Localization and guidance with an embarked camera on a mobile robot

K. ACHOUR* and A. O. DJEKOUNE

Robotics and Artificial Intelligence Laboratory, Development Center of the Advanced Technologies, 128 Chemin Mohamed Gacem, El-Madania, 16075 Algiers, Algeria

Received 28 December 2000; accepted 5 August 2001

Abstract—In this paper, a new method using the Hough transform to correct the drift of the mobile robot CESA is presented. The corrections are made by direct observation. As an illustration, an algorithm implemented is detailed, and experiment results for CESA navigation in our laboratory’s corridor and trajectory generation are given.

Keywords: Robotic vision; mobile robot; Hough transform; navigation strategy; calibration.

1. INTRODUCTION

Autonomous vehicles are being increased used in automated factories and other such reasonably well-structured environments. They are capable of intelligent motion (and action) without requiring either a guide to follow or teleoperator control.

For practical applications in the real world, sensory information is required for autonomous navigation in unstructured environments. Ultrasonic sensors have been employed by many researchers for environment recognition and obstacle avoidance. They are relatively inexpensive and easy to use, but provide limited information of the environment. Moreover, the wide beam-opening angle and specular reflections of ultrasonic waves can cause measurement errors [1].

CCD cameras are analogous to human eyes, which project the three-dimensional (3D) environment to a 2D plane. They can provide much useful information to the vehicle for autonomous navigation. Considerable work has been reported concerning the development of image sensing devices and vision-based navigation algorithms for intelligent autonomous vehicles. Image processing system for extracting

*To whom correspondence should be addressed. E-mail: kachour@cdta.dz
image features in real time was developed in [2]. A feedback control scheme was developed for high-speed motion of an autonomous vehicle. A multisensor perception system for the mobile robot Hilare was developed in [3]. In their system, a scene acquisition module was developed using stereo cameras and a laser range finder. This dynamic vision module was used for robot position correction and feature tracking. Song and Tang proposed a sensor system exploiting double ultrasonic sensors and a CCD camera for environment perception [4]. Several investigators have presented dynamic feature tracking systems [3, 5]. In their approach, the vision subsystem and control subsystem work in parallel to accomplish visual servoing. On the other hand, image sequences have been used for motion detection. Aloimonos proposed a design for obstacle avoidance of mobile robots based on image flow [6].

This paper presents a visual guidance control design with its experimentation for the CESA autonomous vehicle (Fig. 1a). In this experimentation, visual guidance was solved as a dynamic control problem, where real-time image processing was
required to provide feedback signals. As an illustration, we present the CESA navigation experiment results in our laboratory’s corridor and free trajectory generation. The drift corrections are made by direct observation and the rotation/translation parameters are computed taking into account the robot neighborhood parameters. These parameters are obtained by applying the Hough transform (HT) on the scene images acquired by an embarked CCD camera. We have used the HT because it has long been recognized as a robust technique for detecting multi-dimensional features in an image and estimating their parameters. It has many applications, as most manufactured parts contain feature boundaries which can be described by regular curves or straight lines. The main advantage of the HT is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

Algorithms for the autonomous vehicle navigation system are being developed and tested both on a computer-aided design workstation and on the experimental CESA autonomous vehicle.

The CESA vehicle shown in Fig. 1a is an experimental vehicle designed to operate autonomously within a structured office or factory environment. It has been designed with several goals in mind. First and foremost, CESA is a testbed with which to experiment with such things as robot programming language and sensor integration/data fusion techniques. Second, CESA is designed to be low cost, and depend on only one camera and two active telemetric sensors.

CESA is a tricycle configuration with a single front wheel, which serves both for steering and driving the vehicle, and two passive load-bearing rear wheels. The onboard control system uses active telemetric sensors and a vision system to provide the position and heading information needed to guide the vehicle along specified paths between stations. The external, or off-board, part of the system consists of a global path planner which stores the structured environment data and produces the path plans from one station to the next. It also includes a radio communication link between this supervisory controller and the vehicle.

