Robust Detection of 3D Scene Horizontal and Vertical Lines in Conical Catadioptric Sensors

Carlo Pinciroli, Andrea Bonarini and Matteo Matteucci
Department of Electronics and Information, Politecnico di Milano, Milan, Italy

***Abstract*— Recent research in omnidirectional vision is focusing on the extraction of primitives from omnidirectional images, mainly for detecting straight horizontal and vertical edges of the environment where the vision system is located. The aim of this work is to pursue this research direction by proposing two fast algorithms for 3D straight horizontal and vertical edges reconstruction from a single 2D omnidirectional image. The proposed algorithms have been successfully applied to real images providing real-time and robust performance.**

I. Introduction

A great number of optimized methods to detect horizontal and vertical edges of the environment in classical perspective images have been developed. When using a catadioptric image sensor, detecting straight lines becomes a complex task because they are projected onto the image plane in a distorted manner (i.e. they are not straight lines anymore). In this case, if we want to perform the detection, we have two main methods: either antitransform the omnidirectional image to obtain a perspective image, and then apply classic optimized techniques, or work directly on the omnidirectional image.

The former method seems promising, but it has two crucial drawbacks that make it almost always ineffective. The first drawback is slowness: antitransformation is in general a long task, depending on the image size. The second drawback is that antitransformation is not always possible. In fact, with some mirror shapes, such as conical or spheric, antitransformation gives distorted images; it is successful only with mirror shapes satisfying the Single Viewpoint (SVP) constraint, such as hyperboloidal mirrors [1]. Mirrors satisfying the SVP constraint are easy to address because their geometry can be bound to the classic perspective geometry through a projection onto a spheric surface [10] and for these reasons many early researchers focused their efforts on developing algorithms based on SVP-compliant vision systems.

The second method requires to know how horizontal and vertical edges are projected onto the image plane and to devise ad-hoc algorithms to extract these features directly from omnidirectional images. Recent papers, like [12], [9], and this work prove that also non-SVP geometries such as the conical one can be easily modeled and fruitfully exploited to obtain satisfying results in 3D reconstruction of features. In fact, when the mirror does not satisfy the SVP constraint, it is possible to retrieve 3D information about some straight features while this is not possible with classic perspective images.

The aim of this work was to realize a fast and robust system able to detect vertical and horizontal features in the scene from images obtained by a vision system equipped with a conical mirror. Since the conical mirror does not satisfy the Single Viewpoint Constraint, the analysis must be driven directly on omnidirectional images and a study of the geometric properties of this vision system is needed. Besides the well-known property of omnidirectional images that lines parallel to the cone axis are projected to the image plane as radial lines, we proved and exploited the fact that features perpendicular to the cone axis are projected into a specific family of complex curves. In this paper we provide the equation of these complex curves and analyze it in order to realize a fast and general method to detect the corresponding lines. To obtain such result, we derive new equations for conical projection. The main difference between our equations and the already existing ones is the fact that ours are built by exploiting the axial symmetry of the conical projection and so the geometry is reduced to two dimensions, while classical approaches to map a world point (x, y, z) to an image point (u, v) basically consist of a translation, a rotation and a deformation of the world coordinate system.

We propose in Section IV a fast method to retrieve information about vertical features (radial lines). After several tests, some of which are reported in Section VI, this method proved to be more precise than the classical Hough Transform. Moreover, we present in Section V an adaptation of the well known RANSAC algorithm to detect 3D scene horizontal features (i.e., complex curves on the projected image) in a relatively short time. The key features of the proposed algorithm are robustness to noise and the ability to retrieve 3D information about horizontal features from a single static image, while known algorithms need two or more images to perform the same task.

II. State of Art

Different mirror shapes reflect the surrounding world in different ways. The choice of a specific mirror shape, either standard or hybrid, is performed considering various aspects, mainly resolution, manufacturing cost and geometric properties. With respect to these properties mirror shapes can be classified in two categories: *Single Viewpoint* (SVP, also called *central*) imaging systems, and *Non-SVP* (*non-central*) imaging systems.

In SVP mirror systems, all the lines that the incoming rays reflected to the image plane lie on intersect at a single point called *Effective Viewpoint*. Baker and Nayar [1] found that

the only useful mirror shapes possessing this property are hyperboloidal mirrors (with a classic perspective camera) and paraboloidal mirrors (with a orthographic camera). For this class of vision systems geometric properties are simple to recognize. On the other hand, SVP imaging systems suffer for a number of critical problems. First, SVP constraint leaves little room for changing other optical parameters that could be useful to tune because the relative positions of the mirror and the camera are strictly bound. Due to the curved vertical section of the mirrors spatial resolution is badly distributed: typically the ground or the sky (depending on the orientation of the vision system, whether the convexity is upwards or downwards) and the camera occupy the central part of the omnidirectional image leaving little room on the peripheral part of the image to the surrounding environment. Regarding manufacturing cost, hyperboloidal and paraboloidal mirrors are very expensive because of their eccentricity from a sphere. Conical and spheric mirrors, even though they are not SVP, present lower manufacturing costs and are often used. A method to avoid the limitations of hyperboloidal and paraboloidal mirrors while exploiting the advantages deriving from SVP property was developed by Derrien and Konolige [8]. They approximate SVP property when it would not hold either because the mirror is inherently non-SVP or because the relative positions of the (hyperboloidal, paraboloidal) mirror and the camera are different from those required for SVP.

