Ground Truth and Performance Evaluation of Lane Border Detection

Ali AlSarraf, Bok-Suk Shin, Zezhong Xu, and Reinhard Klette

Computer Science Department, Tamaki Innovation Campus,
The University of Auckland, New Zealand

Abstract. Lane-border detection is one of the best developed modules in vision-based driver assistance systems today. However, still there is a need to improve for challenging road and traffic situations, and also a need to design tools for quantitative performance evaluation.
This paper discusses and refines a method to generate ground truth for lane markings from recorded video, applies two lane-detection methods to such video data, and illustrates then the proposed performance evaluation by comparing calculated ground truth with detected lane positions. This requires performance measures, and they are also proposed in this paper.

1 Introduction

Vision-based driver assistance systems became already standard modules in modern cars, supported by the availability of high-computing power using low voltage only (e.g. purpose-designed FPGA solutions), small and accurate cameras, able to fit in any vehicle, and progress in the methodology of computer-vision solutions.

Lane border detection, being a component of vision-based driver assistance solutions, has been studied for more than twenty years, and there are robust solutions available for road environments where lane markings are clearly visible, such as highways or multi-lane main roads.

Vision-based lane detection supports lane departure warning, lane keeping, lane centring, and so on [1]. Despite the many algorithms and approaches available, an ongoing concern [1, 9] is the lack of a proper ground truth estimation to evaluate the efficiencies and accuracies.

Usually the easiest way, most publications use to validate how well their approach works, is using their naked eye for the validation. Building a ground-truth data base can become really difficult as roads are not generic in any single country alone, not to mention the diversity around the world. Roads can be well built or not, have proper markings or no marks at all, are urban or countryside roads, have solid lane markers or painted dashes lane borders, and so forth. The environment is only one factor, the other factor is the equipment, such as the type of camera used (e.g. image resolution, gray-level or color, bits per pixel, or geometric accuracy).

It appears that generating ground truth for lane detection needs to reflect a lot of parameters, to make sure to create a trust-worthy ground truth. Digitally simulated ground truth was created by Revilloud et al.[10] but they found out that when adapting their lane detection algorithm running to their synthetic ground truth, it did not work very well on real data as their approach mainly focuses on the ground texture for detecting lines. They stated a need for another solution for ground truth generation.

There is another obvious solution: generate ground-truth manually, supported by some graphics routines for drawing lines. However, considering frame rates of 25 Hz at least, and the need to generate ground truth for long video sequences, this would just to be too much a tedious task.

Borkar et al. [2] developed a technique using time slices and splines to generate ground truth from any type of road image sequence recorded in an ego-vehicle (i.e. the vehicle the vision system is operating in). The approach works reasonably well on clearly marked roads, but the involved interaction also comes with the risk of human error and limited usability. In this paper we provide an improvement that makes the ground-truth generation process easier to use, and which also helps to generate ground truth for larger diversity of recorded video data.

Figure 1 already illustrates the subject of this paper. Ground truth and estimated lane borders are both shown in the image on the left, with a magnified window on the right. It is necessary to compare quantitatively ground truth with estimated lane borders.

We found that it was quite easy to re-implement the method proposed by Borkar et al. [2]. Potentially it works as long as there are some markings in the frames which identify the lanes, When selecting points on lane markings and the points are not at the center of the drawn lines, then this may lead to errors. We provide solutions by using standard image processing techniques to make the

Fig. 1. *Left*: Ground truth and calculated lane markings. *Right*: Magnified window of the image shown on the left. The white lines are calculated ground truth, and the yellow dots are calculated lane-border positions. Note that algorithms for lane-border detection not necessarily provide curves; it might be just isolated dots.

whole process easier, and to reduce the errors in of point selection, going thus further towards a fully automated solution.

The paper is structured as follows. Section 2 provides a brief explanation of the drawing technique by Borkar et al. and of our improvements. Section 3 recalls a lane-border detection method, published by Jiang et al. in [6], which we implemented and used for illustrating performance evaluation. Section 4 discusses the may how to quantify performance by introducing measures. Some of the actual experiments are reported in Section 5. Section 6 concludes.

2 Ground Truth by Time Slices

Evaluations of computer vision techniques based on available ground truth became a widely accepted approach for improving methods, for identifying issues and to help to overcome those. Current examples are the KITTI benchmark suite [4, 7] and some of the sequences on EISATS [3, 8]. These are websites offering long video sequences, more that 100 frames each, for testing vision algorithms on recorded road scenes. Lane-marking ground truth has been not yet considered there so far. So far, really challenging video data are only suggested for subjective evaluations of algorithmic performances, such as during the Robust Vision Challenge at ECCV 2012 [11]. Altogether, this seems to illustrate the difficulty of providing usable ground truth for lane detection.

