Vision-based pole-like obstacle detection and localization for urban mobile robots

Stefano Sabatini[1], Matteo Corno[1], Simone Fiorenti[2] and Sergio Matteo Savaresi[1]

Abstract— **Despite the enormous progress of the last years, urban environments still represent a challenge for robot autonomous navigation. This paper focuses on the problem of detecting street pole-like obstacles using a monocular camera. Such obstacles, due to their thin structure, may be difficult to be detected by common active sensors like lasers. This is even more critical for innovative solid state LiDARs like the one employed in this work because, at the actual state, they are characterized by very low angular resolutions. The approach described here is based on identifying poles as long vertical structures in the image and in locating them with respect to the robot using a Kalman filter based depth estimation. This information can then be fused with the information coming from LiDARs realizing a complete obstacle detection module.**

I. INTRODUCTION

In the last three decades the field of autonomous mobile robot navigation has been characterized by an extraordinary development. Some complete robotic systems (e.g. [1], [2]) have been presented in literature for autonomous navigation of outdoor populated environments such as city centers. These preliminary solutions spurred great interest also among industries in the research and development of mobile robots able to autonomously navigate the city center for a variety of applications such as autonomous parcel delivery and surveillance. The 2016 McKinsey report on Transport and Logistics estimates that in ten years the 80% of the last mile parcel delivery will be performed by autonomous robots or drones ([3]). In order for this kind of solutions to become effective, it is clear the need of reducing the cost of the sensor setup requested for autonomus navigation. In the systems described in [1] and [2], laser range finders are the only sensors used for perception and are by far the most expensive components of the system. Recently, a new type of LiDAR called Solid State LiDAR became available on the market: this technology does not have any mechanical moving parts and is at least one order of magnitude cheaper than common LiDARs. Sensors based on this technology are meant to play a big role in the large scale spreading of autonomous robot navigation. At the actual state, these kind of sensors does not guarantee the same performance of the classical lasers sensors: they are characterized by long ranges but poor angular resolutions. In this type of technology, the field of view is divided in detection sectors and it has been verified that an obstacle must occupy a certain amount of the sector to be detected: in this way thin obstacles are not visible unless they are very close to the robot. This fact is critical for a robot equipped with this sensor and designed to navigate urban environments: detecting and safely avoiding street poles becomes a challenge. This paper aims to address this issue using a monocular camera and Computer Vision tools: infact, despite the low commercial price, camera sensors carry a great amount of information regarding the surrounding ambient. On the other hand, compared to laser scanners and their stereo counterpart, monocular cameras requires a more sophisticated processing to provide usable information to a navigation algorithm, most of all they do not directly return distance information.

While the problem of obstacle detection using monocular images has been addressed by many researchers in the past, the more specific problem of thin pole-like obstacle detection has not been extensively treated in literature. General computer vision methods for obstacle detection may fail to detect thin structures or they may result unnecessary complex when the camera is used in conjunction with an active range sensor such as the solid state LiDAR used in this research. Obstacle detection algorithms based on appearance methods such the one used in [4] trie to differentiate between free space and obstacles comparing image pixels with the color appearance of the ground. Such an approach may fail in urban environments where the ground color may vary substantially and, in many cases, is similar to the gray color of most of the street poles. Other approaches, such the one used in [5] and [6], are based on the scale expansion concept where the detection is performed monitoring the change of size of objects in the image plane as they get closer to the camera. A fairly big amount of works on monocular obstacle detection can be grouped under the general family of Structure From Motion methods: the basic idea is to extract 3D information from the spatial and temporal changes occurring in images sequences, see [8]. The full 3D reconstruction problem is unnecessary complex in this context since the objective is to only estimate the location of thin vertical structures. Inspired by the recent work of [9], in this paper pole-like obstacles are detected looking for vertical lines in the image. The extracted lines are then matched in subsequent images and their lateral positions in the image plane are considered as measurements. The distance of the pole from the robot is then estimated using a Kalman filter based observer where the pole depth is considered as an unmeasured state with known dynamics and the camera motion is considered as a known input derived from the fusion of the robot inertial and odometry measurements. A similar reasoning is applied

[1]Dipartimento Elettronica Informazione e Bioingegneria, Politecnico di Milano, via Ponzio 34/5, 20133 Milano, Italy. stefano.sabatini, matteo.corno, sergio.savaresi@polimi.it

[2]Yape srl, Via San Martino 12, 20122 Milano, Italy. simone.fiorenti@e-novia.it

in [10] in a different context, where an ad-hoc nonlinear observer is applied to the estimation of the depth of a marker for visual servoing. Furthermore in this work, the Kalman Kilter parameters are adapted online to account for the varying reliability of the information among the frame: features far from the focus of expansion are considered more reliable for the depth estimation and their corresponding measurement noise variance is lowered.

