Detecting and recognizing centerlines as parabolic sections of the steerable filter response

Petar Palašek, Petra Bosilj, Siniša Šegvić
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb
e-mail: name.surname@fer.hr

***Abstract*—This paper is concerned with detection and recognition of road surface markings in video acquired from the driver's perspective. In particular, we focus on centerlines which separate the two road lanes with opposed traffic directions, since they are often the only markings in many urban and suburban roads. The proposed technique is based on detecting parabolic sections of the thresholded steerable filter response in inverse perspective images. The technique has been experimentally evaluated on production videos acquired from moving service vehicles. The obtained results are provided and discussed.**

I. Introduction

To achieve the desired safety standards, the road administrators are obliged to periodically check the prescribed road conditions in their region. Road safety inspections are usually performed by comparing the current road state with the reference state saved in an information system. Such procedures can detect anomalies such as broken, concealed, time-worn and stolen traffic signs, or erased and badly painted road surface markings.

The prescribed state of the road is conveniently stored in a geographic information system (GIS) based on the concept of georeferenced video. The georeferenced video of the road is acquired with synchronized cameras and position sensors, so that a GPS location is bound to every frame of the video. In this way, the recording of the road state is separated from the actual safety inspection, which allows repeatable and more objective comparisons than in classic scenarios where expert on-site assessment is needed.

Computer vision techniques have been used in solving many problems in traffic, such as automatic detection and characterization of traffic signalization, most often for the sake of safety improvement for all participants in traffic. Many approaches have been proposed for driver assistance that can warn the driver in dangerous situations such as unexpected lane departures. Some of the procedures for road and lane detection and modeling are described in [1], [2] and [3].

In this paper we consider one of the problems that is similar to those described above, detection and recognition of the road centerline. The knowledge of the position and the type of the road centerline could later be used for lane detection, in procedures for recovering road appearance mosaics [4], in driver assistance systems, or for developing completely autonomous vehicles.

This research has been jointly funded by Croatian National Foundation for Science, Higher Education and Technological Development, and Institute of Traffic and Communications, under programme Partnership in Basic research, project number #04/20.

The basic idea of the proposed procedure for detecting, recognizing and tracking of the road centerline is as follows: first, a set of points that lie on the centerline is chosen and approximated with a parabola using the least square method. The parabola model was chosen instead of a line model to get the best possible approximation while keeping the model fairly simple and for easier recognition of the road line type later in the process. Considering that the road centerline can be double (two parallel lines), it is possible that we end up having two approximating parabolas, each one for every detected line. Using the estimated parabola/s, we can easily determine their type by examining the colors of image elements under the parabolas in the binarized image. The estimated parabolas are also used for prediction of future line location, i.e. for tracking of the line.

The paper is structured as follows. First, the preprocessing of the input image is described in section II. Next, we describe the planar perspective transformation in section III. Descriptions of procedures for road centerline detection, recognition and tracking follow in sections IV and V. The experiments and results are discussed in section VI and the conclusion is given in section VII.

II. Preprocessing

An example of the input image acquired from the camera mounted on top of the moving vehicle is shown in Figure 1(a). Before trying to detect the centerline we have to preprocess the input image in such a way that it becomes more convenient for later processing.

A. *Inverse perspective transformation*

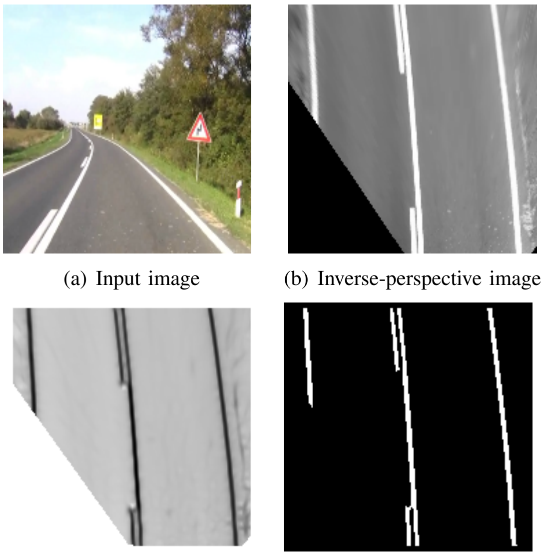
First we apply inverse-perspective transformation to the input image taken from the driver's perspective so that we get an image which would be obtained from a bird's eye view. That is the image we would acquire if we used a camera perpendicular to the road, instead of the camera mounted on top of the vehicle. Inverse-perspective images are useful since in such images the road lines appear as parallel lines with constant width [5]. An example of the inverse perspective image is shown in Figure 1(b).

