THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 18, Number 3/2017, pp. 265–272

AN OMNIDIRECTIONAL STEREO SYSTEM FOR LOGISTIC PLANTS. PART 2: STEREO RECONSTRUCTION AND OBSTACLE DETECTION USING DIGITAL ELEVATION MAPS

Marius DRULEA, Andrei VATAVU, Szilárd MANDICI, Sergiu NEDEVSCHI

Technical University of Cluj-Napoca, Faculty of Automation and Computer Science, Romania
Corresponding author Sergiu Nedevschi, E-mail: sergiu.nedevschi@cs.utcluj.ro

Abstract. This work presents a fisheye lens based omnidirectional stereo sensor. The sensor is installed on the top of an automated forklift. In the first part of our work we have already presented the system overview, the fisheye lenses description and their calibration. The multi-channel rectification led to three pairs of perspective stereo images. By means of a standard reconstruction engine, three disparity images and three associated 3D point sets were generated. In this second part, we present the stereo reconstruction and obstacle detection from the calculated 3D information. The 3D point clouds are projected onto a digital elevation map. Based on this map, we estimate the ground plane. We detect obstacles by labeling and grouping the cells having the height above the calculated ground plane. We have evaluated the obstacle detection rate of the system, the obstacle localization accuracy and the obstacle size estimation accuracy.

Key words: fisheye lens, stereo reconstruction, digital elevation map.

1. INTRODUCTION

Processing raw dense stereo data is a challenging task because of the large volume of information. In order to address the real-time requirements, various solutions were proposed to reduce the amount of stereo data, on the one hand, and to ensure high accuracy and efficiency on the other hand. The existing methods can be split into two main categories. First category of solutions consists in performing the processing tasks in the image space by using the color and disparity information [1, 2, 3]. For example, the authors in [2] compute the projection of disparity information into a so called “V-Disparity” space. The “V-Disparity” image is used to estimate the ground plane and detect the objects above the ground. In [3], the disparity information is used to segment the free-space and detect a set of rectangular vertical entities, called stixels. The second category of solutions implies working directly in 3D. Usually, the 3D information is transformed into more compact grid-based data structures such as Digital Elevation Maps [4] or Occupancy Grids [5], [6]. Occupancy grid based representations were first introduced by Elfes [7] in the context of sonar-based mapping and navigation. However, nowadays, occupancy grids are widely used for various tasks such as object detection and tracking, path planning, sensor fusion and simultaneous localization and mapping. Digital Elevation Maps can be regarded as an improved grid-based representation where, besides the occupancy value, each cell is also described by its height information. Unlike the other environment modeling solutions, this type of intermediate representation is more suitable for crowded environments. The resulting compact 2.5D model can be easily used by the subsequent processing steps that need both high accuracy and high performance.

In our project we use a digital elevation map in order to compactly represent the 3D world. In the first part of the paper [8] we have divided the fisheye stereo images into three pairs of rectified images. We call them channels. For each channel we have generated three disparity images and the associated 3D point clouds. We use a single grid of elevations and not independent grids for each channel. The 3D points from all the channels are therefore projected onto the common grid. We then estimate the ground plane using the height information stored in the grid cells. In this step, we use a RANSAC approach followed by a least-squares refinement. The grid cells are classified as ground or object cells based on the distance to the ground plane. The obstacle cells are finally clustered together to form the obstacles in the scene.

The rest of the paper is organized as follows. Section 2 presents the stereo reconstruction algorithm. Section 3 presents our grid representation. In Section 4 the ground plane is estimated using RANSAC and it is further refined in Section 5. Section 6 presents the classification of the cells as obstacles or road cells and Section 7 presents the detection of the obstacles in the grid. The experimental evaluations of the system are presented in Section 8. Section 9 concludes the paper.

