectory of the "edges" of the vehicular shape (lines highlighted in Figure 5). These depend on the direction of the lane change. For instance, a lane change from the shoulder to the middle lane has a different "edge direction" than the opposite maneuver. In the epoch in Figure 5 it can be seen how the two trajectories of the "edges" of the truck shape have very different directions (in red), whereas in an actual lane change the trajectories of the vehicular shapes' edges need to be parallel (in green).

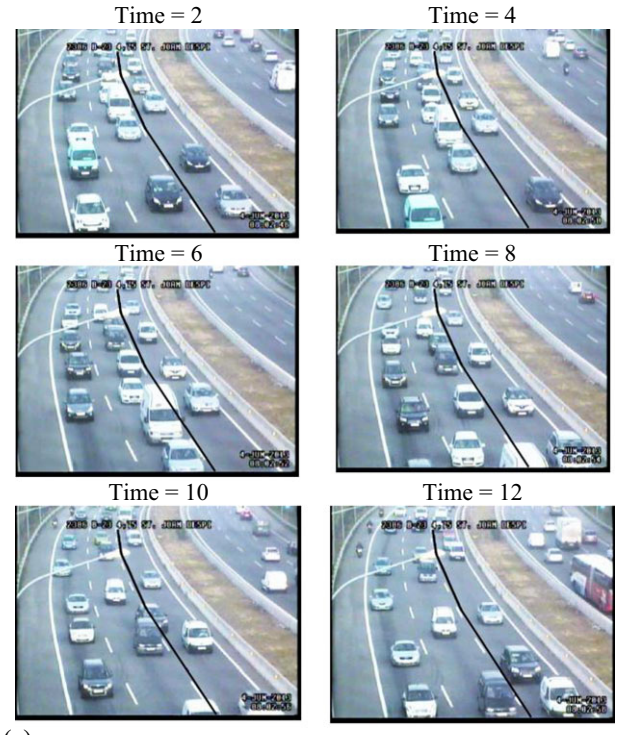


FIGURE 4 Occlusion of the scanline. (a) Sample of video frames every 2 s. The scanline is plotted as a solid black line. (b) The corresponding epoch

Note. $\bar{s}_{(x,y)}$ refers to the arc-length of the parametric curve $s_{(x,y)}$, which defines the scanline; times are in seconds from the start of the epoch.

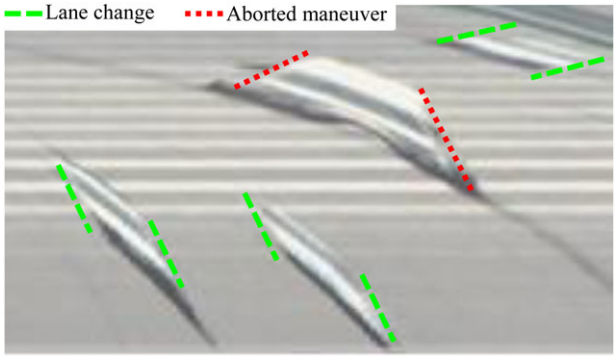


FIGURE 5 Example of an aborted lane-changing maneuver *Note*. The trajectories of the "edges" of the vehicular shapes are marked in green for actual lane changes (i.e., parallel trajectories) and in red for aborted maneuvers (i.e., different directions of the edges' trajectories).

3.4 | Camera settings for high-quality recordings

After analyzing the application of the method to different cameras and conditions (see Section 4), it is concluded that camera settings are critical for achieving a good performance of the method. Cameras must be focused in order to obtain the best possible sharpness. Furthermore, as the width of the scanline is one pixel, the camera resolution directly affects the identification of lane changes on the epoch. The higher is the video resolution, the clearer the epoch. The minimum recommended video resolution is 480×360 pixels, although lower resolutions (e.g., 320×240) can still be used. The frame rate per second (fps) defines the resolution of the epoch in the time axis. A rate of 24 fps or higher is recommended, although it is possible to start counting lane changes from 10 fps. The camera framing is crucial, especially if using low-resolution camera settings. The entire image needs to be focused on the freeway pavement, where lane-changing takes place. Only "pavement" pixels are useful. More open framings, including part of the sky, for example, add nothing, and can be seen as wasting scarce resources (i.e., pixels of the video). The visible freeway length must be much longer than the length taken by the typical lane-changing maneuver. To give an approximate value, at least 50 m of freeway must be captured. Longer lengths are encouraged in order to ease the differentiation of occlusions from lane changes. In addition, in order to minimize occlusions, the scanline should appear in the camera framing as vertical as possible. This means that straight freeway segments aligned with the camera position are preferred. Finally, it is strongly recommended to use the highest possible recording point and a small FOV. Even though that, given a fixed recording height, a larger FOV would cover more road distance, this would imply larger distortion and perspective deformation. So, the recommendation is to try to cover the maximum freeway length from the furthest point possible, in order to keep a small FOV (see Appendix A). Given the previous recommendations, when recording from a fixed pole close to the freeway, a correct video framing usually comprises a stretch starting at least 100 m away from the camera position and ending up to 600 m further away.

Not only technical aspects must be considered for the camera configuration and framing. Environmental factors also have an impact. Clear daylight conditions are necessary. The method should not be applied to night recordings, because the vehicle headlights usually dazzle the camera. A similar effect happens during dawn and dusk, so it is recommended not to point the cameras toward the sun. Bad weather, such as fog or rain, can also lead to low-quality video recordings if the image is so blurry that vehicles cannot be identified.

3.5 | GQI of the recording

Given the previous recommendations, it would be useful to determine, beforehand, the adequacy of a camera framing and video quality for the application of the proposed method.

Automatic methods to detect deficient video quality exist, like the one proposed in Tsai and Huang (2010) based on a machine learning algorithm. However, these generalist methods lack the ability to assess the specific framing requirements of the epoch-based methodology proposed here. Therefore, in order to objectively assess the previous recommendations regarding the image resolution and framing, and to quantify the quality of a recording for this specific application, an ad hoc indicator is defined. The proposed index evaluates six different quality factors (see Table 1); three for the quality of the image (i.e., brightness, clipping, and contrast) and another three for the quality of the framing (i.e., scanline alignment, coverage, and number of unique pixels). The indicator returns a GQI in terms of a percentage, being 100% the maximum possible quality for a recording. The detailed methodology used to compute this indicator is presented in Appendix B.

As an example, Figure 6 shows the framing for seven different cameras, which were used to monitor lane-changing. Among them, the high-definition (HD) camera shows the best image quality and framing (GQI = 88%). Then follow cameras 2305 (52.9%), 2306 (42.7%), 2310 (36.9%), and 2304 (19.2%). Cameras 2309 (0.0%) and 2312 (0.0%) do not fulfill the minimum requirements to apply the methodology with reliability. It is important to stress the fact that, the better the quality of the recording, the shorter the visual intervention in the method. A bad configuration can even invalidate the method, as in Cameras 2309 and 2312. So, each camera configuration should always be tested for a few minutes to assess the resulting epoch.

3.6 | The GUI

To facilitate and enhance the visual identification and counting of lane changes from the epochs a GUI has been

TABLE 1 Quality factors addressed in the proposed Global Quality Index (GQI) of the recording

	Indicator	Factor addressed	Description
Video image	q_1	Brightness	Penalizes extremely dark videos.
	q_2	Clipping highlights	Penalizes overexposure (i.e., when the intensity in a certain area falls over the max. intensity that can be represented).
	q_3	Contrast	Penalizes poor contrast (i.e., poor differences in luminance that makes objects indistinguishable).
Framing	q_4	Scanline alignment	Penalizes scanline deviation from the vertical direction in the video frame.
	q_5	Scanline coverage	Penalizes short scanlines.
	q_6	Scanline resolution	Penalizes scanlines with few pixels.
	GQI	Global Quality Index	Harmonic average of the normalized versions of the previous six partial components.

