

Fig. 2. (a) On-board camera-view; (b) Bird-eye view

be explained in section 2.5, and this is the main reason why we have chosen this reference system. Furthermore, the size of the bird-eye (256x128) is much smaller than the original image (640x480), so that its processing is considerably faster.

However, the bird-eye view presents calibration problems. If the extrinsic calibration parameters of the vision system – i.e. its position and orientation in world coordinates – are not well calculated, the flat road assumption is violated, and the bird-eye view image will show converging or diverging lines instead of parallel ones. This leads to a bad calculation of the lane position and the lane orientation. In order to overcome these problems, an auto-calibration algorithm based on evolutionary techniques is used [5]. This algorithm gives a first estimation of the extrinsic parameters of the vision system, and is run when the cameras are installed in the vehicle. Thereafter, the pitch angle is corrected in every frame by detecting the height of the horizon, as explained in section 2.6.

2.2 Road Model

The road model comprises two parts, the road geometry (linear, parabolic, etc.) and the road type (one-way, two-way or freeway with a variable number of lanes). As has been said in Sect. 1.1, many geometric road models have been extensively researched, but there is little emphasis in road type interpretation. This algorithm is designed to automatically classify the road lines, detect the number of lanes, and track them. Thus, the main contribution of this paper is the automatic road type detection. At present, the algorithm works in freeways with a variable number of lanes.

With regard to road geometry, in this paper we consider the road to be straight for three main reasons. First, straight lines are faster to detect and faster to track than higher order models. They can be robustly and quickly extracted with the Hough Transform, a technique that can hardly be applied to more complex models in real time. Second, it eases other processes such as auto-calibration, the tracking of the pitch angle, and, above all, the road lines classification. Finally, it is a reasonable approximation in the nearby region of the road.

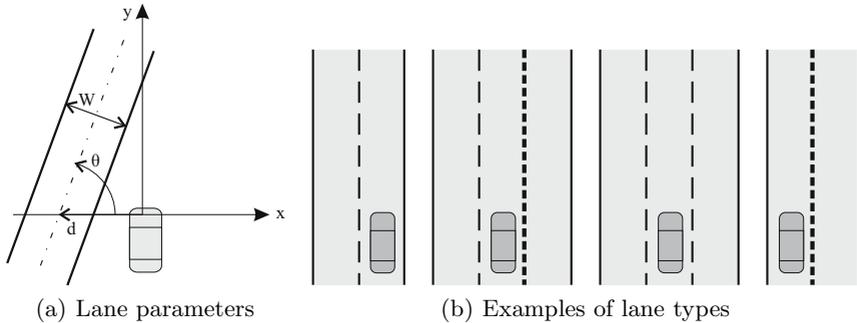


Fig. 3. Road model

The geometric road model is shown in Fig. 3. It has three parameters: d is the distance to the center of the ego-lane, θ is the yaw of the vehicle with regard to the lane, and W is the lane width. The road type model considers the road as a freeway with up to three lanes, in which the lane boundaries can be one of three types: *continuous*, *broken* or *merge*.

2.3 Road Markings Detection

This step extracts from the original image the pixels that are candidates to belong to a road line. Road lines can be considered as bright bands over a darker background. As the lane curvature is small in the nearby region of the road, these lines are mainly vertical in the bird-eye view image of the road. Therefore, the search for pixels that belong to road markings consists of looking for dark-bright-dark transitions in the horizontal direction.

The borders of the image are extracted with a spatial filter based on the ideas of the Canny border extractor, which offers a good signal-noise ratio, compared to other border extractors. This filter uses the intermediate steps of the Canny filter to estimate the orientation of the border, and is used to obtain a horizontal gradient image. Thus, the borders that are not essentially vertical are discarded.

Figure 4 shows how road markings produce two opposite peaks within a certain range of distances in a row of the gradient image. The algorithm scans the horizontal gradient image row by row, searching for a pattern composed of a pair of peaks of opposite sign which are spaced a distance equal to the line width. The line width is considered to be between ten and fifty centimeters in world coordinates. When this pattern is found, the middle point is labeled as a road marking.

