A Smart vision system for advanced LGV navigation and obstacle detection

Bertozzi Massimo, Bombini Luca, Broggi Alberto and Coati Alessandro[1]

***Abstract*— This article presents the VisLab solution for obstacle detection and navigation support for unmanned vehicles in industrial environments. Although the literature contains many examples to tackle this problem, this solution can be considered innovative as it improves traditional laser-based systems. The proposed system is composed by two sub-systems. The first one is an obstacle detection system, which also allows the detection of hanging obstacles, within a 3D monitored area. This solution outperforms the original laser scanner based system used for safety which was limited to bi-dimensional areas only. Another vision system is used for tracking a guideline on the ground, that solves problems of localizations and drifts that sometimes can happen using the laser and vehicle odometry only.**

After a long testing phase, the system is actually installed in a modern industrial warehouse in Parma in order to finally estimate its robustness and reliability.

I. INTRODUCTION

Laser Guided Vehicles (LGVs) represent one of the main tools in a modern industrial warehouse for automatic goods movement. These vehicles are the most suitable and flexible solution for these kinds of scenarios. This type of unmanned vehicle uses laser beam reflections into the workspace for self-localization through triangulation.

In the last years, LGVs (fig. 1a) have increased their operation capabilities to cover much more complex market requests. For example, in some applications, the area dedicated to LGV navigation is very close to workers or this area is shared by other human driven fork-lifts: in these cases it is necessary to increase the on-board safety systems. In addition, in many other applications, LGVs have to drive into a drive-in rack (fig. 1b), really close to shelves or pallets with a limited support by laser guidance, due to possible occlusions; in this case is necessary to increase the precision in the drive-in operation.

The VisLab solution to these problems is the use of machine vision for detecting hanging obstacles and following lines even in situations where the laser guidance fails. Different kinds of problems have to be taken into account: different light conditions, damaged line on the ground, and moving obstacles with different shapes. Moreover, other several constraints such as hardware costs, real-time performance, and the robustness of the proposed solution have to be considered as well.

[1] represents a previous study based on stereo vision, providing different functionalities: in this case obstacles are detected by building a disparity map of the scene in front of the camera and removing the floor; moreover, navigation is improved by recognizing artificial patterns (at least three) placed into the work area or lines painted on the floor. The proposed approach follows a similar approach to detect hanging obstacles, but supports navigation in a different manner: a Line Detector (LD) is used to independently detect lines in images acquired by two cameras; the two results are then fused to obtain a reliable information. One of the problems in 3D reconstruction techniques based on image disparity is the presence of inconsistent "image to world" projections, due to wrong matches; this usually happens in presence of repetitive patterns or reflections of light on wet floor; [12] and [11] propose different disparity validity metrics that could be used by OD, to reduce faults.

In a different system, LD is obtained using the Hough transform or iterative techniques of model fitting like RANSAC [4]. The proposed approach is based on RANSAC as well; initially it detects line borders and then tries to couple them for line presence validation, unlike [5] that uses a different model to directly detect road line borders.

This paper is organized as follow: section II presents an overview of hardware setup designed for this application, section III describes the navigation algorithms, section IV discusses the stereo vision technique used by detect hanging obstacles while section V provides the system test and result; conclusions follow in section VI.

(*a*) (*b*)

Fig. 1. (a) LGV with stereo system and illuminator on the top of vehicle and safety laser scanner (PLS) in the bottom; (b) drive-in rack path.

[1]are with VisLab, Department of Information Engineering, University of Parma, 43124 Parma, Italy bertozzi, bombini, broggi, coati at vislab.it

(*a*) (*b*)

Fig. 2. Setup of the Vision Systems: (a) the LD system with two different pitchs, (b) the OD stereo head with illuminator.

II. Hardware setup

The whole system has been implemented on an industrial PC. The PC is equipped with a IEEE1394-B firewire board with 4 ports. Stereo images are acquired by a pair of IEEE1394-B cameras equipped with 3.5 mm, 1/3" lenses, while images for line detection come from IEEE1394-A cameras with same lenses. The LGV handles image acquisition independently triggering the two vision systems. The stereo vision system used for OD is mounted on the top of the LGV (fig. 2b), monitoring the below area up to 7 m, while LD cameras are mounted on the side (fig. 2a). Communications between LGV guidance system and the vision systems can take place using Ethernet or exploiting 16 digital IOs available on the industrial PC.

