# Learning to segment roads for traffic analysis in urban images

Marcelo Santos, Marcelo Linder, Leizer Schnitman, Urbano Nunes and Luciano Oliveira

***Abstract*— Road segmentation plays an important role in many computer vision applications, either for in-vehicle perception or traffic surveillance. In camera-equipped vehicles, road detection methods are being developed for advanced driver assistance, lane departure, and aerial incident detection, just to cite a few. In traffic surveillance, segmenting road information brings special benefits: to automatically wrap regions of traffic analysis (consequently, speeding up flow analysis in videos), to help with the detection of driving violations (to improve contextual information in videos of traffic), and so forth. Methods and techniques can be used interchangeably for both types of application. Particularly, we are interested in segmenting road regions from the remaining of an image, aiming to support traffic flow analysis tasks. In our proposed method, road segmentation relies on a superpixel detection based on a novel edge density estimation method; in each superpixel, priors are extracted from features of gray-amount, texture homogeneity, traffic motion and horizon line. A feature vector with all those priors feeds a support vector machine classifier, which ultimately takes the superpixel-wise decision of being a road or not. A dataset of challenging scenes was gathered from traffic video surveillance cameras, in our city, to demonstrate the effectiveness of the method.**

## I. Introduction

Road segmentation is one of those tasks which may provide important information for applications of in-vehicle perception [1], [8], [3] or traffic surveillance [9], [4], [5]. In the former, advanced driver assistance, lane departure and aerial incident detection can be cited as examples; in the latter, automatically region segmentation in urban scenes to speed up flow analysis, detection of driving violations and contextual information to improve object detection are also some examples. Methods and techniques for road segmentation can be applied in both fields, although for different purposes.

In traffic analysis from surveillance camera, where we are particularly interested in, the work of Melo et al. [5] exploits vehicle motion trajectory to detect highway lanes. To estimate the trajectory, a merge-and-split algorithm is performed by means of a Kalman filter followed by a random sample consensus (RANSAC) in order to avoid the bias due to vehicle lane changes. Highway lanes are categorized as entry, exit, primary or secondary, after the use of non-metric distance functions and a simple directional indicator. In [8], Shin et al. classify vehicle by means of a road lane detection method. For this latter, a Hough transform is applied over a Sobel edge detector. In fact, to detect vehicles on the road, a 3D object model was fitted over the objects in the image, and the lane detector helped in this process. Another example of lane detection can be found in [7], where Lai et al. extract lane information by using orientation and length features of lane markings and curb structures. Those features are obtained in a 3D space, after a proper camera calibration. A system for vehicle classification that explores lane detection can be found in [4], where Hsieh et al. rely on this information in order to remove shadow from a line-based method. Specifically, to actually segment the road, in [9], Helala et al. detect road boundary by a superpixel extraction with a confidence score assigned for each cluster (superpixel). After ranking each high confident cluster, road boundary is defined. Chung et al [10] proposes a four-step method to segment a road by background subtraction, foreground extraction, background pasting and road localization.

Our aim is to segment road regions from the remaining parts of an image in order to support posterior traffic flow classification. Road segmentation in this context will aid the final system to narrow down the effective image region to be analysed, being interesting that it runs on-the-fly in order to be applicable in online situations. The outline of the proposed system is depicted in Fig. 1. Differently from the aforementioned methods, ours is a learning-driven one, based on a robust superpixel segmentation and extraction of multiple priors, which are synergistically combined road cues. This concept turns the method adaptive even in case of lack of vehicle motion, which is a type of feature very exploited in many methods [4], [5]. Our proposed method starts with a fast background modelling technique based on a median filter. This is intended to subtract the background in order to proceed with the rest of the method. After `N` frames, the resulting background model is segmented by an edge-density-based superpixel detection, where actually each feature will be computed, taking into consideration a gray level amount computation, a texture homogeneity extraction, horizon line detection and motion estimation. Finally, a vector comprised of all those feature feeds a support vector machine (SVM) to make the decision for each superpixel to belong to a road or not. During the training of the classifier, several examples were given considering the presence of all or some of the priors, yielding the proposed system to be robust to a large spectrum of situations.

