A Robust Uncalibrated Visual Compass Algorithm from Paracatadioptric Line Images

Gian Luca Mariottini[1] and Stefano Scheggi, Fabio Morbidi, Domenico Prattichizzo[2]

[1] Dept. of Computer Science and Engineering
University of Minnesota
4-192 EE/CS Building, 200 Union St. SE,
Minneapolis, MN 55455, USA
Email: gianluca@cs.umn.edu
[2] Dipartimento di Ingegneria dell'Informazione
Università di Siena
Via Roma 56, 53100 Siena, Italy
Email: {scheggi,morbidi,prattichizzo}@dii.unisi.it

Abstract. Due to their wide panoramic field of view, paracatadioptric cameras are becoming ubiquitous in many robotic applications. A challenging problem consists in using these vision sensors as a visual compass, that is to exploit the image data solely to provide an estimate of their rotational motion when mounted on a mobile/humanoid robot. Existing visual compass algorithms are difficult to implement in real scenarios since they assume known camera calibration parameters and known geometry of image features (e.g. correspondence between points or parallelism among between lines).
In this paper we present a new multi-view property for paracatadioptric cameras called *disparity-circles constraint* and we use it to design a visual compass algorithm to estimate the $z-$axis camera rotation angle. The proposed algorithm only uses the image projection of 3-D parallel lines and is suitable for real-time implementation. Those 3-D lines that are parallel are shown to be automatically detectable via a RANSAC implementation of the algorithm. Real-data experiments conducted with a paracatadioptric camera mounted on a mobile robotic platform prove the robustness of the proposed approach.

1 Introduction

In the last few years we witnessed a growing interest in autonomous robot navigation for which is essential the correct estimation of the robot pose (orientation and translation). However, some of the most widely used on-board sensors suffer from many limitations. For example, standard GPSs have a substantial error (of order of 10 m) and require line of sight to the satellite constellation, which rules out operation in many indoor, underwater and also urban environments. IMUs (Inertial Measurement Units) provide an estimate of axes rotation angles through double integration of sensed accelerations. This means that even small errors will be integrated over time and then result in large localization error over long paths.

Due to the above limitations, passive vision sensors represent an appealing alternative: they are cheap and provide information of the surrounding environment richer

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

than other traditional devices (e.g. laser range finders, sonars). In this paper, we are interested in *paracatadioptric cameras* consisting in a coupling between a parabolic mirror (*catoptric*) and a refractive (*dioptric*) lens [4]. Due to this coupling, their field of view is wider than standard pinhole cameras so that a higher number of features can be observed. It is worth highlighting herein that, despite these sensors have been extensively studied in the literature [2, 8], several new geometric properties have yet to be discovered: in particular, all those related to the multi-view observation of other features than points, e.g., lines (common in many man-made environments) [1].

A challenging problem consists in using paracatadioptric cameras as a *visual compass*, that is to provide an estimate of the camera rotational motion from the image data solely, when mounted on a mobile/humanoid robot. Many applications can benefit from a visual compass strategy, e.g., autonomous navigation in unknown environments of single and multiple robots [16, 19], visual servoing [3], SLAM [5], real-time generation of 3-D models using mobile cameras [7], etc. A visual compass algorithm has been proposed in [12] and uses lines in a single pinhole view to exploit vanishing points. However, camera calibration parameters are assumed known and an initialization stage is needed to estimate the camera orientation with respect to the scene. Another visual compass algorithm for calibrated pinhole camera has been proposed in [17] and it allows one to retrieve full 3-axis orientation from the observation of a high number of image points. A correspondence-free approach to the computation of camera orientation has been proposed in [13]. Nevertheless, the internal camera calibration parameters are supposed known and the algorithm (at its present stage) is not suitable for real-time implementation.

As an original contribution, in this paper we present a new geometrical property, called *disparity-circles constraint*, which relates two paracatadioptric views of (at least) two 3-D parallel lines. This property is the core of a visual compass algorithm which uses image data solely for retrieving the $z-$axis rotation angle of a paracatadioptric camera with respect to a reference view (e.g., the first acquired frame). The algorithm only exploits 3-D parallel lines, does not need any prior on the camera orientation, is fully uncalibrated and suitable for real-time implementation. We integrated RANSAC to make the algorithm able to automatically select which lines are parallel in 3-D (inliers) and to robustly estimate the camera rotation angle.