III. Navigation system

The purpose of the LD algorithm presented in this paper is to detect and follow a guideline on the ground, even with the presence of noise like wet floor, line erosions, nuts, screws or other material that can occlude or blur the line (fig. 3). A further advantage brought by this system is the ability to correct the error drift of the vehicle odometry, when laser cannot work as in cluttered areas (see fig. 1b).

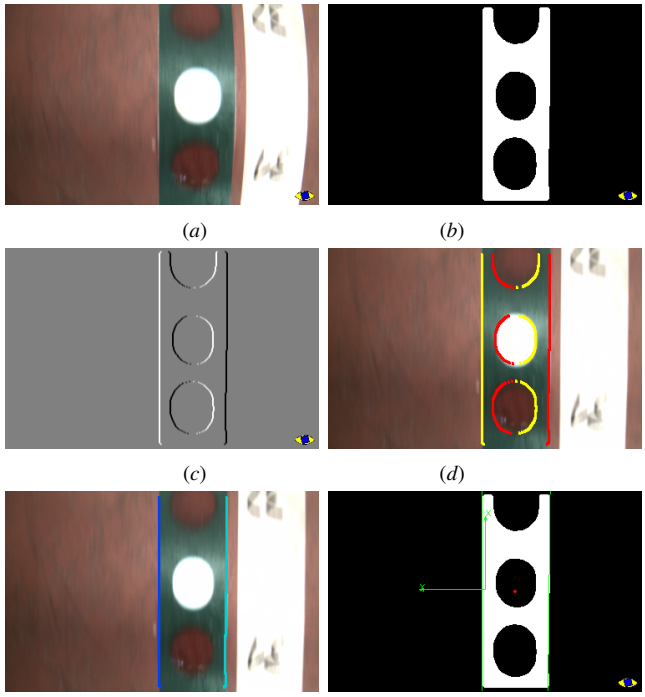
The following details the LD algorithm.

A. Line extraction

The main idea is to detect left and right borders of straight lines into the images and match them with respect to a line model. To do this, the following steps are accomplished:

- *Preprocessing* – A Bayer image at 752×480 pixel resolution is acquired when LGV triggers the camera, and is converted into a color image. To properly detect straight lines, the distortion introduced by optics [2] is removed and then subsampled at half resolution to reduce computation time.
- *Processing* – The color image is filtered for the specific line color and its vertical edges are emphasized. The line model is extracted using an iterative technique, namely RANSAC, that uses the straight line equation (1) in implicit form to retrieve the best fitting. In case of success, the barycenter of the line is estimated with a precision less than 5 mm.

Fig. 3. Example of line detection in presence of noise. The white background doesn't affect the LD results.



(*a*) (*b*) (*c*) (*d*) (*e*) (*f*)

Fig. 4. LD processing: (a) distorted input image, (b) HSV mask, (c) border extraction, (d) edge detection, (e) line edges extraction and (f) barycenter evaluation with representation of world reference system.

$$ax + by + c = 0 \tag{1}$$

The LD algorithm has been designed to adapt easily to detect lines with different width and color. Using a color space transformation from RGB to HSV space [3] it is possible to binarize the input RGB image (fig. 4b), getting rid of all informations that are inconsistent with the searched color. Erosion and expansion are then applied to refine the result.

A fast way to detect vertical edges and, at the same time, discriminating between left and right edges, is to filter the image with a Sobel filter 3×3 with sign (fig. 4c), that considers the type of image transitions: low to high (LH) or HL. The detection of vertical edges only does not allow the detection of horizontal lines because no sufficient points could be used to estimate the line model; this is not a limitation because some constraints are used: this solution is designed to support the LGV during movement into a rack

like in fig.1b where the unmanned vehicle arrives in front of the segment, so orthogonality condition are not expected to be found. Therefore the algorithm can easily estimate the centroid of the line with a simple difference between the two intersections L and R for a predefined row in the image (see fig. 5b).