Summarizing, this paper proposes and experimentally validates a computationally efficient method to detect and estimate the distance of street pole-like obstacles using a monocular camera; this information can be fused with the Solid State LiDAR measurements to guarantee a safe navigation of urban environments. Although the computer-vision tools employed here are quite well known in the literature, the novelty of the work resides in the particular application that finds its motivation in the novel sensor setup employed. The paper is organized as follows: In Section II, the experimental setup is presented and the detection problem formulated. Section III recalls the mathematical background regarding the projection camera model. Section IV presents the overall pole detection algorithm and finally in section V the method is validated with experimental data.

II. EXPERIMENTAL SETUP AND PROBLEM FORMULATION

The robot used in this work is depicted in Figure 1. It is a two wheeled self-balancing robot designed to perform autonomus parcel delivery in urban environments. Its sensing apparatus consists of solid-state LiDARs and monocular cameras characterizing a substantial price reduction with respect to solution based on mechanical laser scanners.

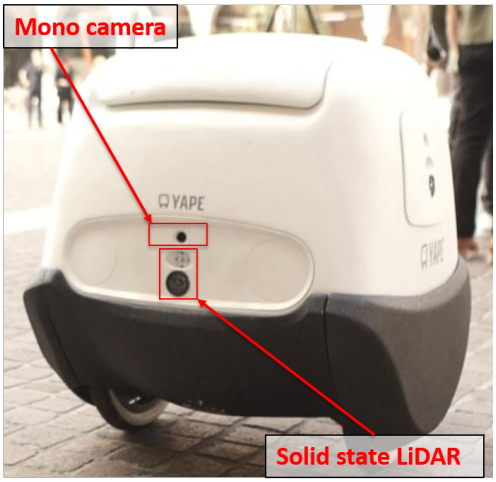

Fig. 1. Picture of the robot used in this work. Mounting positions of the LiDAR and camera are highlighted.

Four 2D solid-state LiDARs are employed (one for each side), the model is the LeddarVu8 by Leddartech [11]. Compared to common mechanical laser scanners, this sensor does not present any rotating parts but it employs the so-called flash LiDAR technology that enables the detection of objects within a field of view of 100 ° and a range of 34 m. The field of view is divided into eight independent detection cones resulting in an angular resolution of 12.5°. Although the sensor has proven reliable and accurate, a major drawback has been encountered during experimental tests: thin obstacles cannot be detected unless they are very close to the robot. More specifically, it has been experimentally found that the detection of an obstacle happens only if it occupies at least 25% of the width of the corresponding detection cone. For this reason, a safe detection and avoidance of thin obstacles such as street poles becomes very critical: it has been verified that a street pole such the one used for road signs is detected when the distance from the robot is less than 1 meter. In the considered setup, where a self balancing robot is employed, the situation is even more critical: due to its particular open-loop unstable dynamics, the braking maneuver is quite slow since the robot has to tilt considerably to produce a braking force and at the same time keeping the balance. If only LiDAR were used for obstacle avoidance, a very stringent limit on the cruise velocity of the robot would be necessary in order to perform a safe stop or an avoidance maneuver in case of pole detection. In order to fill this sensing gap, the position of thin vertical pole-like object must be estimated using the monocular camera mounted right above the LiDAR (see Figure 1). As a guideline requirement, the detection system must be able to perform the detection of the pole at a distance of at least 2.5 m that is approximately the braking distance at the robot maximum speed. The camera is a commercial webcam Logitech c920 characterized by 78° of diagonal field of view and a focal length of 3.67 mm able to stream images at 30 fps. The wheel motors provide wheel rotational velocity and an inertial measuring unit mounted at the center of the wheelbase provides 3D accelerations and rotational speeds. In the rest of the paper, camera motion is considered to be known as in other works on feature depth estimation (e.g [13] and [10]). This is a strong assumption but, since our focus is about obstacle avoidance and not about enviromental mapping, only a short term memory of the feature relative position is necessary: a well calibrated odometry fused with IMU gyroscope can guarantee satisfactory camera motion estimation in the short term. The approach described in the following sections makes use of the fact that street poles can be represented in the image as long vertical straight edges. The method identifies and estimate the position of these edges tracking their movement in the image plane. Posed in this way, the method is not able to distinguish between pole-like obstacles and large obstacles with long straight vertical edges: this is is not an issue with the current setup because LiDARs are perfectly capable to detect large obstacles so that the information coming from the two sensors are perfectly consisent and can be fused in a common local obstacle map as it is proposed in [12].