Still considering video surveillance, road segmentation in urban scenes presents additional challenges in contrast to

This work was partially supported by FCT-Portugal, under grant PTDC/EEA-AUT/113818/2009, and Fundação de Amparo a Pesquisa do Estado da Bahia (FAPESB-Brazil), under grant 6858/2011. Marcelo Santos, Marcelo Linder, Leizer Schnitman and Luciano Oliveira are with Intelligent Vision Research Lab, Federal University of Bahia, Brazil, {marceloms,linder,leizer,lrebouca}@ufba.br, and Urbano Nunes is with Institute of System and Robotics, Coimbra University, Portugal, urbano@isr.uc.pt.

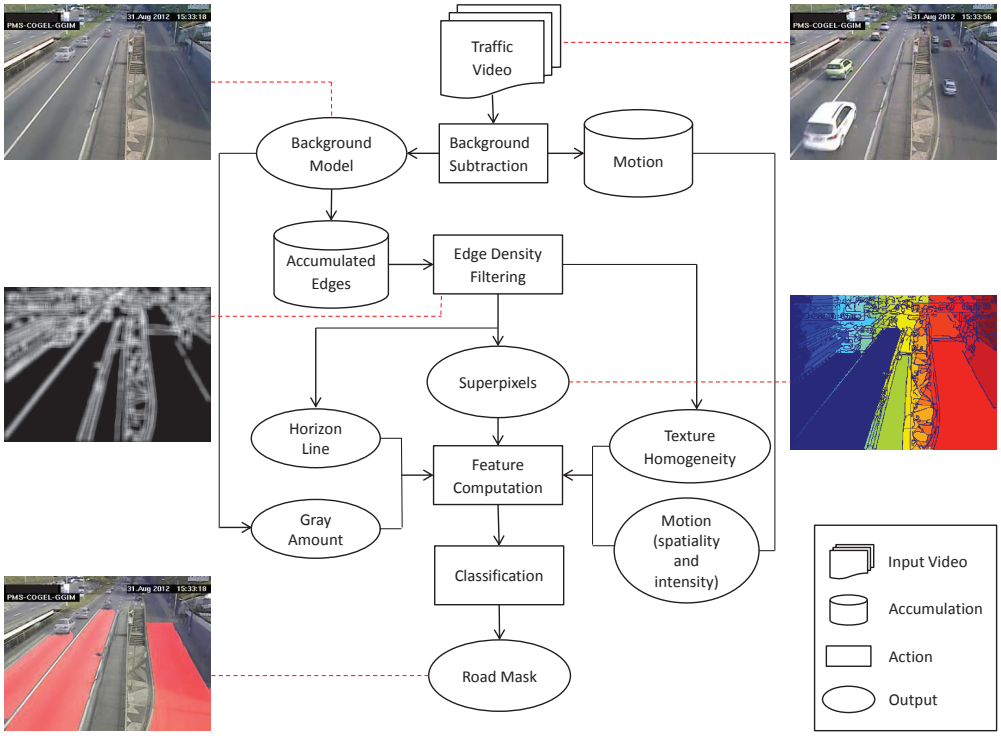


Fig. 1. System outline. The proposed system is comprised of two specific modules: superpixel detection, by a novel method of edge density estimation, and classification of each superpixel, clustering the similar ones. On each superpixel, features of color (gray amount), texture homogeneity (also, edge density analysis) and motion estimation (based on background subtraction) are extracted, composing a feature vector to feed an SVM. On bottom right, the legend indicates the meaning for each frame in the framework: input video, temporal analysis with pixel accumulation in a set of frames, an actual action and the outputs (top-down).

highway scenes: (i) it shows more clutter scenes, with the presence of many vehicles stopped in the roads; (ii) roads do not have only parallel lines which meet in infinite; and (iii) pedestrians in the scenes can turn the problem more complicated since they can also make part of the motion analysis, causing the method to make errors in the boundaries of the road. By extracting a set of cues, which define the object road under many circumstances, we are still able to treat the problem more effectively.