B. *Steerable filter response*

Next, the response of a steerable filter based on the second derivative of a Gaussian is calculated on the obtained inverse-perspective image. An example of a steerable filer response is shown in Figure 1(c).

C. Binarization

The response of the steerable filter is then binarized using a parameterized threshold, so that the color of all image elements that belong to road surface lines is set to white and the color of all other elements to black. An example of the resulting binarized image is shown in Figure 1(d).



(a) Input image (b) Inverse-perspective image

(c) Steerable filter response (d) Binarized image

Fig. 1: An example of input image preprocessing with all the steps shown.

III. Planar projective transformation

Planar projective transformation, or *homography* can be used as a method for transforming any image of a planar object in a single plane into an image of the same object taken from a different perspective (e.g. the camera in the different position). To apply a homography to points in a plane, we represent them in *homogenous* coordinates, where $\vec{x} = [x_1, x_2, x_3]^T$ is the homogenous representation of the point $[\frac{x_1}{x_3}, \frac{x_2}{x_3}]^T$ from $\mathbb{R}^2$. A planar projective transformation can then be written as:

$$\vec{x'} = \mathbf{H}\vec{x}, \tag{1}$$

where $\vec{x}$ and $\vec{x'}$ represent the point in homogenous notation before and after the transformation, and the matrix $\mathbf{H}$ is a non-singular 3×3 matrix given by:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \tag{2}$$

Points in homogenous notation can also be interpreted as projections of points from $\mathbb{R}^3$ space on to the projective plane $\mathbb{P}^2$ acqured by intersecting the ray that goes from the origin of $\mathbb{R}^3$ through the observed point with the projective plane. All the points on a single ray, represented by vectors $k[x_1, x_2, x_3]^T$ may be thought of as represented with a sigle point in $\mathbb{P}^2$, where the points with the homogenous coordinate $x_3 = 0$ are considered *points at infinity* [6].

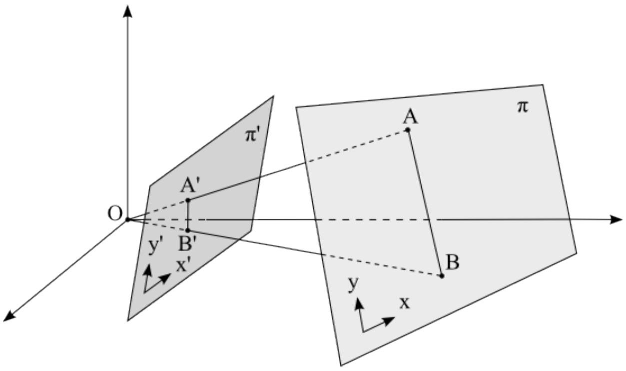


Fig. 2: A visualization of a planar projective transformation.

A homography can now be viewed as a transformation from the representation of points in a certain projective plane to another projective plane, as shown in Figure 2. With this, we can define a homography as an invertible mapping h from $\mathbb{P}^2$ to itself such that three points $\vec{x_1}, \vec{x_2}, \vec{x_3}$ lie on the same line if and only if $h(\vec{x_1}), h(\vec{x_2}), h(\vec{x_3})$ do. A more detailed explanation of projective plannar mappings can be found in [6], but it is clear even from this definition that every such mapping has it's inverse and that we can, given a image taken from a perspective, obtain the original picture by applying the inverse homography.

As the perspecting imaging distorts shapes, parallel lines on scene are not parallel in the image but instead converge to a single finite point called their *vanishing point*, which can be seen on Figure 1(a) where the projected road surface lines are not parallel. To obtain an image suitable for centerline detection, where all the lines are parallel (cf. Figure 1(b)), a rectifying transformation (realized through inverse perspective mapping) must be applied to the input image. This can be easily achieved since the computation of the rectifying transformation does not require knowledge of any of the camera's intrinsic parameters or the pose of the plane [6][1]. A technique for recovering the matrix describing the rectifying transformation is presented in [4].

Various optimization techniques have been applied in order to achieve real time performance of the perspective transformation implementation. Most success has been obtained by exploiting coarse grained parallelism based on Open MP application programming interface [7], and applying strength reduction optimization tehniques at appropriate places. The final optimized version is roughly 4 times faster than the initial, straightforward implementation[2].