III. MATHEMATICAL BACKGROUND

The objective of this section is to recall the necessary mathematical background regarding the perspective camera

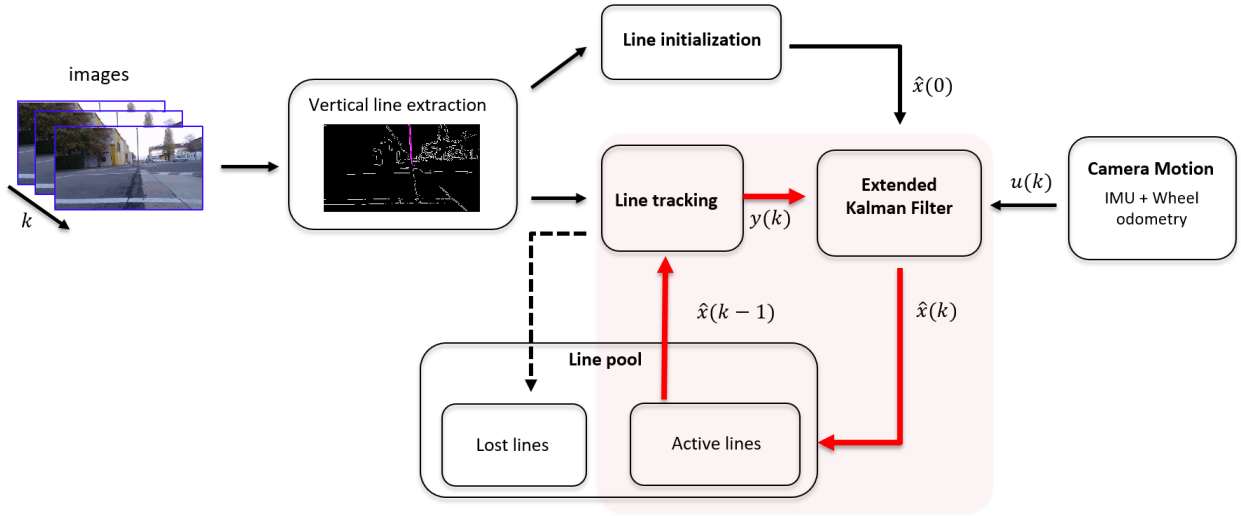

Fig. 2. Overall scheme of the pole detection algorithm. The recursive Kalman filter loop is highlighted in red.

model and to derive the differential equations upon which the pole depth estimation algorithm is designed. The formulation is based on concepts described in [10]: the main idea is to find a mathematical expression that relates the camera motion (hence the robot motion) to the motion of features in the image plane. Figure 3 illustrates the three main coordinate systems of interest for the description of the problem: the world fixed frame, the camera frame and the image frame.

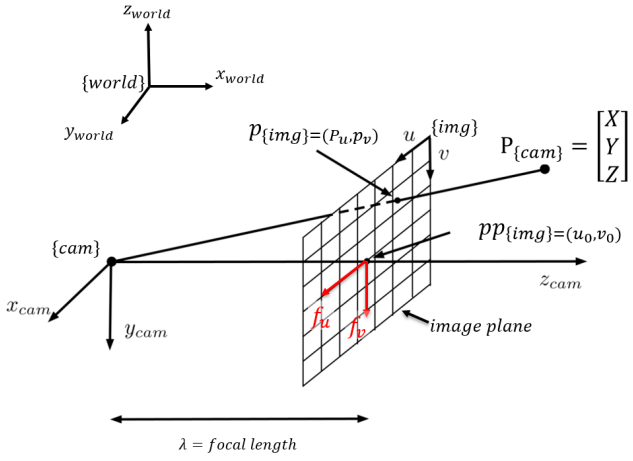

Fig. 3. World, camera and image frames definition.