The rest of this paper describes in details the proposed method as well as results over a dataset gathered from a set of video surveillance cameras installed in our city (Salvador).

## II. Edge-Based Background Modelling

In traffic surveillance images, the road is the main part of the background. Therefore road segmentation methods usually start with a model of the background relatively clean (without foreground). From videos, the more common way of yielding that is by means of background subtraction (BS), which aims at separating foreground objects from the background. BS strategies range from simple inter-frames difference to more complex methods, like Mixture of Gaussians (MoG), which tries to model the background pixels by a mixture of normal distributions (please refer to [18] for a survey).

Since the focus of our algorithm is to provide a traffic analyser system with road information, low complexity is desirable to release computational resources for the traffic analyser module. To cope with this requirement, we have used a simple median filtering as a BS technique. The median filtering consists in modelling the background pixels by a median of their intensity along a frame sequence. However, since it is necessary to accumulate the history of the pixel intensities to calculate the median, it is memory bound. In our work, to overcome the memory requirement problem, we use an approximated median value as proposed in [17]:

$$BG_k = \begin{cases} BG_{(k-1)} + 1, & \text{if } FG_k > 0\,, \\ BG_{(k-1)} - 1, & \text{if } FG_k < 0 \end{cases} \quad (1)$$

where BG and FG denote background and foreground, $FG_k = Frame_k - BG_{(k-1)}$, and $k = 1, 2, ...N$ frames.

The method in [17] presents a drawback: if the foreground objects are moving slowly or are stopped in respect to the road, pixels in foreground gradually appears as background. To tackle this problem, we have adopted the following strategy: for each RGB background updated by the median filter, a Canny edge detector is applied and the edges are accumulated along a frame sequence. At the end, after normalizing the edge accumulator by the frame number, we then select the more stable edges in that sequence, i.e., edges that remain in most of the frames. Fig. 2 shows the effect of selecting stable edges for the background. The idea is that background objects generate more stable edges than foreground ones, that way yielding a more representative

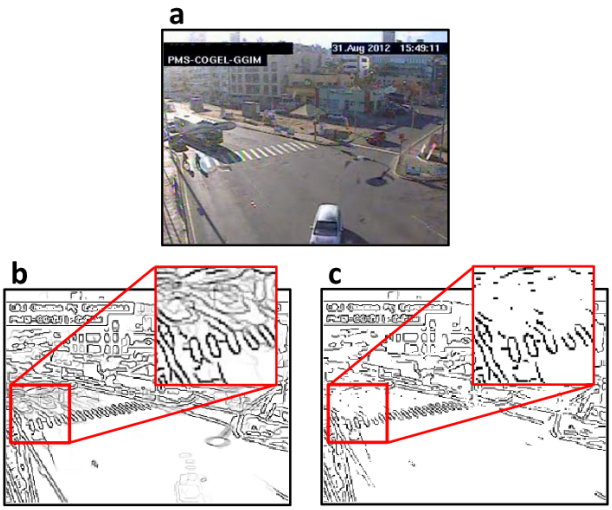


Fig. 2. Background Modelling. In (a), an RGB background update provided by the median filter; (b) shows the edges accumulated along a sequence - the more the edges are considered stable, the more they appear in the final result; and (c) shows the resulting edge-based background model after selecting the more stable edges.

(edge-based) model. This is specially convenient for our algorithm since our superpixel detection method is fed with only an image with edges, as will be shown in the next section.

## III. Superpixel detection by edge density estimation

Superpixel detection is based on image oversegmention. The rationale relies on a way of subdivide the image in a number of regions significantly smaller than than the number of pixels, in order to reduce the complexity of subsequent image processing steps. It is reached by grouping locally similar pixels in more meaningful segments, which can be used in feature extraction and classification, instead of performing these tasks in the pixel itself (refer to [11], [14] for methods in this field).

Methods to detect superpixels rely mostly on pixel-wise color distance [15] or graph-based [16] techniques, turning them computationally expensive to be applied on-the-fly. Furthermore, some of these techniques do not provide satisfactory edge adherence, which is quite critical for road segmentation (an example of a method that deals with edge adherence in a relatively fast way can be found in [11]).

