Self-calibration of an On-Board Stereo-vision System for Driver Assistance Systems

Juan M. Collado, Cristina Hilario, Arturo de la Escalera and Jose M. Armingol
Intelligent Systems Lab
Department of Systems Engineering and Automation
Universidad Carlos III de Madrid, 28911, Leganés, Madrid, SPAIN
E-mail: {Cristina.Hilario,JuanManuel.Collado,JoseMaria.Armingol,Arturo.delaEscalera}@uc3m.es

Abstract

Vision-based Driver Assistance Systems need to establish a correspondence between the position of the objects on the road, and its projection in the image. Although intrinsic parameters can be calibrated before installation, calibration of extrinsic parameters can only be done with the cameras mounted in the vehicle.

In this paper the self-calibration system of the IVVI (Intelligent Vehicle based on Visual Information) project is presented. It pretends to ease the process of installation in commercial vehicles. The system is able to self calibrate a stereo-vision system using only basic road infrastructure. Specifically, only a frame captured in a straight and plane stretch of a road with marked lanes is required. Road lines are extracted with the Hough Transform, and used as a calibration pattern. Then, a genetic algorithm finds the height, pitch and roll parameters of the vision system. The user must run this algorithm only once, unless the vision system is replaced or reinstalled in a different position.

1 Introduction

Companies are showing an increasing interest in Driver Assistance Systems (DAS). Many of them have declared its interest in commercialize vision-based DAS in the short term. These vision-based DAS need to establish a correspondence between the position of the objects on the road, and its projection in the image. Thus, it is essential a calibration process in order to obtain the parameters of the vision-system.

A vision-system has intrinsic and extrinsic parameters. Intrinsic parameters are those related to the camera-optic set, and may be pre-calibrated in factory or before being installed on-board in the vehicle. However, extrinsic parameters (position and orientation of the stereo system, as showed in figure 2(a)) can only be calibrated after installation. Calibration of extrinsic parameters can be done in factory, by technical operators, or afterward, by the user. The algorithm proposed in this paper is useful in both cases.

Obtaining the extrinsic parameters has several problems. On one hand, due to the variability of vehicles, the space of possible positions and orientations is big enough to prevent using initial guesses. On the other hand, any calibration process need a calibration pattern, and when the stereo systems is already installed on-board, the pattern must be seen from the cameras. Most current systems use a calibration pattern which is painted on the road [1], painted on the hood of the vehicle [2], or in a moving object [3].

The algorithm presented here intends to ease installation of stereo-vision systems, by taking advantage of road structure, i.e, using the structured, and partially known, road environment as a calibration pattern.

In short, the main goal of this work, is to design and implement a self-calibration algorithm for stereo-vision systems boarded on vehicles, with the following requirements:

1. allowing to obtain the main extrinsic parameters needed by the Driver Assistance System of the IVVI, namely, height, pitch and roll.
2. using exclusively objects of the basic road infrastructure.
3. requiring minimum user participation.

The designed algorithm uses the road lane boundaries as a calibration pattern, and a genetic algorithm (GA) [4] as an optimization technique. The user only needs to drive the vehicle through a straight and plane stretch of a road with marked lanes, and indicate the system to initiate the auto-calibration process. This is only needed to be performed once, unless the vision system is changed or reinstalled in a different position.

This algorithm has been designed to support other capabilities of a DAS.
1.1 Previous work

To calibrate monocular systems usually requires to simplify the problem, or certain information about the environment. In [5] the calibration problem is reduced to detect the height and the pitch of the vision system. For the pitch, it is enough to detect the horizon line. However, in order to make distance measures in the image, the user must provide the algorithm with the true distance between two points.

Similarly, in [6] a supervised calibration method is described. It requires the user to supply the width of the lane, and to select from the detected lines in the image, which of them are lane boundaries. As in [5], only the pitch angle is considered.

In [7], the orientation of a moving camera is calculated by analyzing sequences of 2 or 3 images. The camera is supposed to keep the same orientation during movement. The algorithm needs to know the height of the camera, and the displacement of the vehicle between two frames.