This paper is organized as follow, in Section 2, we present the CESA onboard control system. The vision system calibration and the embarked sensors are presented in Section 3. In Section 4, we show the line segment extraction algorithm based on the HT, and the developed and used algorithms for CESA navigation in our laboratory’s corridor. Finally, we present the obtained results and a conclusion.

2. VEHICLE CONFIGURATION

2.1. Mechanical configuration

Figure 1b illustrates the basic structure of the CESA vehicle, a photograph of which is shown in Fig. 1a. It consists of a single steerable drive wheel at the front and two passive rear wheels, i.e., it is a tricycle configuration.

An optical encoder is mounted on the steering motor to provide feedback control. In addition, two optical encoders are mounted on each of two rear, independent non-load-bearing wheels.
The vehicle is powered by two sealed 12 V 60 A batteries which, in the current configuration, provide a useful lifetime of approximately 7 h.

2.2. Computer hardware and software

Control of the vehicle is composed on several modules. Software development is performed on a SUN Spark station running under UNIX. All software is written in C language. The diagnostic data from the vehicle may be downloaded to the host via a radio communication link for analysis.

3. SENSORS

The vehicle is equipped with three different sensors: odometry on each wheel, ultrasonic and infrared sensors, and a vision system. These sensors provide all the navigational sensing information available to the vehicle. They deliver information, which could be used to define the planning tasks strategy of the vehicle and needed to guide it along specified paths between stations without collisions.

3.1. Odometry

Odometry, i.e. the estimation of position and orientation by counting the revolutions of a wheel, dates back to the time of Archimedes [7]. The advantage of odometry is that it is both simple and inexpensive. However, it is prone to several sources of errors: (i) surface roughness and undulations may cause the distance to be over-estimated, (ii) wheel slippage can cause the distance to be under-estimated, and (iii) variations in load can distort the odometer wheels and introduce additional errors.

3.2. Ultrasonic and infrared sensors

The CESA vehicle is equipped with two active telemetric sensors. The proximetric sensors (infrared) at the front detect the presence of an object situated at a distance of less than 1 m and the ultrasonic sensors around detect objects situated at a more important distance for the mobile robot.

3.3. Vision system

The CESA vision system is composed of a camera iVC 500 CCD and a microcontroller-based data acquisition card. The camera delivers images in the CIIR standard (624 lines, 50 Hz). A AD9502 CM video digitalizer performs the conversion of the analog video signal to digital data and separates the synchronization signals (horizontal and vertical).

The digitized image has a resolution of $464 \times 256$ pixels and it is transferred to the supervisory controller at 9600 baud. The 80C51 microcontroller controls the whole operation of the card. The acquisition is performed in real time in 40 ms.

A binary edge-detection algorithm was implemented on a field programmable gate array (FPGA) on this card.
3.3.1. Camera calibration. The CCD camera is modeled on by the pin-hole model [8]. It is sustained by a support and makes an $\alpha$ angle with the horizontal. Considering both the geometric model of the vehicle and the embarked camera (Figs 2 and 3), any binary edge image point $(i, j)$ could be represented in the vehicle coordinate system by the point $(x, y, z)$ by applying the following equation systems [9]:

$$
\begin{align*}
    x &= c_y \sin \alpha + c_z \cos \alpha + dx \\
    y &= c_x + dy \\
    z &= c_z \cos \alpha - c_z \sin \alpha + dz,
\end{align*}
$$

with:

$$
\begin{align*}
    c_x &= -\frac{(c_z - \lambda)}{\lambda} l_x \left( \frac{j}{r_x} - 0.5 \right) \\
    c_y &= -\frac{(c_z - \lambda)}{\lambda} l_y \left( \frac{i}{r_y} - 0.5 \right) \\
    c_z &= \frac{d}{\sin \alpha} + \left( \frac{d}{\sin \alpha} - \lambda \right) \frac{\sin \alpha \cos \beta}{\sin(\alpha - \beta)},
\end{align*}
$$

Figure 2. The camera support.