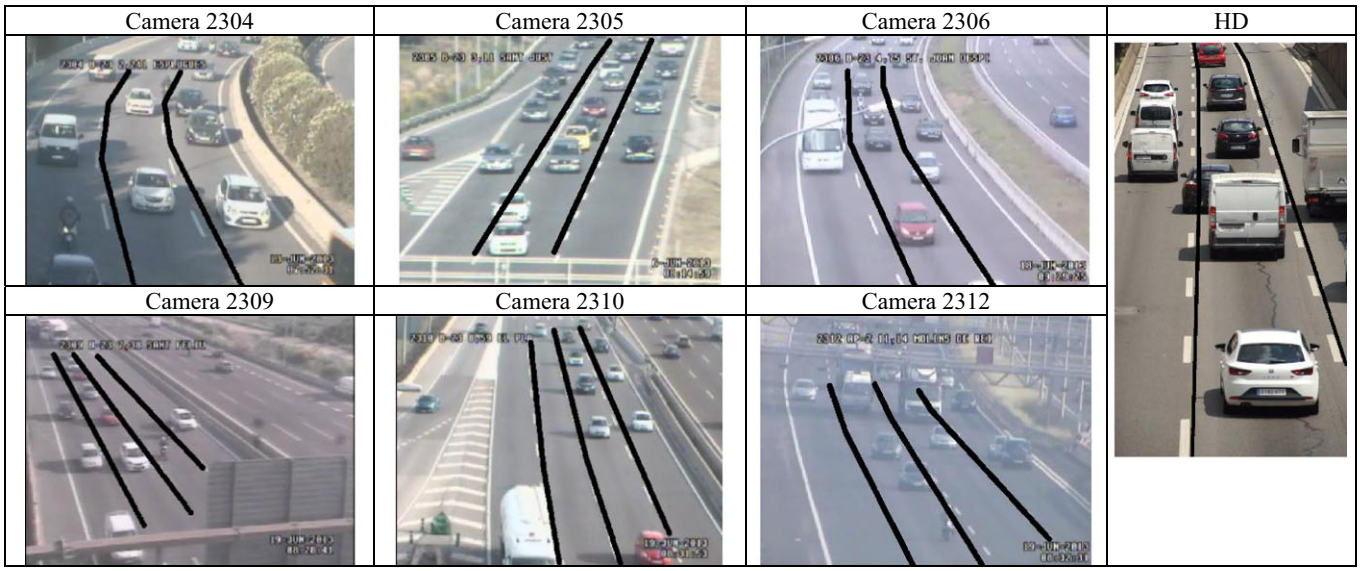


FIGURE 6 Framing and scanlines for seven different camera locations

Note. Camera HD shows excellent framing and quality (GQI = 88.8%); Cameras 2304 (19.2%), 2305 (52.9%), 2306 (42.7%), and 2310 (36.9%) show acceptable framing and quality; Cameras 2309 (0.0%) and 2312 (0.0%) show poor contrast and severe framing problems. They present a low and lateral viewing angle implying a higher rate of occlusions.

coded (see Figure 7; available on http://bit.ly/2yUi08M; the download includes the code to generate the epochs, the GUI, and a manual). With this GUI, the counting of lane changes is done by looking at the epoch (bottom part of the window) and clicking on the lane-change candidate. Then, the video (top of the window) is automatically set to the corresponding frame. If this is still not enough to decide if the maneuver corresponds to a lane change, the GUI allows playing the video around that instant to make the final decision. With the use of the GUI, the method is much more reliable than only using the epoch. The identified lane changes appear in a table at the rightmost part of the window.

4 | APPLICATION TO THE B-23 FREEWAY ACCESSING BARCELONA

The methodology developed in this paper has been applied to count the lane changes in six particular locations on the B-23 freeway accessing Barcelona, in the context of a dynamic speed limit experiment. This is a heavily used freeway with daily recurrent congestion episodes at peak hours. See Soriguera and Sala (2014) for a complete description of the experiment.

Lane-changing activity was to be extracted from 63 hr of standard camera recordings, obtained from different days

FIGURE 7 Graphical user interface

Note. In the epoch lane changes are marked with a blue cross.

during the morning rush, in order to assess the effects of lane-changing in traffic dynamics. In this context, full automation was not cost-effective, even more if one takes into account the reported lack of precision of the automatic imaging techniques. The raw visual procedure was cumbersome, and the presented semiautomatic video processing was selected as an alternative. Nevertheless, full automation based on image preprocessing and machine learning has been implemented in order to test its reliability compared to the proposed method. The details of this implementation can be found in Appendix C. In addition, to assess the full potential of the proposed semiautomatic methodology, 2 hr of HD recording, in the most favorable conditions that could be found, have been included in the analysis.

The framing of the cameras and the scanlines used to create the epochs for each pair of lanes are those shown in Figure 6. These include the six original locations plus the HD recording. The quality of the regular recordings is defined by a resolution of 536 × 400 pixels and a frame rate of 29.97 fps. For the HD recording this is 1080 × 1920 pixels and 50 fps. Note that the HD camera is set on the vertical position to have more pixels on the scanline.

4.1 | Description of the different methods used

Five different methods have been used to count the lane changes. In all of them, a lane change is counted if at least 50% of it happens over the scanline. This means that a lane change can be counted even if the vehicle has started the maneuver before the beginning of the scanline, or finishes it after its end.

The first method, named "ground truth" is a careful count obtained by playing the video once per scanline at the regular speed and pausing or slowing down whenever it is necessary. This count is then refined by searching in the video for all the nonidentified lane changes that have appeared in any of the other methods. This very detailed and accurate counting, unfeasible in the regular practice, is included in order to set a ground truth value.

The second method used, namely "watching the video" is the typical standard visual methodology. It consists in watching the video once for each scanline at double speed. The third method is the "epoch count," obtained simply from the visual inspection of the epoch without any help from the video. The fourth method is the "GUI count" obtained from the epoch with the help of the user interface described in Section 3.6. The last method is the "automatic count" resulting from the fully automatic procedure described in Appendix C. The method automatically detects the lane changes from the epoch, given a previous learning period. The training of the image processing algorithm is done with a sample count from the GUI measurement. This aims to replicate regular practice, where ground truth values are not available.

4.2 | Performance of the different counting methods

In the context of information retrieval and binary classification like lane-changing count, errors arise either when a lane change is missed by the method (i.e., false negative, f_n) or when an identified lane change actually did not happen in reality (i.e., false positive, f_p). In this scenario, performance is assessed in terms of "precision," p, defined as the fraction of relevant identifications (i.e., true positives, t_p) among all the retrieved identifications and "recall," r, the fraction of relevant identifications (t_p) that have been identified over the total amount of existing lane changes. The F_1 score is an integrated performance indicator, computed as the harmonic average of precision and recall. It reaches the best value at 1 (perfect precision and recall) and worst at 0. This is:

$$p = \frac{true\ positives}{total\ identifications} = \frac{t_p}{t_p + f_p}, \tag{1}$$

$$r = \frac{true\ positives}{ground\ truth} = \frac{t_p}{t_p + f_n}, \tag{2}$$

$$F_1 = \frac{2pr}{p+r}. \tag{3}$$

Table 2 shows the obtained results. The first thing to realize is that watching the video (i.e., at double speed, once for each pair of lanes) does not yield a perfect measurement as one could think. The accuracy measured in terms of the F_1 *score* ranges from 0.86 to 0.98 without any clear evidence of the source of such variation. This unexplained variance should not be surprising in this visual intensive task, as the analyst's level of attention very likely fluctuates during the processing time.

