Efficient stereo vision for obstacle detection and AGV Navigation

Rita Cucchiara, Emanuele Perini, Giuliano Pistoni
Dipartimento di Ingegneria dell'informazione, University of Modena and Reggio Emilia, ITALY, Vision-e srl, ITALY
cucchiara.rita@unimore.it, {perini, pistoni}@vision-e.it

Abstract

The paper presents the design and implementation of a system of autonomous navigation of a mobile robot with stereo vision

1. Introduction

One of the main pools of automation engines in modern industrial warehouses is the automatic transport of the goodies in the environment. AGVs (autonomous guided vehicles) can often represent the most suitable and flexible solution. In the last years, several type of vehicles and mobile robots have been realized and many approaches have been proposed for semi- and fully autonomous navigation.

AGVs generally exploit sensor-based information: as well as lasers, infrared and PIR sensors, cameras are very often exploited with computer vision techniques to provide self-localization, obstacle detection and other navigation tasks. De Souza and Kak in 2002 presented a complete survey for computer vision based autonomous navigation [5]. Here, many proposals are reviewed, which process single or couple of images for map-based, map-building-based and mapless navigation.

In this paper, we address map-based autonomous navigation of mobile robots for industrial applications, based on visual sensors only. The goal is to create an integrated framework, sufficiently efficient to run on standard PCs in real-time, capable of providing self-localization, map correction, navigation and obstacle detection for the autonomous guide of simple AGVs in indoor or outdoor environments.

The proposal fully exploits stereo vision, integrates and improves the state of the art of computer-vision algorithms and exploits the multimedia expansion of Intel instruction set Architecture in order to optimize the software speed.

Using the images captured from a stereo camera, the system calculates the distance between the vehicle and artificial patterns placed either on the floor either on the wall and uses this information to correct the position readed from the wheel encoders. Moreover, self-localization is improved with pattern recognition of painted lines on the floor. Then, the disparity map of the scene in front of the camera is builted and after an operation of “floor removal”, it’s able to localize possible 3D obstacles.

The main novelty of the work is the interleaved integration of different techniques for both localization and obstacle detection, with an efficient optimization, that allows a real-time autonomous navigation of a mobile robot, controlled by a standard PC. Due to the low computational costs, and the limited hardware requirement, the prototype can be suitably adopted in many industrial applications.

The paper is structured as follows. Section 2 reports a brief list of related works in the topic of Automated Guided Vehicle. Section 3 describes the system proposed and the implemented computer-vision techniques. Section 4 describes the improvements of the algorithms and the provided system tests; conclusions follow in Section 5.

2. Related works

The general challenges of mobile vehicle navigation can be summarized with three questions: where am I, where am I going and how should I get there [1]; in order to answer these questions, the system should know the output of a lot of sensors and mix them to obtain the most exactly position in the environment [5]. Some sensors, such as lasers, cannot be used outdoor or can be dangerous for humans. Others have a low precision. Visual sensors, such as cameras in visible spectrum, have a wide usability and flexibility. Therefore, computer vision techniques applied to images acquired in real time by cameras mounted on mobile robots could give an efficient solution to the fore mentioned questions.

Vision based navigation is spreading in industrial applications; nevertheless, up to now, the high complexity of the algorithms and the severe computational costs limited de facto a wider diffusion.

In this topic two main groups of techniques can be devised: the relative position measurements, and the absolute position measurements; in the first group, the
odometry-based techniques [2,3] are mostly adopted, where two or more encoders measure the wheel rotation and/or steering orientation. Some system use also inertial navigation, like gyroscopes and accelerometers to measure the rate of rotation and acceleration. These methods can be affected by errors and imprecision, which increase during the time and often can derive.

In the second group, sensor-based techniques that provide absolute localization are accounted: for instance, active beacons [4] compute the position of the vehicle from measuring the direction of incidence of three or more actively transmitted beacons; model matching approaches [6][17], compare the information acquired from the onboard sensors to the information stored in the map of the environment to estimate the vehicle’s absolute position.

In addition, there are techniques based on natural or artificial landmark recognition where distinctive features or artificial patterns are mapped in a known location in the environment; laser range finders [7], or cameras [8] are exploited. The natural landmarks are distinctive features present in the environment, detectable without the need of initial setting. They must be known in advance. Sometimes, very simple features are used and they can be easily confused each other; instead the main advantage of the artificial landmarks is that they can be designed for optimal detecting even under adverse environmental condition. An example of artificial landmark detection is given in [11]. We use an improved approach, based on [11], to provide the so-called incremental localization [5]: the integration of pattern and landmark detection corrects an absolute position computed with odometry.