A static point P of coordinates $\begin{bmatrix} X & Y & Z \end{bmatrix}$ expressed in the camera frame can be projected into the image plane through the pinhole camera model resulting in the point p expressed in the image frame:

$$p = \begin{bmatrix} p_u \\ p_v \end{bmatrix} = \begin{bmatrix} \lambda & 0 & u_0 \\ 0 & \lambda & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} \tag{1}$$

Where λ is the camera focal length in pixels and u_0 and v_0 are the principal point coordinates in pixels; these parameters can be obtained through a camera calibration procedure. It is then convenient to express a point in the image plane with respect to a reference centered on the camera principal point defining $f_u = p_u - u_0$ and $f_v = p_v - v_0$. Considering now a moving camera in a 3D static environment, it is useful to map the motion of the points projected in the image plane to the known motion of the camera. In order to do this, the following relation can be used:

$$\begin{bmatrix} \dot{f}_u \\ \dot{f}_v \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{Z} & 0 & \frac{f_u}{Z} & \frac{f_u f_v}{\lambda} & -(\lambda + \frac{f_u^2}{\lambda}) & f_v \\ 0 & -\frac{\lambda}{Z} & \frac{f_v}{Z} & \lambda + \frac{f_v^2}{\lambda} & -\frac{f_u f_v}{\lambda} & -f_u \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \tag{2}$$

where $\mathbf{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}$ and $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}$ are the linear and angular velocities of the camera with respect to the world frame expressed in the camera frame. Equation (2) is called optical flow equation and is of fundamental importance in Computer Vision, its derivation is explained in [13]. It should be noticed that features speed in the image plane are related to the camera translational velocity through their inverse depth $1/Z$. The estimation of poles depth relies on this dependency.

IV. SYSTEM OVERVIEW

The overall scheme of the algorithm is depicted in Figure 2. The algorithm takes as inputs a sequence of images streamed from the camera and its linear and angular velocities estimated from the robot IMU and wheel encoders. For each new incoming frame, vertical lines are extracted as features representing pole-like obstacles and fed to a Kalman filter as measurements. The Kalman filter has the objective of estimating poles depth and therefore their position relative to the robot. The extracted vertical lines are then tracked in subsequent frames in order to refine the estimate. In the following, the feature extraction step and the Kalman Filter step are described in detail.

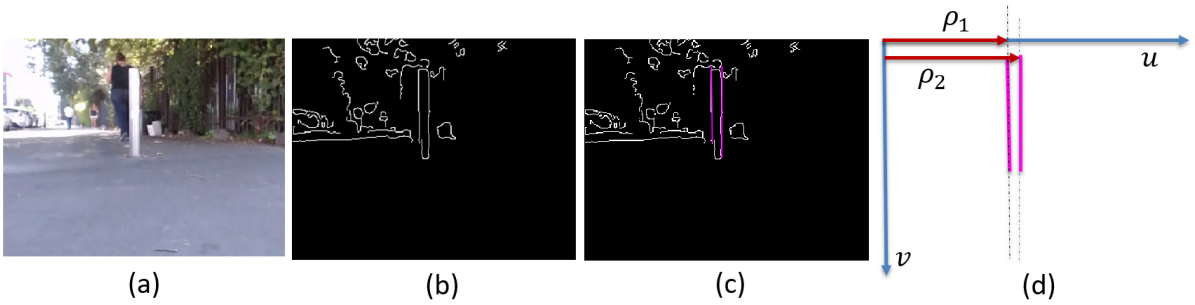

Fig. 4. Image processing chain for line extraction: **a)** undistorted camera image **b)** Canny edge detection **c)** Line extraction through Hough transform **d)** the vertical approximation of each extracted lines is taken, each line is defined only by the ρ parameter.