Other systems use a pattern painted on the floor to calibrate a binocular system. In [1] the pattern is used to calibrate each camera separately. The equations are linearized and solved with a recursive minimum square technique, and the algorithm outputs the displacement and rotation of the stereo-system respect to the world. Other works have a supervised stage, where, once the grid pattern has been captured by a binocular system, the user selects the intersections of the grid lines [2]. It follows an unsupervised stage, where the parameters are refined with an iterative process, assuming that the images captured from both cameras come from a flat textured surface. This technique is also used to correct small drifts of the extrinsic parameters due to the movement of the vehicle, but, instead of a flat textured surface, it uses several markers placed on the hood of the vehicle that can be easily detected by the cameras.

In [8], no special calibration pattern is needed, and the “v-disparity” algorithm is used to sequentially estimate roll, pitch and yaw parameters of the vehicle.

This paper presents a new approach, able to estimate simultaneously the roll, pitch, and height of a stereo vision system, without the need of an artificial calibration pattern.

2 Algorithm Description

The main idea of this algorithm is to use the road lane boundaries as a calibration pattern. First, we start from a straight and plane stretch of a road (figure 1(1)). Then, we capture an image with both cameras (figure 1(2)). Next, pixels belonging to road lanes are detected, and lines are extracted with the Hough Transform. After that, assuming a possible set of extrinsic parameters, the perspective transformation is reversed (figure 1(3)). If the extrinsic parameters coincide with the true ones, the images projected from both cameras onto the road plane should be identical, but, if the parameters are wrong, then the road lines will not match neither be parallel. A genetic algorithm tries to find a set of extrinsic parameters that makes the bird-eye view obtained from right and left camera be coherent. In other words:

1. all the road lines appear completely parallel,
2. the road lines projected from right camera match the lines projected from the other.

Therefore, the genetic algorithm evaluates the fitness of the possible solutions following the two criteria above mentioned. The fitness function will be detailed in subsection 2.3.

2.1 Stereo-vision System Model

The perspective transformation requires a mathematical model of the complete vision system. This model consists of two parallel cameras joined together with a solid rod, so that they are separated a distance $b$. Thanks to the rectification of the images [9], both cameras can be considered to be identical, and perfectly aligned. The origin of the reference system is located in the middle point of the rod (figure 2(b)).

The projection of any point of the road onto each camera can be calculated using homogeneous matrices, which allows to represent advanced transformations with a single matrix.

In order to obtain the matrix that represents the perspective transformation between road plane and camera plane, several steps are followed:
First the coordinate system is raised a height $H$, and rotated an angle $\beta$ (roll) and $\alpha$ (pitch) over the $OY'$ and $OX$ axes, respectively. Second, the coordinate system is displaced $b/2$ or $-b/2$ (for the right or left camera, respectively) over the $OX$ axis. Next, the point is projected along the $OY$ axis, according to the pin-hole model. Finally, the coordinates are translated from millimeters to pixels, and the origin translated to the optical center of the CCD.

Multiplying these matrices, the global transformation matrix is obtained:

\[
T = \begin{bmatrix}
K_u \cdot f & C_u & 0 & K_u \cdot f \cdot b/2 + C_u \\
0 & 1 & 0 & 0 \\
0 & C_v & -K_v \cdot f & C_v \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

where

- $S$ and $C$ stand for the sine and cosine functions,
- $f$ is the focal distance,
- $K_u$ and $K_v$ refer to the pixel size, expressed in pixel/mm,
- $C_u$ and $C_v$ are the optical center in pixels.

The matrix dimensions can be reduced to $3 \times 3$, if the second row (that corresponds to the depth coordinate, irrelevant in images), and the third column (that corresponds to the height coordinate, always zero if the world is flat), are eliminated.

The resultant matrix $M_{persp}$ stores all the necessary information to perform the perspective transformation:

\[
\begin{bmatrix}
u \\ v \\ s'
\end{bmatrix} = M_{persp} \cdot \begin{bmatrix}
x \\ y \\ 1
\end{bmatrix}
\]

where $s'$ is a scale factor.