In case of using the epoch, the processing time is drastically reduced. Less than 10% of the video duration is needed to count the lane-changing between every pair of adjacent

TABLE 2 Lane-changing counts from different methods and their relative errors

Camera	Time (min)	GT[a] Count	Watch the video[b]			Epoch[c]				GUI[d]				Automatic[e]			Camera
			p	r	F_1[f]	p	r	F_1[f]	Savings[g]	p	r	F_1[f]	Savings[g]	p	r	F_1[f]	GQI[h]
HD	31	133	0.94	0.95	0.95	0.90	0.96	0.93	89%	0.98	0.97	0.97	71%	0.83	0.68	0.75	88.8%
2305f[i]	30	257	0.98	0.85	0.91	0.99	0.79	0.88	87%	1.00	0.98	0.99	56%	0.85	0.75	0.80	52.9%
2305c[i]	30	437	0.99	0.92	0.95	0.98	0.84	0.91	78%	0.98	0.95	0.97	46%	0.85	0.71	0.77	52.9%
2306	60	241	1.00	0.85	0.91	0.89	0.67	0.76	88%	0.98	0.95	0.96	75%	0.70	0.71	0.70	42.7%
2310	30	187	0.85	0.86	0.86	0.95	0.47	0.62	88%	0.98	0.99	0.99	58%	0.87	0.46	0.60	36.9%
2304	60	165	0.89	0.88	0.88	0.78	0.47	0.58	87%	0.99	0.85	0.91	74%	0.78	0.34	0.48	19.2%
2309	15	88	1.00	0.95	0.98	0.78	0.20	0.32	84%	1.00	0.80	0.89	−11%	–	–	–	0.0%
2312	15	91	0.96	0.80	0.87	0.67	0.31	0.42	83%	1.00	0.51	0.67	−33%	–	–	–	0.0%

[a]GT stands for "Ground Truth."
[b]"Watch the video" count by watching the video at double speed.
[c]"Epoch" is visually counting by only looking at the epoch.
[d]"GUI" is using the graphical user interface.
[e]"Automatic" is using the proposed automatic image recognition technique.
[f]See Section 4.2 for the definition of the performance indicators: "precision" p, "recall" r, and "F_1 score."
[g]"Savings" refers to the time savings of human visual intervention with respect to "watch the video" (i.e., at double speed). For "Automatic," savings are 100%.
[h]GQI stands for the "Global Quality Index" of each camera.
[i]For the Camera 2305 two consecutive 30 min intervals were tested. The first one (i.e., 2305f) is in free flowing, while the second (i.e., 2305c) is mostly congested with three strong stop&go waves.

lanes. The main reason for this reduction is that visual time is spent only when some activity happens (i.e., a spot in the epoch), avoiding wasting time and energy in time windows without lane-changing activity. Because the analyst is more active when using this method, he is more likely to hold the attention during the video processing. In spite of this, the accuracy achieved with this method is worse than watching the video, and only of the same order of magnitude (i.e., $F_1 > 0.8$) in cameras with high quality of recordings. This low accuracy is mainly due to the difficulty in differentiating lane changes from small occlusions or shadows, especially at the edges of the epoch. In most of cases this leads to an underestimation of the lane-changing activity (i.e., increase of the false negative rate, implying especially lower recall values).

Using the GUI allows a better interpretation of the epoch, bringing clear benefits in terms of accuracy. In most of the cameras the GUI outperforms the "watching the video" method in terms of accuracy, investing less than half of the visual processing time. It has been found that when two lane changes happen at different locations but almost at the same time, only one is generally detected by watching video.[1] In contrast, using the GUI both can be clearly identified, leading to a significantly better count when a lot of activity happens. In conclusion, the GUI is an accurate and fast method if the recording fulfills a minimum quality (i.e., GQI > 20%). Otherwise, if the quality of the recording is extremely poor, the GUI does not help, either in reducing the visual processing time, or in achieving higher accuracies with respect to watching the video. Cameras 2309 and 2312 are an example of this undesirable situation.

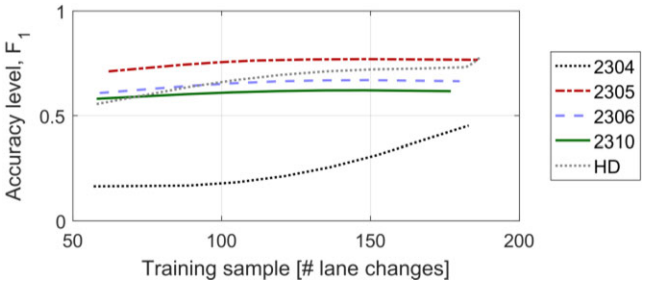


FIGURE 8 Evolution of error in the automatic method for different training sample sizes

Regarding the fully automatic procedure, a lower accuracy is the price one has to pay in order to avoid completely the visual intervention. The implemented machine learning algorithm does not improve the performance of the human analyst in the identification of lane changes from the epoch, and the obtained accuracy is in general worse, with values of $F_1 = 0.80$, at best. In addition, the accuracy of the results rapidly deteriorates with the reduction of the quality of the recording, so that, if the quality of the recording is not good (i.e., GQI > 40%), results cannot be considered reliable (i.e., p or r below 0.5). The effects of the length of the training sample on the accuracy of the method have also been analyzed (see Figure 8). Results show that longer training samples slightly improve accuracy when the default accuracy of the automatic method is very bad, because of an insufficient quality of the recording. In contrast, if the default accuracy is already on its higher levels, extending the training sample does not contribute in further improvements. In conclusion, if the quality

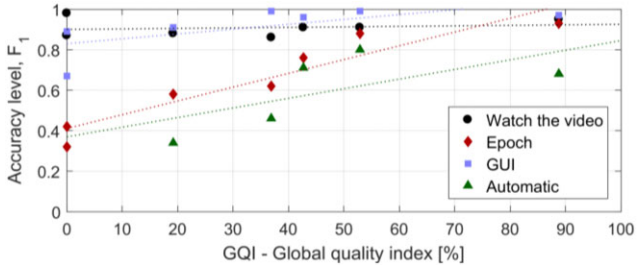


FIGURE 9 Evolution of error with the quality of the recording
Note. Tendency lines are obtained from least squares regression to data.

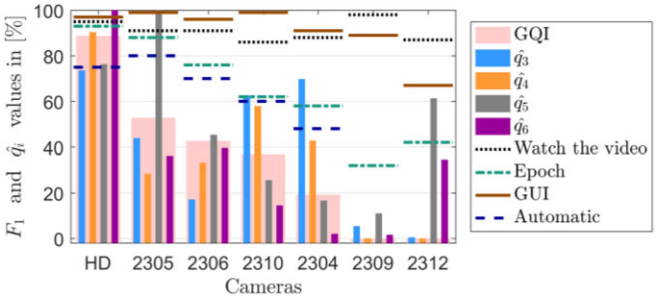


FIGURE 10 Detailed video quality indicators and resulting F_1 accuracy metric
Note. $\hat{q}_1$ and $\hat{q}_2$ are not displayed since they are 100% for all cameras. GQI stands for the Global Quality Index.

of the recording is good, the minimum training sample (e.g., 50 lane changes) suffices.

Results in Table 2 also show that the accuracy of the methods is not significantly affected by different traffic regimes (i.e., free-flowing or congested). This can be seen by comparing the results in Camera 2305f and 2305c.