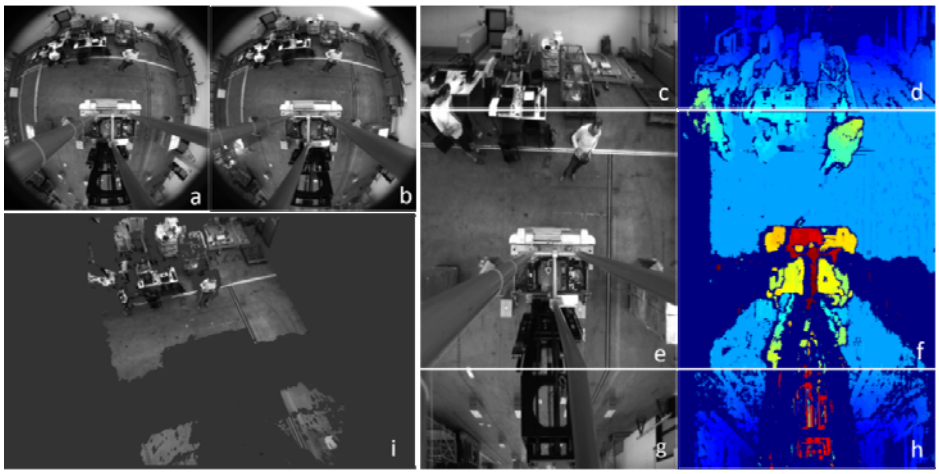

Fig. 1 – a) Left fisheye image; b) right fisheye image; c), e), g) the division of the fisheye image into three channels: the front, central and back rectified images; d), f), h) the disparities for each left-right pair of rectified images; i) the common 3D view of all the channels.

2. STEREO RECONSTRUCTION ALGORITHM

For each rectified pair of images provided by the multi-channel rectification module, we apply the stereo matching and reconstruction engine described in [9–11]. We have chosen this algorithm because it was successfully applied in a driving assistance application and because of its real-time performance on CUDA enabled GPUs. We have tuned the parameters of the algorithm to our setup. Figure 1 d), f), h) shows the disparity images for the rectified pairs given as example.

The stereo matching algorithm [10] is a SGM stereo engine and uses the census transform in the matching cost. The census transform is applied over a window of 9x9, but only 16 pixels are selected for the census signature. A 2-bit census is used and therefore the census signature is 32 bits in size. The matching function between two census signatures is the hamming distance. The matching costs are aggregated over a 5×5 window. A modified semi-global matching technique is applied over the cost volume. The modified SGM algorithm penalizes large disparity discontinuities using a gradient based adaptive penalty. The classical penalty of small disparities transition is no longer used. The best disparity is then selected using a winner-takes-all (WTA) strategy with confidence threshold. The confidence measure is defined as the ratio between the best and the third cost. In order to further filter the wrong disparities, the algorithm performs a left-right consistency check of disparities. In order to estimate the disparities with sub-pixel accuracy, an interpolation function adapted to this particular stereo algorithm is used [11].

Each disparity image channel gives a cloud of 3D points. The 3D points are computed w.r.t. the coordinate system of the virtual imagers. With respect to the optical axis of the sensor, the front, central and back imagers are rotated by 90^o, 0^o and -90^o respectively. For convenience, we define the "approximate local coordinate system of the AGV" as follows: this system is parallel with the front virtual imager and it is approximately on the ground. The exact local coordinate of the AGV is parallel to the ground. Our defined system of coordinates is an approximation because the stereo sensor can be tilted, its optical axis is only approximately perpendicular to the ground. The height of the camera w.r.t. to the ground is only approximately known. Moreover, the poles sustaining the stereo sensor oscillate while the vehicle moves. We rotate and translate all clouds of points into this approximate local coordinate system of the AGV. The unified 3D scene for the selected scenario is shown in Figure 1 i). We use this transformation simply to have a unified space for all the 3D points. The actual ground plane is calculated in the next chapter.

3. GRID BASED REPRESENTATION

We use a grid of 0.1m × 0.1m cells. The size of the grid is 165 × 70, which covers the 16.5 m × 7 m area around the AGV, as depicted in [8], Fig. 2. Each cell m_{xz} has a position *(x,z)*, which is the coordinate of the cell in the grid. For each cell we store the number N_{xz} of 3D points in the cell, the minimum height Y_{min}^{xz} and the maximum height Y_{max}^{xz} of the points that fall in the given cell. $P_{min}^{xz} = (X, Y_{min}^{xz}, Z)$ represents the 3D point corresponding to the minimum height Y_{min}^{xz} and $P_{max}^{xz} = (X', Y_{max}^{xz}, Z')$ is the 3D point corresponding to the maximum height Y_{max}^{xz}. We denote the type of the cell with c_{xz}, where $c_{xz} \in \{Object, Ground\}$. With these notations, the cell is represented as:

$$m_{xz} = (x, z, N_{xz}, Y_{min}^{xz} Y_{max}^{xz}, P_{min}^{xz}, P_{max}^{xz}, c_{xz}) \tag{1}$$