IV. Detection of the road centerline

In this section the procedures used for detection of the road centerline are described. We tried to make the procedures as robust as possible, as we wanted the centerline detection to work even in situations where there are shadows present on the road.

[1]The statement holds if radial lens distortion can be disregarded, which is true for most images.

[2]Evaluation was conducted on a system with an i7-860 processor with 4 cores and hyperthreading.

A. Extracting a set of putative centerline points

As described in section II, the input image is the binarized version of the inverse-perspective image acquired from a camera mounted on top of the vehicle. White pixels on this image are those that presumably lie on road surface lines, and all other pixels are black (based on the response of the steerable filter). For further analysis we have to choose a set of points we then try to approximate with a parabola.

As we are only interested in the road centerline, there is no need to analyze the whole image. Instead, we can reduce our observation only to the area called the region of interest (*ROI*). The horizontal offset of the *ROI* is defined by the parameter x_s and the *ROI* width is defined by the parameter d_{ROI}.

In the beginning, the region of interest is bounded by two lines

$$y_1 = x_s - d_{ROI} \tag{3}$$

and

$$y_2 = x_s + d_{ROI}. \tag{4}$$

The region of interest is later changed in a way described in subsection V-B.

A magnified hypothetical region of interest is shown in Figure 3. We analyse transitions from black to white and from white to black color in each n-th row in the region of interest. The coordinates of every point that is determined to be the central point of the white area in current row is added to the set of chosen points. In this example, the set of points was chosen from every 3rd row and the chosen points are marked in green color.

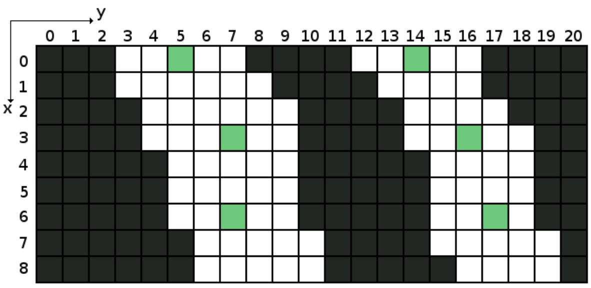


Fig. 3: Part of a hypothetical region of interest and the extracted set of putative centerline points marked green.

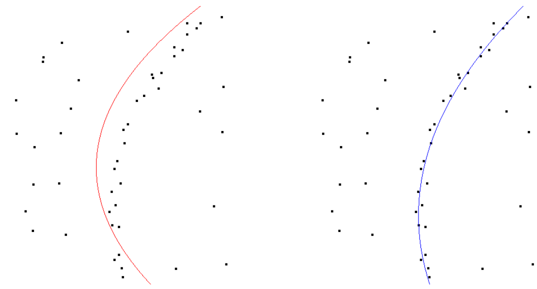
B. Fitting a parabola to the set of extracted points

We want to approximate the chosen set of points with a parabola, and we want the approximating parabola to be as good as possible. We can say that the parabola approximates the set of points well if it passes near enough all of the points from the set. To get the best approximating parabola for a given set, we want to minimize the distances between the parabola and all of the points from the set. We have to keep in mind that the set of chosen points may contain outliers, points that don't belong to the population we are modeling. To get rid of the outliers we use a Random Sample Consensus algorithm [8].

1) Filtering outliers using RANSAC algorithm: To assure that the set of points we are approximating with a parabola consists only of inliers, we use a Random Sample Consensus (RANSAC) algorithm. In general, RANSAC algorithm generates n models using randomly chosen subsets of the given input set, compares them and determines which one is the best model.

In our case, we use the RANSAC algorithm to choose 3 random points from the set in every of n iterations and fit a parabola to them (generate a model) using the least square method (described in the following section). If the current generated model is better than all of the models generated in previous iterations, we memorize the current model as the best and continue with the algorithm.

The criteria used to decide between two parabola models are 1) the number of points with a distance from the parabola model less than d and 2) the average distance of those points from the parabola model. The points with an offset less than d are the inliers of the set. If two models have the same number of inliers, the model with smaller average offset is better.



(a) Least square method (b) RANSAC algorithm

Fig. 4: A comparison of a hand-chosen set of 50 points approximated with a parabola using the least square method and using a RANSAC algorithm.