2.4 Adaptative Road Lanes Detection

Next, the Hough Transform is used to detect straight lines. Compared to other model fitting methods, the Hough Transform is very robust as it uses global

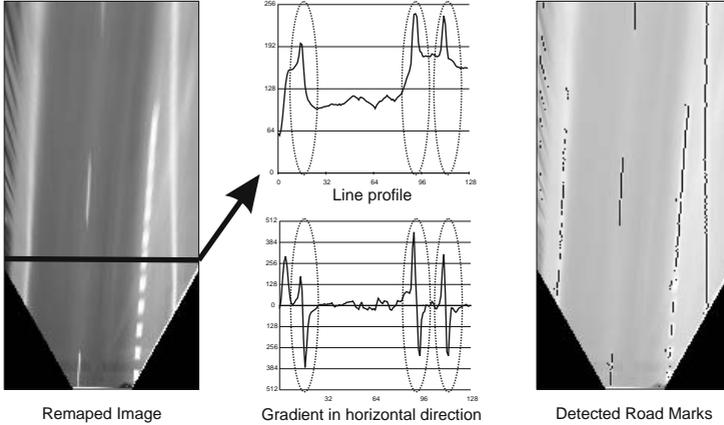


Fig. 4. Detection of pixels belonging to road markings

information. Then, it can easily detect the road lines even though they are broken or partially occluded. In addition, when the model is simple and the image is small, it is fast enough to be applied in real time.

The usual ρ - θ parameterisation is used for straight lines. As the lines are mainly vertical, in the accumulator matrix the parameter θ is constrained to the range $[-15^\circ, +15^\circ]$. Once the accumulator is calculated, only some regions of interest (ROIs) are scanned for local maximums. The ROIs are delimited with the predictions of the Kalman filter. This fact speeds up computation, and avoids interferences with other features outside of the search region.

Kalman filter is used to track five variables: the lateral position (d) and speed (\dot{d}) of the vehicle with regard to the center of the ego-lane, the orientation (θ) of the vehicle respect to the lane, the angular speed ($\dot{\theta}$), and the lane width (W). The width and height of the ROIs – i.e. the interval in ρ and θ – are calculated from the confidence interval of the lateral position and the orientation of the vehicle, respectively.

For the first frame, only two ROIs are considered. These ROIs are big enough to contain the right and left boundaries of the ego-lane. If several lines are found in the same ROI, the algorithm tries to match each line or one ROI with a line of the other ROI, i.e., find the opposite lane border, which should have the same orientation. The best match is used as the initial observation that will be tracked. If no lines can be matched, the most voted line is used. For subsequent frames, if several lines are found, Kalman filter will select the observation that gives the best χ^2 -test result.

Once the ego-lane has been detected, its lane boundaries are classified in continuous, broken or merge (as explained in Sect. 2.5). When a line is identified as broken or merge, an additional ROI is created to look for a new road line that should be at the same side, and separated a distance equal to the lane width (Fig. 5).

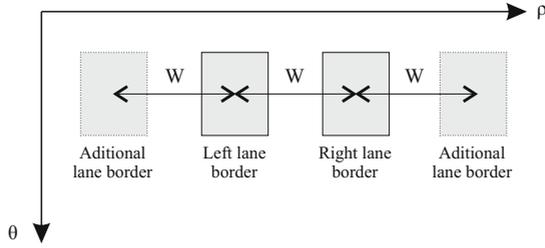


Fig. 5. Regions of interest in the accumulator of the Hough Transform

2.5 Road Lines Classification

The extracted lines are classified in the different types of lines that are found on roads. The main difficulty of this task is the lack of international standardization of the length and frequency of the white stripes in broken lines. However, most roads have three basic line types already mentioned, namely: *continuous*, *broken* and *merge*.

In order to explain this stage of the algorithm, the three lines showed in Fig. 6 will be used as examples. Each of them represent one of the three classes that are being considered.

The intensity line profile for each detected line (right column of Fig. 7) is not a good data to feed the frequency analysis, because its appearance changes substantially with the environment conditions. Besides, the resolution of the bird-eye view in the distance is poor. This effect represents an inconvenience in the merge lines, which appear blurred far ahead and could even look like a continuous one (Fig. 7(a)). Besides, the power spectrum (left column of Fig. 7) presents a tiny peak at the specific frequency of the merge line.