Given the previous results, it is interesting to compare the different sensitivities of the analyzed methods with respect to the quality of the recordings. On the one hand, watching the video results in accuracy levels around $F_1 \approx 0.9$ on average, and this is almost insensitive to the quality of the recording. On the other hand, the accuracy of all the other methods, based on the epoch, is strongly dependent on the GQI. In these methods, accuracy increases with the quality of the recordings. Figure 9 illustrates these facts, and could be used to support the selection of the video processing technique, given the GQI of the recording and the required accuracy of the results.

Further detail is provided in Figure 10, which shows the effects of each individual factor (i.e., $\hat{q}_i; i = 1 \div 6$; see Table 1 and Appendix B) in the GQI and eventually in the accuracy level, F_1. From Figure 10 it can be concluded that a poor performance in one particular quality indicator cannot be compensated by better results in other indicators. For instance, poor image quality (i.e., brightness – $\hat{q}_1$, clipping – $\hat{q}_2$, or contrast – $\hat{q}_3$) is not compensated by a better image framing (scanline alignment – $\hat{q}_4$, length – $\hat{q}_5$, or resolution – $\hat{q}_6$) or vice versa. So, all individual quality factors need to be

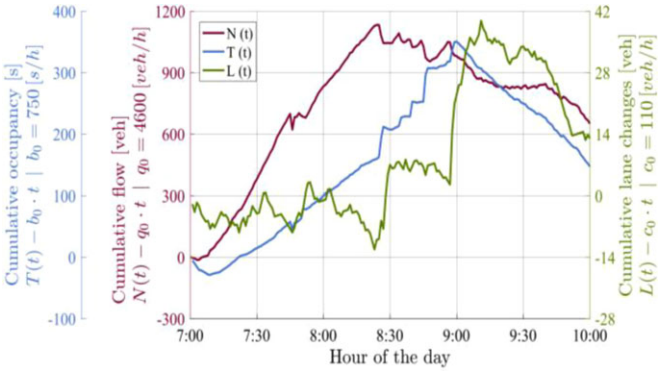


FIGURE 11 Oblique cumulative count (N), occupancy (T), and lane-changing (L) curves
Note. Lane-changing data from Camera 2306; flow and occupancy data are measured by a loop detector on the same freeway segment; the subtracted background flow is 95% of the average. See Cassidy and Windover (1995) for a detailed description of the oblique cumulative count method.

considered when setting up the video cameras for the experiment. These evidences support the multiplicative structure of the GQI (see Appendix B) and lead to the recommendations for the camera settings provided in Section 3.4.

4.3 | Findings from the measured lane-changing data

To illustrate the applicability of the lane-changing data obtained using the proposed semiautomatic methodology (GUI method), the results of a preliminary traffic analysis are presented here. For instance, the data extracted from the video recordings allow empirically proving that the lane-changing rate peaks during traffic state transitions (i.e., from free flowing to congestion, or vice versa).

This can be seen in Figure 11, where oblique cumulative count (N-curve), cumulative occupancy (T-curve), and cumulative lane-changing (L-curve) with respect to time are plotted. Using these curves, the transition between free flow and congestion is identified by an increase of the occupancy (slope of the T-curve) simultaneously with a decrease of the flow (slope of the N-curve). This happens slightly before 8:30 in Figure 11. In turn, the recovery of free-flowing conditions is identified by an occupancy drop (i.e., around 9:00 in Figure 11). It can be seen that in both situations, the lane-changing rate peaks (slope of the L-curve).

Furthermore, the perspective correction method developed in this paper allows locating the lane changes in their real position on the freeway. Figure 12 illustrates the spatial distribution of the lane-changing data, and shows how the solid line, between the central and shoulder lanes, reduces lane-changing rates. Also, just after the end of the solid line, the void left by vehicles exiting at the off-ramp produces an increase of the lane-changing rates in this area.

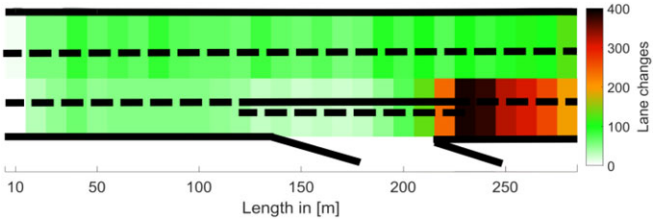


FIGURE 12 Location of lane changes

Note. Data from Camera 2305; traffic goes from left to right; solid line on the pavement bans lane-changing from the middle to the shoulder lane at the off-ramp location.

While lane-changing is a complex drivers' decision process and further research would be needed to model the phenomenon, this simple analysis only aims to show interesting research directions supported by the data obtained with the proposed methodology. In addition, the application of the proposed method could allow other "microscopic" applications. For instance, the perspective correction method could be used to determine the duration in time and space of each lane-change maneuver, by analyzing the shape of the "stain" on the epoch. This would open up new research directions.

5 | CONCLUSIONS

On the one hand, visually extracting traffic data from video recordings requires multiple visualizations. This task is time consuming and extremely tedious. On the other hand, for relatively short applications like the ones in a research context, full automatic methods are overcomplicated, costly, and do not provide the required accuracy. In between semiautomatic methods do have an opportunity. They are simple video processing tools, aimed to ease and speed up the visual process avoiding the complexities and inaccuracies of complete automation.

The proposed semiautomatic method is based on the transformation of the video recordings, obtained from a fixed camera, to an adequate standstill image, called "epoch," from where to visually identify lane changes. The method, which has been implemented on a user-friendly GUI, speeds up the completely visual procedure by concentrating human intervention only on the time windows where some activity happens. For the sake of comparison in the accuracy of the results, the visual identification of lane changes from the epochs has been automated using a machine learning algorithm, achieving a fully automated tool in this case.

The proposed semiautomatic method was to be applied to 63 hr of recordings from seven cameras on a Spanish freeway. Two of these cameras did not meet the minimum recording quality in order to apply the method with reliability. This left "only" 45 hr to process. Using the typical visual procedure, the extraction of the lane-changing data from the videos would have taken 50 hr of labor, as the video needs to be watched once for every pair of lanes at a maximum of double speed (note that sections with three and four lanes were analyzed). In contrast, it took only 7 hr by using the GUI, resulting in 43 hr of labor saved (i.e., 86% of "visual" processing time saved). In addition, this has been achieved with an increased accuracy in the lane-change count and providing the precise time and location of the lane changes, which is not common in visual methods.

Regarding the accuracy of the methods (in terms of the F_1 score), while the completely visual procedure leads to accuracy levels of $0.86 \div 0.98$ for different cameras, by using the GUI semiautomatic method this accuracy is always above 0.95, provided that the recording meets some minimum quality standards (i.e., minimum of 35% in the developed GQI out of 100% for an optimal recording). Also, for recordings with a rather poor quality (e.g., GQI = 20%), the accuracy drops only to 0.90, still above the results achieved with the visual procedure (i.e., 0.88 for this quality of recording). In spite of this, the recommendations regarding the video quality and framing are crucial in order to successfully use the method. The better the video quality, the easier, faster, and more reliable the counting. It should also be noted that an extremely poor quality of the recordings (i.e., GQI = 0%) invalidates the reliable use of the semiautomatic method, leaving the visual procedure as the only alternative.

Regarding the fully automated implementation of the method, two conclusions are obtained. First, the accuracy of the results deteriorates, as expected, with F_1 values around 0.80 in the best cases. Second, the accuracy of the results depends much more strongly on the quality of the recordings, so that, for recordings with GQI < 40%, a reliable count cannot be obtained. Therefore, automatic methods should only be applied when the quality of the recordings is high and low precision data suffices.