In order to fulfil the aforementioned requirements (fast computation for online applications and high edge adherence), we propose a new superpixel detection. Focusing on the processing time, our method is based on just three usually fast tasks: an edge detection followed by a linear filtering and a morphological operation. For the former, a Canny detector was used as described in the previous section, where from an RGB frame sequence (Fig. 3 (a)) an edge-based background model is produced (Fig. 3 (b)). The motivation to use edges as basis for superpixels is because they essentially represent strong dissimilarities in the image, delimiting most homogeneous regions, which ultimately are good candidates to be superpixels. However, in most of the cases, the direct use of the edges is not practical due to discontinuities that often occur along them, merging different regions erroneously. To address this issue, we perform a spatial linear filtering in the edge image by means of the following filter, so-called edge density

$$ED = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{2}$$

where the edge density ED is a simple arithmetic median calculated in the neighborhood N of the pixels $p_i$.

The idea behind this filter is that although edge discontinuities are unset pixels in a binary image, its local edge density is not null. This way, by applying a suitable threshold over the filter kernel, we can expand the edges in order to reconstruct their broken regions covered by the filter, as can be seen in Fig. 3 (c). In our work we have used a threshold equal to 0.1 in a 9×9 filter kernel with stride of 1 pixel. On the other hand, the expanded edges lose the adherence with the real contour of the objects in the original image. To cope with this situation, a thinning morphological operation is then applied, iteratively refining the borders, however without generating new discontinuities. Since the edges are expanded by N-1, and the thinning operation performs symmetrically in both sides of it, thus (N-1)/2 iterations are necessary. Fig. 3 (d) shows the refined edges after the thinning process.

Finally, to reabsorb some small superpixels that eventually are found by Canny, we reinsert the original edges, through an OR logical operation in those regions filled with clusters of edges which were glued by the edge density filtering. The resulting superpixels of this last step are shown in Fig 3 (e).

## IV. Learning to segment a road

Most of the works, which aims at road segmentation, focuses on one or two road priors (e.g., color, texture or motion). This means that if one of the priors are not presented in the image, road information is lost. On the other hand, computing a lot of features, many times, requires an exceptional effort in terms of computational resources, making it difficult to run on-the-fly. Our method integrates four different types of priors - horizon line, texture homogeneity, gray amount and motions. In order to keep the computational load low, simple-but-efficient strategies to compute those features were developed, as will be described along this section.

### A. Road priors

In Figure 4, the importance of the four priors are depicted as heat mappings, which show the contribution of each prior on the final classification. As we have treated the problem of road segmentation as a learning one, indeed, during the training, we have chosen to feed the model with several different situations where the presence of the priors could happen total or partially. In practice, each prior is computed

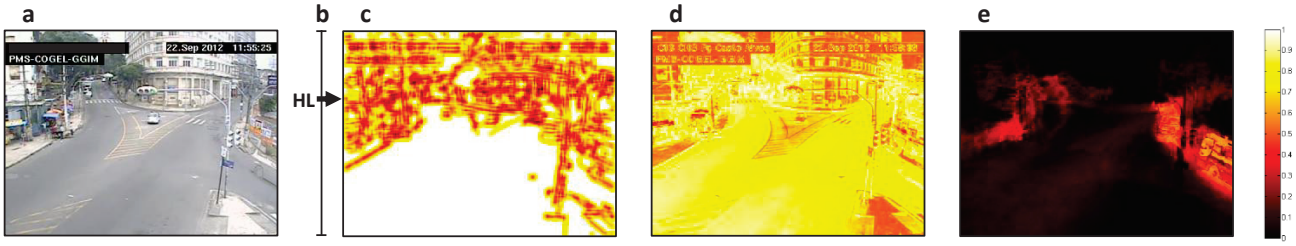


Fig. 4. Heat map illustrating the contributions of each road priors (the closer to one, the hotter, as in the legend on the most right). In (a), the original image; (b) horizon line; (c) texture homogeneity, (d) gray amount; (e) motion.