It is more robust to obtain the line profile from the thresholded image given by the road markings detection step (Fig. 6(b)), which is showed on the left side

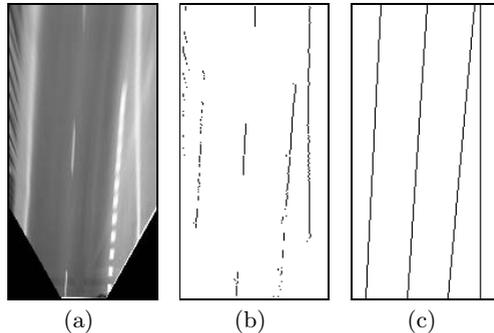


Fig. 6. (a) Remapped image; (b) Detected Road Markings; (c) Lines detected by Hough Transform

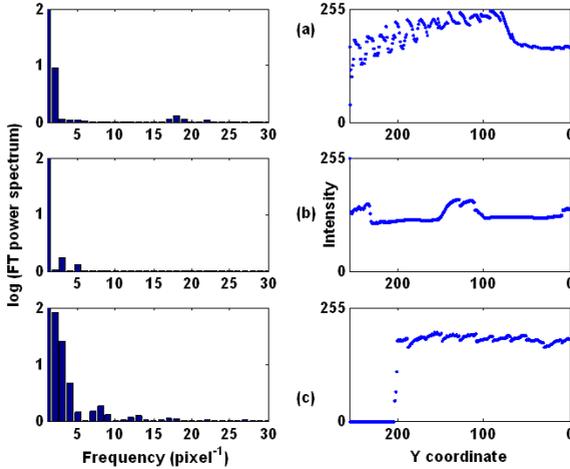


Fig. 7. (right) line profile extracted from the intensity image (Fig. 6(a)); (left) Power spectrum of the Fourier analysis; (a) merge line; (b) broken line; (c) continuous line

of Fig. 8. Again, the right side of the figure shows the power spectrum of Fast Fourier Transform applied to the line profile vector. The results show that a clear sharp peak appears in the Fourier Transform power spectrum when the line is broken, and that the value of the frequency associated to that peak gives the line type (broken or merge). These peaks are showed on the left side of Fig. 8(a) and Fig. 8(b) with arrows pointing at them. No significant peaks are present when the line is continuous (Fig. 8(c)). It can now be seen that the peaks are sharper and much easier to detect. It has been heuristically found that only the first 21 frequencies are significant in this analysis.

Thus, the classification is performed by scanning the first 21 frequencies. Two requirements are needed in order to classify a line as broken or merge:

1. In the first place, a peak must be found within a certain range of frequencies. Two different frequency ranges have been specified. The broken vertical lines on the left side of Fig. 8(a) and Fig. 8(b) show the limits for merge and broken lines, respectively.
2. In second place, the peak must overcome a threshold, which depends on the frequency interval, as the height of the peak decreases as the frequency increases. On the left side of Fig. 8, a horizontal line shows the threshold for each frequency interval. Figure 8(c) shows that no peak exceeds the threshold in neither of the specified frequency ranges when the line is *continuous*.

2.6 Pitch Angle Correction

The extrinsic parameters of the vision system are calculated during installation, but these parameters suffer small drifts during driving, specially the pitch angle

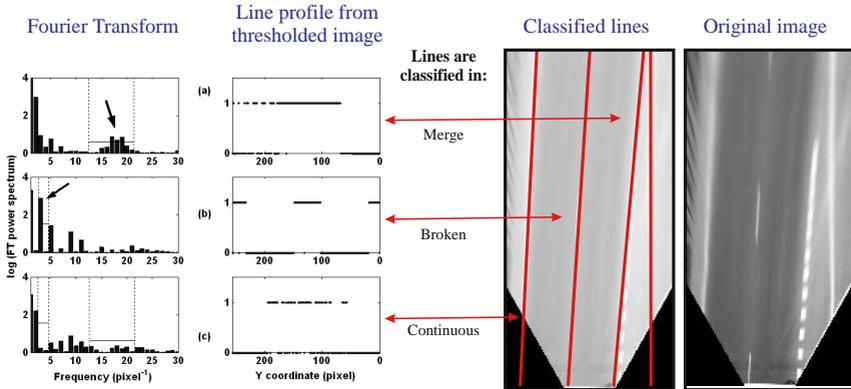


Fig. 8. Fourier analysis for the line profile; (a) merge line; (b) broken line; (c) continuous line

due to the usual swaying of the vehicle, e.g. in sudden braking, dips, etc. This is the most critical parameter because it notably distorts the bird-eye view obtained through the perspective transformation.

In order to correct the pitch angle, the image is processed twice. First, two lane boundaries are detected and its intersection point calculated. This point should belong to the horizon line. These lines do not need to be classified since they are only used to estimate the horizon height. Then the pitch angle (ϕ) is given by the equation:

$$\phi = \arctan\left(\frac{y_{horizon} - y_{center}}{f}\right) \quad (1)$$

where: $y_{horizon}$ is the y coordinate of the horizon line in pixels,
 y_{center} is the y coordinate of the center of the CCD in pixels, and
 f is the focal distance in pixels.