To calculate the distance between a point $T(x_0, y_0)$ and the parabola $y = a_2x^2 + a_1x + a_0$, we choose a point $P(x, y)$ on the parabola and minimize the length of the line segment $\overline{TP}$. The length of the line segment $\overline{TP}$ depends on the x coordinate of the point P and can be calculated as:

$$\begin{aligned} d(x) =& [a_2^2x^4 + 2a_2a_1x^3 + x^2(1 + a_1^2 - 2a_2q) - \\ & 2x(x_0 + a_1q) + x_0^2 + q^2]^{\frac{1}{2}} \end{aligned} \tag{5}$$

where $q = y_0 - a_0$. To minimize the length of the line segment $\overline{TP}$, we calculate the derivative of $d(x)$ and equalize it with zero. To simplify the calulation, we calculate the derivative of the squared expression:

$$\begin{aligned} \frac{d}{dx}d(x)^2 = & 4a_2^2x^3 + 6 * a_2a_1x^2 + \\ & 2x(1 + b^2 - 2a_2q) - 2(x_0 + a_1q) = 0. \end{aligned} \tag{6}$$

The solutions of equation 6 are potential x coordinates of the point P on the parabola. To find the distance of the point T from the parabola, we calculate the value of $d(x)$ for all of the roots of equation 6 and choose the minimal value. To calculate the roots of a polynomial of the third degree, the Cardano's formula is used.

Finally, after using RANSAC to find the best parabola model and determine the most probable set of inliers, we use the least square method to determine the parameters of the parabola that best approximates the initially chosen set of points.

2) The least square method: By using the least square method we minimize the sum of squared offsets from the points from the set we are approximating to the approximating parabola. To simplify the calculation, we use vertical offsets instead of the perpendicular offsets distance. For a parabola defined by the equation $y = a_2x^2+a_1x+a_0$, the vertical offset from the point $T(x_t, y_t)$ to the parabola is

$$d_v = a_2x_t^2 + a_1x_t + a_0 - y_t. \tag{7}$$

Our goal is to find the parameters a_2, a_1, a_0 that minimize the sum of squared vertical distances between all of the points from the set and the parabola, that is, the vector $\vec{a}$ given by:

$$\vec{a} = \underset{\vec{a}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(a_2x_i^2 + a_1x_i + a_0 - y_i\right)^2. \tag{8}$$

In matrix form we can write this as

$$\vec{a} = \underset{\vec{a}}{\operatorname{argmin}} \left(\begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right), \tag{9}$$

or more briefly

$$\vec{a} = \underset{\vec{a}}{\operatorname{argmin}} \left(\mathbf{X}\vec{a} - \vec{y}\right). \tag{10}$$

We can calculate the vector $\vec{a}$ from the equation

$$\vec{y} = \mathbf{X}\vec{a}, \tag{11}$$

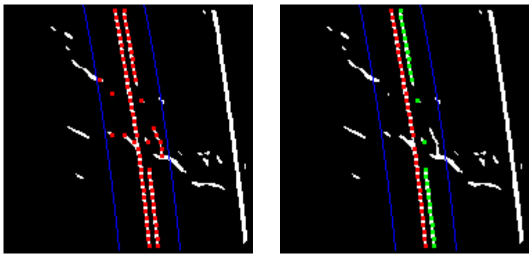
using the generalized inverse of the matrix $\mathbf{X}$:

$$\vec{a} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\vec{y}. \tag{12}$$

3) Detecting the parallel line: As the road centerline can be double (as mentioned before in the introduction), we have to check whether a line parallel to the detected line exists. To do so, we check if the intially chosen set of points contains 5 or more outliers after filtering the set with the RANSAC algorithm. If it does, we try to find a parabola parallel to the detected parabola $y = ax^2+bx+c$, again by using the RANSAC algorithm on the remaining set of points. This time we are only changing the c parameter of the parabola model, because the parameters a and b have to remain the same for the parabolas to be parallel. The rest of the procedure is the same as described in IV-B1 and it results with another set of inliers (as shown in Figure 5) and a parameter c of the parallel parabola, if such a parabola exists.

V. Recognizing and tracking the detected centerline

In this section procedures for recognizing and tracking the detected centerline are described.



(a) Chosen set of points (b) Two subsets of inliers determined by the RANSAC algorithm

Fig. 5: An example of finding two subsets of inliers in a given set of points using the RANSAC algorithm. The subsets are later approximated by parabolas using the least square method.