For each 3D point *P(X,Y,Z)*, the corresponding grid cell *(x,z)* position is computed by scaling down its lateral *X* and longitudinal *Z* coordinates. The minimum height values Y_{min}^{xz} and the associated points P_{min}^{xz} are used as the point hypotheses for ground-plane model computation, while the maximum height values Y_{max}^{xz} describe the elevation of the associated grid cell.

In order to detect the ground plane we use a RANSAC (RANdom SAmple Consensus) approach combined with a Least-Squares refinement step. The approach is described in the following chapters.

4. PLANE ESTIMATION USING RANSAC

The RANSAC algorithm is a robust method for fitting a given model to a set of noisy measurements containing outliers. The least-square minimization alone is sensitive to outliers. Even a single incorrect reconstructed 3D point can introduce a considerable bias in the estimated result. Instead of directly computing the model parameters by using the whole measurement space, the RANSAC technique iteratively generate random samples by selecting subsets of the original observation. At each iteration, the selected hypothetical inliers are used to instantiate a candidate model. Then all the other observations are tested against the candidate model and a score is computed. The sample with the highest score (the maximum number of inliers) is saved.

In order to estimate the ground surface we use a plane model defined by the following general equation:

$$A'x + B'y + C'z = D' \tag{2}$$

The plane equation (2) can be written in the form of three independent parameters:

$$y = -Ax - Cz + D \tag{3}$$

where $A = A'/B', C = C'/B', D = D'/B'$.

The plane parameters are computed in our case by using the RANSAC technique. We select the cells with the minimum height values $Y_{\min}$ within a predefined range. The corresponding P_{min} 3D points of the selected cells are the inputs to our RANSAC procedure. We denote this set as

$$S = \{P_{min}^{xz} \mid -h_1 \le Y_{min}^{xz} \le h_2\} \tag{4}$$

The parameters h_1 and h_2 depends on the cameras height w.r.t. to the ground, which is approximately known. We relax these parameters to enable $\pm$ 60 cm inaccuracy in the initial estimation of the camera position. The exact position is known after the calculation of the ground plane.

The RANSAC algorithm for the ground plane model iteratively repeats the following steps until a satisfactory solution is found or up to a maximum number of iterations. We have set the maximum number of iterations to 200.

Sampling. This step is a random selection of three points from the set S.

Plane estimation. We estimate the ground plane parameters from the three sampled points. The estimated parameters form a ground plane candidate.

Estimate the consensus set. The distances of all 3D points in the set S to the candidate plane model are computed. The points with a distance less than a given threshold are selected as inliers. These points form the consensus set. We have set the distance threshold to 10 cm.

Update. In this step, the obtained number of inliers is compared with the number of inliers of the best consensus set obtained so far. If the current number of inliers is higher, the current plane parameters and the current consensus set become the best ones. In addition, if at least 80% in the set S are inliers of the current plane hypothesis, the solution is considered satisfactory and the algorithm stops. Otherwise, it proceeds with the next iteration.