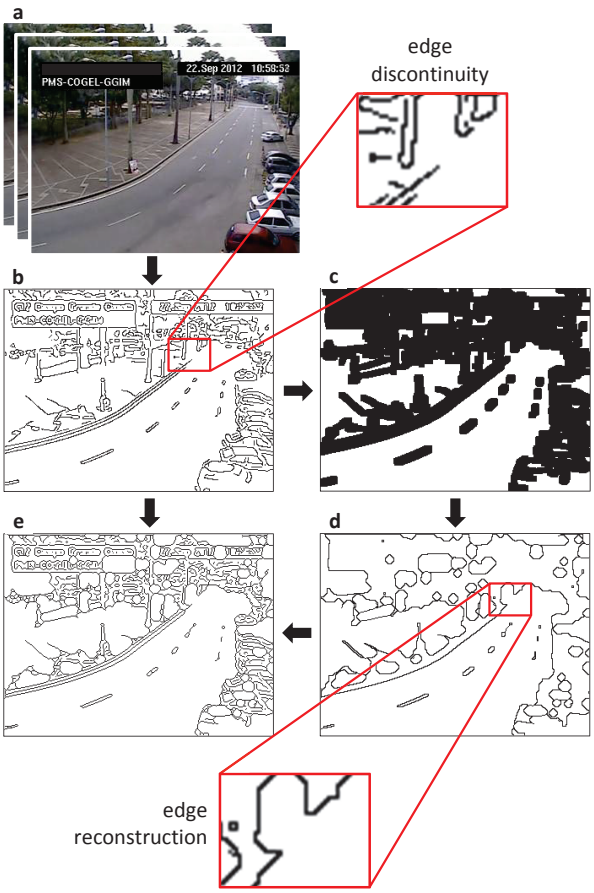


Fig. 3. Superpixel detection. In (a), image provided after background subtraction after $n$ frames; (b) shows the result of the accumulated Canny detector (with an edge discontinuity example) over the subtracted background; (c) shows the result of the spatial filter; (d) illustrates the result of a thinning morphological operation (without edge discontinuities); finally, (e) shows the results of the OR bit-wise operation between Canny detector and the resulting image of the morphological operation.

over a superpixel, which represents the minimum unit to be analysed in our method.

*1) Horizon line estimation:* In [19], the authors evaluate some methods to detect the horizon lines in urban and non-urban scenes. A particular approach is that one based on Gabor filter bank, a technique commonly used to output texture features. From that, the horizon line is estimated as the line with the maximum response to the filters. Despite the authors do not deepen in the hypothesis that motivate this method, it is expectable that, due to perspective effects, the horizon is the most likely place to contain the highest concentration of objects in the image. Consequently, in the horizon line, there also will be the highest concentration of texture, which will maximize the response of the filters.

Our horizon line detection method is inspired by the same idea. However, instead of spending more processing time to perform Gabor, we take the horizon as the line with the maximum edge density (see Fig. 4(b)), which was previously performed in the superpixel detection step. It is straightforward because, once a region has the highest concentration of texture in an image, this also will be more likely to contain edges. After the horizon line is found, we define a prior for each superpixel as a metric of its vertical position related to this line (normalized in the interval [0;1]). In other words, since the road will always be below to the horizon, this prior is proportional to the superpixel part that meets such a condition. Thus, a superpixel fully below has maximum prior value, while a fully above one has prior value equal to zero.

*2) Texture homogeneity:* In order to reach a large view of the roads, traffic surveillance cameras are usually placed in elevated spots, like lamp posts and viaducts. Because the texture is not distance invariant, the road pavement roughness is not caught by the cameras. Thereby the road texture assumes a high homogeneity. When filtered by an edge detector, objects with homogeneous textures yields fewer edges than non-homogeneous ones. This way we can further use the edge density provided by our superpixel method as a texture homogeneity metric. The heat map in Fig. 4 (c) evidences how much this feature can be meaningful for our segmentation purpose, highlighting substantially the road among the other objects. That brings an additional advantage: a simple and fast way of computing a prior from an already computed feature (edge density), used for several different purposes.