The rest of the paper is organized as follows. Section 2 presents basic results and algorithms used through the paper. In Section 3, the disparity circle constraint is introduced and the visual compass algorithm is presented. Simulation and experimental results are reported in Section 4, to show the effectiveness and the practical applicability of the proposed approach. In Section 5, we provide some concluding remarks and highlight the main contributions of the paper.

2 Basics on paracatadioptric projection and circle fitting

In what follows we briefly review the basics on paracatadioptric projection of 3-D lines. These lines project to arcs of circles in the image plane. We then illustrate and compare some of the existing methods to extract from the arcs the parameters of the entire circle in the paracatadioptric image plane.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

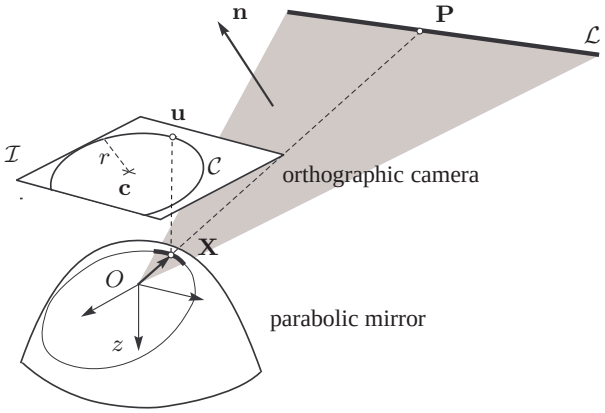


Fig. 1. The interpretation plane through the mirror focus center O and the 3-D line $\mathcal{L}$ intersects the mirror at a curve that is orthographically projected onto the image plane in a circle $\mathcal{C}$ with center $\mathbf{c}$ and radius r.

2.1 Paracatadioptric image of a 3-D line

Fig. 1 reports the imaging model of a paracatadioptric camera with the mirror focus at O. Every scene point $\mathbf{P} \in I\!R^3$ is projected onto the mirror surface at $\mathbf{X} \in I\!R^3$ through O and, finally, via an orthographic projection onto the image plane $\mathcal{I}$ at $\mathbf{u}$. The projection from $\mathbf{P}$ to $\mathbf{u}$ is analytically described by a nonlinear function $\eta : I\!R^3 \rightarrow I\!R^2$ [8,18] that depends on both the camera calibration parameters and the mirror geometry.

Let $\mathcal{L}$ be a 3-D line observed by the paracatadioptric camera. The *interpretation plane* is defined as the plane through $\mathcal{L}$ and the mirror focus O and has normal vector $\mathbf{n} = [n_x \; n_y \; n_z]^T$ (in the mirror frame at O).

Proposition 1. *(Paracatadioptric image of a line [10]) Suppose given the setup of Fig. 1, in which a line $\mathcal{L}$ is observed by a paracatadioptric camera. If $n_z \neq 0$, then the line $\mathcal{L}$ is projected onto an image* circle *$\mathcal{C}$ with center* $\mathbf{c}$ *(pixels) given by:*

$$\mathbf{c} \triangleq \begin{bmatrix} c_u \\ c_v \end{bmatrix} = \begin{bmatrix} u_0 - 2af_\alpha(n_x/n_z) \\ v_0 - 2af_\alpha(n_y/n_z) \end{bmatrix},$$

and radius (pixels)

$$r = 2a\, f_\alpha\, (1/n_z)\,,$$

where a is the focal mirror parameter, (u_0, v_0) the optical center and f_α the focal length of the camera (pixels).

As note in Prop. 1, a singular case can occur when the line to be projected is perpendicular to the camera plane of motion ($n_z = 0$). This occurrence will be however automatically discarded by our visual compass algorithm.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

2.2 Circle fitting algorithms

We illustrate here some basics for the extraction of n circles in the image plane, corresponding to the projection of n 3-D lines.