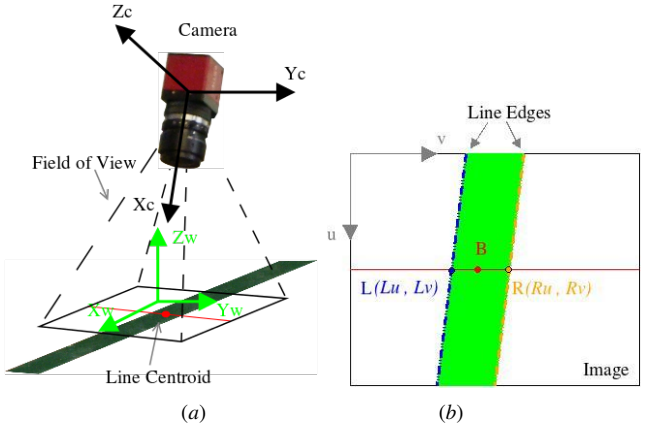


Fig. 5. Reference systems configurations for LD: (*a*) describe line centroid respect world coordinates, (*b*) respect image coordinates.

RANSAC represents an iterative method to estimate the parameters of a model M from a set S of observed data, also in presence of high noise (namely *outlier*). In this case each border represents a set (S_l, S_r) that is processed to extract the coefficients of (1) namely (a_l, b_l, c_l) for the left edge and (a_r, b_r, c_r) for the right edge. This technique does not guarantee to find the absolute optimal model but, if possible, finds a model M_O which is the best fitting among all considered models, in a bounded number of iterations L with respect to the equation (2).

$$L = \frac{\log(p_{fail})}{\log(1 - (p_{good}^N))} \tag{2}$$

Assume N as data items subset of S, from which the parameters of M could be estimated; p_{good} represents the probability of a randomly selected data item being part of a good model and p_{fail} represents the probability that the algorithm will exit without finding a good fit if one exists. L could be find by:

$$\begin{aligned} p_{fail} &= p.\ of\ L\ consecutive\ failures \\ p_{fail} &= (p.\ a\ given\ trial\ is\ a\ failure)^L \\ p_{fail} &= (1\ -\ p.\ a\ given\ trial\ is\ a\ success)^L \end{aligned}$$

$$p_{fail} = (1 - (p_{good})^N)^L \tag{3}$$

Applying the *log* function to both sides of (3) and solving for L it is possible to obtain (2).

The LD uses an horizontal least square fitting for edge reconstruction approach, with a maximum fixed number of iterations equal to 200 and needs the execution of the RANSAC algorithm for two times, one for each edge.

The algorithm fails when one or both edges are not found or the line width exceeds a given interval.

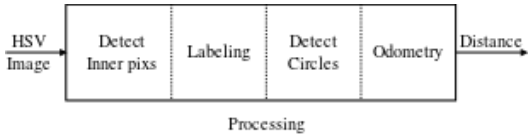


Fig. 6. Odometry Algorithm Steps.

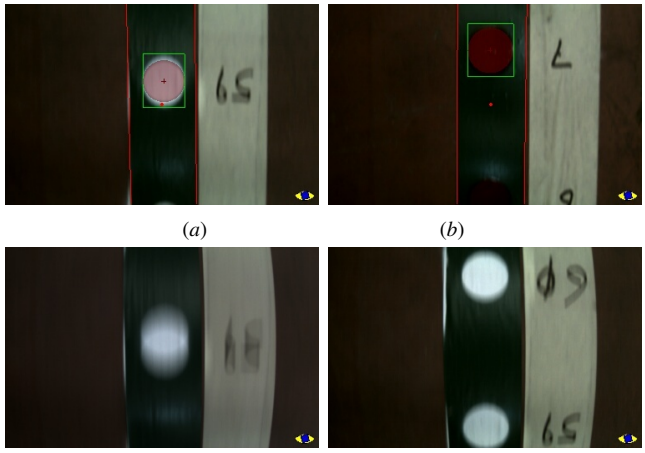


Fig. 7. Examples of circle detection (a-b) and image acquisition (c-d) of different odometry patterns. (c) and (d) show a problem of blured patterns due to too high exposure time respect LGV speed.

B. Visual odometry

The LD system is also used for visual odometry as sketched in fig. 6. In order to enable visual odometry, blobs have been added to lines as reference markers (fig. 7).