All the attempts to perform 3D reconstruction with SVP mirrors need to use a set of images, or need to have a priori information about the structure of the scene. As we will see in this section, non-SVP can be considered a good property for systems aimed at 3D reconstruction of features because the lack of a single viewpoint can be thought as the intrinsic ability of a non-SVP vision system to view the scene from different points and retain some piece of 3D information.

Early non-SVP catadioptric vision systems were based on a spheric mirror [2], [6], or on a conical mirror [14] due to the low cost of such mirrors and the mathematical simplicity of the projection. The advantages of a conical mirror over a curved section mirror (such as spherical) were discussed in [12]. The main merit of the conical mirror is to present straight lines in vertical section yielding to a simple projective geometry and the lack of the typical aberrations due to curved sections (i.e. *spherical aberrations*).

The first system developed with a conical mirror is Yagi and Kawato's COPIS ("Conical Projection Image Sensor") [14]. It is composed by a conventional perspective camera and a conical mirror with the common setup previously described. It was primarily used for map building purposes, and the only features of the scene that it could detect were vertical edges. In all the convex and rotationally symmetric mirrors, vertical edges project onto the image plane as straight lines oriented as radii: therefore, COPIS works as a *goniometric* sensor, that is it detects only the azimuth angle of objects. By known moves and successive measurements of azimuth angles, COPIS can be used to build an environmental map of the surrounding world.

Another well known omnidirectional vision system based on a conical mirror is the SYCLOP ("Conical SYstem for LOcalization and Perception") sensor by Brassart *et al.* [3]. It was equipped with a full body of subsystems dedicated to calibration and map building. Similarly to COPIS, SYCLOP detects from the image only radial lines (vertical edges in the scene). The system exploits a single camera and is able to move along a straight rail: at the beginning of the path an image is taken and stored. Then, the system moves on the other side of the path and takes another picture. The information obtained by the two images is then combined and a raw form of 3D reconstruction is performed.

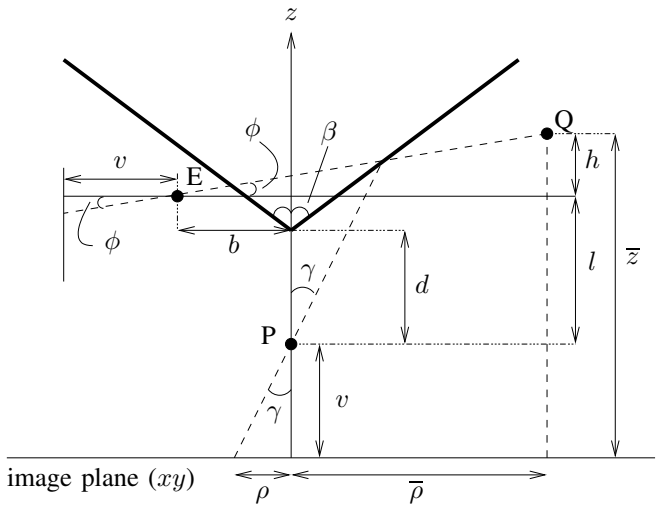
A well known approach for estimation of 3D information of both horizontal and vertical edges of the scene is due to Fiala and Basu [9]. Vertical edges are simply detected with a voting procedure. The method for detecting horizontal edges is more interesting: it is an adaptation of the classical Hough Transform to the case of curves due to non-SVP projection of horizontal edges and is called *Panoramic Hough Transform*. The algorithm estimates the azimuth and the slope of the plane which the horizontal line belongs to, therefore from a single image complete 3D information is not reconstructed. In order to obtain a complete reconstruction, two images are analyzed and their results are combined by a method similar to the epipolar geometry specifically designed for the output of the Panoramic Hough Transform. It gives good results and a satisfying robustness to noise is provided by the classic *Identify and Remove* technique often implemented for real application of the Hough Transform.

Recently, a more refined approach to 3D reconstruction with the SYCLOP sensor was proposed by Cauchois *et al.* [5]. It aims at making SYCLOP a complete vision sensor, able to exploit horizontal edges in the 3D scene as well as vertical. The method maps the omnidirectional image obtained by the sensor into a panoramic image projected onto a cylinder surface. Then, feature extraction is performed. The extracted features are compared to an a priori knowledge base of the structures expected in the surrounding world. Matching between the sensed and the expected structures yields to a good reconstruction of the environment. The method was tested only in an artificial environment whose structure was simple to model.

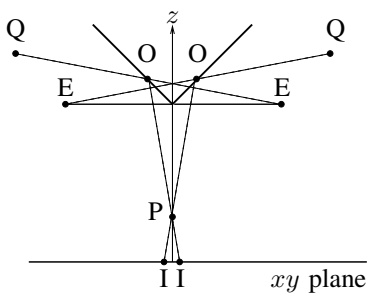
The system proposed in this paper is based on geometric considerations that enable it to estimate complete 3D information of almost any horizontal straight line without a priori knowledge about the surrounding world. At the present time no system is able to perform such task from a single image and without motion. Because of geometric reasons discussed in Section V-B, vertical edges are not totally reconstructed; on the other hand, horizontal edges are completely estimated in a robust, non approximated way from a single static image.