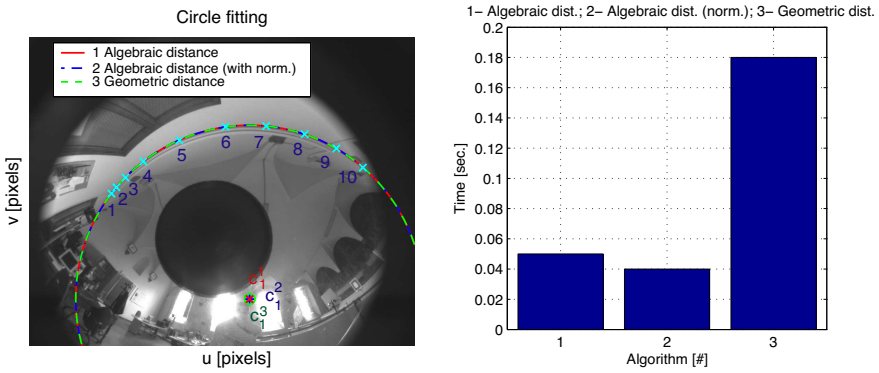
As a first preprocessing step, the image is filtered with a Median filter so that the edges are preserved. A Canny edge detector is then applied to obtain potential line points. In the second step, we run an edge eroding followed by a chaining strategy, to extract only the connected pixels along the strongest edges. (All the above strategy are available in MATLAB and OpenCV). To enhance robustness we extract only those arcs whose perimeter is greater than a certain threshold (in pixels) and sample from them some *control points* $\{\mathbf{u}_j\}_{j=1}^{m}$ with $m \geq 3$. We assume that our points are not affected by outliers. We are herein interested in evaluating the performances of three circle fitting methods for the estimation of the parameter vector $\mathbf{p} \triangleq [c_u \; c_v, \;\; r]^T$ of a circle $\mathcal{C}$ (i.e., its center $\mathbf{c}$ and radius r).

The first method consists in minimizing the *algebraic cost function* [9], i.e. a measure of the distance of the circle form the control points:

$$\sum_{j=1}^{m} \left((\mathbf{c} - \mathbf{u}_j)^T (\mathbf{c} - \mathbf{u}_j) - r^2 \right)^2 . \tag{1}$$

The estimated vector parameters $\mathbf{c}$ and r are computed in closed-form as:

$$\mathbf{c} = -\frac{1}{2}\mathbf{A}^{-1}\mathbf{b} \quad \text{where} \quad \mathbf{b} \triangleq \sum_{i,j,k}^{m} (\mathbf{u}_i^T \mathbf{u}_i - \mathbf{u}_j^T \mathbf{u}_j)(\mathbf{u}_k - \mathbf{u}_i)$$
$$\text{and} \quad \mathbf{A} \triangleq \sum_{i,j,k}^{m} (\mathbf{u}_k - \mathbf{u}_i)(\mathbf{u}_j - \mathbf{u}_i)^T \tag{2}$$



(a) (b)

Fig. 2. Circle fitting comparison. (a) A line is detected in the image plane. A set of $m = 10$ control points is used to fit the circle. (b) Computation times.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

and

$$r = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\mathbf{c}-\mathbf{u}_j)^T(\mathbf{c}-\mathbf{u}_j)}. \quad (3)$$

Due to its simple formulation, this first method is fast and thus suitable to real-time implementations. This kind of algebraic method can be affected by the so-called *high-curvature bias problem* [20] (i.e., high curvature points are less-informative for curve estimation): however, this is not critical in our algorithm since each point on a circle has the same curvature. Anyhow, the minimization based on algebraic cost function has the drawback of being not invariant w.r.t. euclidean transformations (e.g., translations, rotations, etc.).

To address this problem, a slight variation of (1) includes an initial image point *normalization step*: all the image control points $\{\mathbf{u}_j\}_{j=1}^m$ are first translated to the origin by their centroid. The above linear algebraic method is then applied. Finally, the image points are de-normalized. Note that data normalization is always addressed as an essential step [11] and should be not considered optional.

Another strategy for circle fitting consists in minimizing a *geometric cost function*. In the case of curve fitting, the orthogonal distance d_j between a point $\mathbf{u}_j$ and the fitted circle $\mathcal{C}$ is the smallest Euclidean distance among all distances between $\mathbf{u}_j$ and the points on the curve. For a circle, d_j can be measured along the radius direction and can thus be simply written as:

$$d_j(\mathbf{c},r)^2 = (\|\mathbf{c}-\mathbf{u}_j\|_2 - r)^2 \qquad \forall\, j = 1,...,m. \quad (4)$$

The geometric distance minimization problem in (4) can be approached using nonlinear minimization techniques (e.g., Levenberg-Marquardt).