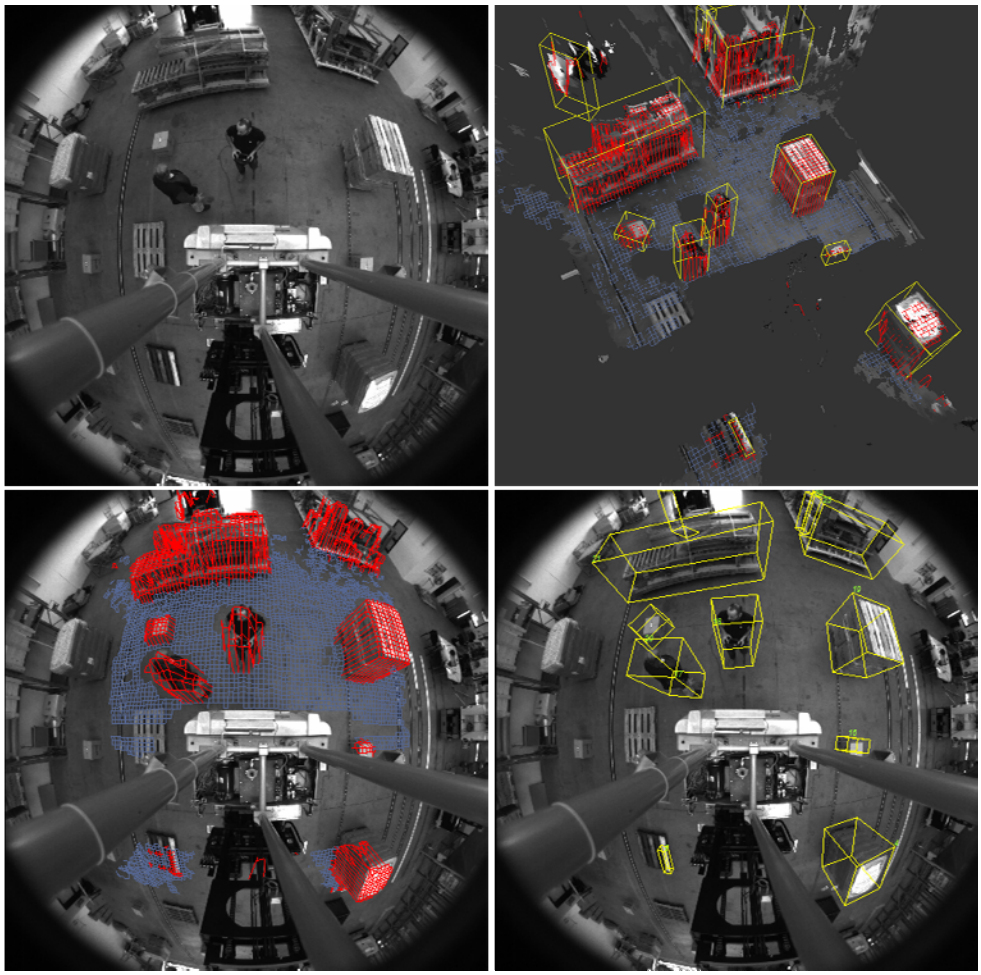

Fig. 2 – *Top left*: the original left fisheye image. *Top right*: the combined 3D scene of all the rectified channels and the elevation map. The cells in blue represents the ground cells, the cells in red represent obstacle cells. The clusters of red cells form the objects, represented as bounding boxes. *Bottom left*: the elevation map projected onto the original image. *Bottom right*: the detected obstacles in the original left image.

5. LEAST-SQUARES PLANE REFINEMENT

The final consensus set $CS \subset S$, provided by the RANSAC approach represents a collection of inliers of a plane hypothesis.

$$CS \subset S, CS = \{p_i(x_i, y_i, z_i) | i = 1..N\} \tag{4}$$

We can improve the result if we further fit a plane over the inliers. We estimate a refined plane model by applying a least-squares minimization. Considering the plane equation described by 0 we define the following objective function over the points in the consensus set CS :

$$\varepsilon(A,C,D) = \frac{1}{N}\sum_{i=1}^{N}\left(Ax_i + y_i + Cz_i - D\right)^2 \tag{6}$$

In order to find the minimum residual error, partial derivatives with respect to *A, B, C* are set to zero:

$$\frac{\partial\varepsilon(A,C,D)}{\partial A} = 0\,,\frac{\partial\varepsilon(A,C,D)}{\partial C} = 0\,,\frac{\partial\varepsilon(A,C,D)}{\partial D} = 0 \tag{5}$$

Then, the parameters *A, B* and *C* are determined by the following relation expressed in the matrix form:

$$\begin{bmatrix} A \\ C \\ D \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i z_i & -\sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i z_i & \sum_{i=1}^{N} z_i^2 & -\sum_{i=1}^{N} z_i \\ -\sum_{i=1}^{N} x_i & -\sum_{i=1}^{N} z_i & N \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\sum_{i=1}^{N} x_i y_i \\ -\sum_{i=1}^{N} y_i z_i \\ \sum_{i=1}^{N} y_i \end{bmatrix} \tag{6}$$

An example with the estimated ground surface is illustrated in Fig. 2.