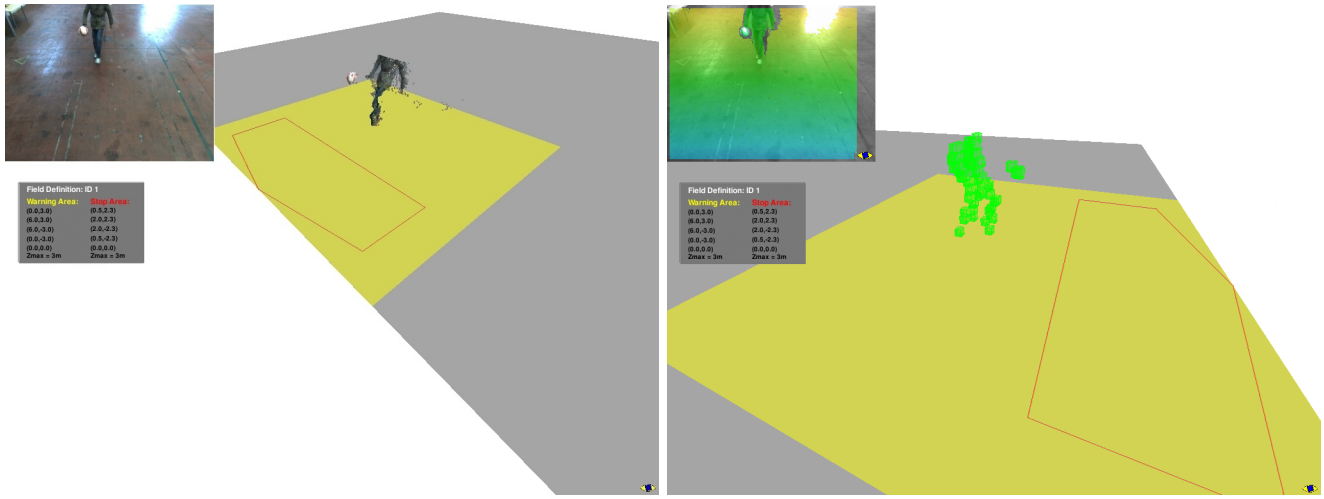
The camera acquires images automatically managing the exposure time and gain, at a 10 Hz rate. Considering a max speed achievable by LGV $v_{max} = 2ms^{-1}$, to prevent capturing of blurred images (fig. 7c-d), shutter exposure has to be set in the $7-9$ ms interval. Also in this case, the illuminator plays an important role when the camera is placed inside the vehicle.

If the LD algorithm detects a valid line, the visual odometry phase can take place. The HSV mask of fig. 4b is used to detect pixels of blobs, considering the complementary part of the HSV mask, inside the line. The interconnected parts of detected labeled blobs are classified and analyzed to spot circular markers: the validation method takes into account the covered area. Odometry is then derived, by counting the number of blobs for each frame.

IV. Obstacle detect system

A limitation of the current security system based on Proximity Laser Scanner (PLS) is the impossibility to detect hanging obstacles, since the laser is installed at a 200 mm height. This limit does not depend on the safety device itself, but by safety regulations that restrict the laser positioning.

The use of passive sensors like cameras could resolve this problem, since it is possible to freely place them without any limitation on position; a stereoscopic vision system mounted



(*a*) (*b*)

Fig. 8. Example of 3D reconstructions of suspended obstacle: (a) hanging obstacle, (b) show a voxel representation of the world and the corresponding disparity image.

on the front of the LGV could then be used to detect suspended obstacles (fig. 8). The approach is based on the search for 3D features; areas that in the 3D world include a sufficiently large number of 3D features, called voxels, are considered as obstacles.

A. Detailed Algorithm Description

Processing takes place according to these steps:

1) Preprocessing: Bayer-patterned images at 656×492 pixel resolution acquired by the cameras are converted to gray-scale and subsambpled at half resolution, then lens distortion is removed and the stereo pair is rectified [2]. A vertical Sobel filter is used to improve the subsequent matching phase [6]. Thanks to the use of the Semi-Global Matching (SGM) approach [7], [10], the disparity image is builded and 3D reconstruction can take place. To enhance the result quality, a smoothing filter is applied.

2) Processing: The main idea is to split the frontal world (7 m deep, 10 m wide and 3 m height area) in many cubes (named voxel) of 100 mm size. The ratio between the number of features and voxels is used to discard voxels that do not present a sufficient number of features. On the basis of these information, the algorithm checks for the status (filled or empty) of two tridimensional polygonal areas, respectively a Warning Area (WA) and a Stop Area (SA) (fig. 8) and alerts the LGV when an obstacle is detected. The union of WA and SA defines the *Field*; the LGV dynamically updates the Field via Ethernet to adapt it to the desidered field of view.