In order to find the best fitting algorithm for our visual compass strategy, we implemented all the above presented methods. Our tests have been performed in MATLAB on a 2.2 GHz Intel core 2 Duo processor, with 4 GB of RAM. The fitted circles are reported in Fig. 2(a). While all the methods give the same fitting (in fact the minimization of (1) and of (4) clearly leads to the same result in the case of circles), however the computational load is different. The algebraic method with normalization is the fastest one (see Fig. 2(b)). In fact, data normalization induces a better conditioning when using SVD in the linear algorithm. Due to the real-time requirements of our application, we then selected the *linear fitting method* with *data normalization*.

To conclude note that, as the camera/robot moves, the detected circles can be tracked in the image plane using a line tracking software, like ViSP [14].

3 Uncalibrated visual compass algorithm

In this section we present our visual compass algorithm for the estimation of the camera rotation angle θ about the $z-$axis. The algorithm is based on the new *disparity circles constraint*, does not assume any a priori knowledge of the camera calibration parameters and only uses the circles automatically detected on the image plane (see Sect. 2.2). The automatic selection of image circles corresponding to 3-D parallel lines has been made possible by integrating RANSAC estimation with the disparity-circles constraint.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

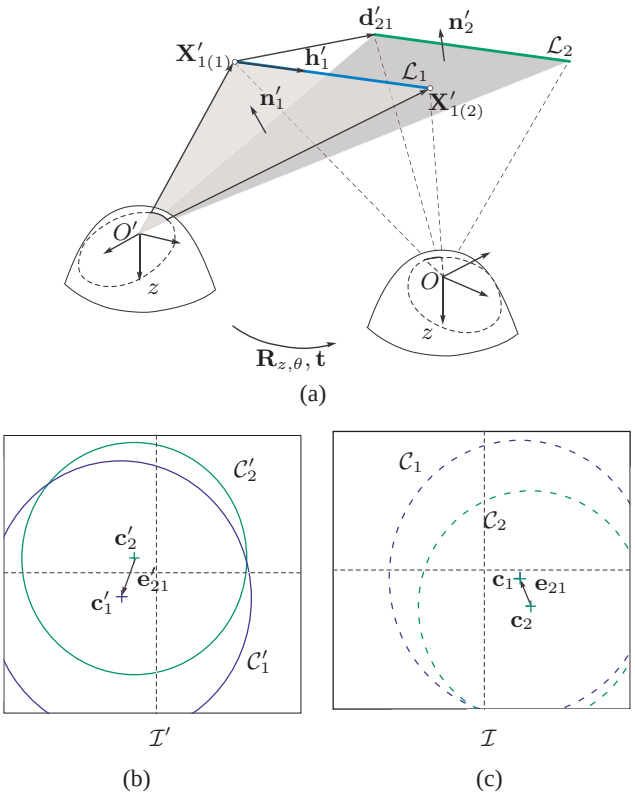


Fig. 3. *Paracatadioptric visual compass*. (a) Two paracatadioptric cameras are rotated by an angle θ about the z-axis and observe two parallel lines $\mathcal{L}_1$ and $\mathcal{L}_2$; (b)-(c) The two lines are projected in each image plane $\mathcal{I}$ and $\mathcal{I}'$, at two circle pairs $(\mathcal{C}_1, \mathcal{C}_2)$ and $(\mathcal{C}'_1, \mathcal{C}'_2)$, respectively. From the circle centers we can compute the vectors $\mathbf{e}'_{21}$ and $\mathbf{e}_{21}$ that are used to retrieve the angle θ.