### 2.2 Extraction of the calibration pattern

Extraction of the calibration pattern is done in three steps. First, in order to obtain the horizontal gradient, the image is correlated with the kernel $[-1 \quad 0 \quad 1]$.

As road lines are white stripes over a darker background, they can be recognized because if the gradient image is scanned row by row, two opposite peaks must appear at a distance equal to the line width. So, when a positive peak can be matched with a corresponding negative peak, the positive peak is marked as a pixel that belongs to a road line.

Several trials were carried out marking the middle point (equidistant to positive and negative peaks), but the results were not satisfactory. This is because, due to perspective effects, the middle point of the horizontal section, do not coincide exactly with the center of the line.

In the second step, the marked pixels are used by the Hough Transform to detect the right and left road lines. The Hough Transform used here has been improved as explained in [10]. The usual $\rho-\theta$ parameterization is used, and the range of the $\theta$ parameter goes from $-\pi$ to $\pi$, allowing $\rho$ to be negative, so that left and right lane borders have different sign in $\theta$ (figure 3(a)). Then, Hough Transform selects the best line in the region of negative angles, and the best in the region of positive angles. But these detected lines are not valid yet as a calibration pattern. Some experiments have been done trying to compare two lines from its parameters, without success. For this reason, what the algorithm compares is two set of points extracted from the road lines, evaluating the detected lines at different heights in the image. The first height is at 90% of the vanishing point height. The subsequent heights are spaced ten pixels, until the bottom of the image is reached.

### 2.3 Fitness function

Each individual of the population of the genetic algorithm represents a position and orientation of the stereo-vision system.

The GA should evaluate that:

- the projections of the pattern from both cameras onto the road are identical,
- road lines are parallel.

Accordingly, these requirements will be evaluated by an error function composed of two terms. The first term evaluates that the two pairs of lines are matched. The algorithm compares two set of points extracted from the road lines, as already explained. Each
an upper limit (1/so that they have the same importance. The use of inverse
right and left borders of the lane. In Eq. (4) and Eq. (5)
The second term evaluates the parallelism between the
from the left camera and right camera, respectively.
Next, E1 and E2 are normalized and transformed into F1 and F2, as in Eq. (6). K is a constant that helps to define an upper limit (1/K), which is the same for both terms, so that they have the same importance. The use of inverse

\[
E_1 = \sum_i \| \vec{x}_{i,\text{left}} - \vec{x}_{i,\text{right}} \|^2 \tag{3}
\]

where \( \vec{x}_{i,\text{left}} \) and \( \vec{x}_{i,\text{right}} \) are the sets of points extracted from the left camera and right camera, respectively.

The second term evaluates the parallelism between the right and left borders of the lane. In Eq. (4) and Eq. (5) \( \theta \) is the angle between the line and the horizontal axis. Subscripts 1 and 2 represent first and second line in the image (i.e. right and left lane border), while subscripts right and left represents right and left camera, respectively.

\[
E_2 = |\vartheta_{\text{right,1}} - \vartheta_{\text{right,2}}| + |\vartheta_{\text{left,1}} - \vartheta_{\text{left,2}}| \tag{4}
\]

\[
E_2 = \min \left( |\vartheta_{\text{right,1}} - \vartheta_{\text{right,2}}|, |\vartheta_{\text{left,1}} - \vartheta_{\text{left,2}}| \right) \tag{5}
\]

Equation (4) does not work. Experiments using this equations shows that it gives a very high error even when the parameters are close to the true ones. In contrast, Eq. (5) gives better results, and it represents that at least there is one pair of lines well aligned.

Both terms \( E_1 \) and \( E_2 \) should be zero (or nearly zero) in the perfect case. Now, a fitness function must be created from them. Experience has proved that it is indispensable for the fitness function to give both terms the same importance, and to have a high resolution when we are close to the true solution.