*3) Gray amount:* Commonly, the largest part of a road is essentially gray. Further, in terms of digital images, the road color usually presents moderate intensity: it is lower than a cloudy sky during the day, for example, which owns a high gray level due to the sunlight behind the clouds; and, conversely, it is often higher than the gray intensity of

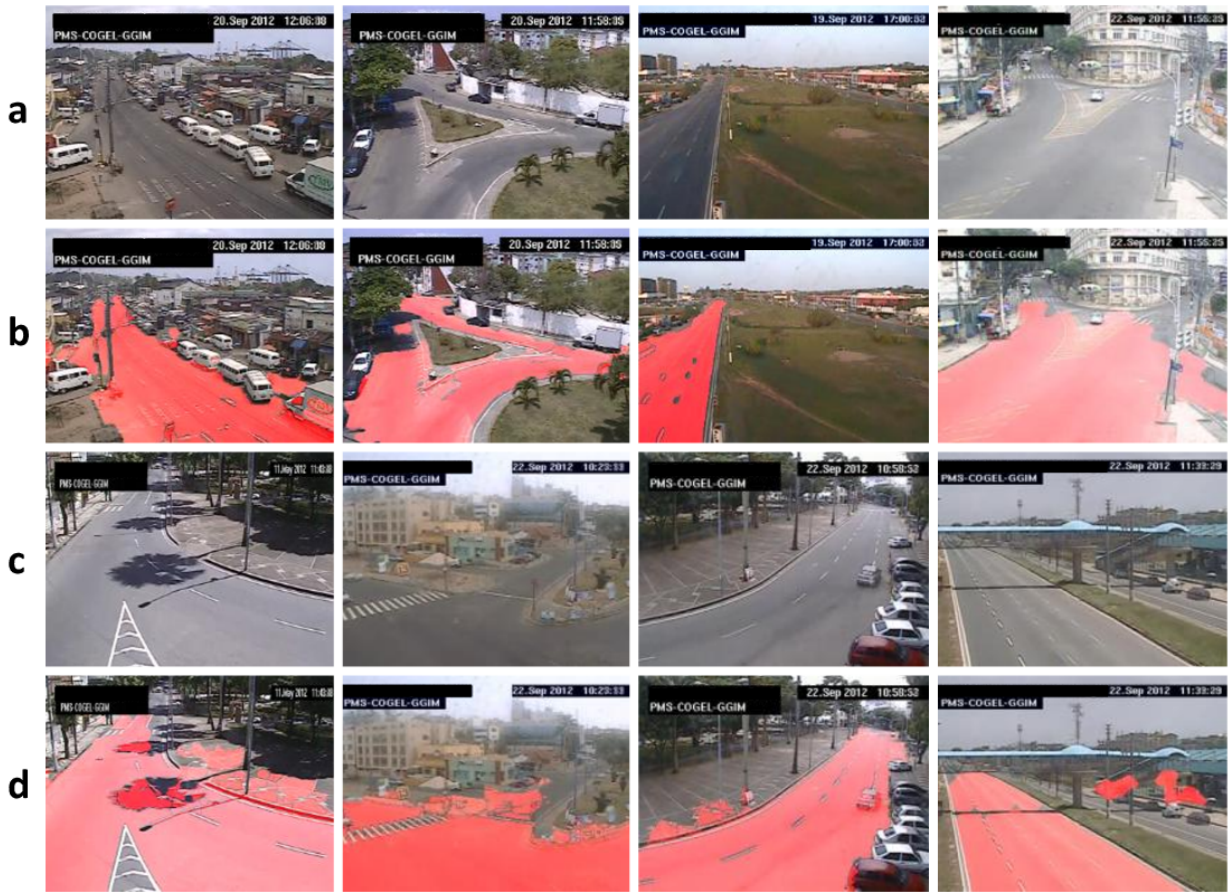


Fig. 5. Some resulting examples. (a) and (c) show original images, while (b) and (d) the correspondent resulting images. The first line shows near perfect results in challenging scenarios; and the last one illustrates some errors of our method (in the shadows and sidewalk).