6. GRID CELL CLASSIFICATION

In this module, we classify all grid cells as *Object* cell or *Ground* cell. For this, we use the distance of the highest point in that cell, $P_{max}^{xz} = (X^{'}, Y_{max}^{xz}, Z^{'})$, to the estimated ground plane.

$$Dist_{xz} = \frac{|A \cdot X^{'} + Y_{max}^{xz} + C \cdot Z^{'} - D|}{\sqrt{A^2 + B^2 + C^2}} \tag{7}$$

The classification rule is very simple: the cells having the distance $Dist_{xz}$ smaller than a threshold are classified as *Ground*. Otherwise the cells are classified as *Objects* cells. In our experiments we have set this threshold to 15 cm. Figure 2 bottom left depicts a sample grid classification. The *Object* cells potentially belong to the obstacles in the scenes. They are further processed to extract the desired objects.

7. OBJECTS DETECTION BY GROUPING THE GRID'S OBJECT CELLS

In this module, we extract the objects from the classified cells in the grid. The object cells are clustered together based on their connectivity. Two cells are connected if their coordinates in the grid differ by at most two. We discard the groups of cells that have less than four cells. The obtained groups of cells represent the objects. For each grid-object we calculate its dimensions and orientation. The width, length and the orientation are calculated using the layout of the corresponding cells in the grid. The height of the object is the maximum height among the corresponding cells. Figure 2 bottom right depicts a sample obstacle detection and Fig. 2 top right shows the classified grid and the detected objects in a 3D view. Each individual object is described by an oriented bounding box, which can be used as primary information for other subsequent processing tasks.

An alternative representation useful especially in obstacle tracking is based on attributed polygonal models extracted by the radial scanning of the Elevation Map [14]. The main idea is to extract a free-form object model by selecting the most visible (not occluded) parts from the camera position. This is achieved by using a scanning axis which extends from the observation point and moves in a radial direction with fixed

steps. At each step, the most visible cell that is classified as object is marked as a delimiter cell. The extracted contours are used to compute polygonal structures so that each individual DEM cluster is described by a separate free-form polygonal model (see Fig. 3).

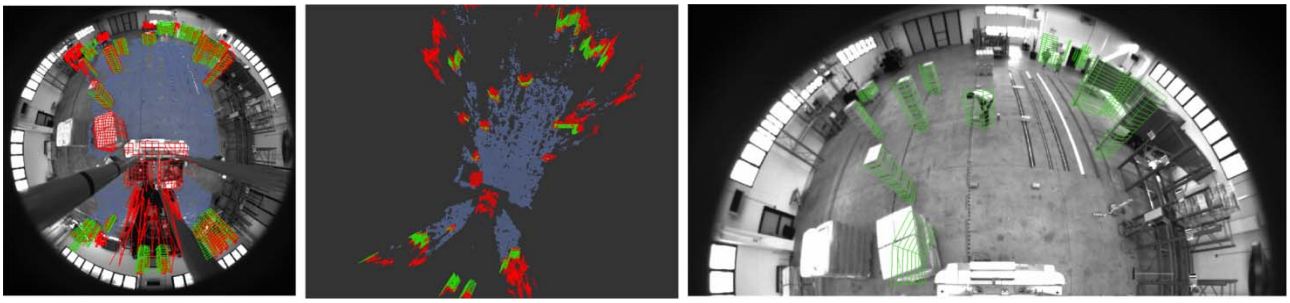

Figure 3 – *Left*: the Elevation Map projected on the left fisheye image. The points are classified as Ground (blue) or Object (red). *Middle*: The 3D view representation of the Elevation Map and of the extracted delimiters. *Right:* the projection of the attributed polygonal delimiters on the left fisheye image.