3.1 Disparity-circles constraint

Suppose given two paracatadioptric views of two parallel lines $\mathcal{L}_1$ and $\mathcal{L}_2$ (Fig. 3(a)). Let the mirror frames be centered in O and O'. Owing to Prop. 1, the lines $\mathcal{L}_1$ and $\mathcal{L}_2$ project in each view at two circle pairs: $(\mathcal{C}_1, \mathcal{C}_2)$ in the first view and $(\mathcal{C}'_1, \mathcal{C}'_2)$ in the second view, respectively (Figs. 3(b)-(c)). Without loss of generality, we will assume that the world frame is coincident with the paracatadioptric mirror frame in O', which we will refer to as *reference*. For the purposes of our work we will suppose that a rigid body motion $(\mathbf{R}_{z,\theta}, \mathbf{t})$ occurs between the two views, being $\mathbf{R}_{z,\theta}$ a 2×2 rotation about the z-axis of an angle θ.

Theorem 1 (Disparity-circles constraint). *Consider the setup in Fig. 3 where the* current *paracatadioptric camera at O is rotated of an angle $\theta \in [-\pi/2,\ \pi/2]$ about its $z-$axis with respect to the* reference *camera at O'. Two 3-D parallel lines $\mathcal{L}_1$ and $\mathcal{L}_2$ are projected in the two image planes $\mathcal{I}$ and $\mathcal{I}'$, in two circles $(\mathcal{C}_1, \mathcal{C}_2)$ and $(\mathcal{C}'_1, \mathcal{C}'_2)$,*

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

respectively (see Figs. 3(b)-(c)). Let their centers (in pixels) be $\mathbf{c}_1, \mathbf{c}_2$ *and* $\mathbf{c}'_1, \mathbf{c}'_2$*, respectively. Under the previous assumptions, the following equation holds true:*

$$\mathbf{e}'_{21} = \mathbf{R}_{z,\theta}\, \mathbf{e}_{21} \tag{5}$$

being $\mathbf{e}_{21} \triangleq \mathbf{c}_2 - \mathbf{c}_1$ *and* $\mathbf{e}'_{21} \triangleq \mathbf{c}'_2 - \mathbf{c}'_1$.

Proof: See the Appendix.

Remark 1 (Extension to n circles). In the case of n parallel lines $\mathcal{L}_i$, $i = 1, ..., n$, the disparity-circles constraint (5) can be generalized as follows:

$$\mathbf{y} = \mathbf{R}_{z,\theta}\, \mathbf{x} \tag{6}$$

where the columns of the matrices $\mathbf{x}$ and $\mathbf{y}$ are composed by all the p possible (non repetitive) combinations of vectors e_{jk} and e'_{jk} between the centers of the n circles on $\mathcal{I}$ and $\mathcal{I}'$, respectively. Hence, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{2\times p}$, with $p = \binom{n}{2}$. Note that (6) corresponds to the *Procrustes problem* which has an elegant closed form solution given by $\mathbf{R}_{z,\theta} = \mathbf{V}\,\mathbf{U}^T$ where $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{M})$ being $\mathbf{M} = \sum_{i=1}^{p} \mathbf{x}_i \mathbf{y}_i^T$. This solution could be used to initialize an iterative estimation method (e.g., Levenberg-Marquardt) for improving the estimation of $\mathbf{R}_{z,\theta}$.

Algorithm 1 resumes the main steps of our *visual compass algorithm* based on condition (6).

The proposed visual compass algorithm has an attractive geometrical interpretation sketched in Figs. 4(a)-(b): first, take two paracatadioptric views in two different poses (a $z-$axis camera rotation and a translation occur between them). Second, fit a circle to each line and then superimpose them as in Fig. 4(c). Compute the center of the four circles $\mathbf{c}'_1$, $\mathbf{c}'_2$, $\mathbf{c}_1$ and $\mathbf{c}_2$: after the computation of the centers of the two circles, the angle θ between the lines $\overline{\mathbf{c}'_1\mathbf{c}'_2}$ and $\overline{\mathbf{c}_1\mathbf{c}_2}$ is exactly the $z-$axis rotation angle θ between the two cameras.