Next, \( E_1 \) and \( E_2 \) are normalized and transformed into \( F_1 \) and \( F_2 \), as in Eq. (6). K is a constant that helps to define an upper limit (1/K), which is the same for both terms, so that they have the same importance. The use of inverse

\[
F_1 = \frac{1}{E_1 + K}; \quad F_2 = \frac{1}{E_2 + K} \tag{6}
\]

The global fitness function cannot be a simple sum of \( F_1 \) and \( F_2 \), because the genetic algorithm tends to maximize only one of them at the expense of the other. Thus the fitness function has been defined as the minimum of the two terms:

\[
Fitness = \frac{1}{K} \min (F_1, F_2) \tag{7}
\]

In the experiments, the constant \( K \) has been empirically set to \( 10^{-4} \), so that the fitness is constrained to the interval \([0, 10^4]\).

### 3 Results

Both tests in synthetic and real environments have been carried out. The behaviour or the algorithm in a real environment has been tested with the IvvI platform (figure 4). IvvI is a research platform for the implementation of systems based on computer vision, with the goal of building an Advanced Driver Assistance System (ADAS). However, a real environment do not allow to evaluate the estimation because it is difficult to measure the true parameters. Thus synthetic images (figure 5) have been used in order to evaluate the performance of the algorithm. These images have been generated through a perspective transformation with the same intrinsic parameters of the cameras mounted on the IvvI (Table I).

Table II shows the parameters of the genetic algorithm (GA), and table III shows the output of the GA and the comparison with the true parameters. For the first set of
parameters, it also includes the error of the estimation and the variance of the estimated parameters calculated from ten executions of the algorithm. The variance of the result is about $10^{-2}$ in height, $10^{-8}$ in pitch, and $10^{-10}$ in roll, that is, the convergence of the algorithm is quite robust.

Fig. 6 shows an example of execution. The figure on the left represents the error of the solution, and the convergence. The figure on the right shows the correspondence between the points projected from the right camera (crosses) and the left camera (circles). It also shows (with dots) the points projected onto the road using the true parameters. In other words, the “dots” represent the true position of the lane on the road, and the “crosses” and “circles” represent the supposed position using the estimated parameters. Error is about 1 meter at 35 meters ahead (3%), and about 5 centimeters at 4 meters (1.3%).

In order to recreate real environment, and to study the sen-

---

### TABLE I

**INTRINSIC PARAMETERS**

<table>
<thead>
<tr>
<th>Image width</th>
<th>640 pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image height</td>
<td>480 pixels</td>
</tr>
<tr>
<td>Focal distance</td>
<td>6.28 mm</td>
</tr>
<tr>
<td>CCD width</td>
<td>7.780 mm</td>
</tr>
<tr>
<td>CCD height</td>
<td>3.589 mm</td>
</tr>
<tr>
<td>CCD X center</td>
<td>342.58 pixels</td>
</tr>
<tr>
<td>CCD Y center</td>
<td>261.43 pixels</td>
</tr>
<tr>
<td>Distance between cameras ($b$)</td>
<td>148.91 mm</td>
</tr>
</tbody>
</table>

### TABLE II

**GA PARAMETERS**

<table>
<thead>
<tr>
<th>Population representation</th>
<th>Real-valued</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>1000</td>
</tr>
<tr>
<td>Number of parents</td>
<td>90% of the population</td>
</tr>
<tr>
<td>Number of children</td>
<td>90% of the population</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>70%</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>1%</td>
</tr>
<tr>
<td>Max. number of generations</td>
<td>50</td>
</tr>
</tbody>
</table>