III. Projection Model

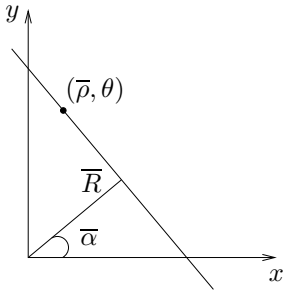
Equations for conical projection were derived by Yagi and Kawato for COPIS [14] [11], by Brassart *et al.* for SYCLOP [3] and by Spacek [12]. Equations by Yagi were specialized



(a) The geometry of the projection of a single point $(\overline{\theta}, \overline{\rho}, \overline{z})$ onto the image plane $z = 0$.



(b) Effective viewpoint for a conical mirror. Q is the world point to project, E is the effective viewpoint, P is the pinhole of the camera, I is the point which Q projects to.



(c) The representation of a line with parameters $\overline{\alpha}$ and $\overline{R}$.

Fig. 1. (a) The geometry of the projection of a single point; (b) effective viewpoints of a conical mirror; (c) the parameters of a line in polar representation.

for their COPIS sensor and so are not suitable to be used to model the projection of 3D scene horizontal lines. Brassart defined equations for conical projection based on a translation, a rotation and a deformation of the three-dimensional scene coordinate system to the two-dimensional image coordinate system. Due to the nature of the model, the proposed equations are not suitable for developing a real-time detector of 3D scene lines. As opposed, the equations for the projection of a single point devised by Spacek are based on simple geometric considerations and exploit the axial symmetry of conical projection. Unfortunately, such equations hold only for the case of half cone tip angle $\beta = \pi/4$. In the following discussion, we present the generalization of this result to $0 < \beta < \pi/2$.

If we look at the projection of a single world point Q, we can define the effective viewpoint as the point which the ray coming from Q should pass through if the mirror was not present. The concept is graphically displayed in Figure 1(b). The image shows that the mirror is rotationally symmetric and that the world point Q, the effective viewpoint E and the pinhole of the camera P are coplanar; therefore it is possible to fix an angle θ on the xy plane and reason in two dimensions. The main axis is identified with axis z and the image plane is located in $z = 0$. With reference to Figure 1(a), exploiting the radial symmetry and so fixing θ, the objective is to find a relation $\rho = \rho(\overline{\rho}, \overline{z})$ where ρ is the distance of the image point from the origin of the image plane, $\overline{\rho}$ is the distance of point Q from the main axis and $\overline{z}$ is the distance of point Q from the plane $z = 0$. The parameters of the vision system are the angle 2β at the tip of the cone, the distance d between the tip of the cone and the pinhole of the camera and the focal distance of the camera v; all of them will appear in $\rho = \rho(\overline{\rho}, \overline{z})$.

The section of a conical mirror depicted in Figure 1(b) shows that for each θ there are two symmetric effective viewpoints E. Rotating the section around the main axis by π, the effective viewpoints form a circular locus instead of a single point, thus the SVP property does not hold. However, there exists a situation in which the SVP constraint is satisfied: in [1] it is proved that if the pinhole of the camera coincides with the tip of the cone, the effective viewpoints collapse to a single point located at the tip of the cone too. But in such situation the only rays projecting onto the image plane are the rays that graze the surface of the cone. In other words, SVP constraint holds in a situation of no practical value.

With reference to Figure 1(a), the circular locus of effective viewpoints has radius b and the distance between the line perpendicular to the main axis passing through E and the camera pinhole P is l. Thus we can find

$$b = d\sin(2\beta) \quad (1)$$

$$l = 2d\sin^2\beta \quad (2)$$

The equations ruling the projection of a single point are:

$$h = \overline{z} - l - v \quad (3)$$

$$\tan\phi = \frac{h}{b + \overline{\rho}} \quad (4)$$

$$\tan\gamma = \tan(2\beta + \phi - \frac{\pi}{2}) = \frac{\tan(2\beta)\tan\phi - 1}{\tan(2\beta) + \tan\phi} \quad (5)$$

$$\rho = v\tan\gamma \quad (6)$$

Algorithm 1 The basic sketch of the algorithm to detect vertical edges.

1: Find the starting point of a line
2: Move from the current point to the next point towards the center
3: Repeat Step 2 until the segment has finished
4: Output the starting and ending points of the segment
5: Repeat from Step 1 until points are finished

with the constraint

$$\rho \geq 0 \tag{7}$$

The above constraint is needed since ρ is a distance; points $Q(\overline{\rho}, \overline{z})$ having $\rho < 0$ are world points not visible in real omnidirectional images. So, also from a physical point of view, $\rho < 0$ is not meaningful.

It is well-known that straight lines parallel to the cone axis (typically vertical lines) project onto the image plane as straight lines radially oriented. The projection of straight lines perpendicular to the cone axis (typically horizontal lines) is on the other hand more complicated, as the following discussion will show.

The equations for the projection of a straight horizontal line are reported. The differences between the new set of equations and the set given for a single point are the insertion of the equation of the projection of an horizontal line onto the plane $z = \overline{z}$ (Equation (8), see also Figure 1(c)) and the substitution of Equation (4) with Equation (9):

$$\overline{\rho}(\theta) = \frac{\overline{R}}{\sin\overline{\alpha}\sin\theta + \cos\overline{\alpha}\cos\theta} \tag{8}$$

$$\tan\phi(\theta) = \frac{h}{b - \overline{\rho}(\theta)} \tag{9}$$

with

$$\theta \in \{\overline{\alpha} + \frac{\pi}{2} < \theta < \overline{\alpha} + \frac{3\pi}{2} \mid \rho(\theta) \geq 0\} \tag{10}$$

Equation (8) is the equation of a straight line on the horizontal plane $z = \overline{z}$ in polar coordinates. Equation (9) is needed for geometrical reasons: in fact, a point at a certain angle θ is projected by the conical mirror to an image point at angle $\theta + \pi$. So, in order to plot the curve due to the projection of a line, the angles to consider are in the range specified by (10). When experimenting with various cones and horizontal lines, a small number of different shapes can be seen, as the images in Figure 2 witness.