Finally, it has also been found that both, the semiautomatic and the fully automatic methods, perform equally well (or equally bad) in free-flowing and in congested traffic regimes.

It remains as further research the application of the method using stable aerial footage, with the benefits of reduced perspective issues and without occlusions. Also, the method could be adapted to be used with recordings taken from a probe vehicle, exploding the benefits of a moving observer in relating the lane-changing behavior with the properties of the traffic flow.

ACKNOWLEDGMENTS

Authors acknowledge the collaboration of the *Servei Català de Trànsit*, the Catalan traffic administration for the provision

of the required video recordings. The work of Adrià Torres in the counting of lane changes and his feedback is also gratefully acknowledged. This research has been partially funded by the Spanish Ministry of Economy and Competitiveness (TRA2016-79019-R/COOP).

ENDNOTE

[1] Three different analysts suffered this problem, and all of them agreed that it is intrinsic to the "watching the video" method.

REFERENCES

Adeli, H., & Karim, A. (2005). *Wavelets in intelligent transportation systems*. New York, NY: John Wiley and Sons.

Alireza, A., Karami-Mollaie, M., & Baleghi Y. (2012). Object tracking using adaptive object color modelling. *Proceeding of 4th Conference on Information and Knowledge Technology, Babol, Iran*, 848–852.

Angel, A., Hickman, M., Mirchandani, P., & Chandnani, D. (2003). Methods of analyzing traffic imagery collected from aerial platforms. *IEEE Transactions on Intelligent Transportation Systems*, *4*(2), 99–107.

Avery, R., Wang, Y., & Zhang, G. (2007). Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. *Transportation Research Record*, *1993*(1), 138–147.

Bhaskar, P. K., & Yong, S. P. (2014). Image processing based vehicle detection and tracking method. *IEEE 2014 International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia*, 1–5.

Buisson, C., Villegas, D., & Rivoirard, P. (2016). Using polar coordinates to filter trajectories data without adding extra physical constraints. *Proceedings of the 95th Annual Meeting of the Transportation Research Board, Washington, DC*.

Campbell, R. (2012). *An analysis framework for evaluation of traffic compliance measures* (PhD thesis). University of California, Berkeley, CA.

Campbell, R., & Skabardonis, A. (2013). Analysis framework for evaluation of traffic compliance measures. *Transportation Research Record*, *2364*, 71–79.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(6), 679–698.

Cassidy, M. J., Jang, K., & Daganzo, C. F. (2010). The smoothing effect of carpool lanes on freeway bottlenecks. *Transportation Research A*, *44*(2), 65–75.

Cassidy, M. J., & Windover, J. (1995). Methodology for assessing dynamics of freeway traffic flow. *Transportation Research Record*, *1484*, 73–79.

Cheung, S. C. S., & Kamath, C. (2005). Robust background subtraction with foreground validation for urban traffic video. *Eurasip Journal on Applied Signal Processing*, *2005*(14), 2330–2340.

Coifman, B., Beymer, D., McLauchlan, P., & Malik, J. (1998). A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C*, *6*(4), 271–288.

Coifman, B., & Li, L. (2017). A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset. *Transportation Research Part B*, *105*, 362–377.

Coifman, B., Mishalani, R., Wang, C., & Krishnamurthy, S. (2006). Impact of lane-change maneuvers on congested freeway segment delays: Pilot study. *Transportation Research Record*, *1965*, 152–159.

Duret, A., Ahn, S., & Buisson, C. (2011). Passing rates to measure relaxation and impact of lane-changing in congestion. *Computer-Aided Civil and Infrastructure Engineering*, *26*(4), 285–297.

Federal Highway Administration. (2006). *Interstate 80 Freeway Dataset Factsheet*. https://www.fhwa.dot.gov/publications/research/operations/06137/

Federal Highway Administration. (2015). *Next Generation Simulation (NGSIM)*. Retrieved from http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

Ghosh-Dastidar, S., & Adeli, H. (2003). Wavelet-clustering -neural network model for freeway incident detection. *Computer-Aided Civil and Infrastructure Engineering*, *18*(5), 325–338.

Karim, A., & Adeli, H. (2003). Fast automatic incident detection on urban and rural freeways using wavelet energy algorithm. *Journal of Transportation Engineering, ASCE*, *129*(1), 57–68.

Knoppers, P., Van Lint, H., & Hoogendoorn, S. (2012). Automatic stabilization of aerial traffic images. *Proceedings of the Transportation Research Board 91st Annual Meeting, Washington, DC*.

Kumar, P., Ranganath, S., Weimin, H., & Sengupta, K. (2005). Framework for real-time behavior interpretation from traffic video. *IEEE Transactions on Intelligent Transportation Systems*, *6*(1), 43–53.

Laval, J., Cassidy, M. J., & Daganzo, C. F. (2007). Impacts of lane changes at merge bottlenecks: A theory and strategies to maximize capacity. In *Traffic and Granular Flow' 05* (pp. 577–589). Berlin, Heidelberg, Germany: Springer.

Mak, C. L., & Fan, H. S. (2006). Single-station algorithm using video-based data for detecting expressway incidents. *Computer-Aided Civil and Infrastructure Engineering*, *21*(2), 120–135.

Malinovskiy, Y., Zheng, J., & Wang, Y. (2009). Model-free video detection and tracking of pedestrians and bicyclists. *Computer-Aided Civil and Infrastructure Engineering*, *24*(3), 157–168.

Masoud, O., & Papanikolopoulos, N. P. (2001). A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Transactions on Vehicular Technology*, *50*(5), 1267–1278.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. *The 13th British Machine Vision Conference University of Cardiff*, 2.5 September 2002, David Marshall & Paul L. Rosin (Editors), 384–396.

Menendez, M., & Daganzo, C. F. (2007). Effects of HOV lanes on freeway bottlenecks. *Transportation Research Part B*, *41*(8), 809–822.

Montanino, M., & Punzo, V. (2013a). Making NGSIM data usable for studies on traffic flow theory: A multistep method for vehicle trajectory reconstruction. *Transportation Research Record*, *2390*, 99–111.

Montanino, M., & Punzo, V. (2013b). *Reconstructed NGSIM data*. Retrieved from www.multitude-project.eu/exchange/101.html

Montanino, M., & Punzo, V. (2015). Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transportation Research Part B*, *80*, 82–106.

Morris, B., & Trivedi, M. (2006a). Robust classification and tracking of vehicles in traffic video streams. *IEEE Intelligent Transportation Systems Conference*, *Toronto, ON, Canada*, 1078–1083.

Morris, B., & Trivedi, M. (2006b). Improved vehicle classification in long traffic video by cooperating tracker and classifier modules. *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, Sydney, Australia*.

Nguyen, H. T., & Smeulders, A. W. (2004). Fast occluded object tracking by a robust appearance filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(8), 1099–1104.

Patire, A. (2010). *Observations of lane changing patterns on an uphill expressway* (PhD thesis). University of California, Berkeley, CA.

Patire, A., & Cassidy, M. J. (2011). Lane changing patterns of bane and benefit: Observations of an uphill expressway. *Transportation Research Part B*, *45*(4), 656–666.

Punzo, V., Borzacchiello, M., & Ciuffo, B. (2011). On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data. *Transportation Research Part C*, *19*(6), 1243–1262.

Qu, X., & Wang, S. (2015). Long-distance-commuter (LDC) lane: A new concept for freeway traffic management. *Computer-Aided Civil and Infrastructure Engineering*, *30*(10), 815–823.

Shaoqing, M., Zhengguang, L., Jun, Z., & Chen, W. (2009). Real-time vehicle classification method for multi-lanes roads. *4th IEEE Conference on Industrial Electronics and Applications*, *Xi'an, China*, 960–964

Soriguera, F., & Sala, M. (2014). Experimenting with dynamic speed limits on freeways. *Procedia Social and Behavioral Sciences*, *160*, 35–44.