A. Feature Extraction

The main characteristic of street poles is their tall and narrow aspect ratio. The idea of the algorithm is based on identifying poles in the image searching for vertical lines. Images are taken from the camera at a resolution of 480x270 and the lens distortion is corrected using the parameters obtained from the camera calibration. The image processing chain is illustrated in Figure 3: as a first step images are converted in gray-scale and filtered with a Gaussian filter to reduce noise. Then edges are extracted using a Canny edge detector (Figure 4b). From the edge binary map, the Hough transform is calculated in order to detect lines. A straight line is represented in the Hough space using two parameters ρ and θ corresponding respectively to its perpendicular distance from the origin and to the angle that the perpendicular makes with the u axis. Since the Hough transform is quite computationally intensive and we are only interested in identifying vertical straight lines, the Hough transform is computed only for a small range of angles around zero ($\theta = \begin{bmatrix}-3 & 3\end{bmatrix}$, in our implementation). To reduce the number of detections per single image, only lines longer than a certain threshold are extracted as features (Figure 4c). The detected lines are then approximated with their perfectly vertical counterpart (setting $\rho' = \rho cos(\theta)$ and $\theta' = 0$): in this way a line is described only with the related ρ parameter that defines its position along the u axis of the image frame (Figure 4d).

It must be noticed that the described method is a very general detector: not only vertical lines corresponding to poles are usually extracted. It happens frequently that large objects with straight vertical boundaries are detected (e.g buildings edges). Classifying this edges as obstacles is actually correct and this information can be fused with measurements from the LiDAR that is perfectly able to detect large obstacles.

B. Depth Estimation

Once vertical lines are extracted from the image, their position with respect to the robot must be estimated to inform the local path planner of the presence of potential obstacles that may not be visible by the LiDAR.

The depth estimation is carried out through an Extended Kalman Filter based on a simplified version of Equation (2): the objective here is to describe the movement of the extracted vertical lines in the image plane as a result of the camera motion. First, the assumption of planar camera motion is made: it is assumed that the camera moves parallel to the xy_{world} defined in Figure 3, hence neglecting the robot pitch and roll movements. This assumption greatly simplifies the problem because it decouples the two equations in (2), consequently only the first equation, that describes the points velocities in the u axis, is relevant. Basically, under this assumption, the extracted vertical straight lines only translates along the u axis of the image frame while the robot moves. The idea is then to infer the distance looking at their projection on the image plane. Figure 5 shows a top view of a possible robot-pole configuration: the depth Z is the objective of the estimation problem, f_{u_1} and f_{u_2} are the positions on the image plane of the extracted edges, v and ω are the wheelbase speed and robot yaw rate that together describe the motion of the robot on the ground plane. it

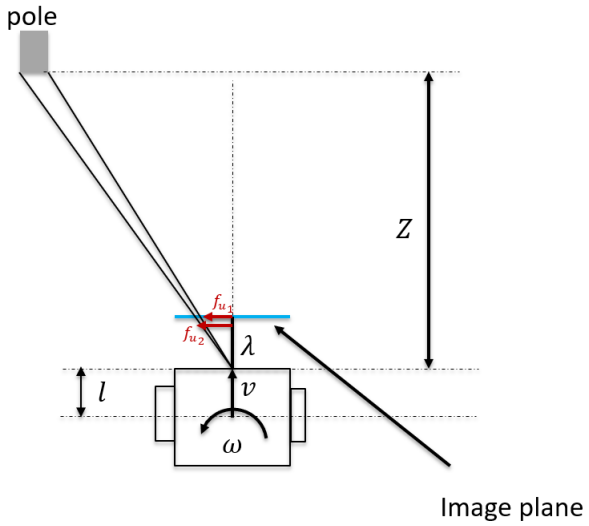

Fig. 5. Top view of a possible robot-pole configuration.