Another problem of mobile vehicle navigation is the obstacle detection; the approaches proposed in literature can be divided in two groups independently from the used sensors: the former is based on the 3D reconstruction of the environment [9] and the latter is based on homography techniques [10]. In [10] to detect possible obstacle on the road, the homography is followed by a line detector (straight or curved) to recognize the road lines and a simple and efficient obstacle detection with an algorithm called “Inverse Perspective Mapping” (IPM) using a couple of images. In this work, we use homography and line detection as in [10]. Conversely, IPM is substituted with a more complex obstacle detection method, since IPM is not suitable in indoor or small environment where i) many obstacles could be present at the same time or ii) some obstacles cannot be completely visible by both cameras or iii) apparent obstacles such as the walls of open doors can be detected. In the matter of fact IPM does not detect obstacle but only the boundaries of disparity between floor and obstacles. An obstacle is detected by coupling the two peaks of disparity. In previous conditions, the peaks coupling is not always straightforward. For this reason, IPM can be very useful only in some specific cases with single large obstacle, such as a car in front of the autonomous moving vehicle.

In [15], a proposal for an autonomous robot using disparity map is proposed. One of the purpose of this work is to remove the floor form the calculated disparity map. This allows the detection of the obstacle map in front of the vehicle after “floor removal”. We adopt a proposal similar to [15] integrated with the line detection [10] and the landmark detection [11] to make self localization and obstacle detection efficient and robust.

3. The proposal

As introduced, in this project we use basically the odometry to define the relative vehicle’s position on the environment, as in most industrial AGV. Then, stereo vision provides incremental localization and error correction in the position. Self-localization joints two techniques: artificial line detection and self landmarks recognition. The results of line detector correct the absolute angle of vehicle. The landmarks positions correct the absolute coordinates. Line and landmark detection are interleaved to improve the speed of the system. Conversely, at each frame, a fast stereo disparity map algorithm [14] [12] is used: after a floor removal operation, the relative position of obstacles over the vehicle’s path can be computed. Fig. 1 gives an overview of the computer vision techniques adopted.

3.1 Epipolar calibration
Stereo visioning requires the pair of stereo cameras to be registered, such that corresponding epipolar lines are identified. Classical epipolar geometry is recalled the picture of Fig. 2. Registration needs a calibration procedure: an efficient method is described in [13]. In our application, the cameras are pre-calibrated by manufacturer and the epipolar calibration parameter are stored in the flash memory of the cameras. The software driver provide calibration during the acquiring procedure.

![Epipolar geometry](image)

**Figure 2: Epipolar geometry**

### 3.2 Self similar landmarks recognition

A function \( f : \mathbb{R}^+ \rightarrow \mathbb{R} \) is \( p \)-similar for a scale factor if

\[
p, 0 < p < 1, \text{ if } f(x) = f(px) \quad \forall x > 0
\]  

(1)

Self-similar intensity patterns, generated by (1) with a fixed \( p \), are attractive for landmark recognition since the property of \( p \)-similarity is invariant to scale: this makes the recognition robust and insensitive to variable distance between cameras and patterns [11].

The periodic “square wave” function \( S \)

\[
S(x) = \begin{cases} 
0, & x - [x] < \frac{1}{2} \\
1, & x - [x] \geq \frac{1}{2}
\end{cases} = \lfloor 2(x - [x]) \rfloor
\]  

(2)

has the propriety that \( S(x+1) = S(x) \) and \( S(x + \frac{1}{2}) = 1 - S(x) \) i.e, it is a periodic function that is similar under a translation of 1 and anti-similar under a translation of \( p \).

It’s possible to transform \( S \) into a \( p \)-similar, \( \sqrt{p} \)-antisimilar function \( s_p \) by substituting \( \log_p x \) for \( x \).

\[
s_p(x) = S(\log_p x) = \lfloor 2(\log_p x - \lfloor \log_p x \rfloor) \rfloor
\]  

(3)

In Fig.3 is indicated an example of Self-similar pattern generated using the function (3).

![Self-similar Landmark](image)

**Figure 3: Self-similar Landmark**

If at least three landmarks are visible in both images captured from the stereo camera as in Fig. 4, the depicted distances D1, D2 and D3 can be calculated, using the information available with epipolar geometry.

The algorithm detects all possible landmarks that have a sufficient size not to be confused with other natural or artificial patterns.

Using the information about the position of the artificial landmarks in the space, the angles \( \alpha, \beta \) of Fig.4 can be derived and the absolute position of the vehicle on the map can be computed.

![Landmarks triangulation](image)

**Figure 4: Landmarks triangulation**

This computed position is affected by an intrinsic stereo camera error (see Fig 5) This error depends on camera-landmark distance and many other factors like resolution, focal length and base-length. Accordingly, the correction of the position of the vehicle with this absolute localization, w.r.t. the relative position
computed with odometry, is acceptable only if the delta correction is less than the error possible at that distance.

In many cases, just only one landmark could be visible. Therefore, we integrate this information with another measure computed by searching linear patterns on the floor.

In standard vehicle application it's possible to assume that the disparity decreases from bottom to the top image. In most of common situation objects lie to the floor and this assumption permits to restrict the minimum and maximum disparity allowed, depending to the y value of the image.

The coupled adoption of landmarks and lines detectors make the system particularly robust and flexible. During autonomous navigation, either lines, landmarks or both can be detected to provide localization.