Having to deal with a low cost hardware, but at the same time having to ensure real-time performance, the subsampling of the 3D point cloud coming from the DSI-to-World transformation leads to a double benefit: the number of points to be processed decreases, now substituted by each centroid coordinates of the voxels, and areas with a scarce feature content are discarded and do not require to be processed.

Contiguous voxels can be fused together to form a convex polygon. This step is performed using the PCL library [8], [9]. Also in this case it is possible to take into account obstacles with a volume greater than a certain threshold only.

B. DSI extraction and filtering

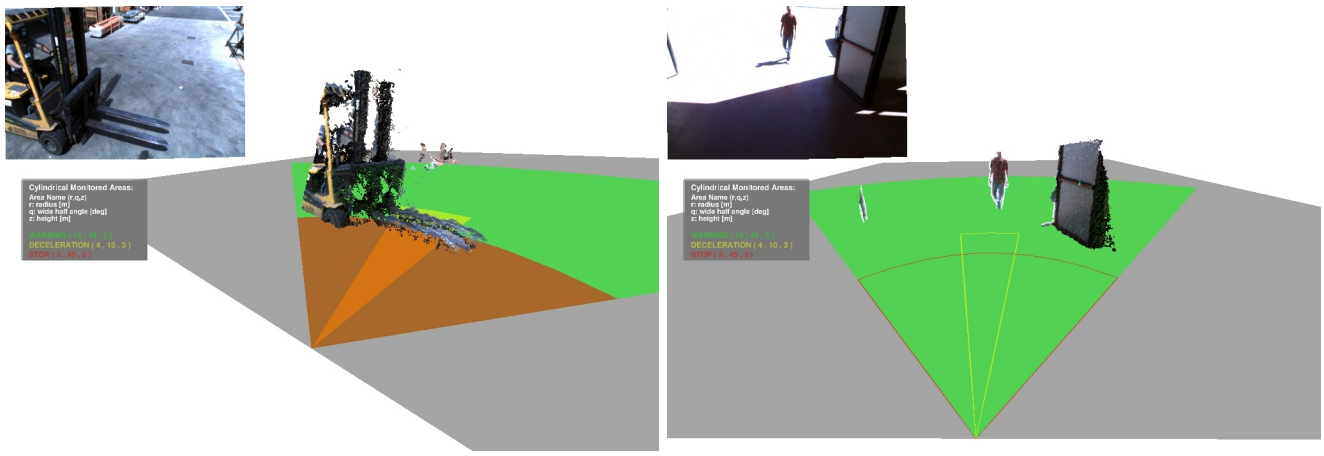
The proposed approach for building DSI images follows the SGM technique [7], [10]. In order to fully exploit the parallel processing capabilities of modern multi-core CPUs and reach real-time frame rates, a multi-threaded SIMD processing scheme has been devised exploiting the XMM CPU capabilities. The most time-consuming step of the SGM algorithm is the path accumulation, since it must be performed for each pixel, disparity and path. To speed up the processing, for each path direction, the pixels are split into several independent slices that are concurrently processed in parallel. Moreover only the accumulated values is saved in memory, while temporary data needed for the incremental processing is kept into the CPU registers[1].

A post-processing filter that analyzes a 3×3 window around each DSI element is used to compare the disparity values to neighborhood ones and to discard *isolated* values. This filter is used to eliminate the noise component of salt and pepper type that occurs in case of poor illumination or occlusions of any of the two stereo cameras.

C. Illumination control

The management of the exposure time of the camera is one of the main problems that must be taken into account in the development of a vision system that must operate in places where there may be different illumination conditions. The camera must be able to handle situations in presence of little or too much light or in quickly passing from a situation

[1]Disparity search ranges of up to 128 are supported; over this threshold not enough XMM CPU registers are available.



(*a*) (*b*)

Fig. 9. Examples of obstacle detection results, considering different situations: (a) hanging obstacles, (b) detection in a saturated area.

to another (fig. 9b). To solve this problem two different solutions are used.

The first one is the ability to control shutter and gain (VPs) range values of the camera, when they can vary in automatic mode thanks to the embedded auto exposure algorithm. To prevent the presence of trails left by moving objects within the scene, and to leave sufficient time for the processing, a limit on the maximum exposure time is used (about 30-35% of trigger period). The gain can vary without restrictions, but in darkness conditions it amplifies the image noise. To cope with this problem the system uses a smooth filter on DSI image.