is convenient to perform a change of coordinates and to estimate the inverse depth $d = 1/Z$ instead of Z. In the light of the above considerations, combining the first equation in (2) and expressing the depth dynamics as a function of the

camera motion, the following non linear dynamical system can be formulated in the state-space form:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} x_1 x_2 & -(\lambda + \frac{x_1^2}{\lambda} - l\lambda x_2) \\ x_1^2 & -\frac{x_2 x_1}{\lambda} \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{aligned} \tag{3}$$

where the state vector corresponds to $x = \begin{bmatrix} f_u & d \end{bmatrix}^T$, the measurement is $y = \rho - u_0$ where ρ is the result of the feature extraction described in Section IV-A and the input is $u = \begin{bmatrix} v & \omega \end{bmatrix}^T$. A linearized observability analysis shows that the depth of a line becomes unobservable in absence of camera translation (camera rotation by itself does not carry any information about depth) or in case of camera translation in the exact direction of the considered line ($x_1 = f_u = 0$) (i.e. the feature perfectly lies in the so-called focus of expansion of the image). In general, the depth of lines close to the focus of expansion is weakly observable and as they move away from it, they move faster in the image and, intuitively speaking, their depth becomes more and more observable enhancing the convergence properties of the filter. This fact inspired an adaptive tuning of the filter measurement noise variance: The value of R can be lowered as features get closer to the edges of the frame indicating that measurements are more reliable far from the focus of expansion. This adaptive tuning of the filter has been implemented with an R value exponentially decreasing towards the edges of the image:

$$R = ae^{-b|x_1|} \tag{4}$$

the objective here is to have the same amount of noise in the estimation independently from the line position. Having an estimate of the feature depth, it is possible to recover the feature position in the camera plane $P_{cam} = \begin{bmatrix} Z\tan(\alpha) & 0 & Z \end{bmatrix}^T$, where α is the feature angular position that, given the camera horizontal FOV and the image width W in pixels, can be calculated as:

$$\alpha = \frac{FOV f_u}{2W} \tag{5}$$

The equation of the standard Extended Kalman Filter are not reported here, the reader is referred to [16] for any detail about the filter formulation,

C. Line tracking

For each incoming frame, vertical lines are extracted. Each extracted line can be either a new feature that was never observed before or associated to an already tracked feature in the active line pool (see Figure 2). Since vertical lines are described by a single parameter, the line association is very simple: first, for each feature in the active pool, a prediction of the position in the image is computed based on the previous frame $\hat{\rho}_j(k|k-1) = \hat{x}_1(k|k-1) + u_0$. In a second step the extracted features at the current frame $\rho_i(k)$ are matched with their closest prediction $\hat{\rho}_j(k|k-1)$. Finally, the actual matching is performed only if the 1D euclidean distance between the measurement and the prediction is below a certain threshold $\delta\rho_{th}$:

$$|\rho_i(k) - \hat{\rho}_j(k|k-1)| < \delta\rho_{th} \tag{6}$$

If an extracted feature is not associated with any active feature, a new state is initialized. For each new feature the depth is initialized at a value of 15 meters with an a large P_0. Of course a huge error in the estimated depth reflects on the goodness of the feature prediction $\hat{\rho}_j(k|k-1)$ and consequently on the tracking performance, possibly causing tracking failures. The problem has been handled making the distant threshold $\delta\rho_{th}$ dependent on the uncertainty of the estimate at the current instant evaluated from the filter error covariance matrix P: the more uncertain is the feature prediction the bigger the area where to look for the matching. The initialization parameters are considered in this sense as tuning parameters. It can happen that, due to occlusions or sudden changes in light conditions, for multiple frames features corresponding to a pole that is still in the camera FOV are not detected. In this case, lines that are not matched with any measurement are kept active and projected ahead in a open loop manner: eventually they will be tracked again in future frames. If an active feature is not matched for a certain number of consecutive frames, it is deleted from the feature pool.

V. EXPERIMENTAL RESULTS

In this section, the goodness of the described pole detection algorithm is demonstrated against experimental data. A specific experiment has been designed to tune the Kalman fiter: the robot was driven towards three different street poles located at the same depth but at three different angular position with respected to the camera: one pole close to the image center (that is the focus of expansion in this case of pure translation motion), one in the right part of the image ($f_u > 0$) and one in the left part of the image ($f_u < 0$). For this preliminary experiment a large obstacle was positioned behind the poles in order to enable LiDAR detections even at long distances and to have a ground truth measurement of the depth. The performance of the poles depth estimation can be appreciated in Figure 6: solid lines represent the performance of the filter with a constant tuning, while dashed lines are the result of the adaptive Extended Kalman Filter. Both tunings are able to converge at the true distance before the pole gets closer than 4 meters. It can be notice how, without the adapting tuning, the convergence speed and noise depend on the position of the pole in the image (shown in the bottom plot): the more distant from the image center, the faster and less noisy the filter convergence. On the other hand, with the adaptive filter, the three estimates present the same noise level: of course this slows down the convergence time of lines that are close to the focus of expansion (yellow dashed line in Figure 6) because their measurements is considered less reliable than the others for the reasons explained in Section IV-B.