Building ground-truth data base for lane sequences requires generating curves indicating where the actual lane border is located in an image. Borkar et al. create an image called a *time slice* by selecting points from each frame of the given sequence, and apply spline interpolation for those selected points to generate ground truth. See Figure 2. In this paper we illustrate results for the same data set as used by Borkar et al. in [2], Sequence 1 consists of 1372 frames, each having a resolution of 640×500 pixels.

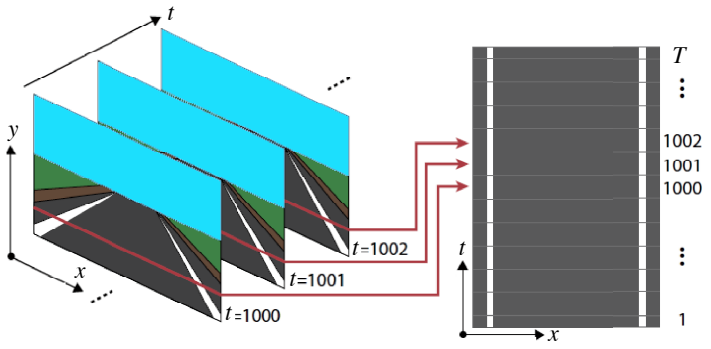

Fig. 2. A time slice is created by using a stack approach, combining single rows of pixels from each image frame into one sequence. This figure follows a figure given in [2], but using modified notation.

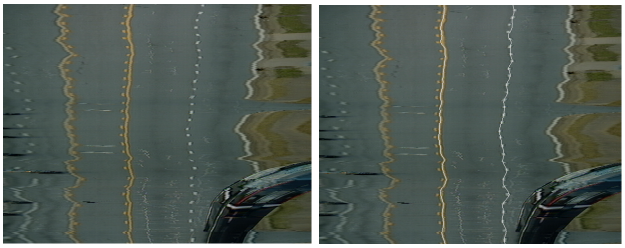

Fig. 3. Samples of generated time slices. The images have 497 rows, meaning that 497 frames contributed each one row to those generated time slices.

For generating $n > 0$ time slices, as defined in [2] for any given video sequence, we extract n rows of pixels from each frame at fixed row locations. The distance between subsequent rows should be around 20 to 30 pixels, assuming a standard 640×480 VGA image format. For n, a value between 3 and 5 is reasonable.

Each of the n fixed rows, accumulated over time, defines a time slice: the row from Frame 1 goes into the bottom-most row, the row from Frame 2 into the next, and so forth.

See Figure 3 for examples of two time slices calculated for the same image sequence but for different fixed rows. After creating $n > 0$ different time slices, [2] suggests to manually select a number of points on the left and right lane borders, and to apply curve fitting using cubic spline interpolation. This generates new

Fig. 4. Results of applying curve fitting by interpolating manually selected points to the time slice shown on the right in Figure 3. The interpolated curves are shown as thin white curves following the lane markings.

Fig. 5. Final results (white curves) after propagating points from time slices into the image sequence and interpolating those points. The image is a frame from one of the data sets provided by Borkar et al., see [2].

points between the few chosen points and connects them together; see the thin white curves shown in Figure 4.

After repeating the same curve generation on each of the $n > 0$ time slices, [2] propagates all the created points from those time slices into corresponding locations (on the fixed rows) in the original frames. The the authors of [2] re-apply curve fitting once again, by interpolating curves in the recorded frames. This finally generates the proposed ground truth data for each data sequence. Figure 5

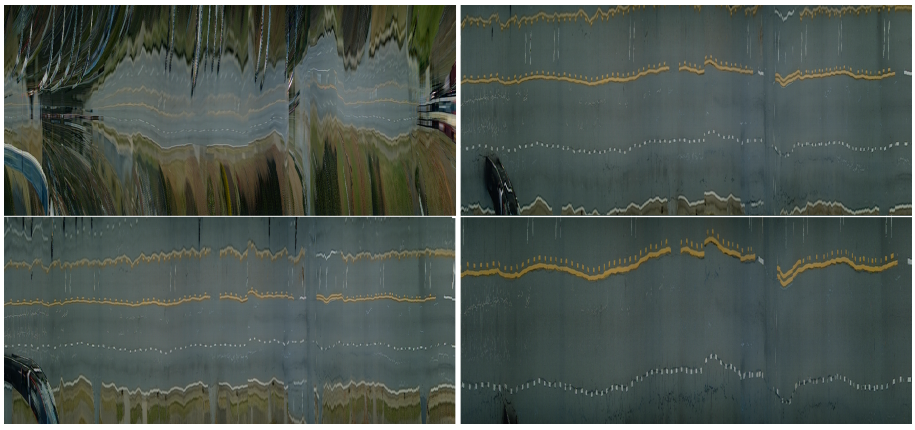

Fig. 6. Generated time slices from Sequence 1 of the used data set, using rows 245 (left, top), 290 (left, bottom), 320 (right, top), and 380 (right, bottom).