IV. Detection of Vertical Lines

The main idea of the algorithm we propose to detect vertical edges is reported in Algorithm 1. It is an adaptation of the well-known Canny's edge detector.

Steps 1 and 2 require to detect all the possible edge points. The gradient of the image in the radial direction is computed with the appropriate Sobel mask in the radial direction. For each point $(\bar{x}, \bar{y})$, the angle between $(\bar{x}, \bar{y})$ and the center of the area to filter is computed. Then, the Sobel mask related to the closest angle is chosen to be used for edge detection in $(\bar{x}, \bar{y})$.

A point $(\bar{x}, \bar{y})$ is claimed to be the starting point of a possible segment when the gradient in $(\bar{x}, \bar{y})$ is greater than a given threshold, called the *high threshold*. Then, in Steps 2 and 3, for each point in the radial direction towards the center the condition to continue is that for each point the gradient must be greater than a second threshold called the *low threshold*. When this condition is not satisfied, the segment is considered as finished.

The use of two thresholds allows to have a finer control on the detection. In fact, the high threshold impacts directly on the number of lines found. The higher is the threshold, the lower is the number of lines found. On the other hand, the low threshold influences the length of the found edges: the lower the threshold, the longer the edges. Fine tuning of this two thresholds is in general not a simple task, but experience proved that, once suitable values are found, the effectiveness of the algorithm is not affected by slight changes in light conditions.

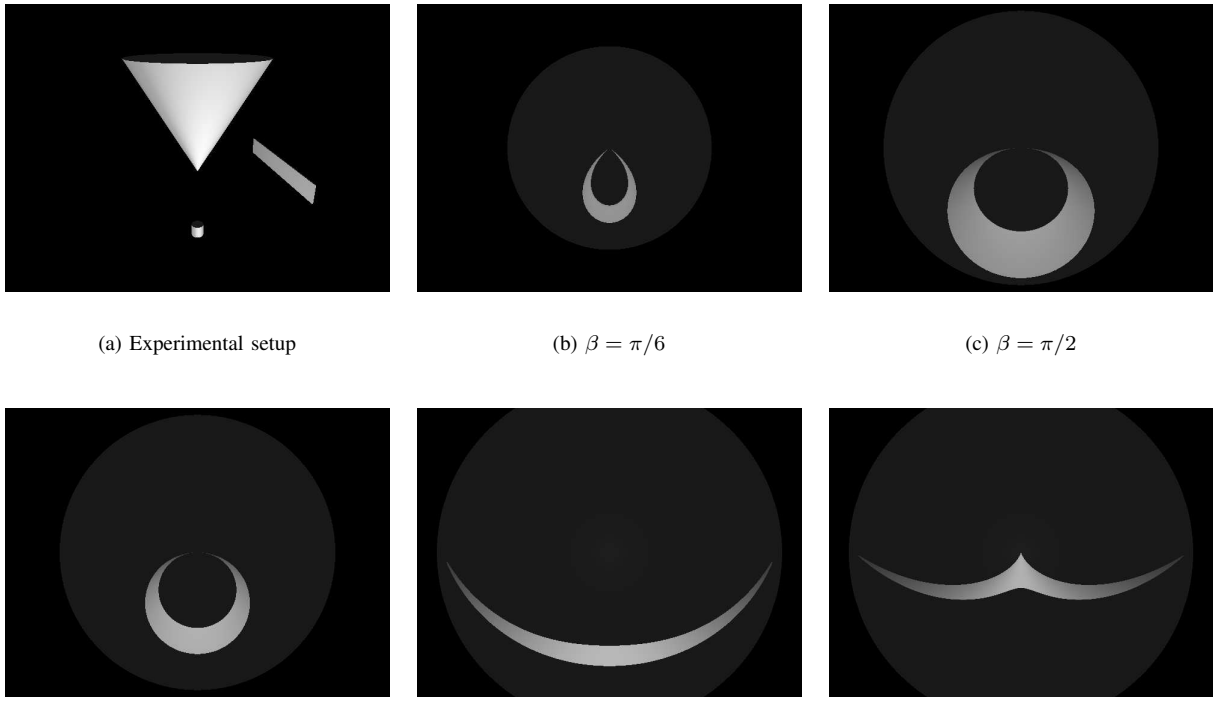
Practical considerations lead to necessary improvements of the presented algorithm. The main concern when filtering real images is the presence of noise or too many details of no interest. Low-pass filtering is helpful to avoid, at least partially but with good efficacy, the impact of this concern. Another problem is that usually, with real images, sets of edges are found when only one real edge exists due to the application of the Sobel masks. In order to fix this problem, a fast clustering algorithm that groups the found edges with respect to their angle and distance to the center of the circular area to filter is applied. Then, each cluster is substituted with a single edge, that is usually the longest edge or the mean edge of the cluster.