A. Recognizing the detected centerline

After we have estimated the coefficients of the parabola approximating the road centerline, we use a simple algorithm to determine the type of the line (continuous or dashed).

We will say that the detected line is continuous if it isn't discontinued, i.e. if all the image elements under the approximating parabola in the binarized image are white. We can conclude that the decision whether the line is continuous or dashed can be made by counting the black pixels under the parabola in the binarized image. If the number of black pixels under the parabola is greater than some threshold value then the line is considered dashed, and if the number of black pixels is less than the threshold value, the line is considered continuous.

B. Tracking the detected centerline

As described in subsection IV-A, the set of points chosen for approximating the road centerline with a parabola is chosen from the region of interest. As the lines in the image acquired from the camera change their positions through time, depending on the position of the vehicle on the road, it could happen that the centerline leaves the region of interest. In that case the line wouldn't be detected. To avoid this situation, the region of interest also has to change through time, in such a way that it always contains the road centerline.

Tracking of the detected line is realized by moving the borders of the region of interest depending on the location of the parabola approximating the line. If we define the approximating parabola in the current frame as

$$y = a_2x^2 + a_1x + a_0, \tag{13}$$

the region of interest in the next frame is going to be bordered by two parabolas

$$y_1 = a_2x^2 + a_1x + a_0 - d_{ROI} \tag{14}$$

and

$$y_2 = a_2x^2 + a_1x + a_0 + d_{ROI}, \tag{15}$$

where d_{ROI} represents the width of the region of interest.

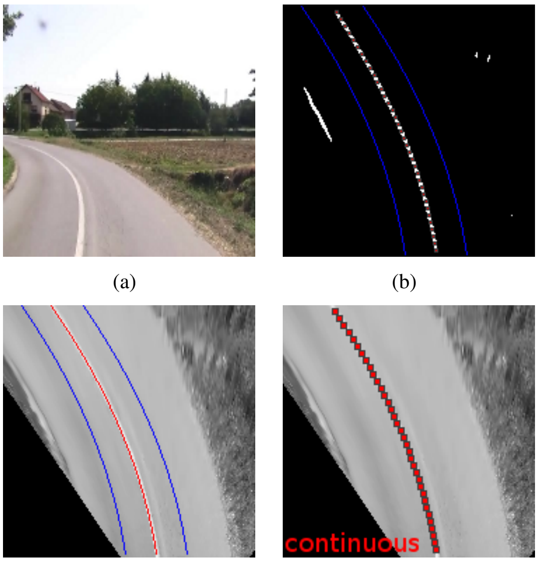
We can see that the lines that border the region of interest in the first frame (as mentioned in IV-A) are in fact parabolas with coefficients $a_2 = 0$, $a_1 = 0$ and $a_0 = x_s \pm d_{ROI}$.

VI. Experiments and results

In this section some results of experiments on real sequences of images are shown and described. Some problems that occured in detecting and tracking the centerline are also shown.

A. Successful detection and recognition of the centerline

In figures 6 and 7 the results of successful detection and recognition of a continuous and a double road centerline are shown. The example shown in Figure 6 justifies the use of a parabola model instead of a line model, and the example shown in Figure 7 demonstrates the robustness of the proposed procedure when there are shadows present on the road. Successful detection of the line resulted in successful line tracking in subsequent frames.



(a) (b) (c) (d)

Fig. 6: An example of a successful detection of a continuous road centerline in a curved road: input image (a), set of points chosen with RANSAC (b), a parabola estimated with the least square method (c), the result (d).

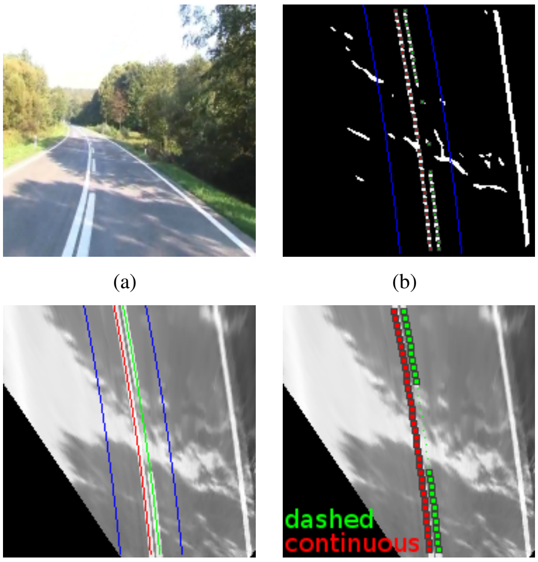
B. Problems with centerline detection

In Figure 8 an example of non-existing line detection is shown (a *false positive* detection). The false positive detection occurred because the steerable filter detected the edge of a passing car as a line on the road, and the centerline is recognized as a double line.