Algorithm 1 Uncalibrated Paracatadioptric Visual Compass Algorithm

1: Let be given a *reference* paracatadioptric image $\mathcal{I}'$ taken at O' and a set of n 3-D parallel lines $\mathcal{L}_i$, $i = 1, ..., n$.
2: In $\mathcal{I}'$, use the linear circle fitting method with data normalization (see Sect.2.2) to detect all the centers $\mathbf{c}'_i$ of the n circles $\mathcal{C}'_i$. From the circle centers compute the vector $\mathbf{y}$ (recall Remark 1).
3: Let be given a *current* paracatadioptric image $\mathcal{I}$ taken at O, rotated by $\mathbf{R}_{z,\theta}$ and translated of $\mathbf{t}$.
4: In $\mathcal{I}$, use the linear fitting method of Sect.2.2 to detect all the centers $\mathbf{c}_i$ of the n circles $\mathcal{C}_i$. From the circle center determine the vector $\mathbf{x}$.
5: Determine the matrix $\mathbf{M} = \sum_{i=1}^{p} \mathbf{x}_i \mathbf{y}_i^T$, and do $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{M})$. The camera rotation matrix is given by

$$\mathbf{R}_{z,\theta} = \mathbf{V}\,\mathbf{U}^T.$$

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

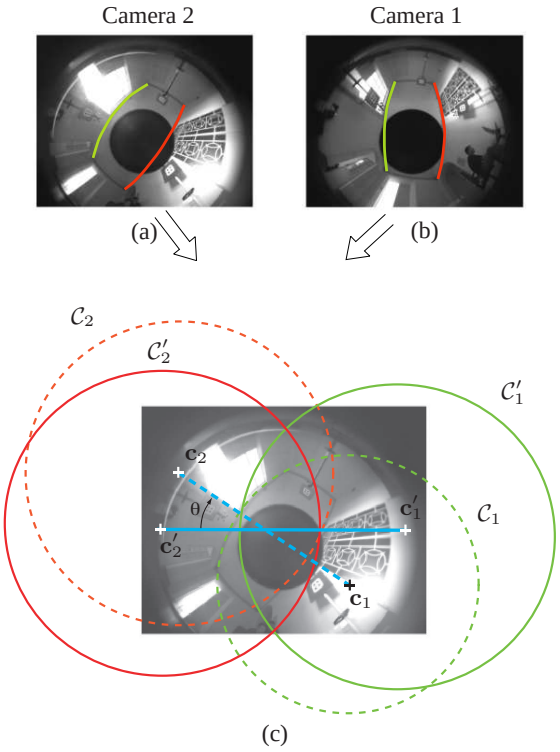


Fig. 4. *Illustration of the visual compass algorithm*: two roto-translated views (a) and (b) of two parallel lines are superimposed and circles are fitted to the lines (c). Circle centers are joined with segments to obtain the angle θ between the two views (a) and (b).

3.2 Robust visual compass

RANSAC (RANdom SAmple Consensus) [6] is an iterative method to estimate the parameters of a model from random samples of a minimal set of data (the centers of two circles, for the case of the disparity circle constraint). The model to be satisfied in the visual compass is the disparity circle constraint (5), whose scalar parameter to be estimated is the camera angle θ. A basic assumption in RANSAC is that the data are composed of a sufficient number of "inliers" (i.e., data who fit the model) and "outliers" (i.e., data who do not). RANSAC works properly if the inliers are more than the 50% of all the data.

Our RANSAC implementation of the visual compass algorithm counts the number of data verifying (5) up to a specified threshold and finally retains the best consensus. At the end of the iterations, RANSAC also provides a classification of data into inliers and outliers. In our case this is particularly appealing since at the end of all the iterations the circles will be *automatically* classified into "parallel" and "non-parallel/singular", where "singular" corresponds to the case of vertical edges, i.e. $n_z = 0$ (see Sect.2.1).

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

Finally note that, for n circles, the (maximum) number of iterations made by RANSAC will equal the number of possible (non-repetitive) associations of circle pairs among the two images $\mathcal{I}$ and $\mathcal{I}'$, i.e. $\binom{n}{2}^2$.

Some concluding remarks are in order at this point.

- The proposed visual compass algorithm is *uncalibrated* in the sense that it does not require any prior knowledge of the internal camera parameters (i.e., both lens and mirror parameters).
- At least *two 3-D parallel lines* are sufficient for computing θ. Note that the existence of parallel lines is a reasonable assumption in many man-made environments (e.g. streets, rooms, corridors). Image circles corresponding to 3-D parallel lines are automatically detected by the RANSAC implementation of the visual compass algorithm.
- The visual compass algorithm, which makes use of the *disparity-circles constraint* to estimate the camera rotation angle θ,is computationally sound and suited for realt-time implementation.
- The algorithm still works when *no translational displacement* exists, i.e., only a pure rotation occurs between the two views. Other existing strategies, like the ones that estimate and decompose the essential matrix, fail in the case of zero translation.