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *Fort Collins, CO*, 246–252.

Sun, D. J., & Elefteriadou, L. (2012). Lane-changing behavior on urban streets: An "in-vehicle" field experiment-based study. *Computer-Aided Civil and Infrastructure Engineering*, *27*(7), 525–542.

Sun, D. J., & Kondyli, A. (2010). Modeling vehicle interactions during lane-changing behavior on arterial streets. *Computer-Aided Civil and Infrastructure Engineering*, *25*(8), 557–571.

Tsai, Y., & Huang, Y. (2010). Automatic detection of deficient video log images using a histogram equity index and an adaptive Gaussian mixture model. *Computer-Aided Civil and Infrastructure Engineering*, *25*(7), 479–493.

Uke, N., & Thool, R. (2013). Moving vehicle detection for measuring traffic count using OpenCV. *Journal of Automation and Control Engineering*, *1*(4), 349–352.

Zheng, J., Wang, Y., Nihan, N. L., & Hallenbeck, M. E. (2006). Detecting cycle failures at signalized intersections using video image processing. *Computer-Aided Civil and Infrastructure Engineering*, *21*(6), 425–435.

Zheng, Z., Zhou, G., Wang, Y., Liu, Y., Li, X., Wang, X., & Jiang, L. (2013). A novel vehicle detection method with high resolution highway aerial image. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *6*(6), 2338–2343.

How to cite this article: Sala M, Soriguera F, Huillca K, Vilaplana V. Measuring traffic lane-changing by converting video into space–time still images. *Comput Aided Civ Inf*. 2019;1–18. https://doi.org/10.1111/mice.12430

APPENDIX A

Video recordings suffer from perspective deformation and distortion. Appendix A presents the assessment of the effects of distortion together with an analytical method to correct perspective deformation. Table A1 summarizes the variables and parameters involved. Figures A1 and A2 provide their graphical definition.

A1 EFFECTS OF DISTORTION ON A FLAT SENSOR

Figure A2 illustrates the effects of distortion, considering a camera with a distance r between the flat sensor and the

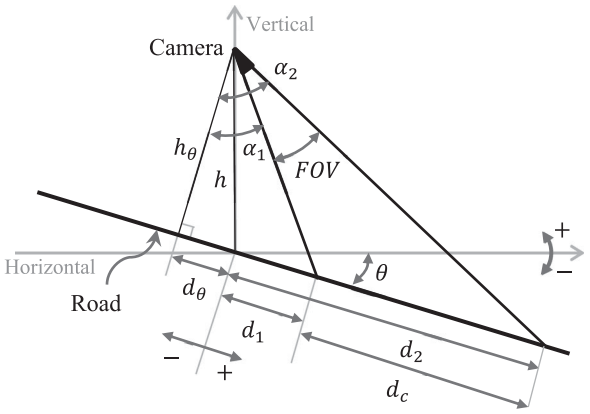


FIGURE A1 Graphical definition of parameters

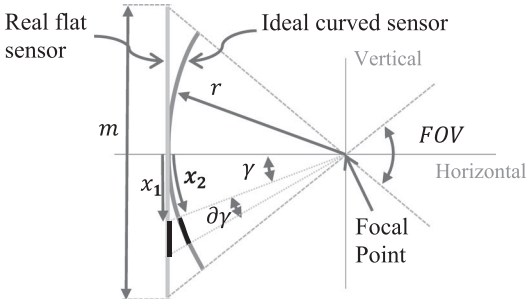


FIGURE A2 Distortion effects on a flat sensor

focal point. In the theoretical ideal curved sensor, the ray of width $\partial\gamma$ encompasses the same number of pixels for all γ values. In contrast, in the real flat sensor, the number of pixels encompassed by $\partial\gamma$ depends on γ, being maximum for $\gamma = \pm FOV/2$ and minimum for $\gamma = 0$. This effect is known as image distortion.

From Figure A2 it can be seen that $x_1 = r \cdot \tan(\gamma)$, while $x_2 = r \cdot \gamma$. Then, the increase in x_1 and x_2 when γ increases by $\partial\gamma$, is expressed by the derivatives $x_1' = \frac{\partial x_1}{\partial\gamma}$ and $x_2' = \frac{\partial x_2}{\partial\gamma}$, as computed in Equations (A1) and (A2). Equation (A2) shows that in the ideal sensor this magnitude is constant and does not depend on γ.

$$x_1' = r \cdot (1 + \tan^2(\gamma)) \tag{A1}$$

$$x_2' = r \tag{A2}$$

Distortion, e_s, is expressed as the relative difference between x_1' and x_2' (i.e., the differential error), which depends on γ, as expressed in Equation (A3). The average error for a given FOV is computed as the integral of Equation (A3) over the whole range of γ:

$$e_s(\gamma) = \frac{x_1' - x_2'}{x_2'} = \tan^2(\gamma). \tag{A3}$$

Figure A3 shows average and maximum distortion as a function of different FOVs. Because small FOVs are recommended for this application, the effects of distortion are small and can be neglected. For the case study presented in Section 4, the maximum FOV is 0.27 rad, resulting in an average and maximum distortion of 0.08% and 1.87% of the pixel size, respectively.

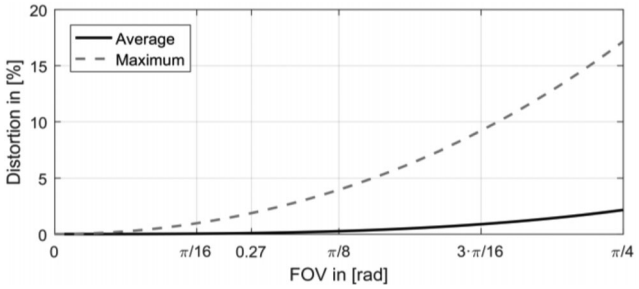


FIGURE A3 Maximum and average distortion values for different FOVs on a flat sensor

Note. The values of the FOV have been limited to $\pi/4$ as wider FOVs have little sense for this application.

A2 THE PERSPECTIVE-CORRECTED SCANLINE

Perspective deformation happens because as α increases (i.e., more distant view), a larger stretch of freeway is seen within a constant $\partial\alpha$. In order to develop a simple correction method

TABLE A1 Definition of variables and parameters in the perspective correction method

Var.		Units	Description
α		(rad)	Angle between the ray of vision and the orthogonal to the road.
d		(m)	Straight distance from the base of the camera to the point where the ray of vision reaches the road.
γ		(rad)	Angle between the bisector of the field of view (FOV) and the ray of vision. $\gamma = \alpha - (\alpha_1 + \alpha_2)/2$
$\partial\gamma$		(rad)	Differential of FOV.
Par.	**Units**	**Value***	**Description**
FOV	(rad)	0.106	Field of view (total angle of vision of the camera framing).
α_1 α_2	(rad)	1.431 1.537	Angle between the lower (α_1) or upper (α_2) ray of vision and the orthogonal to the road; $\alpha_1 \leq \alpha \leq \alpha_2$; $\alpha_2 - \alpha_1 = FOV$.
h	(m)	8	Camera height from the base.
d_R	(m)	179	Length of the scanline on the road.
d_c	(m)	178.2	Length of the scanline projection on the camera FOV plane. $d_c \leq d_R$.
d_1 d_2	(m) (m)	57 235	Straight distance from the base of the camera to nearest (d_1) or furthest (d_2) point where the FOV reaches the road. $d_1 \leq d \leq d_2$; $d_2 - d_1 = d_c$
θ	(rad)	$4.01 \cdot 10^{-2}$	Slope of the road with respect to the horizontal.
$\partial\alpha$	(rad/pixel)	$5.54 \cdot 10^{-5}$	Angular value of one pixel in the original scanline.
v_f	(m/s)	22.22	Traffic free-flow speed.
fps	(fps)	50	Video frames per second.
g	(pixels/frame)	1	Perspective correction factor.
m	(pixels)	1226	Number of vertical pixels in the original epoch.
n	(pixels)	403	Number of vertical pixels in the perspective corrected epoch.