Fig. 7. Interpolated curves for row 380 after using our automated point location approach.

shows ground truth generated this way; see also Figure 1 which illustrated such type of ground truth already together with lane detection results applying an adaptation of the algorithm of [6], briefly described in the next Section.

The approach of [2] is easy and reasonably time-efficient in generating ground truth data on different types of video sequences. The problem we experienced is the uncertainty in selecting points in the time slices.

These points should be ideally in the center of the lane borders. See Figures 5 and 4 for visible deviations from an the ideal line. The index of the fixed row (i.e. further down in the frame, close to the ego-vehicle, or further up, thus away from the ego-vehicle) has impacts on accuracy: the further away the more likely that manually located points are not supporting ground truth curves being in the center of lane markings. Figure 6 shows result for four time slices for row indices 245, 290, 320, and 380.

Fig. 8. *Left*: Generated ground-truth curve with the original interactive approach. *Right*: Generated ground truth when using our edge-operator based automated approach.

Our approach: We decided to apply a Canny edge detector on generated time slices and use an automated center correction between detected (left and right) edges of lane borders to find the center points. This not only provides automated point selection, it also improves the accuracy.

See Figure 7 for the automatically generated result for row 380 and compare with Figure 6, right, bottom. Differences are also illustrated in Figure 8.

3 Lane-Border Detection

For performing a comparison of ground truth with calculated results by using a lane-border detector, we decided for using two totally different techniques. The first is an adaptation of the technique described in [6]. This method uses a lane border model, originally proposed in [12], defined by isolated left and right lane border points in one image row. See Figure 9.

Assume we detect points P_L and P_R in one image row for locations of a left and a right lane border. Angle 2α at an assumed fixed height H above the center P_C, being half-way between P_L and P_R, and slope angles β_1 and β_2 at P_L and P_R (with lane borders in a calculated bird's eye view of the frame) define three more parameters of the used lane model. Altogether, the x-coordinates of P_L and P_R, α, β_1, and β_2 define a 5-dimensional space for locating a pair of lane border points in one image row. [6] suggested to use a particle filter for propagating such

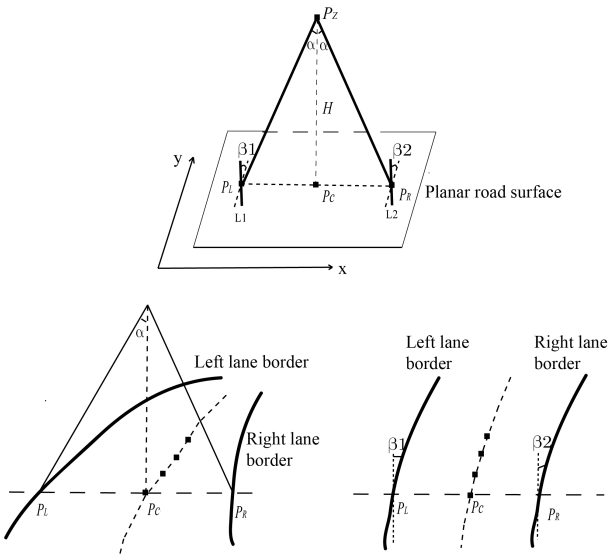

Fig. 9. Lane model based on isolated points [12]. *Top*: model, assuming a planar road. *Bottom, left*: Sketch of a perspective 2D lane view in an input image. *Bottom, right*: Sketch of a bird's-eye image of the lane. See text for further explanations. Reproduction of a figure by courtesy of the authors of [5].