Figure 3. Image transverse section.
and

\[ \beta = \tan^{-1}\left( \frac{l_y}{\lambda} \left( \frac{i}{r_y} - 0.5 \right) \right), \]

where \( r_x \) is the column number of the screen (pixel), \( r_y \) is the row number of the screen (pixel), \( l_x \) is the width of image plan (m), \( l_y \) is the height of image plan (m), \( \lambda \) is the focal distance of the used lentil, \( d \) is the vertical distance between the camera and the ground (mm), \( \alpha \) is the camera slant angle measured with the horizontal, and \((dx, dy, dz)\) is the component of the vector from the reference point of the mobile robot to the image plan center of the camera (mm).

4. LOCALIZATION AND NAVIGATION STRATEGY

4.1. Robot’s localization

The current vehicle position is obtained by interpreting the visual information extracted from binary edge images. The vehicle displacement is then insured by a position control in the Cartesian space (Fig. 4). The control position is realized by ‘static look and move’ [9]. The measurement errors in Fig. 4 are the result of the corridor’s surface roughness and undulations, wheel slippage and active telemetric sensor interference. These errors drift the vehicle from its desired position, which in our case is the middle of the corridor (robot’s position consign).

The used attributes are the segments. They are used for mobile robot localization and navigation as in [9, 10] because they faithfully produce the robot environment evolution and present a better compromise among the different primitives in terms of hardiness and time processing. The segments are obtained by the HT on images directly at the output of the vision system, because it is a well-known method for detection of parametric curves in binary images, and it was recognized as an important means of searching for objects and features in binary images. It converts

![Figure 4. Position control.](image-url)
the problem of detecting collinear points, or lines, in feature space to the problem
of locating concurrent, or intersecting, curves in parameter space.

Vehicle localization in the corridor consists of defining its reference center
coordinates \((X_G, Y_G)\) (Fig. 1b) with regard to both the left and right corridor borders
extremities extracted from the scene image, by using the two above-mentioned
equation systems (1) and (2).

4.2. Image segmentation and corridor borders extraction

From binary edge images, the HT allows us to extract pertinent information like
lines, curves, etc., according to the needs of applications. In our case, we seek
to detect all line segments contained in an image. The HT used in the normal
parameterization of a line in an image given by [11, 12]:

\[ x \cos \theta + y \sin \theta = \rho, \]  

where \(\rho\) is the normal distance of the line to the origin, \(\theta\) is the angle between \(\rho\) and
the \(x\)-axis, and \((x, y)\) are points on the line (Fig. 5).

The parameters of all lines going through a point \((x_i, y_i)\) in the image constitute
a sinusoidal curve in the parameter space, given by (Fig. 5):

\[ x_i \cos \theta + y_i \sin \theta = \rho. \]  

The sinusoidal curves corresponding to collinear points of a line with parameters
\((\rho_j, \theta_j)\) will cross each other at the point \((\rho_j, \theta_j)\) in the parameter space. So line
parameters can be obtained from the crossing-point of sinusoidal curves in the
parameter space.

The HT of a point \((x_i, y_i)\) is performed by computing \(\rho\) from (4) for all \(n\) values of
\(\theta_k\) into which \(\theta\) is quantized \(0 \leq \theta < \pi\). The values of \(\rho\) are quantized in \(n_\rho\)
intervals of width \(\rho_k\). In this way a quantized sinusoidal curve is obtained and along
the quantized curve each cell is incremented with one ‘1’ value. This procedure is
repeated for all points of the binary edge image.

The line segment extraction from the binary edge image is performed by searching
the collinear points in the binary edge image. These points show up as peaks in the

![Figure 5. The line segment parameter.](image-url)
parameter space (Fig. 6). So, for each peak, we eliminate the effect of all points belonging to this peak in the parameter space [9].

The result of the line segment extraction is a line segment list where each line segment is stored with the following attributes:

- An index: its position in the line segments list.
- $\rho$: its normal distance value relative to the image coordinate system.
- $\theta$: its orientation value in this coordinate system.
- $(x_1, y_1)$ and $(x_2, y_2)$: its extremities coordinates.