*"Value" stands for the particular application to the HD camera (see Figure 2).

for the perspective deformation, two assumptions are made: (i) no camera distortion (i.e., ideal curved sensor), and (ii) a constant grade of the road, θ. These assumptions approximately hold for the present application, because distortion can be neglected for small FOVs, and road grades are smooth in the relatively small distances of the scanline.

Equation (A4) is derived from the geometry in Figure A1, and allows computing d, the distance from the base of the camera to any point seen with an angle α. In turn, Equation (A5) computes the rate of change of this distance for an incremental change in α:

$$d_{(\alpha)} = h_\theta \cdot \tan(\alpha) - d_\theta, \tag{A4}$$

$$d'_{(\alpha)} = \frac{\partial\left(d_{(\alpha)}\right)}{\partial\alpha} = h_\theta \cdot \left(1 + \tan^2(\alpha)\right), \tag{A5}$$

where $h_\theta = h \cdot \cos(\theta)$ and $d_\theta = h \cdot \sin(\theta)$.

Because distortion effects are neglected, it can be assumed that all pixels in the scanline represent a constant $\partial\alpha$. So, using linear interpolation, it is possible to construct a vector identifying the α_i's for all the pixels in the original scanline, where $i \in 1 \div m$, being m the number of pixels in the original scanline. Note that under these assumptions, $\partial\alpha = (\alpha_2 - \alpha_1)/m$. Then, using Equation (A5), the distance corresponding to the α_i pixel in the original scanline is computed as $d_i = f \cdot d'_i \cdot \partial\alpha$, where f is a correction factor to take into account that the total distance of the scanline projection on the FOV plane is smaller than the real one.

This is:

$$\sum_1^m d'_i \cdot \partial\alpha \approx d_c \leq d_R. \tag{A6}$$

The inequality in Equation (A6) turns to equality only in case of a straight scanline with a parallel camera alignment (FOV). In other cases $d_c < d_R$. So, the correction factor, f, defined in Equation (A7), needs to be applied in order to obtain the real-world distance represented by each pixel:

$$f = \frac{d_R}{d_c}. \tag{A7}$$

Recall that the objective of perspective correction is that each pixel in the corrected scanline covers the same real-world distance. This distance can be expressed as d_R/n, where n is the number of pixels in the corrected scanline. While n can be an arbitrarily selected integer, in order to obtain epochs where lane changes are clearly identified by vehicle-shaped stains over the pavement background, it is recommended for n to be at least the number of frames taken by a vehicle to travel the whole scanline length at free-flow speed. This is:

$$n = g \cdot \frac{d_R}{v_f} \cdot fps, \tag{A8}$$

where $g \geq 1$ is a user-defined perspective correction parameter. The larger g, the higher the epoch, but the more deformed the vehicular shape in the epoch.

Given the real-world distance encompassed by each pixel in the original scanline d_i, which is a function of α_i $\forall i = 1 \div m$, and the constant real-world distance to be covered by pixels in the corrected version of the scanline, d_R/n, the perspective correction algorithm works as follows:

- If $d_i < d_R/n$, the density of pixels per unit length in the original scanline is too high and some pixels need to be skipped in the corrected epoch.
- If $d_i > d_R/n$, the density of pixels per unit length in the original scanline is too low and some pixels need to be replicated. This situation, which happens in the furthest part of the scanline (especially if the FOV is large) is undesirable because it implies lack of information and results in blurry parts of the epoch. For the recommended small FOVs, the worst pixel on the original scanline with the largest d_i might need to be replicated up to four times.

The algorithm in Figure A4 summarizes the method.

APPENDIX B

An ad hoc recording Global Quality Index (GQI) is developed in Appendix B. The index includes six different quality parameters: three for video quality (i.e., brightness, clipping, and contrast) and three devoted to the epoch framing (i.e., scanline alignment, coverage, and number of unique pixels).

Inputs:	• The m pixels of the original scanline and their α_i's $\forall i = 1 \div m$ • d_R/n • $d_i = f \cdot d'_i \cdot \partial\alpha$
Outputs:	• The n pixels of the perspective corrected scanline

```
count = 0
for all i = 1 ÷ m
    count = count + d_i
    while count > 0.5 · (d_R/n)
        add the pixel i to the corrected scanline
        count = count − (d_R/n)
    end
end
```

FIGURE A4 Algorithm for the pixel selection in the perspective corrected scanline

Note. *count* represents the part of the distance of the pixel in the corrected scanline covered at each iteration.

B1 VIDEO IMAGE QUALITY INDEXES

The proposed three quality indexes of the recording are based on its luminance histogram, which represents the frequency of a given luminance value in an image. The normalized cumulative luminance curve, L, is defined as the cumulative sum of the histogram divided by its total value, as expressed in Equation B1:

$$L(i) = \frac{\sum_{j=1}^{j=i} h_j}{\sum_{j=1}^{j=l} h_j}, \tag{B1}$$

where $h(i)$ is the frequency of a given luminance bin i; $i \in [1, l]$, ordered from darker to brighter levels, and l is the number of different luminance values in the image. Note that a typical 8 bit image or video includes a total of 256 possible different values (i.e., $l = 256$). Note that L is a monotonically increasing function and $L(i) \in [0, 1]$.

Table B1 summarizes the formulas to obtain q_1, q_2, and q_3, the three image quality indexes. Brightness is addressed by q_1 and it is computed as the sum of the first half of the cumulative luminance curve. For darker images, q_1 will be higher. q_2 assesses the highlights clipping of the image and looks for overexposed areas. q_2 is smaller when there are larger overexposed areas in the image (i.e., higher percentage of luminance in the brightest 16 bins). Finally, q_3 assesses the contrast of the image by computing the number of bins encompassing 80% of the cumulative luminance. The larger this luminance range, the better the image contrast.

Because q_1, q_2, and q_3 are related to image quality, and a video is a succession of images, the quality indicator needs to be applied to a sample of frames. One frame every minute of video is considered. The quality indicator is then computed as the average of the values obtained in the sample.

TABLE B1 Video quality indexes and their lower and upper bound thresholds: q_{j_l}, q_{j_u}

Indicator	Definition	q_{j_l}	q_{j_u}
q_1	$\sum_{i=1}^{\frac{l}{2}} 1 - L(i)$	$\frac{l}{4}$	$\frac{l}{3}$
q_2	$L(l-16)$	0.8	0.9
q_3	$b - a$ where $a = \text{Min}(a)/L(a) \geq 0,1$ $b = \text{Min}(b)/L(b) \geq 0,9$	$\frac{l}{4}$	$\frac{l}{2}$
q_4	$\beta = -\frac{\sum_1^k (p_i \cdot \beta_i)}{\sum_1^k p_i} \cdot \frac{1}{m_f}$	−30°	−5°
q_5	$\frac{d_R}{v_f}$	2 s	10 s
q_6	*Unique pixels*	25·g	250·g

Note. d_R is the real length of the scanline; v_f is the free-flow speed of traffic; g is the perspective correction factor.