8. RESULTS AND EVALUATIONS

We have evaluated the obstacle detection rate and the accuracy of the detection. In the test session we have used static ground-truth objects. The AGV however, was allowed to move among the objects since it sends its position at a frequency of 10 Hz. The AGV's localization system was already available. It is achieved using a navigation laser-scanner and its accuracy is of +- 1cm [12]. We have used the AGV's localization system to precisely mark a grid of 5 × 4 positions on the ground. In some of the positions in the grid we have installed pallets and boxes. The size of each ground-truth object was manually measured. The height of the objects varies from 40 cm to 2m. The AGV then took one or two tours among the objects. We have repeated this process three times, by using other objects in different positions. In total, we have used about 25 distinct obstacles. Figure 4 depicts a sample from the evaluation of the system.

We have added the ground-truth objects to the software evaluation module. The ground-truth objects are described by their global coordinates w.r.t. the warehouse and their dimensions. The rest of the evaluation steps are automatic. At each frame, we rotate and translate the ground-truth objects into the approximated local coordinate system of the AGV, defined in [8] at the end of section 5. For the evaluation at the current frame, we only keep those ground-truth objects and those detected objects that fall in the 16.5 m × 7m region of interest around the AGV, as shown in [8] Fig. 1.

Then we match and compare the selected ground-truth objects with the selected objects detected by our system. The matching is based on the proximity criteria. We associate each ground-truth object to the closest detected object. The distance has to be lower than 1m. The localization, width, height and length errors are measured for each matched ground-truth object. We also count the total number of ground-truth objects and the number of detected ground-truth object.

Table 1 shows our measurements. The cumulated number of ground-truth objects in all 932 frames and which fall in the defined AGV's proximity is 7615. Among them, a number of 6 930 ground-truth objects were detected, corresponding to a detection accuracy of 91%. The localization error is 28 cm on average. The width, height and length are estimated with an average error of 28,7 and 51 cm respectively. The results in the test session presented in this paper are better than those achieved in the first evaluation of the system, presented in [13]. The improvements are mainly due to enhanced software.

The errors vary with the distance from the vehicle. Table 2 and Fig. 5 depict this evolution. The length error increases from 30 cm to 1m at 10 m from the vehicle. This happens due to inaccuracies in the estimation of the disparity map. The transition from the objects to the background should be theoretically sharp. The stereo matching algorithms however cannot perfectly estimate this transition and introduces some slight smoothness at object borders. A small disparity error at distance turns into a larger depth error. The localization, the width and the height errors are quite stable in time, they increase only slightly. The false positive object detections and the errors in the distance, height and width of the objects affect the subsequent path planning module negatively as the maneuvering would be based on incorrect information. Tracking is

used in order to stabilize object detection in terms of accuracy of estimated parameters, in the rate of detection and also to reduce the false positive rate.