V. Detection of Horizontal Lines

As already described, the projection onto the image plane in general produces different complex curves, depending on the parameters of the vision system (mainly, but not uniquely, on its tip angle 2β) and the position of the horizontal edge (i.e, its distance from the main axis and its position with respect to the extremes of the lateral field of view). This reasoning suggests to use the proposed equations to devise a general algorithm able to detect 3D scene horizontal edges. The algorithm should be fast and robust to noise in order to be used successfully in real world applications. The idea to realize such an algorithm while enforcing all the constraints coming from the fast circle detector of Chen and Chung [7].

A. Random Sample Consensus (RANSAC) Algorithm

The RANSAC algorithm is used for robust fitting of models. Robustness is meant as good tolerance to outliers in the experimental data. Since RANSAC is a randomized estimator, the fitted model is correct only with a certain probability. The algorithm is very simple but powerful. The input of the algorithm is a set of data samples. The objective is to model the provided data estimating the model parameters. RANSAC is organized in two distinct phases:



(a) Experimental setup (b) $\beta = \pi/6$ (c) $\beta = \pi/2$

(d) $\beta = \pi/2$ (e) $\beta = \pi/3$ (f) $\beta = \pi/3$

Fig. 2. Curves due to the projection of horizontal lines at different elevation and distance with different cone angles.

1) Pick randomly n samples with uniform probability. n should be the smallest number of samples needed to estimate a model of the data set.
2) Evaluate the quality of the estimated model on the full data set. In this step various methods can be followed. It is common to count the number of inliers and to keep as the final model the estimated model with the greater support. The support samples are then removed from the set. Since typically the parameters estimated by RANSAC are not very precise, often parameters are recomputed with a least square fit on the support data.

Phase 1 and Phase 2 are repeated until a proper stop condition is met. Usually a good stop condition is to terminate the computation when the probability to find new right models is below a given threshold value. Obviously, the nature of the condition and the value of the threshold completely depend on the type of application RANSAC is used for.

In our case, the data set is the edge-detected version of the original image. Each candidate edge point is a sample. As derives from [4], four points are needed in order to identify a curve. A point is claimed to support the estimated curve if it lies on it or it is located at a sufficiently small distance ϵ to the curve.

B. 3D Reconstruction of Horizontal Lines

As we stated before, while vertical lines cannot be completely reconstructed, with non-SVP mirrors the projection of horizontal lines retains complete 3D information. The key aspect is the shape of the interpretation surface of a curve in the image plane, that is the surface composed by the points in the 3D scene that project onto the curve on the image plane. If this surface can contain an infinite number of straight lines, as in the case of a plane, then it is impossible to reconstruct from a single image the line that generated the curve onto the image plane. Otherwise, if the interpretation surface contains only one line (two if we consider the main axis of the vision system, always present because of the axial symmetry, and not counting the viewing rays), a complete reconstruction is possible without ambiguity. The most general and non-planar interpretation surface containing at least two straight lines (discarding the main axis and the viewing rays) is a *ruled quadrics*.

Thus, if we want to estimate all the parameters of an horizontal line, we should check whether the interpretation surface of its projection is a ruled quadrics or not. If it is a ruled quadrics, then reconstruction is impossible as well as in the planar case; otherwise, reconstruction is possible. A fundamental theorem by Caglioti and Gasparini [4] states that when omnidirectional images are obtained by a conical mirror whose tip angle 2β is included in the range $(0, \pi)$, it is possible to reconstruct *almost any* straight line of the 3D scene; the only lines that cannot be reconstructed are vertical lines (as we already saw) and lines pushed towards infinity. Therefore, horizontal lines can be safely detected in the most common cases.

C. The Algorithm for Horizontal Lines

In order to implement the two phases of the RANSAC algorithm as described in Section V-A, the projection model of Section III is needed. In fact, for Phase 1 a procedure to estimate an horizontal line with $n = 4$ candidate points is required. Phase 2 can be performed by devising an algorithm able to detect whether an edge point lies or not on the estimated line.

1) Estimation of an Horizontal Line: Three distinct image points P_1, P_2 and P_3 lying on the same curve identify three distinct viewing rays v_1, v_2 and v_3, that in turn identify a single ruled quadrics. Since a ruled quadrics contains infinite lines, a fourth point P_4 belonging to the curve is needed to localize a line univocally.

The equation of a viewing ray v_i can be written as the intersection of two planes: a vertical plane Π_i^1 passing through the origin with slope $\tan\theta_i$ on xy and a plane Π_i^2 perpendicular to Π_i^1 passing through the z axis in $(v + l + b\tan\phi_i)$ with slope $-\tan\phi_i$:

$$\begin{aligned}\Pi_i^1 &: -x\sin\theta_i + y\cos\theta_i = 0\\ \Pi_i^2 &: x\cos\theta_i + y\sin\theta_i + z\cot\phi_i - (b + \cot\phi_i(v+l)) = 0\end{aligned}$$

for $i = 1, 2, 3, 4$. $\cot\phi_i$ can be computed using Equation (6) and inverting Equation (5):

$$\tan\gamma_i = \frac{\rho_i}{v} \qquad \cot\phi_i = \frac{\tan(2\beta) - \tan\gamma_i}{\tan(2\beta)\tan\gamma_i + 1}$$

where (ρ_i, θ_i) are coordinates of the image points P_i. To make notation easier, let

$$\begin{aligned}a_i &= \cos\theta_i & b_i &= \sin\theta_i\\ c_i &= \cot\phi_i & d_i &= -(b + \cot\phi_i(v+l))\end{aligned}$$

thus obtaining

$$\begin{aligned}\Pi_i^1 &: b_i x - a_i y = 0\\ \Pi_i^2 &: a_i x + b_i y + c_i z + d_i = 0\end{aligned}$$