buildings, where shadows can commonly be found. In order to use, as much as possible, the road color information as a prior, it is convenient to extract this feature from a background model, avoiding noise due to occlusions. To do that while saving time, our algorithm reuses the RGB background model, previously generated by the median filtering (see Sec. 2). So, the gray amount is estimated as follows

$$
\begin{aligned}
G' &= \frac{|r-g| + |r-b| + |g-b|}{3} \\
G'' &= \left| \frac{(r+g+b)}{3} - 0.5 \right| \\
G &= \frac{G' + G''}{2}\,,
\end{aligned}
\tag{3}
$$

where $G'$ denotes the average among the differences of the RGB channels, which lately shows how gray is a pixel, while $G''$ represents how far is the pixel gray level from the diagonal of RGB cube. The average between $G'$ and $G''$ is the so-called gray amount. Fig. 4(d) shows the gray amount in heat mapping; the hotter, more moderate gray level.

*4) Motion:* While cars are passing on the road, their motions are an obvious cue to identify the road in the image. For instance, Melo et al. [5] use a Kalman filter and a kmeans segmentation approach to track vehicles, clustering their trajectories, from which the lane geometry is estimated. In practice, since the usage of such techniques require a reasonable computational effort, there is no gap to compute further features. Thus, this type of method only works in situations where there are cars moving on the road, and one can track them without occlusion. Also, directly estimate the road shape through vehicle trajectories is not a trivial geometric problem. Because of this, road segmentation methods based on vehicle tracking are difficult to apply in challenging scenarios (e.g., non parallel multiple roads, intersections, sharp turns), as is the case of our dataset.

In our work, owing to the utilization of techniques with low computational complexity, the motion estimation can be performed along with the extraction of other (static) features. This makes the algorithm more robust to deal with partial lack of priors. Other benefit is that vehicle tracking and trajectory analysis are not necessary, since road geometry is previously determined by means of superpixels. To estimate the motion, we utilize the foreground resulting from the background subtraction described in Section II. The motion prior is computed by accumulating a foreground

mask along several frames, and finally normalizing it by $N$, i.e. the number of frames in a sliding window. The mask area determines the spatial component of the prior, i.e., it separates the image in two kinds of regions: with or without motion. Additionally, the value accumulated by each pixel in the mask along the frames, i.e., the frequency which each pixel appears in the foreground, reveals the motion intensity. At the end, the motion is generalized for the superpixels by taking the mean of the motion intensities of the pixels within the superpixel. Fig. 4 (e) shows the motion intensities calculated from an instance of our video dataset. The hottest regions correspond to where the more intense traffic occurs.

## V. Experimental analysis

To assess the performance of the method, a set of 17 videos from different cameras and with an average of one minute per video was collected. This video set represents a total of 4,536 frames of different types of urban roads under challenging conditions: side cars, shadows, lighting change, non-structured roads, etc. A total of 6,691 superpixels were detected in those images, where 2,230 frames were used for training, and 4,461 frames for testing an SVM classifier. After superpixel classification, the results were matched with the ground-truth annotations in a pixel-wise manner. Unfortunately, the correlated works found until now do not provide neither their codes nor datasets to allow comparative analysis. Thus, our method was self-compared considering different SVM kernels. The results are summarized in Table I .

According to Table I, accuracy, precision and recall for all kernels are very similar (with exception of precision and recall in Poly-2 kernel). This indicates that the data is very close to a linear separability (since linear kernel presents accuracy very close to the others). By using a computer with a core i5 processor, 4G of RAM and a Matlab implementation, time to classify each frame was approximately 100 ms with no relevant difference among the kernels used. Fig. 5 illustrates some examples of results: the first line with near perfect classification results, and the last one with examples of misclassification in shadow and sidewalk areas.

The results show the effectiveness of our method regarding precision and speed, which are of underlying process in real-life applications.