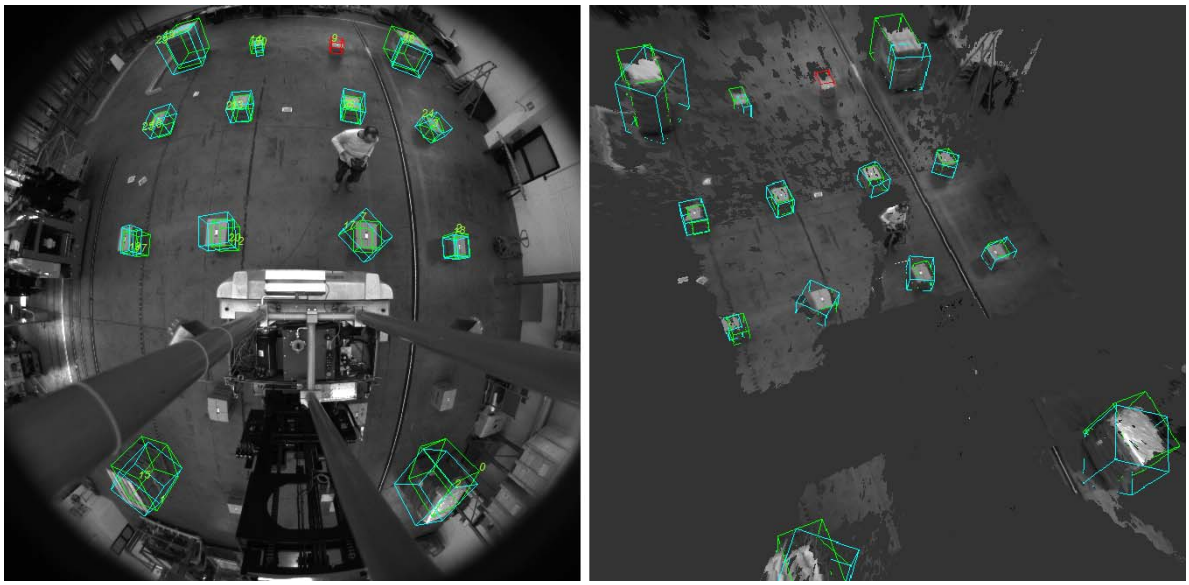

Fig. 4. – Samples from the evaluation of the omnidirectional stereo sensor. *LEFT*: evaluation objects in the fisheye image. *RIGHT*: evaluation objects in 3D view. *Green*: detected ground-truth object, with ground-truth location, size and orientation. *Cyan*: the ground-truth object as it was detected by the sensor. *Red*: undetected ground-truth object.

Table 1 SUMMARY OF SYSTEM EVALUATION

detection rate	91%
mean localization error	28 cm
mean width error	28 cm
mean height error	7 cm
mean length error	51 cm
ground truth objects	7615
detected ground-truth objects	6930
number of frames	932

Table 2 EVOLUTIONS OF MEASUREMENTS BY DEPTH

depth in m	0.5	1	2	3	4	5	6	7	8	9	10
localization error (cm)	23	23	20	21	29	25	24	35	35	40	37
width error (cm)	25	27	26	25	24	21	24	29	35	43	43
height error (cm)	15	11	8	8	8	8	9	14	14	10	11
length error (cm)	34	33	30	40	42	35	47	70	87	109	96

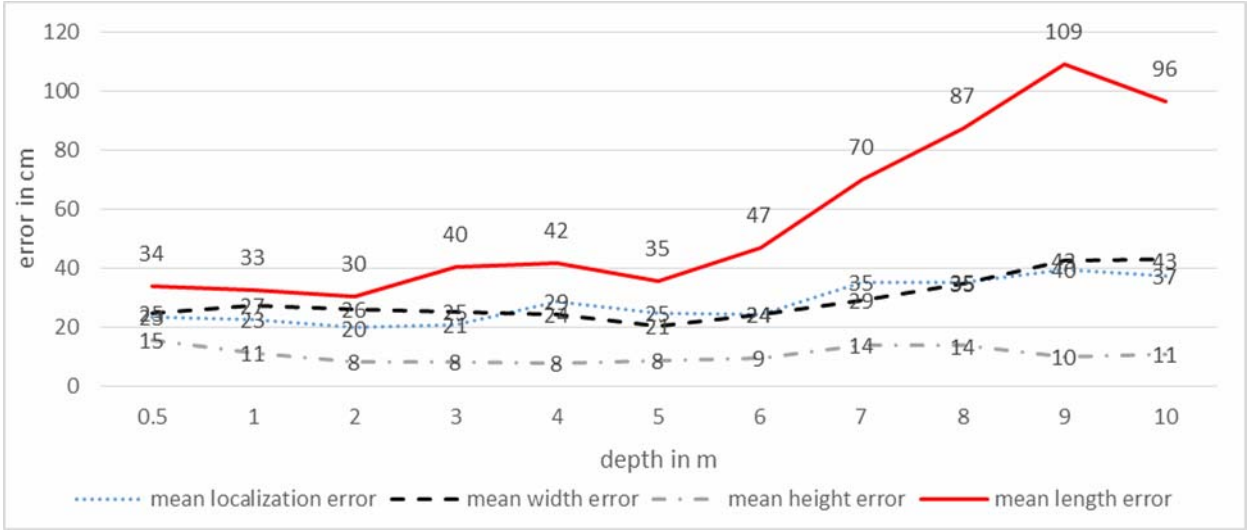

Fig. 5 – The evolution of the errors by depth.

The original resolutions of the fisheye images are 2048 × 2048. The resolutions of the rectified central images are 1024 × 1024 and the resolutions of the rectified front and back images are 1024 × 512. The running time of the entire system with the complete processing pipeline is 130 ms/frame.