We model the corridor by its two left and right borders situated at the corridor’s walls (Fig. 7). Their identification among other segments contained in the observed scene must satisfy the following conditions:

- The corridor’s borders are always oblique.
- The corridor’s right border orientation $\theta$ belongs to the interval $]0, \pi/2[$.
- The corridor’s left border orientation $\theta$ belongs to the interval $]\pi/2, \pi[$.
- The normal distance $\rho$ of the corridor’s border is minimal.
- The detected corridor’s borders belong to the floor.

This selection is very important because one detection error can generate a poor control position and therefore poor trajectory generation for the vehicle.

4.3. Navigation

To permit its displacement in the corridor, it is necessary that the floor should not be damaged and no obstacles should be present on the path. The CCD camera provides a very detailed description of the robot environment. Therefore, the mobile robot can be in one of the possible situations that is illustrated in Fig. 7.

The observed scene is segmented with the HT and then the corridor borders identification algorithm is started. This algorithm gives the polar coordinates $(\rho_R, \theta_R, \rho_L$ and $\theta_L)$ and the extremities coordinates $(P_{R1}, P_{R2}, P_{L1}$ and $P_{L2})$ of each detected corridor’s border. These coordinates are necessary for the robot’s position calculation and path execution.
By referring from its station, the vehicle follows a path that permits it to move from its station toward the next target station by fixing the end of the corridor (Figs 8 and 9).

The target position \((X_{\text{Dest}}, Y_{\text{Dest}})\) in the robot coordinate system is computed according to the detected corridor’s borders coordinate extremities \((P_{R1}, P_{R2}, P_{L1} \text{ and } P_{L2})\) (Fig. 9). It is formulated by the following expression:

If two corridor’s borders are detected:

\[
X_{\text{Dest}} = \frac{X_{P_{R1}} + X_{P_{R2}} + X_{P_{L1}} + X_{P_{L2}}}{4},
\]

\[
Y_{\text{Dest}} = \frac{Y_{P_{R1}} + Y_{P_{R2}} + Y_{P_{L1}} + Y_{P_{L2}}}{4}.
\]
If one corridor’s border is detected:

\[
X_{\text{Dest}} = \frac{XP_1 + XP_2}{2},
\]
\[
Y_{\text{Dest}} = \frac{YP_1 + YP_2}{2} + (-1)^n L_{\text{arg}},
\]  

(6)

with

\[
n = \begin{cases} 
0 & \text{if the right corridor’s border is detected,} \\
1 & \text{if the left corridor’s border is detected,}
\end{cases}
\]

and $L_{\text{arg}}$ is the half robot thickness.

Three operations are therefore necessary for onboard control of the vehicle:

- A first deviation ($\theta_{D1}$ or $\theta_{G1}$) toward the target station in the corridor.
- The distance $D$ to travel in order to arrive at this target station.
- A second deviation ($\theta_{D2}$ or $\theta_{G2}$) so that the vehicle fixes the corridor’s end.

These operations are calculated from the following equations:

The first deviation $\theta_1$:

\[
\theta_1 = \tan^{-1}\left(\frac{|Y_{\text{Dest}}|}{X_{\text{Dest}}}\right) = \begin{cases} 
\theta_{G1} & \text{Left deviation if } Y_{\text{Dest}} > 0, \\
\theta_{D1} & \text{Right deviation if } Y_{\text{Dest}} < 0.
\end{cases}
\]

(7)

The second deviation $\theta_2$:

\[
\theta_2 = \begin{cases} 
\theta_{G2} & \text{Left deviation,} \\
\theta_{D2} & \text{Right deviation.}
\end{cases}
\]
If only the right corridor’s border \((\rho_A, \theta_A)\) is detected:

\[
\begin{align*}
\theta_{G2} &= \begin{cases} 
\theta_A - \theta_{G1} & \text{if } \theta_{G1} \leq \theta_A, \\
\theta_A + \theta_{D1} & \text{if } \theta_{D1} \leq \theta_A,
\end{cases} \\
\theta_{D2} &= \theta_{G1} - \theta_A & \text{if } \theta_{G1} > \theta_A.
\end{align*}
\]