With the updated pitch angle, the bird-eye view is regenerated, now with correct parameters. The corrected image is processed according to the steps explained in previous sections.

3 Results

This algorithm has been tested in the IvvI platform. IvvI (Fig. 9) is a research platform for the implementation of systems based on computer vision, with the goal of building an Advanced Driver Assistance System (ADAS). It is equipped with a stereo-vision system composed of two B&N progressive scan cameras used for road, vehicle, and pedestrian detection, a color camera, used for traffic signs detection, a GPS to measure speed, and a processing system composed of two Pentium IV computers.

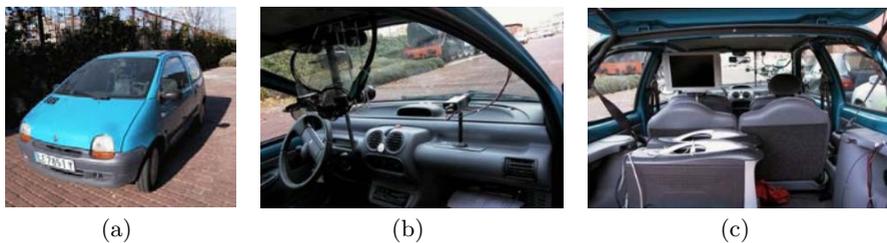


Fig. 9. (a) IvvI vehicle; (a) vision system; (c) processing system

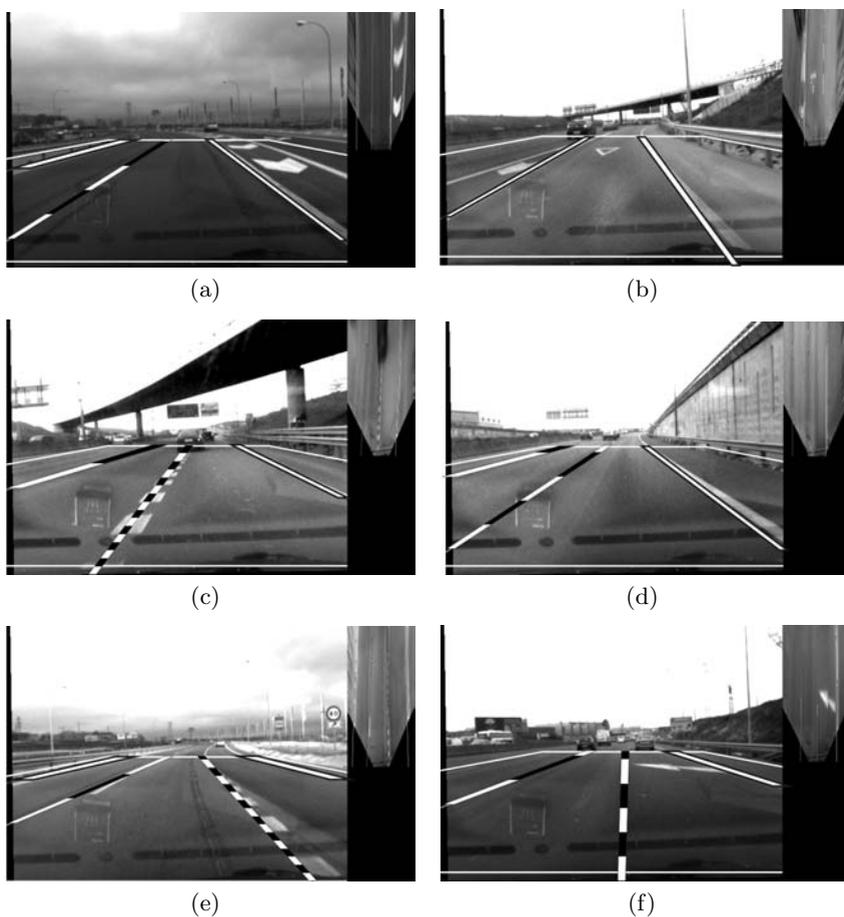


Fig. 10. Examples of detected lanes. On each example, the remapped image is at the top-right corner. The remapped region of the original image is delimited with a white line.

Figure 10 shows some examples of the execution of the algorithm. It can be seen how adjacent lanes are searched when broken or merge lane boundaries are detected.