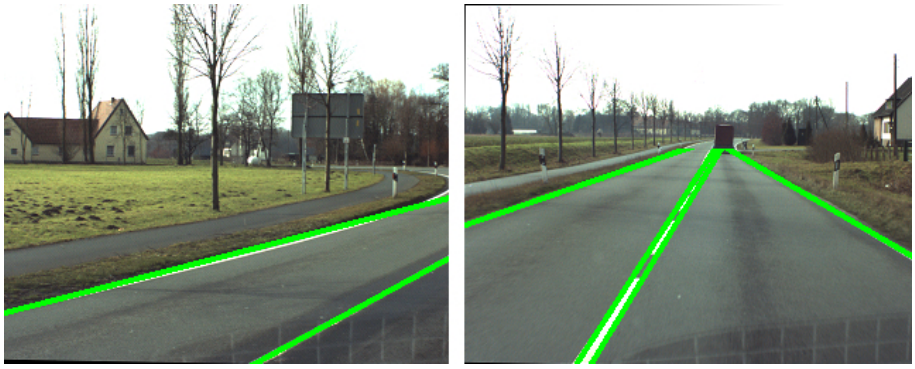

Fig. 10. Illustration of our Hough-transform based lane border detection method, using data from EISATS [3]. In this case we even detect edges of lane borders, e.g. clearly visible in the image on the right and the middle lane border.

pairs of lane border points upward in the recorded frame. The row-component of the Euclidean distance transform (REDT) is used for detecting center pixels P_C, defining P_L and P_R this way, and the particle filter uses then previous (lower) row results and the results of REDT for calculating pairs in subsequent (higher) rows of a frame. The algorithm also uses temporal propagation of detected pairs from Frame t to Frame $t + 1$.

Figure 1 already illustrated (by yellow dots) located pairs of border points using this algorithm.

As a second lane border detection method, we also have an implementation of a Hough-transform based lane border detection algorithm, approximating lane borders by line segments; see Figure 10. We skip details here.

Both outputs are qualitatively different by having isolated pairs of points in the method following [6], but connected curves for the Hough-based method. We discuss performance measures for the isolated points (which extends to connected curves: every connected digital curve is actually a sequence of isolated pixels) and for connected curves such as the Hough-transform.

4 Performance Measures

In this section we describe two measures for comparing ground truth with calculated lane borders. In each frame we only consider an interval of relevant rows, with indices between $y_{\min}$ and $y_{\max}$; see Figure 11.

First we describe a measure for comparing isolated points, as of relevance for the method proposed in [6]. For each row y, with $y_{\min} \leq y \leq y_{\max}$, we have the following cases to be considered for ground truth:

1. ground truth provides points on a left and a right lane border (case GTB)
2. ground truth only provides a point on the left border (case GTL)
3. ground truth only provides a point on the right border (case GTR)

Fig. 11. Relevant rows while driving on a planar road. We are not interested in rows too close to the ego-vehicle, and not in rows too far away.

4. ground truth provides no point in this row y (case GTN)

Analogously, we also have the cases BDB, BDL, BDR, and BDN for border detection (BD). For example, in Figure 10, left, we do not even see the right lane border the ego-vehicle is driving in at that moment. We consider cases as GTL, GTR, and GTN as being defined by the circumstances, not to be penalized.

In row y, with $y_{\min} \leq y \leq y_{\max}$, we use the notation $x_{y,L}^{GT}$ and $x_{y,R}^{GT}$ for detected ground truth points (if they exist at all), and $x_{y,L}^{BD}$ and $x_{y,R}^{BD}$ accordingly for the studied border detection algorithm. The error $E_{IP}(y,t)$ for the case of a BD algorithm providing isolated points (IP) in row y, with $y_{\min} \leq y \leq y_{\max}$, and considering Frame t of the input sequence, is defined as follows:

$$E_{IP}(y,t) = \begin{cases} ||(x_{y,L}^{GT}, x_{y,R}^{GT}) - (x_{y,L}^{BD}, x_{y,R}^{BD})||_1, & \text{if cases BDB and GTB} \\ ||x_{y,L}^{GT} - x_{y,L}^{BD}||_1 + \tau, & \text{if cases BDL and GTL or GTB} \\ ||x_{y,R}^{GT} - x_{y,R}^{BD}||_1 + \tau, & \text{if cases BDR and GTR or GTB} \\ \tau, & \text{if cases BDN but not GTN} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Value τ is a penalizer for inconsistency of BD results with GT. Due to the given resolution of 600×500, we decided for $\tau = 10$. Of course, the L_1-distances (i.e. absolute values) in Equation (1) can also be replaced by another distance measure such as a Mahanalobis distance.