If only the left corridor’s border \((\rho_B, \theta_B)\) is detected:

\[
\theta_{D2} = \begin{cases} 
\theta_{G1} + \pi - \theta_B & \text{if } \theta_{G1} \leq \pi - \theta_B, \\
\pi - \theta_B - \theta_{D1} & \text{if } \theta_{D1} \leq \pi - \theta_B,
\end{cases} \\
\theta_{G2} = \theta_{D1} - \pi + \theta_B & \text{if } \theta_{D1} > \pi - \theta_B.
\]

If both left and right corridor’s borders \((\rho_A, \theta_A)\) and \((\rho_B, \theta_B)\) are detected: Let

\[
\Delta \theta = \frac{\theta_B - \theta_A}{2},
\]

\[
\begin{align*}
\theta_{G2} &= \begin{cases} 
\theta_{D1} + \theta_B - \Delta \theta - \frac{\pi}{2} & \text{if } \theta_B \geq \pi - \theta_A, \\
\theta_{D1} + \theta_A + \Delta \theta - \frac{\pi}{2} & \text{if } \theta_B < \pi - \theta_A \text{ and } \left(\frac{\pi}{2} - \theta_A - \Delta \theta\right) < \theta_{D1}, \\
\theta_B - \theta_{G1} - \Delta \theta - \frac{\pi}{2} & \text{if } \theta_B \geq \pi - \theta_A \text{ and } \left(\theta_B - \frac{\pi}{2} - \Delta \theta\right) \geq \theta_{G1},
\end{cases} \\
\theta_{D2} &= \begin{cases} 
\frac{\pi}{2} - \theta_A + \theta_{D1} - \Delta \theta & \text{if } \theta_B < \pi - \theta_A \text{ and } \left(\frac{\pi}{2} - \theta_A - \Delta \theta\right) \geq \theta_{D1}, \\
\theta_{G1} - \theta_B + \Delta \theta + \frac{\pi}{2} & \text{if } \theta_B \geq \pi - \theta_A \text{ and } \left(\theta_B - \frac{\pi}{2} - \Delta \theta\right) < \theta_{G1}, \\
\theta_{G1} - \theta_A - \Delta \theta + \frac{\pi}{2} & \text{if } \theta_B < \pi - \theta_A.
\end{cases}
\end{align*}
\]

Once these three operations are executed, a new corridor’s image acquisition is made and then segmented. After identification of the corridor’s borders, the CESA vehicle localizes itself again and then follows a path provided by the off-board part of the system which always permits to it to stay at the middle of the corridor (robot’s consign position). The cycle restarts until the CESA vehicle arrives at its final station.

5. EXPERIMENTAL RESULTS

The CESA vehicle localization and navigation procedures outlined in Sections 3 and 4 have been implemented on a computer-aided design workstation. The program is written in C language and runs on a Sun Sparc station. The current version of the program requires as input data the state information from the vehicle. It produces as output the run operations downloaded via radio waves to the onboard real-time
Table 1. The CESA navigation experimental results

<table>
<thead>
<tr>
<th>Image</th>
<th>The polar coordinates of the detected borders</th>
<th>The course $D$ (mm)</th>
<th>First deviation $\theta_1$</th>
<th>Second deviation $\theta_2$</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Right $(\rho_R, \theta_R)$</td>
<td>Left $(\rho_L, \theta_L)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Res1</td>
<td>98, 50$^\circ$</td>
<td>117, 130$^\circ$</td>
<td>7168</td>
<td>1.17$^\circ$ on the left</td>
<td>1.66$^\circ$ on the right</td>
</tr>
<tr>
<td>Resd1</td>
<td>72, 50$^\circ$</td>
<td>125, 120$^\circ$</td>
<td>7314</td>
<td>2.68$^\circ$ on the left</td>
<td>7.68$^\circ$ on the right</td>
</tr>
<tr>
<td>Resg1</td>
<td>115, 60$^\circ$</td>
<td>100, 140$^\circ$</td>
<td>7180</td>
<td>0.78$^\circ$ on the right</td>
<td>10.78$^\circ$ on the left</td>
</tr>
<tr>
<td>Resg3</td>
<td>—</td>
<td>-10, 120$^\circ$</td>
<td>7136</td>
<td>7.65$^\circ$ on the right</td>
<td>52.35$^\circ$ on the right</td>
</tr>
</tbody>
</table>

system. The diagnostic data from the vehicle may be downloaded to the supervisory controller for analysis.