Furthermore, several experiment were performed in urban conditions, an example is presented in Figure 7. In the top

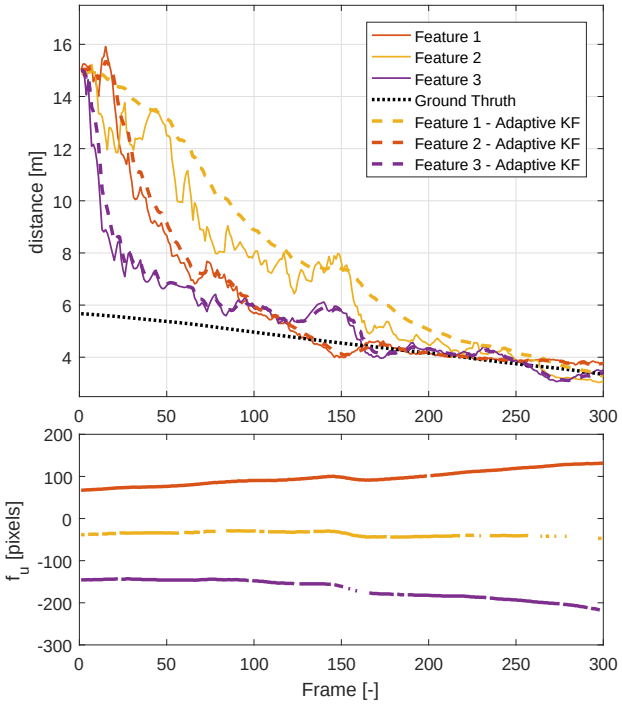

Fig. 6. Pole depth estimation results. The experiment has been made with three different poles (one for each feature in the plot) placed at the same depth. The ground truth distance is measured through the LiDAR. Results with and without the adaptive tuning of the Kalman Filter are plotted for comparison: The values of $a = 25$ and $b = 0.03$ are used for the adaptation of the R matrix presented in (4). The constant $R = 0.1$ is used for the standard Kalman filter.

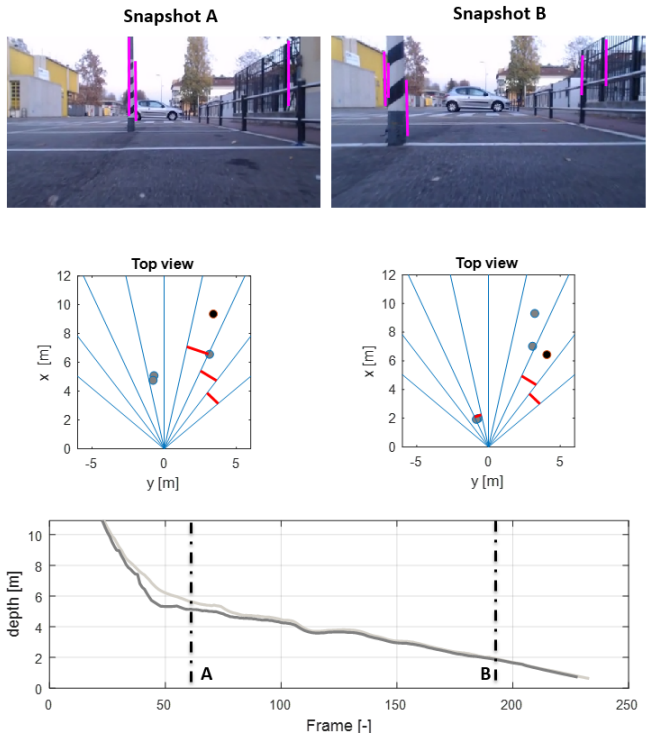

Fig. 7. Pole Detection in a urban context. Two snapshot of teh robot motion are highlighted: when the robot is at 6 meters from the pole (snapshot A) and when it is very close to the pole (snapshot B). **Upper layer**: camera images together with the extracted features. **Middle layer**: Top view of the scenario in the camera system of reference. Red lines corresponds to the LiDAR detections, grey dots to the vision-based poles estimated position. **Lower layer**: street lighting pole depth estimation in time, instants corresponding to the two snapshot are highlighted.