The parametric form of v_i can be now computed:

$$x_i = t_i \qquad y_i = \frac{b_i}{a_i}t_i \qquad z_i = -\frac{1}{a_i c_i}t_i - \frac{d_i}{c_i}$$

Since we are looking for horizontal lines, let $z = \overline{z}$ for all the viewing rays:

$$x_i = -a_i c_i\left(\overline{z} + \frac{d_i}{c_i}\right) \tag{11}$$

$$y_i = -b_i c_i\left(\overline{z} + \frac{d_i}{c_i}\right) \tag{12}$$

$$z_i = \overline{z} \tag{13}$$

In this way, we found the (x, y) coordinates of points Q_i of v_i intersecting the horizontal plane $z = \overline{z}$. When points Q_i are all lined up, then the horizontal line is found. Actually, two lines will be found: the fourth point will be used to distinguish the right solution. Furthermore, in real cases, we should not expect points Q_i to be perfectly lined up.

In order to find the desired horizontal line, first we find the line passing through Q_1 and Q_2, and then we impose it to pass through point Q_3. The result, after calculations and simplifications that we omit for brevity, is the following quadratic equation:

$$M\overline{z}^2 + N\overline{z} + O = 0 \tag{14}$$

where

$$\begin{aligned}M &= c_1c_2(a_2b_1 - a_1b_2) - c_1c_3(a_3b_1 - a_1b_3) +\\ &\quad + c_2c_3(a_3b_2 - a_2b_3)\\ N &= (a_2b_1 - a_1b_2)(c_2d_1 + c_1d_2) +\\ &\quad -(a_3b_1 - a_1b_3)(c_3d_1 + c_1d_3) +\\ &\quad (a_3b_2 - a_2b_3)(c_3d_2 + c_2d_3)\\ O &= d_1d_2(a_2b_1 - a_1b_2) - d_1d_3(a_3b_1 - a_1b_3) +\\ &\quad + d_2d_3(a_3b_2 - a_2b_3)\end{aligned}$$

The above equations can be expressed in a more compact way as a scalar product of vectors. Letting

$$\begin{aligned}A_{ij} &= a_jb_i - a_ib_j & B_{ij} &= c_id_j + c_jd_i\\ C_{ij} &= c_ic_j & D_{ij} &= d_id_j\end{aligned}$$

and

$$\begin{aligned}A &= [A_{12} \quad -A_{13} \quad A_{23}] & B &= [B_{12} \quad B_{13} \quad B_{23}]\\ C &= [C_{12} \quad C_{13} \quad C_{23}] & D &= [D_{12} \quad D_{13} \quad D_{23}]\end{aligned}$$

then

$$M = AC^T \qquad N = AB^T \qquad O = AD^T$$

It is important to note that Equation (14) is the signed distance (discarding a proportional factor) of point Q_3 from the line passing through Q_1 and Q_2.

In order to state if solutions are present, we can check whether $N^2 - 4MO > 0$: if so, then points Q_1, Q_2 and Q_3 are all perfectly lined up. Otherwise, only one or no solutions are present, and we must find the minimum (or maximum) point of the parabola. The fourth image point P_4 is useful in both cases to distinguish the right line.

In fact, when $N^2 - 4MO > 0$, a fast and straightforward way to proceed is to find the two candidate horizontal lines, find their intersections points with the viewing ray v_4 and check which one coincides with point $Q_{4(1)}$ computed for $\overline{z}_{(1)}$ or with point $Q_{4(2)}$ computed for $\overline{z}_{(2)}$. Once the right intersection point is found, $\overline{z}$ is univocally identified. Unfortunately, this method deserves great care for the application we are developing. In fact, as we saw before, neither $Q_{4(1)}$ nor $Q_{4(2)}$ can be expected to lie exactly on one line. In real cases, it could happen (and it is quite frequent) that v_4 is very close to one of the other v_i: with exact calculations there is no problem, but with an approximated approach this could lead to wrong results. This problem can be overcome by imposing a sufficient minimum distance between the four edge points P_i to pick and by defining a maximum point-line distance when checking if a point Q_i lies on a candidate line.

On the other hand, when $N^2 - 4MO \leq 0$, we just need to find the $\overline{z}$ of the minimum of the parabola of Equation (14). From a geometrical point of view, we are looking for the point Q_3 on v_3 closest to the estimated line. Once this point is found, we compute the distance of point Q_4 to the estimated line. If both points Q_3 and Q_4 are close enough to the estimated line, then the line is accepted; otherwise is rejected.

If the found $\overline{z}$ identifies a line that can be accepted, substituting $\overline{z}$ into Equations (11) and (12) for $i = 1, 2$ allows to reconstruct completely the horizontal line:

$$px + qy + r = 0 \quad (15)$$

$$p = (y_1 - y_2) \quad (16)$$

$$q = -(x_1 - x_2) \quad (17)$$

$$r = (x_1 y_2 - x_2 y_1) \quad (18)$$

so the parameters to identify are

$$\overline{R} = \frac{|r|}{\sqrt{p^2 + q^2}} \quad (19)$$

$$\overline{\alpha} = \arctan\left(\frac{q}{p}\right) \quad (20)$$

2) Criterion for a Point to Belong to a Line: Phase 2 of the RANSAC algorithm requires to evaluate whether a sample supports the evaluated model or not. In our case, we should check whether an edge point approximately lies on the estimated horizontal line. Such test should be very fast, since often the points to consider are many.