The whole algorithm takes about 100 milliseconds in a Pentium IV, including rectification of the image – the image needs to be rectified because it comes from the left camera of a stereo-vision system –, perspective transformation (twice, due to the correction of the tilt angle), Hough Transform and road lines classification. Thus, it runs at about 10 fps. Higher rates can be achieved if the correction of the tilt angle is used to remap the next frame, instead of the current one.

4 Conclusions and Perspectives

In this paper, the Road Detection and Interpretation module of the Advanced Driver Assistance System for the Ivvi project, has been presented. It is able to track the ego-lane and automatically identify lane boundary types and detect adjacent lanes if present. It can process a video sequence at nearly real time.

Detection and tracking of the road lanes is robustly performed. Also, the road line classification works reasonably good. However, this parameter of the model should be also tracked in the future, in order to filter some spurious misclassifications.

Likewise, the performance can be enhanced if interaction with other modules of the Ivvi is implemented, especially with the vehicle detection one [8]. Lane position helps vehicle detection by giving an idea of the regions of the image susceptible of containing a vehicle, and the estimated size of the vehicle depending on the image position, which is related to the distance to the camera. It also helps to know if a vehicle is likely to be oncoming or out-coming depending on the lane where it is and the road type. Finally, the vehicle detection module can help the lane detection module to avoid analyzing the areas of the image occupied by other vehicles.

References

1. R. Aufrère, R. Chapuis, and F. Chausse. A model-driven approach for real-time road recognition. *Machine Vision and Applications*, 13(2):95–107, November 2001.
2. A. Broggi. Robust real-time lane and road detection in critical shadow conditions. In *IEEE International Symposium on Computer Vision*, pages 353–358, Coral Gables, Florida, November 19–21 1995. IEEE Computer Society.
3. N. W. Campbell and B. T. Thomas. Navigation of an autonomous road vehicle using lane boundary markings. In D. Charnley, editor, *1st IFAC Int. Conference on Intelligent Autonomous Vehicles*, pages 169–174. Pergamon Press, 1993.
4. R. Chapuis, R. Aufrère, and F. Chausse. Accurate road following and reconstruction by computer vision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 3 of 4, December 2002.
5. Juan M. Collado, Cristina Hilario, Arturo de la Escalera, and Jose M. Armingol. Self-calibration of an on-board stereo-vision system for driver assistance systems. In *IEEE Intelligent Vehicle Symposium*, Tokyo, Japan, June, 13–15 2006.

6. D. DeMenthon. A zero-bank algorithm for inverse perspective of a road from a single image. In *IEEE International Conference on Robotics and Automation*, pages 1444–1449, Raleigh, NC, April 1987.
7. E. D. Dickmans and B. D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 14 of 2, pages 199–213, February 1992.
8. C. Hilario, J. M. Collado, J. M. Armingol, and A. de la Escalera. Pyramidal image analysis for vehicle detection. In *IEEE Intelligent Vehicle Symposium*, pages 87–92, Las Vegas, Nevada, U.S.A., June 6-8 2005.
9. Paul V.C. Hough. Machine analysis of bubble chamber pictures. In CERN, editor, *International Conference on High Energy Accelerators and Instrumentation*, pages 554–556, 1959.
10. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. Kluwer Academic Publishers.
11. K. Kanatani and K. Watanabe. Reconstruction of 3-d road geometry from images for autonomous land vehicles. In *IEEE Transactions on Robotics and Automation*, volume 6 of 1, February 1990.
12. K. Kluge and S. Lakshmanan. A deformable-template approach to lane detection. In *Intelligent Vehicles '95 Symposium., Proceedings of the*, pages 54–59, 25-26 September 1995.
13. R. Risack, P. Klausmann, W. Küger, and W. Enkelmann. Robust lane recognition embedded in a real-time driver assistance system. In *IEEE International Conference on Intelligent Vehicles*, pages 35–40, 1998.
14. B. Southall and C.J. Taylor. Stochastic road shape estimation. In *8th IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 205–212, 7-14 July 2001.
15. Y. Wang, D. Shen, and E.K. Teoh. Lane detection using spline model. *Pattern Recognition Letters*, 21(8):677–689, July 2000. *Pattern Recognition Letters*, vol.21, no.8, July 2000. p. 677-689.
16. Y. Wang, E. K. Teoh, and D. Shen. Lane detection and tracking using b-snake. *Image and Vision computing*, 22:269–280, July 2004.
17. A. L. Yuille and J. M. Coughlan. Fundamental limits of bayesian inference: order parameters and phase transitions for road tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):160–173, February 2000.