### TABLE III

**RESULTS OF THE GA**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Height (mm)</th>
<th>Pitch (rad) (deg)</th>
<th>Roll (rad) (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>1500</td>
<td>0.170</td>
<td>9.74</td>
</tr>
<tr>
<td>Estimated</td>
<td>1458</td>
<td>0.169</td>
<td>9.69</td>
</tr>
<tr>
<td>Variance</td>
<td>1.34</td>
<td>9.10$^{-11}$</td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td>42.0</td>
<td>0.001</td>
<td>0.05</td>
</tr>
<tr>
<td>True</td>
<td>1500</td>
<td>0.170</td>
<td>9.74</td>
</tr>
<tr>
<td>Estimated</td>
<td>1484</td>
<td>0.168</td>
<td>9.62</td>
</tr>
<tr>
<td>True</td>
<td>1100</td>
<td>0.170</td>
<td>9.74</td>
</tr>
<tr>
<td>Estimated</td>
<td>1065</td>
<td>0.168</td>
<td>9.64</td>
</tr>
</tbody>
</table>

### TABLE IV

**RESULTS WITH GAUSSIAN NOISE ADDED TO ORIGINAL IMAGES**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Height (mm)</th>
<th>Pitch (rad) (deg)</th>
<th>Roll (rad) (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>1500</td>
<td>0.170</td>
<td>9.74</td>
</tr>
<tr>
<td>5% noise</td>
<td>1455</td>
<td>0.169</td>
<td>9.65</td>
</tr>
<tr>
<td>10% noise</td>
<td>1677</td>
<td>0.170</td>
<td>9.72</td>
</tr>
<tr>
<td>20% noise</td>
<td>1665</td>
<td>0.170</td>
<td>9.71</td>
</tr>
</tbody>
</table>

---

Fig. 5. Road projected onto the cameras of the stereo-vision system

Fig. 6. GA execution example; (+) points projected from right camera; (○) points projected from left camera; (·) points projected using true parameters
Fig. 7. Results with noise added to original images; (left) convergence of the GA and error of the solution; (right) projection of the pattern in world coordinates, where: “+” are points projected from right camera, “◦” are points projected from left camera, and “·” are points projected using true parameters.

Fig. 8. Lines detected and used to generate the calibration pattern.
TABLE V
RESULTS FROM A SEQUENCE OF TEN CONSECUTIVE FRAMES C A P T U R E D F R O M A  C A R D R I V I N G AT 80K M/H

<table>
<thead>
<tr>
<th>Frame</th>
<th>Height</th>
<th>Pitch</th>
<th>Roll</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(m)</td>
<td>(rad)</td>
<td>(deg)</td>
</tr>
<tr>
<td>1</td>
<td>0.951</td>
<td>-0.0020</td>
<td>-0.11</td>
</tr>
<tr>
<td>2</td>
<td>0.999</td>
<td>-0.0032</td>
<td>-0.19</td>
</tr>
<tr>
<td>3</td>
<td>0.926</td>
<td>-0.0072</td>
<td>-0.41</td>
</tr>
<tr>
<td>4</td>
<td>0.970</td>
<td>-0.0074</td>
<td>-0.43</td>
</tr>
<tr>
<td>5</td>
<td>1.020</td>
<td>-0.0131</td>
<td>-0.75</td>
</tr>
<tr>
<td>6</td>
<td>1.047</td>
<td>-0.0131</td>
<td>-0.75</td>
</tr>
<tr>
<td>7</td>
<td>1.028</td>
<td>-0.0091</td>
<td>-0.52</td>
</tr>
<tr>
<td>8</td>
<td>1.006</td>
<td>-0.0055</td>
<td>-0.31</td>
</tr>
<tr>
<td>9</td>
<td>1.034</td>
<td>-0.0078</td>
<td>-0.45</td>
</tr>
<tr>
<td>10</td>
<td>0.986</td>
<td>-0.0078</td>
<td>-0.45</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.997</td>
<td>-0.0076</td>
<td>-0.44</td>
</tr>
<tr>
<td>Std.</td>
<td>0.039</td>
<td>0.0036</td>
<td>0.2074</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0047</td>
</tr>
</tbody>
</table>

Fitting a curved model will make the algorithm work not only in straight sections of road, but in curved sections too.

Acknowledgments

This work was supported in part by the Spanish Government through the CICYT project ASISTENTUR (Grant TRA2004-07441-C03-01).

References