### 3.5 Disparity Map

The Census algorithm is a stereo correspondence algorithm implemented in our system: it is described in detail in [12]. The Census algorithm is chosen because it provides a robust, fast a non-parametric dense disparity map. Rather that parametrically comparing intensity value across images, the Census algorithm performs a transform on the input images based on intra-image comparison and then use the transformed images to determine the correspondence. The first step of the algorithm is to apply the Census transform to the left and right image. Given these images, the best matching pixel $P'$ for a given pixel $P$ is determined to be one of the minimal accumulated Hamming distance between $P$ and $P'$.

In order to restrict the range of disparity value, the hypothesis that the maximum disparity decreases proportionally from bottom to top image is stated. This permitted to reduce the noise of the disparity map, or to have the same noise level using an half-resolution disparity map.

### 3.6 Obstacle Detection

In our approach of obstacle detection, “obstacles” are defined as all those blobs sufficiently large that don't lie to the floor. To estimate the floor and remove it from the disparity map, the information from the y direction is exploited: in the matter of fact, the disparity of the floor increases linearly. This consideration allows a labelling of increasing disparity and the floor segmentation. Segmented floor can be subtracted to the disparity map to have only obstacles.

After a blob segmentation, it is possible to assign a disparity and a depth to each object. For the nearest
object on the trajectory of vehicle, the time to collision is computed. Fig. 6 describes two examples of obstacles detected by the mobile robot.

![Figure 6: Obstacle detection](image)

**4. Test**

The proposed system was tested with a Bumblebee Stereo camera mounted on a Pioneer 3DX robot, using a PC with a 3.2 GHz P4 Intel Processor.

The Self-similar pattern recognition algorithm of [11] has been modified, to allow the detection in wide range of scale. This is obtained by considering a search window of variable dimension.

In this manner landmarks can be placed on floor as well as on the walls. This is very important in industrial environment where walls are not always visible. The code has been improved using the SSE3 P4 Intel instruction set. The SSE3 optimization permits a valuable computational time reduction, or having the same time available as in [11], allows an improvement in the global accuracy with a higher stereo resolution and a higher number of scanlines.

With this improvement, using a 640x480 image, the system finds correctly all patterns in a range of distance from 0.8 to 6 meters. This range depends on the camera placement and the camera orientation: as shown in Fig.8 camera is mounted at less than a fifty centimetres to the floor.

To test the system, we have programmed the robot guide in a simple path that the vehicle repeats three times: the path is about 2x2 meters. Correction is carried out with an association of the angle computed with line detection and the detection of a single self-similar landmark.

In Fig 7 is possible to see the difference between the path covered with the proposed system correction (left) and the path covered using the common odometry position system only (right) is indicated. After 3 cycles the angular difference is 17.1 degree and the position difference is about 26 cm.

![Figure 7: Comparison between corrected (left) and uncorrected (right) path](image)

The obstacle detection system is been tested using some artificial obstacle placed on the vehicle test path. The system finds correctly obstacle height more than 10 centimetres in a range of distance from 1 to 4 meters.

Table 1 gives a measure of the computational time required, in average for each step of the navigation task. Here, only the computer- vision steps are accounted. The majority of computational costs are caused by the landmark detection: as previously introduced, this part has been optimized for Intel architecture in order to limit the computational time and to have a high precision. Computational times are not strictly proportional to the pixel number: images with one quarter of resolution need about a half of computational time only.

<table>
<thead>
<tr>
<th>Operation (msec)</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>640x480</td>
</tr>
<tr>
<td>1 Grab+Calibration</td>
<td>75</td>
</tr>
<tr>
<td>2 Landmark recognition</td>
<td>160</td>
</tr>
<tr>
<td>3 Obstacle Detection*</td>
<td>140</td>
</tr>
<tr>
<td>4 Line Detection</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 1: Time measures (in msec) * Obstacle detection is made at half resolution.
In the system the obstacle detection are interleaved with landmarks recognition (frame A) and line detector (frame B); the results frames rate are summarized in Table 2. The system provides a fps of about 3fps and 5.5 fps for images of 640x480 and 320x240 resolution, respectively.

<table>
<thead>
<tr>
<th>Frame (fps)</th>
<th>Resolution</th>
<th>640x480</th>
<th>320x240</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame A (1,2,3)</td>
<td>2.66</td>
<td>5.26</td>
<td></td>
</tr>
<tr>
<td>Frame B (1,3,4)</td>
<td>3.77</td>
<td>6.25</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Frame rates

5. Conclusion

In this work we have implemented a working prototype of AGV guided by a standard PC for localization and obstacle detection. It is mainly used in map-based autonomous navigation even if the same techniques can be used for absolute mapless navigation. The prototype is tested for industrial applications, like the autonomous product transport. The project has been funded by Regione Emilia Romagna, Italy PRIITT project in collaboration with ItalVision srl.

6. References

9. R Labayrade, D Aubert, JP Tarel 2002 Real Time Obstacle detection in stereovision on non flat road geometry through v-disparity representation. . Procs. IEEE Intelligent Vehicles Symposium 2002