In Figure 9 an example is shown where a road centerline exists, but it isn't detected (a *false negative* detection). The false negative detection occurred because of low visibility of the line on the road and a poorly chosen threshold value for binarization. A solution for this problem is proposed in the conclusion.

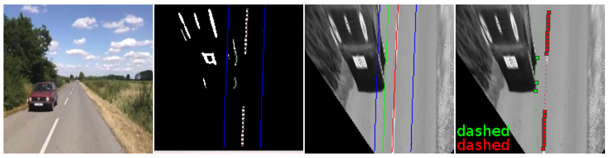
C. Problems with centerline tracking

A problem that occurs in some situations when tracking the detected line is shown in Figure 10. Due to a low number of points detected in the region of interest, the parabola approximated using the chosen set of inliers is tilted to one side, which results in region of interest shifting from the road centerline to the left or right border



(a) (b) (c) (d)

Fig. 7: An example of successful detection and recognition of a double centerline with shadows present on the road.



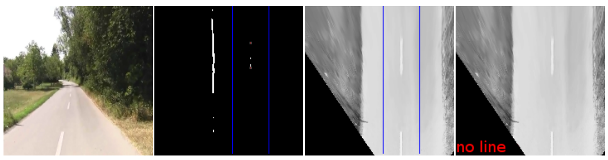
(a) (b) (c) (d)

Fig. 8: An example of a false positive detection of the road centerline due to a passing vehicle.

line of the road. After the region of interest is shifted to the border line, the border line is tracked until the end of the video. A solution for this problem is also proposed in the conclusion.

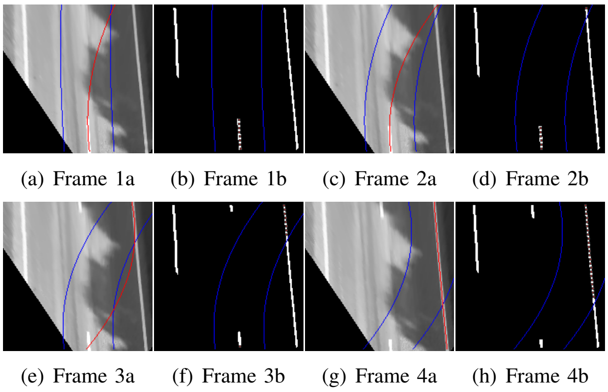
D. Rating the results of the proposed procedure

To rate the proposed procedure, an experiment was performed on a real video of the road acquired from a moving vehicle. Every 50th frame from the video was choosen for processing, resulting in a total of 1500 processed frames. The result of each processed frame was then subjectively labeled as correct if the centerline was correctly detected and recognized, or wrong if there were errors in the result. The experiment resulted in 90% corectness, with 149 frames being erroneously processed. The most of wrongly processed frames contained road surface markings that were out of the scope of this paper, such as hatched markings or zebra crossings. The other



(a) (b) (c) (d)

Fig. 9: An example of a false negative detection of the road centerline.



(a) Frame 1a (b) Frame 1b (c) Frame 2a (d) Frame 2b

(e) Frame 3a (f) Frame 3b (g) Frame 4a (h) Frame 4b

Fig. 10: Shifting of the region of interest.

part of wrongly processed frames were those with falsely recognized centerline types because of a badly chosen binarization threshold or low quality of the centerline marking on the road. Some of the frames were wrongly processed because of a car passing by (as shown in Figure 8).

E. Demonstration video

A short demonstration video of the proposed procedure is available online at: `http://www.zemris.fer.hr/~ssegvic/pubs/centerlinePreliminary101015.avi`

A frame from the demonstration video is shown in Figure 11.