To illustrate the benefits of our approach, we have tested these procedures in CESA navigation in our laboratory’s corridor. The vehicle moves with a velocity of approximately 0.48 m/s within an area of about 1.2 × 20 m. The vision system output image consists of 464 × 256 pixels and two gray levels/pixels. It is then segmented by the HT with 10$^\circ$ step sizes for the $\theta$ dimension and one pixel for the $\rho$ dimension.

Table 1 illustrates the experimental results for four corridor station images (Figs 10–13) taken at four different stations in the corridor. The results have been executed with the following parameters:

- The focal distance of the camera, $\lambda = 25$.
- The camera slant angle that makes with the horizontal, $\alpha = 9^\circ$.
- The vertical distance between the camera and the ground, $d = 550$.
- The column number of the screen, $r_x = 464$.
- The row number of the screen, $r_y = 256$.
- The component of the vector active from the reference point of the mobile robot to the image plan center of the camera, $(dx, dy, dz) = (317, 0, 240)$.

In the Figs 10–13, we represented the detected corridor’s borders by the full segments, the target station in the corridor by a small square and the orientation that the vehicle must have once at this target station by a small triangle.

During this experience, the mobile robot is completely autonomous, but this autonomy is sensitive both to environment lighting and the corridor’s borders detection algorithm. Several algorithms have been developed for solving this problem: (i) the algorithm developed by [13] to restore the gray levels images to obtain some best edge images and best segmentation, and (ii) the algorithm
Figure 10. Image taken by the camera when CESA is in the middle of the corridor.

Figure 11. Image taken by the camera when CESA is in the middle and near to the right wall of the corridor.

developed by [6, 14, 15] using the stereo matching technique to compute the 3D model of the environment to allow to the mobile robot to control its trajectory and eventually have automatic guidance. These algorithms are now the in the experimentation stage in our laboratory.
Figure 12. Image taken by the camera when CESA is in the middle and near to the left wall of the corridor.

Figure 13. Image taken by the camera when CESA is in the middle and facing the left wall of the corridor.

6. CONCLUSION

Guidance and control procedures to enable a robot vehicle to accurately and autonomously navigate around a structured environment, given a set of operations from the host supervisory controller, have been described. These procedures have been implemented in an experimental robot vehicle ‘CESA’. They permit the vehicle to correct its drift by direct observation using a unique embarked CCD camera.
The rotation/translation parameters are computed taking into account the robot neighborhood parameters. They are obtained by the HT from the environment scene images. The HT is used because it is a well-known method for the detection of parametric curves in binary images, and it was recognized as an important means of searching for objects and features in binary images.

During this procedure, the mobile robot is completely autonomous, but this autonomy is sensitive both to environment lighting and the corridor borders detection algorithm. Several algorithms have been developed for solving to this problem, e.g. the algorithms to restore the observed scene and to compute the 3D model of the mobile robot environment.

REFERENCES

ABOUT THE AUTHORS

A. Oualid Djekoune was born in Algiers (Algeria) in October 1969. He received the Electronic Engineering degree from the Electronic Institute of USTHB, Algiers, in 1993 and the Magister degree in signals and systems in real time from USTHB in 1997. Today, he works as a Researcher in vision equipment in the Robotics and Artificial Intelligence Laboratory of the Development Center of the Advanced Technologies.

Karim Achour was born in Tizi-Ouzou (Algeria) in 1958. He received his PhD degree in Electrical and Computer Engineering from Rennes University (France) in 1987, the MS Degree in Electrical Engineering from ENSM Nantes (France) and the BS at the Polytechnic School in Algiers (Algeria). He is the author of numerous publications for conference proceedings and journals. He has been active in research on perception, computer vision and pattern recognition. He has participated in many conferences as an author, a panel member and a session chairman. He has been a reviewer for many conferences. He was a Head of the Robotics and AI laboratory in the High Research Center in Algiers.