9. CONCLUSIONS

In this second part of our work we have presented the stereo reconstruction and obstacle detection modules of the fisheye stereo sensor and its evaluations. The obstacle detection component uses as input three clouds of 3D points obtained by a GPU optimized SGM stereo reconstruction algorithm. The clouds of points are generated from fisheye images, as described in [8]. All the points are projected onto a digital elevation map. We detect the ground plane using a RANSAC procedure on the points with minimum heights in the digital elevation map. The found plane is refined using a least square refinement over the final inliers. The cells are then classified as belonging to the ground or to an obstacle based on the distance of the cell height to the ground plane. The obstacles are extracted by grouping the obstacle cells together. For evaluations, we have installed ground-truth objects at known locations. The obstacles were static, but the vehicle has moved among them. We have obtained a detection rate of 91% and a good localization, width and height accuracy.

ACKNOWLEDGMENTS

This work was supported by the research project PAN-Robots funded by the European Commission, under the 7th FP Grant Agreement n. 314193. The background knowledge comes from the MULTISENS project code PNII-ID-PCE-2011-3-1086, funded by the Romanian Ministry of Education and Research.

REFERENCES

1. RABE C., FRANKE U., GEHRIG S., *Fast detection of moving objects in complex scenarios*, Proc. of 2007 IEEE Intelligent Vehicles Symposium, pp. 398–403.
2. LABAYRADE R., AUBERT D., TAREL J., *Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation*, Proc. of 2002 IEEE Intelligent Vehicles Symposium, **2**, pp. 646–651.
3. PFEIFFER D., FRANKE U., *Efficient representation of traffic scenes by means of dynamic stixels*, Proc. of 2010 IEEE Intelligent Vehicles Symposium, pp. 217–224.
4. ONIGA F., NEDEVSCHI S., *Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection*, IEEE Trans. on Veh. Technol., **59**, pp. 1172–1182, 2010.
5. NGUYEN T.-N., MICHAELIS B., AL-HAMADI A., TORNOW M., MEINECKE M., *Stereo-Camera-Based Urban Environment Perception Using Occupancy Grid and Object Tracking*, IEEE Trans. on Intell. Transp. Syst., **13**, pp. 154–165, 2012.
6. LATEGAHN H., DERENDARZ W., GRAF T., KITT B., EFFERTZ J., *Occupancy grid computation from dense stereo and sparse structure and motion points for automotive applications*, Proc. of 2010 IEEE Intelligent Vehicles Symposium (IV), pp. 819–824, 2010.
7. ELFES A., *Sonar-based real-world mapping and navigation*, IEEE J. Robot. Autom., **3**, 249–265, 1987.
8. DRULEA M., VATAVU A., MANDICI S., NEDEVSCHI S., *An omnidirectional stereo system for logistic plants. Part 1: calibration and multi-channel rectification*, Proc. Rom. Acad., 2016.
9 HALLER I., PANTILIE C., ONIGA F., NEDEVSCHI S., *Real-time semi-global dense stereo solution with improved sub-pixel accuracy*, Proc. of 2010 IEEE Intelligent Vehicles Symposium, pp. 369–376.
10. PANTILIE C., NEDEVSCHI S., *SORT-SGM: Sub-pixel Optimized Real-Time Semi-Global Matching for Intelligent Vehicles*, IEEE Trans. on Vehicular Technology, **61**, *3*, pp. 1032–1042, 2012.
11. HALLER I., NEDEVSCHI S., *Design of Interpolation Functions for Subpixel-Accuracy Stereo-Vision Systems*, IEEE Trans. on Image Process., **21**, 889–898, 2012.
12. REINKE C., BEINSCHOB P., *Strategies for contour-based self-localization in large-scale modern warehouses*, Proc. of 2013 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 223–227.
13. DRULEA M., SZAKATS I., VATAVU A., NEDEVSCHI, S., *Omnidirectional stereo vision using fisheye lenses*, Proc. of 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 251–258.
14. VATAVU A., NEDEVSCHI S., ONIGA F., *Real Time Object Delimiters Extraction for Environment Representation in Driving Scenarios*, Proc. of ICINCO RA, 2009, pp. 86–93.

Received, October 10, 2015