The idea is to reuse the results obtained from the previous Phase to save computation. Given an edge point $P_i(x_i, y_i)$, the first task is to compute ρ_i, $\sin\theta_i$ and $\cos\theta_i$:

$$\rho_i = \sqrt{x_i^2 + y_i^2} \qquad \cos\theta_i = \frac{x_i}{\rho_i} \qquad \sin\theta_i = \frac{y_i}{\rho_i}$$

From Section V-C.1, we know parameters p, q and r of the vertical plane of the horizontal line (Equations (16), (17) and (18)). To exploit Equation (8), we can note that

$$\cos\overline{\alpha} = \frac{p}{\sqrt{p^2 + q^2}} \qquad \sin\overline{\alpha} = \frac{q}{\sqrt{p^2 + q^2}}$$

and since $\sqrt{p^2 + q^2}$ was already computed to evaluate $\overline{R}$ (Equation (19)), the effort to perform this task is very small.

Then, we can use Equations (8), (9), (5) and (6) to find $\hat{\rho}_i$ with a small computation effort. After that, we compare ρ_i and $\hat{\rho}_i$: if they are approximately equal, then the current edge point is considered as part of the estimated horizontal line. Mathematically, the condition is

$$|\rho_i - \hat{\rho}_i| < \epsilon$$

with ϵ defined in a proper way by the user.

3) Stop Conditions: Only part of the points detected by computing the gradient are really edge points. This means that generally there are some points that can not support any estimated line. Therefore, when only these points remain, the algorithm must stop.

In order to detect if only useless (i.e. noise) points are left, we introduce two parameters. The first parameter is called MIN_POINTS: in Phase 2, the support of each line is built and all the corresponding samples are removed. When the number of samples left are smaller than MIN_POINTS, the computation ends because we state that there are too few points to support new lines.

Furthermore, a second parameter MAX_FAILURES is introduced and a corresponding variable FAILURE is managed. A failure occurs whenever the estimation process cannot be concluded either when the four points P_1, P_2, P_3 and P_4 do not identify a line or when the other points do not sufficiently support the estimated line.

Every time a failure occurs, variable FAILURE is incremented by 1; if a line is accepted, FAILURE is set to zero. After some lines are accepted, the remaining points are likely to be noise, and so FAILURE increases. When the number of consecutive failures increases, the probability that a line can be supported by the remaining points quickly becomes small. Therefore, when FAILURE is greater than a certain parameter value MAX_FAILURES, the computation be can safely terminated. The combined effects of the two parameters MIN_POINTS and MAX_FAILURES completely identify the conditions of termination of the adapted RANSAC algorithm.

VI. Experimental Results

Experimental evaluation is aimed at demonstrating the ability of the described algorithms both to detect straight horizontal and vertical lines of the environment and to reconstruct the exact position of horizontal lines.

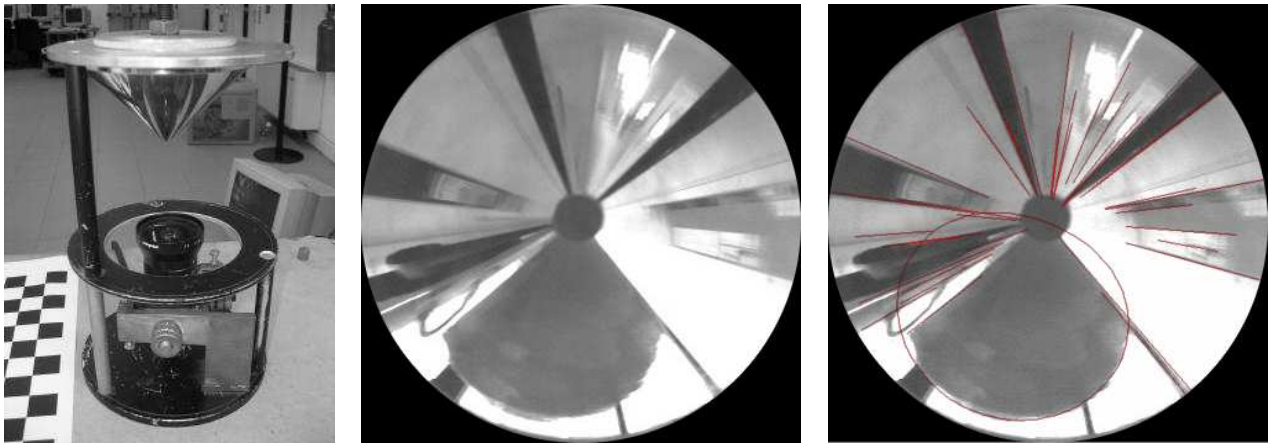
A calibration procedure is not available at the moment for the vision system in Figure 3(a), therefore the value of the parameters needed by the algorithms were manually measured when possible (e.g., the cone angle that is $\pi/2$) or the nominal values were chosen (e.g., the focal length v of the camera).