TABLE I

Performance evaluation with different SVM kernels

| SVM kernel | Accuracy | Precision | Recall |
|---|---|---|---|
| Poly-3 | 0.65 | 0.76 | 0.81 |
| Poly-2 | 0.67 | 0.89 | 0.74 |
| Linear | 0.66 | 0.71 | 0.90 |
| RBF | 0.70 | 0.76 | 0.89 |

## VI. Conclusion

In this paper, we presented a fast and efficient method for road segmentation. The novelty of the method resides in a superpixel-based segmentation and simple-but-effective strategies to allow the extraction of multiple road features, on-the-fly. It makes the method more robust than the state-of-the-art ones to deal with challenging scenarios (e.g., non parallel multiple roads, intersections, sharp turns) under different conditions (e.g., side cars, shadows, lighting change, non-structured roads). Furthermore, a new dataset with 17 videos containing the mentioned scenarios and conditions will be available. As future work, we intend to refine the motion prior, addressing the issue of noise caused by non vehicle objects which are moving in the scene.

## References


[1] T. Kühnl, F. Kummert and J. Fritsch, *Monocular road segmentation using slow feature analysis*, in IEEE Intelligent Vehicles Symposium, pp. 800–806, 2011.

[2] J. Alvarez and A. Lopez and R. Baldrich, *Illuminant invariant model-based road segmentation*, in IEEE Intelligent Vehicles Symposium, pp. 1155–1180, 2008.

[3] L. Zhang and E. Wu, *A road segmentation and road type identification approach based on new-type histogram calculation*, in IEEE International Congress on Image and Signal Process, pp. 1–5, 2009.

[4] J. Hsieh, S. Yu, Y. Chen and W. Hu, *An automatic traffic surveillance system for vehicle tracking and classification*, in IEEE Transactions on Intelligent Transportation Systems, pp. 175–187, 2006.

[5] J. Melo, A. Naftel, A. Bernardino and J. Santos-Victor, *Detection and classification of highway lanes using vehicle motion trajectories*, in IEEE Transactions on Intelligent Transportation System, pp. 188–200, 2006.

[6] V. Kastrinaki, M. Zervakis and K. Kalaitzakis, *A survey of video processing techniques for traffic applications*, in Image and Vision Computing, pp. 359-381, 2003.

[7] A. Lai and N. Yung, *Lane detection by orientation and length discrimination*, in IEEE Transactions on System, Man, Cybernetics - Part B, pp. 539–548, 2000.

[8] W. Shin, D. Song and C. Lee, *Vehicle classification by road lane detection and model fitting using a surveillance camera*, in Journal of Information Processing Systems, vol. 2, pp. 52–57, 2006.

[9] M. Helala, K. Pu and F. Qureshi, *Road boundary detection in challenging scenarios*, in IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2012.

[10] Y. Chung, J. Wang, S. Chang and S. Chen, *Road segmentation with fuzzy and shadowed sets*, in Asian Conference on Computer Vision, pp. 294–299, 2004.

[11] R. Achanta, . Shaji, K. Smith and A. Lucchi, P. Fua and S. Süsstrunk, *SLIC Superpixels compared to state-of-the-art superpixel methods*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 2274 – 2282, 2012.

[12] J. Shi and J. Malik, *Normalized cuts and image segmentation*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 888–905, 1997.

[13] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson and K. Siddiqi, *TurboPixels: fast superpixels using geometric flows*, in IEEE Transctions on Pattern Analys and Machine Intelligence, pp. 2290–2297, 2009.

[14] O. Veksler, Y. Boykov and P. Mehrani, *Superpixels and supervoxels in an energy optimization framework*, in European conference on Computer Vision, pp. 211–224, 2010.

[15] D. Comaniciu and P. Meer, *Mean shift: A robust approach toward feature space analysis*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 603–619, 2002.

[16] P. Felzenszwalb and D. Huttenlocher, *Efficient graph-based image segmentation*, in International Journal of Computer Vision, pp. 167–181, 2004.

[17] N. McFarlane and C. Schofield, *Segmentation and tracking of piglets in images*, in Machine Vision and Applications, pp. 187-193, 1995.

[18] S. Cheung and C. Kamath, *Robust techniques for background subtraction in urban traffic video*, in Visual Communications and Image Processing, pp. 881-892, 2004.

[19] C. Herdtweck and C. Wallraven, *Horizon estimation: perceptual and computational experiments*, in ACM Symposium on Applied Perception in Graphics and Visualization, pp. 49–56, 2010.