B2 VIDEO FRAMING QUALITY INDEXES

In Table B1, q_4, q_5, and q_6, are the three quality indexes devoted to the quality of the video framing. The alignment of the scanline is addressed by q_4, and it is computed as the average angle, β, between the scanline and the FOV plane (i.e., the vertical in the video frame). The lower β, the better the alignment. If the scanline is defined as piecewise linear, β is easily computed as a weighted average of the angle, β_i, exhibited by each one of the k parts of the scanline, where the weights are their respective length in pixels, p_i. In addition, a correction factor, m_f, is included in q_4. In general, $m_f = 1$, unless the camera is located over the freeway median. In such case, $m_f = 1.25$ and the misalignment is less penalizing. This is because the light vehicles in the median lane are less likely to occlude the scanlines.

The free-flow travel time required to travel the whole scanline length is denoted by q_5. The longer this time, the better the framing, because the relative effect of the boundaries is reduced and more lane changes can be counted.

Finally, q_6 takes into account the number of unique pixels that compose the perspective-corrected scanline. Recall that, in the corrected scanline, in the furthest zones of the epoch some pixels may be repeated. This happens when either the recording has a high FOV and/or a small resolution. Even if the epochs are not corrected this amount of "unique pixels" is a good indicator of the framing quality, taking into account the scanline free-flow travel time, the number of frames per second, the video resolution, and the perspective deformation.

B3 THE GQI

The GQI is obtained as the harmonic average of the normalized versions, $\hat{q}_j$, of the previous six partial components. This is $\hat{q}_j \in (0, 1)$; $j = 1 \div 6$. In order to normalize the partial quality indexes, upper and lower bound thresholds are defined (i.e., $q_{j\_u}; q_{j\_l}$) and linear interpolation is used in between (see Equation (B2)). Finally, the GQI is computed as the weighted harmonic average represented in Equation (B3). Framing quality has a stronger impact than image quality in the global quality of the recording for this specific lane-changing monitoring application:

$$\hat{q}_j = \max\left(0, \min\left(1, \frac{q_j - q_{j\_l}}{q_{j\_u} - q_{j\_l}}\right)\right); j = 1 \div 6, \tag{B2}$$

$$\text{GQI} = \sqrt[4]{\sqrt[3]{\hat{q}_1 \cdot \hat{q}_2 \cdot \hat{q}_3} \cdot \hat{q}_4 \cdot \hat{q}_5 \cdot \hat{q}_6}. \tag{B3}$$

Table B2 shows the results of the application of the video quality indexes to the seven cameras used in the present paper (see Figure 6).

TABLE B2 Video quality indexes applied to the cameras used in the pilot test

Camera	$\hat{q}_3$	$\hat{q}_4$	$\hat{q}_5$	$\hat{q}_6$	*GQI*
HD	73.6%	90.4%	76.3%	100.0%	88.8%
2305	43.9%	28.4%	100.0%	36.2%	52.9%
2306	17.1%	33.3%	45.3%	39.6%	42.7%
2310	62.6%	58.0%	25.6%	14.5%	36.9%
2304	69.8%	42.8%	16.6%	2.1%	19.2%
2309	5.4%	0.0%	11.0%	1.6%	0.0%
2312	0.5%	0.0%	61.3%	34.6%	0.0%

Note. $\hat{q}_1$ and $\hat{q}_2$ are not displayed since they are 100% for all cameras.

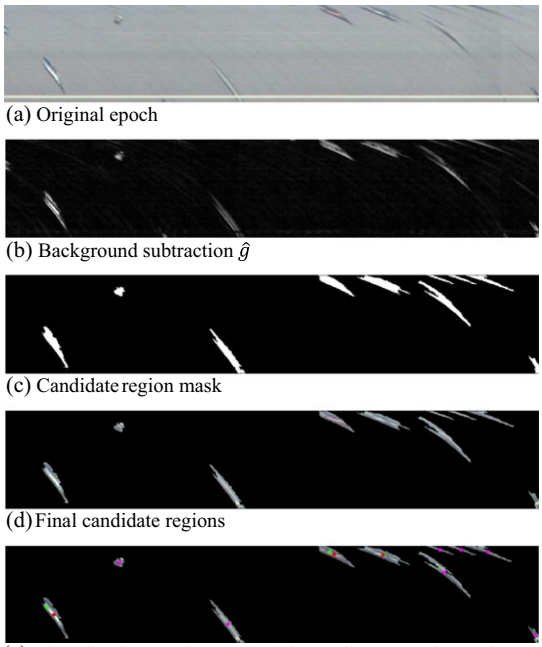
APPENDIX C

Appendix C describes the fully automatic method implemented for detecting lane changes from the epochs. Lane changes appear in the epochs as regions with shapes, colors, and geometric characteristics that are very different from the pavement background. The automatic method analyzes the image to find distinctive areas that differ from the background, characterizes these regions in terms of visual descriptors, and decides if they correspond to a lane change or not using a binary classifier.

C1 IMAGE PREPROCESSING AND CANDIDATE EXTRACTION

The epoch (i.e., a color image; see Figure C1a) is first converted to a grayscale image g of size $h \times w$ (h height, w width) in units of pixels. Next, an estimate of the background is obtained by computing a column vector b of size $h \times 1$, where each element i in b is the median of the pixel values in the corresponding row in g: $b(i) = median_j\,(g(i,j))$. The image background (i.e., the pavement) is eliminated by subtracting the vector of median values from each image column $\hat{g}\,(i,j) = g(i,j) - b(i)$, $\forall i,j$ (see Figure C1b). A morphological closing filter is used to connect small regions corresponding to the same component of the image that may be disconnected after removing the background

The following step is the candidate extraction process, using a segmentation technique called Maximally Stable Extreme Regions (Matas, Chum, Urban, & Pajdla, 2002). The objective is to detect image regions that may correspond to lane changes. This algorithm finds homogeneous components of high contrast with respect to the surrounding area. Finally, post-processing morphological filters are applied to fill holes, remove thin leakages, and filter out small regions. Figure C1c shows the result of this process, where white areas correspond to candidate regions. Figure C1d presents these regions filled with the original image values (pixel-wise product between the epoch C1a and the binary mask C1c).



(a) Original epoch

(b) Background subtraction $\hat{g}$

(c) Candidate region mask

(d) Final candidate regions

(e) Classification result. True positive and true negative regions are depicted with red and magenta dots, respectively, while green dots correspond to ground truth lane changes

FIGURE C1 Automatic epoch processing for lane-change detection. (a) Original epoch, (b) Background subtraction $\hat{g}$, (c) Candidate region mask, (d) Final candidate regions, (e) Classification result.

C2 CLASSIFICATION

After the preprocessing stage, a set of candidate regions is obtained. A binary classifier is then used to discriminate between positive (i.e., lane change) and negative (i.e., occlusion, shade, or other artifacts) regions.

The classification is done with a support vector machine (SVM) model. The classifier hyperparameters (width of the Gaussian kernel and regularization parameter) are selected by cross-validation, minimizing the F_1 measure. The features used by the classifier are: area of the region, region orientation, mean value of each color component in the region (using the YCbCr color space), histogram in each color component, horizontal and vertical projections of the region mask, and lower and upper limits of the region bounding box. The same set of features is used in all the experiments (i.e., for all cameras).

One classifier is trained for each camera. The data used to train each SVM are obtained applying the preprocessing and region extraction procedures to epochs computed from a portion of the video and using annotations from the GUI to label each region as positive or negative. It is important to note that the classifiers are not trained using real ground truth labels but noisy labels provided by the GUI, as would be the case in real practice. This issue might have a negative impact on the accuracy metrics reported for the fully automatic method. Following the example, Figure C1e presents the classification results.