Moreover, the VPs remain synchronized on the two cameras: each time the master camera acquires a frame with auto exposure algorithm, same parameters are forced on the slave camera.

The second solution is based on the use of an illuminator with high intensity LEDs (as seen in fig. 1a), in order to increase and normalize the quantity of light present in the scene.

V. Tests e results

The industrial PC used as production platform is equipped with an Intel Core™ 2 Duo @ 2.56 GHz processor and 4 GB RAM.

TABLE I
Processing Times for Line Following

Algorithm step	avg [ms]	min [ms]	max [ms]	std [ms]
1xPreproc	30.1	10.1	79.1	12.0
1xProc	23.8	11.7	68.4	9.95
RANSAC	12.1	3.94	44.9	7.52

Tables I and II show the system performance. For the LD algorithm the most time consuming operation is represented by color image reconstruction (DeBayer). In case of success, the barycenter of the line is estimated with a precision less than 5 mm. The line centroid oscillations detected by the frontal camera are shown in fig. 10. This is due to the accuracy of both the LD and the LGV control system to adhere to the ideal trajectory.

TABLE II
Processing Times for Obstacle Detector

Algorithm step	avg [ms]	min [ms]	max [ms]	std [ms]
Preproc	34.9	25.7	55.6	4.69
Proc	3.52	3.4	10.1	1.12
Total	39.3	28.8	64.3	4.79

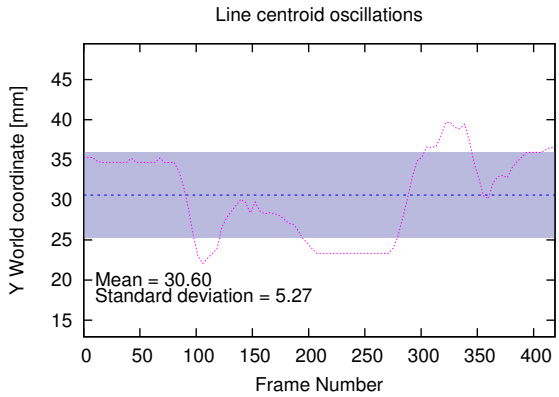


Fig. 10. Example of line centroid oscillations (Y component) detected by the frontal camera, using a red line 2 cm thick. Data are referred to camera world coordinates.

For the OD algorithm the most time consuming operation is represented by the DSI image: in the preprocessing time it accounts for 63% of the time.

To qualitatively evaluate the OD accuracy, the results in presence of different obstacles with different size, shape (ex. boots, ball, boxes, stair, post reflective) or distance (from 1 m up to 6 m) were recorded and compared with laser scanner. Results are shown in Table III.

Another result, shown in Table IV, refers to three different tests of detection repeatability of a dark cube (fig. 11) at three

TABLE III

PERFORMANCE METRICS FOR OBSTACLE DETECTOR

Obstacle Type	Dimensions (WxHxD) [mm]	PLS distance [mm]	OD distance [mm]	Diff. [mm]
Ball	Ray 200	720	725	5
Tire boots	160x380x500	1440	1446	6
Black toolbox	450x380x180	1690	1634	57
Foam cube	500x500x500	2360	2291	69
Carton box	350x300x300	4350	4333	17
Reflective Cylinder	Ray 50 H 700	5320	5334	14

Fig. 11. Example of dark cube used in breaking tests. The white lines on the floor representing the LGV stop positions in different tests.

different speeds. With 2 ms^{-1} it is necessary to enlarge the Stop Area of a 67% factor for collision avoidance.

TABLE IV

PERFORMANCE METRICS FOR OBSTACLE DETECTOR IN BRAKING PHASE

speed [m/s]	*Warning Area* [mm]	*Stop Area* [mm]	*Min Impact Distance* [mm]	*Braking Range Distance* [mm]
1	3000	1500	1360	40
1,5	3000	1500	310	50
2	3000	2500	130	100

Fig. 12 shows different types of noise rejected by the LD algorithm, while as we seen fig. 8 and fig. 9 shows different light conditions and 3D reconstructions of detected obstacles in different monitored areas.