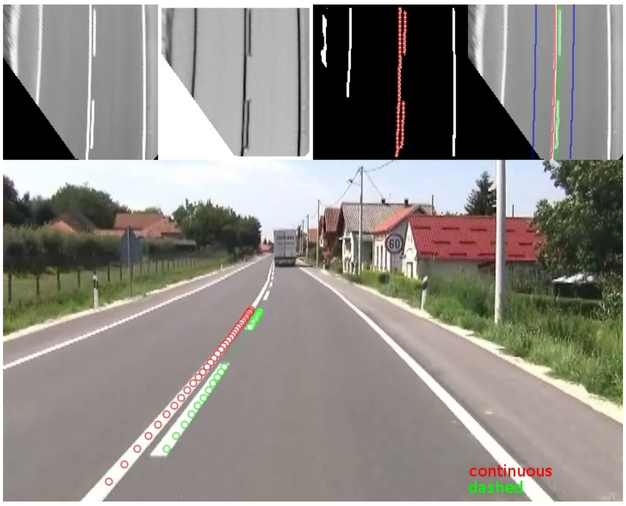


Fig. 11: A frame from the demonstration video of the proposed procedure available online.

VII. Conclusions and future work

The proposed procedure for detecting, recognizing and tracking the road centerline achieves satisfactory results if the conditions on the road are good enough (the road centerline is clearly visible) and the parameters are chosen well.

The binarization threshold may produce a great effect on the results of detection and recognition. As the road surface lines aren't of the same quality on all parts of the road and because the road isn't always under the same illumination, we can conclude that a single threshold value for binarization can't give good results in all of the sections of the road. A possible solution to this problem lies in using adaptive binarization methods, where the binarization threshold value would depend only on the current frame being processed, thus absolving us of the need for a hardcoded threshold. This approach might result in correct detection of the road surface line in situations such as the one shown in Figure 9, where the line wasn't detected due to a poorly chosen binarization threshold. One of the ideas for finding an adaptive binarization threshold is to observe the relation between the number of inliers and and the total number of points in the set depending on the chosen threshold. The binarization would then be performed with a number of different threshold values, and the final binarization threshold would be the one that resulted in the best relation of the number of inliers and total number of points in the set. Another approach would be to use a method such as the Otsu's method for adaptive thresholding.

The second problem that came up during experimenting on real sequences of images is shifting of the region of interest from the road centerline to border lines of the road. The reason why this happens is because the parabola model isn't appropriate for approximation of the road line in some situations, such as the situation shown in Figure 10. To solve this problem, a line model can be used along with the parabola model. After comparing the approximating line and parabola, the model which better approximates the chosen set of points would be used for determining the borders of the region of interest in the following frame. For even better results, the orientation of the steerable filter should be considered when modeling the road surface line.

Adding a parameter for the width of the road centerline could possibly solve the problem shown in Figure 9. Also, by adding a parameter for the width of the road, the procedure described in this paper could be extended for tracking the edges of the road and for detecting the road lanes. The recognition of the centerline could be improved by measuring the visibility and the contrast of the line and it's surroundings.

References

[1] J. C. McCall and M. M. Trivedi, "Robust lane detection and tracking in challenging scenarios," in *IEEE Transactions on Intelligent Transportation Systems, Vol. 9, No. 1*, March 2008, pp. 16–26.

[2] C. R. Jung and C. R. Kelber, "Lane following and lane departure using a linear-parabolic model," *Image and Vision Computing*, vol. 23, no. 13, pp. 1192 – 1202, 2005.

[3] Z. Kim, "Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation," in *IEEE Transactions on Intelligent Transportation Systems, Vol. 7, No. 1*, January 2006, pp. 20–37.

[4] I. Sikirić, K. Brkić, and S. Šegvić, "Recovering a comprehensive road appearance mosaic from video," in *MIPRO 2010*, Opatija, Croatia, May 2010.

[5] S. Šegvić, K. Brkić, Z. Kalafatić, V. Stanistavljević, D. Budimir, and I. Dadić, "Towards automatic assessment and mapping of traffic infrastructure by adding vision capabilities to a geoinformation inventory," in *Proceedings of MIPRO*, Opatija, Croatia, 2009.

[6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision, 2nd Edition*. Cambridge University Press, 2004.

[7] M. Rasmussen, M. Stuart, and S. Karlsson, "Parallelism and scalability in an image processing application," *International Journal of Parallel Programming*, vol. 37, pp. 306–323, 2009.

[8] C. Sunglok, K. Taemin, and Y. Wonpil, "Performance evaluation of RANSAC family," in *20th British Machine Vision Conference*, 2009.