The proposed algorithms are robust to a poor calibration with real images. The test image is reported in Figure 3(b) and the result of the application of the algorithms is in Figure 3(c): all the radial lines and the long curve in the image are correctly detected without previous low-pass filtering.

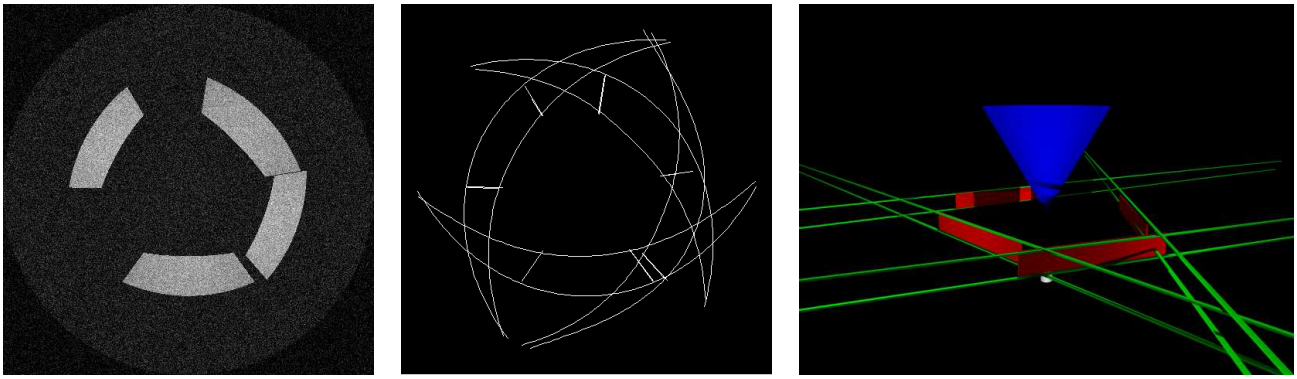
The second test is performed to prove the ability of the proposed algorithm to correctly reconstruct the position of the horizontal lines of the 3D scene and to show the generality of the algorithms proposed. In order to properly validate the obtained results with the right expected values, synthetic images were used. Gaussian noise was added to further test robustness. The simulated vision systems is composed by a camera whose intrinsic parameters are known and a cone with tip angle equal to $2\pi/3$.

VII. Conclusions and Future Work

In this paper we presented two algorithms for 3D scene horizontal and vertical straight line detection. The algorithms proved to be robust to noise and poor calibration. The algorithms are parametric with respect to the most important aspects of the vision system: the tip angle, the base radius of



(a) The real vision system. (b) Real image (c) Lines detection



(d) Sensed image (e) Lines detection (f) Reconstructed horizontal lines

Fig. 3. Images used for experimental evaluation and obtained results.

the conical mirror, the distance between the tip and the pinhole of the camera, and the focal length of the perspective camera.

Calibration of the vision system is an important open issue. The results obtained by the algorithms without calibration are encouraging, but, in order to exploit them in a real world application, automatic calibration is fundamental. For this reason, current research is devoted to devise a robust calibration method and to generalize the projection model of Section III to include small offsets from the ideal relative positioning of the camera and the conical mirror. Further future work involves a fast least square fit of the estimated parameters of the horizontal lines or an improved approach to 3D line reconstruction and inliers detection [13].

REFERENCES

[1] S. Baker and S. K. Nayar, "A theory of catadioptric image formation," in *Proc. of the 6th International Conference on Computer Vision*, 1998.

[2] S. Bogner, "Introduction to panoramic imaging," in *IEEE SMC Conference*, October 1995.

[3] E. Brassart, L. Delahoche, C. Cauchois, C. Drocourt, C. Pegard, and E. M. Mouaddib, "Experimental results got with the omnidirectional vision sensor: Syclop," in *Proc.of OMNIVIS '00*, June 2000.

[4] V. Caglioti and S. Gasparini, "On the localization of straight lines in 3d space from single 2d images," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 1129–1134.

[5] C. Cauchois, E. Brassart, L. Delahoche, and A. Clerentin, "3d localization with conical vision," in *Proc. OMNIVIS '03*, 2003.

[6] J. R. Charles, "How to build and use an all-sky camera," in *Astronomy Magazine*, April 1987.

[7] T.-C. Chen and K.-L. Chung, "An efficient randomized algorithm for detecting circles," in *Computer Vision and Image Understanding*, 2001.

[8] S. Derrien and K. Konolige, "Approximating a single viewpoint in panoramic imaging devices," in *Proceedings of the IEEE Workshop on Omnidirectional Vision*, June 2000, pp. 85–90.

[9] M. Fiala and A. Basu, "Hough transform for feature detection in panoramic images," in *Pattern Recognition Letters*, 2002.

[10] C. Geyer and K. Daniilidis, "Catadioptric projective geometry," in *International Journal of Computer Vision*, December 2001.

[11] S. Kawato, Y. Yagi, and S. Tsuji, "Real-time omni-directional image sensor (copis) for vision-guided navigation," in *IEEE Transactions on Robotics and Automation*, no. 10, 1994.

[12] L. Spacek, "Omnidirectional catadioptric vision sensor with conical mirrors," November 2003.

[13] S. Teller and M. Hohmeyer, "Determining the lines through 4 lines," *Journal of Graphics Tools*, 1999.

[14] Y. Yagi and S. Kawato, "Panorama scene analysis with conic projection," in *IEEE International Workshop on Intelligent Robots and Systems*, 1990.