VI. CONCLUSIONS AND FUTURE WORKS

The line following and obstacle detection system presented in this paper were successfully employed during a real-time experimentation in an industrial environment. The approach presented can be applied into several dynamic scenarios diversified by illumination condition and moving obstacles.

The line detection approach for drive-in rack operations could be improved using a different line model in RANSAC for fitting the line instead of line borders. The stereo vision approach for obstacle detection increases the safety of LGV, being able to detect obstacles that are missed by the laser scanner based systems. The robustness of the devices was tested for a long time obtaining successful results. This work represents the basis and the prerequisite for a classification of detected obstacles to increase LGV safety and efficiency because it could adapt speed in presence of close workers or when another LGV is detected nearby.

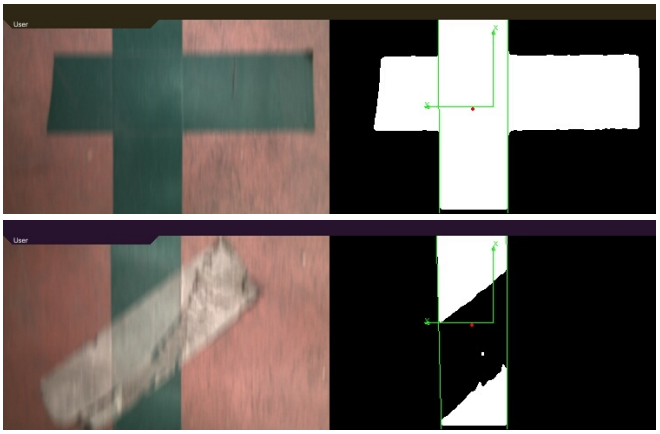


Fig. 12. Example of line detection in presence of noise.

ACKNOWLEDGMENT

This work has been supported by Elettric80 s.p.a.. This company is a global provider of end-of-line automation solutions which increase operational profitability and efficiency. VisLab would like to thank Elettric80 for its support during all the experimentation phase providing the LGV and the experimentation infrastructure, as well as hints and suggestions on the scenarios.

REFERENCES

[1] Rita Cucchiara, Emanuele Perini, Giuliano Pistoni, *Efficient Stereo Vision Obstacle Detection and AGV Navigation*, 14th International Conference on Image Analysis and Processing (ICIAP 2007).

[2] Frédéric Devernay, Oliver Faugeras, *Straight line have to be straight*, Machine Vision and Applications, vol 13, pp.14-24, 2001.

[3] Max K. Agoston (2005). *Computer Graphics and Geometric Modeling: Implementation and Algorithms*. London: Springer. ISBN 1-85233-818-0. pp. 300306.

[4] Martin A. Fischler, Robert C. Bolles, *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*, Volume 24 Issue 6, June 1981 ACM New York, NY, USA.

[5] D. Fontanelli, M.Cappelletti, D. Macii, *A RANSAC-based Fast Road Line Detection Algorithm for High-speed Wheeled Vehicles*, pp. 1-6, Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE.

[6] H. Hirschmüller, S. Gherig, *Stereo matching in the presence of sub-pixel calibration errors*, in Intl. Conf. on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 437-444.

[7] H. Hirschmüller, *Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information*, in Intl. Conf. on Computer Vision and Pattern Recognition, vol. 2, San Diego, CA, USA: IEEE Computer Society, June 2005, pp. 807-814.

[8] Radu Bogdan Rusu, Steve Cousins, *3D is here: Point Cloud Library (PCL)*, ICRA 2011, pp.1-4, Willow Garage, Menlo Park, CA, USA.

[9] Radu Bogdan Rusu, *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*, PhD Thesis, Technischen Universität München eingereicht.

[10] Alberto Broggi, Michele Buzzoni, Mirko Felisa, Paolo Zani, *Stereo obstacle detection in challenging environments: the VIAC experience*, ITSC 2011.

[11] Gijs Dubbelman, Wannes van der Mark, Johan C. van den Heuvel, Frans C.A. Groen, *Obstacle Detection during Day and Night Conditions using Stereo Vision*, IROS 2007.

[12] Alberto Broggi, Stefano Cattani, Elena Cardarelli, Brad Kriel, Michael S. McDaniel, Hong Chang, *Disparity Space Image's Features Analysis for Error Prediction of a Stereo Obstacle Detector for Heavy Duty Vehicles*, ITSC 2011.