layer of the figure the camera image along with the extracted vertical lines is presented, notice the presence of a big pole for street lighting on the left. In the middle layer, a top view of the scenario in the camera system of reference is shown: the LiDAR detection cones (blu lines) and the corresponding detections (red segments) are represented together with the object detected by the vision-based algorithm (gray dots). In the bottom plot the time history of the depth estimation for the street lighting pole is presented. The robot is driven towards the car visible in the image and two snapshots are highlighted in the figure: in snapshot A, the robot is at around 6 meters from the pole. While the LiDAR does not return any object detected in that direction, the vision based detection algorithm is able to provide a reasonable estimation of its depth. Only when the robot is at less than 2 meters from the pole, the LiDAR detects an obstacle on the left side that overlap perfectly with the vision based position estimation confirming again the goodness of the depth estimation.

VI. CONCLUSIONS

In this paper a pole-like obstacle detection algorithm based on monocular vision has been presented and validated experimentally. The output of the algorithm can be used as support to an obstacle detection system based on active sensors such as laser scanners. The position estimation of a pole that lies perfectly on the focus of expansion is problematic since the system described in (3) becomes unobservable in that situation. Future improvement could deal with addressing this issue designing a specific trajectory that enables the depth estimation when such situation is recognized.

REFERENCES

[1] Kummerle, Rainer, et al. "Autonomous robot navigation in highly populated pedestrian zones." Journal of Field Robotics 32.4 (2015): 565-589.

[2] Trulls, Eduard, et al. "Autonomous navigation for mobile service robots in urban pedestrian environments." Journal of Field Robotics 28.3 (2011): 329-354.

[3] Joerss, M., et al. "Parcel Delivery: The Future of Last Mile." McKinsey & Company (2016).

[4] Ulrich, Iwan, and Illah Nourbakhsh. "Appearance-based obstacle detection with monocular color vision." AAAI/IAAI. 2000.

[5] Wedel, Andreas, et al. "Realtime depth estimation and obstacle detection from monocular video." Joint Pattern Recognition Symposium. Springer, Berlin, Heidelberg, 2006.

[6] Mori, Tomoyuki, and Sebastian Scherer. "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles." Robotics and automation (icra), 2013 ieee international conference on. IEEE, 2013.

[7] Bruhn, Andrs, Joachim Weickert, and Christoph Schnrr. "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods." International journal of computer vision 61.3 (2005): 211-231.

[8] zye?il, Onur, et al. "A survey of structure from motion*." Acta Numerica 26 (2017): 305-364.

[9] Zhou, Chen, et al. "Fast, Accurate Thin-Structure Obstacle Detection for Autonomous Mobile Robots." arXiv preprint arXiv:1708.04006 (2017).

[10] De Luca, Alessandro, Giuseppe Oriolo, and Paolo Robuffo Giordano. "Feature depth observation for image-based visual servoing: Theory and experiments." The International Journal of Robotics Research 27.10 (2008): 1093-1116.
[11] website: www.leddartech.com, 2018
[12] Nuss, Dominik, et al. "Fusion of laser and monocular camera data in object grid maps for vehicle environment perception." Information Fusion (FUSION), 2014 17th International Conference on. IEEE, 2014.
[13] Matthies, Larry, Takeo Kanade, and Richard Szeliski. "Kalman filter-based algorithms for estimating depth from image sequences." International Journal of Computer Vision 3.3 (1989): 209-238.
[14] Zhang, Zhengyou. "A flexible new technique for camera calibration." IEEE Transactions on pattern analysis and machine intelligence 22.11 (2000): 1330-1334.
[15] Morbidi, Fabio, and Domenico Prattichizzo. "Range estimation from a moving camera: an immersion and invariance approach." Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, 2009.
[16] Bishop, Gary, and Greg Welch. "An introduction to the Kalman filter." Proc of SIGGRAPH, Course 8.27599-23175 (2001): 41.