This error measure provides multiple options for penalizing inconsistencies between the BD algorithm and the ground truth. We only state here formally

one option, defined by the use of only all those rows $y_{0,t} < y_{1,t} < \ldots < y_{m_t,t}$, with $y_{\min} \leq y_{i,t} \leq y_{\max}$ for $i = 0, \ldots, m_t$ where we have both GTB and BDB. This gives us the error

$$E_{BD} = \frac{1}{T} \sum_{t=1}^{T} \left[\frac{1}{m_t + 1} \sum_{i=0}^{m_t} E_{IP}(y_{i,t}, t) \right] \tag{2}$$

for method BD for the whole sequence of T frames.

As mentioned earlier, because the lines provided by the Hough transform are also composed of isolated points we can actually apply this measure for both the method following [6] as well as for the Hough-transform based method.

For completeness we also briefly outline our second performance measure, designed for comparing a *curve* provided by GT with a *curve* provided by BD. Here we interpolate given BD data by a Spline curve in the interval $[y_{\min}, y_{max}]$ of rows. There are several cases to be considered, such as an algorithm providing isolated points does not contribute border points "close" to minimum or maximum values $y_{\min}$ or y_{max}, or BD provides left and right approximations of border lines (defined by a very short distance between bot approximated lines), such as illustrated in Figure 10. In this case we interpolate between both approximating curves for being able to compare with the GT curve which represents the center line of a lane border marking.

5 Experiments and Discussion

We tested the proposed way of ground truth generation and the proposed performance measures on different data, using both the isolated-point method of [6] as well as the Hough-transform based method as options for BD.

We selected five different sequences from different sources (data provided by [2] and data available on [3]). Interestingly, we obtained individual measurements for those sequences which come with different characteristics, defined by frame resolution, used camera, and bit per pixel, but also by road, traffic, light, contrast, or weather conditions.

We can add tables here, but believe that these are not very illustrative for the reader. In summary, non of both border detection methods was the all-time winner. Not surprising, the Hough-transform based method preforms better on straight roads having clear lane border markings. The isolated-point technique appears to be better suited for the more challenging conditions.

6 Conclusions

We proposed an improvement for the method proposed in [2] for generating ground-truth for lane borders in recorded on-road video data. The proposed addition works fine in general for going towards an automated process, and also for identifying centers of lane border markings.

We defined two performance measures and applied those for two different lane border detection algorithms, representing different methodological approaches. The evaluation is in consistency with statements, for example, in paper [8], that the diversity of situations requires adaptive selections of techniques for optimizing analysis results.

The proposed framework can be used to identify better algorithms which correspond in their performance to a given situation or scenario.

References

1. Bar Hillel, A., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: A survey. *Machine Vision Applications*, **???**:1–19 (2012)
2. Borkar, A., Hayes, M., Smith, M.T.: An efficient method to generate ground truth for evaluating lane detection systems. In Proc. IEEE Int. Conf. Acoustics Speech Signal Processing, pages 1090–1093 (2010)
3. EISATS benchmark data base. The University of Auckland, `www.mi.auckland.ac.nz/EISATS` (2013)
4. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In Proc. IEEE Int. Conf. Computer Vision Pattern Recognition, pages 3354–3361 (2012)
5. Jiang, R., Klette, R., Wang, S., Vaudrey, T.: New lane model and distance transform for lane detection and tracking, In Proc. CAIP, LNCS 5702, pages 1044 – 1052 (2009)
6. Jiang, R., Klette, R., Vaudrey, T, Wang, S.: Lane detection and tracking using a new lane model and distance transform, *Machine Vision Applications*, **22**:721–737 (2011)
7. KITTI vision benchmark suite. Karlsruhe Institute of Technology, `www.cvlibs.net/datasets/kitti/` (2013)
8. Klette, R., Krüger, N., Vaudrey, T., Pauwels, K., Hulle, M., Morales, S., Kandil, F., Haeusler, R., Pugeault, N., Rabe, C., Lappe, M.: Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database. *IEEE Trans. Vehicular Technology*, **60**:2012–2026 (2011)
9. McCall, J.C., Trivedi, M.M.: Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Trans. Intelligent Transportation Systems*, **7**:20–37 (2006)
10. Revilloud, M., Gruyer. D., Pollard, E.: Generator of road marking textures and associated ground truth applied to the evaluation of road marking detection. In Proc. IEEE Int. Conf. Intelligent Transportation Systems, pages 933 – 938 (2012)
11. Robust Vision Challenge at ECCV 2012. See `hci.iwr.uni-heidelberg.de/Static/challenge2012/` (2012)
12. Zhou, Y., Xu, R., Hu, X., Ye, Q.: A robust lane detection and tracking method based on computer vision. *Measurement Science Technology*, **17**:736–745 (2006)