4 Experiments

4.1 Simulation results

All the simulation results reported in this section, have been obtained using the Epipolar Geometry Toolbox (EGT) for MATLAB [15]. We implemented the unified catadioptric model by Geyer and Daniilidis [8] to simulate the paracatadioptric camera projection, modeled as a projection to a sphere and then to a plane (see Fig. 5(a)).

In the first simulation experiment, the visual compass algorithm is applied to the nominal case (i.e., no image noise) with four parallel lines $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_4$ and $\mathcal{L}_5$, while $\mathcal{L}_3$ is perpendicular to the camera plane of motion. All these lines are imaged in two paracatadioptric views in O and O' (see Fig. 5(a)). The camera in O has been placed with $\mathbf{R}_{z,\theta}$ and $\mathbf{t} = [7,\ 7,\ 0]^T$, with $\theta = \pi/4$. Without loss of generality, we assumed that O' coincides with the origin of the reference frame. The RANSAC version of the visual compass algorithm provides an exact estimate of $\theta = \pi/4$. This means that the algorithm has been able to discard the outlier (i.e., the projection of $\mathcal{L}_3$). Figure 5(b) reports the resulting *association matrix*: the columns and rows of this matrix contain the indexes (i, j) and (i', j') of all the possible associations between the circles $(\mathcal{C}_i, \mathcal{C}_j)$ and $(\mathcal{C}'_i, \mathcal{C}'_j)$ in the current and reference images. As a measure of association, for each pair association (i, j)-(i', j') we simply counted the number of times the angle estimated by this pair has been voted by the other pairs (the "consensus") in all the other possible associations (or RANSAC iterations). In Fig. 5(b), different gradations of grey have been employed: a black area corresponds to a high consensus, i.e. a high association, while a white area to an absence of association. For instance, at the intersection $(1, 2) - (1', 2')$ a high association is obtained (black area) meaning that the circles $(\mathcal{C}_1, \mathcal{C}_2)$ and

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

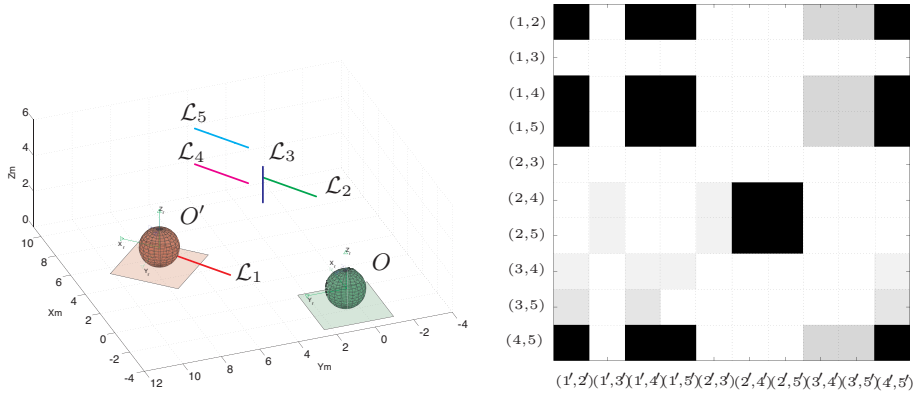
$(\mathcal{C}'_1, \mathcal{C}'_2)$ are highly associated (i.e. they are indeed parallel in 3-D). On the other hand, all the associations involving the line $\mathcal{L}_3$ give a *small* association, meaning that this line is not parallel to the other ones and must be ruled out.

In order to test the robustness of our algorithm to noisy data, we added gaussian noise with standard deviation σ to the image. Fig. 6 reports the mean and the standard deviation of the estimation error $|\hat{\theta} - \theta|$, over a set of 250 realization of the same experiment, for a fixed σ and number of control points. Note that we used an increasing number (from 3 to 7) of image control points (used for the circle fitting algorithm), and σ from 0 to 3 pixels. From Fig. 6, we see that the higher the number of control points, the lower are the mean and standard deviation errors. Moreover, we observe that an increase of noise power does not correspond to an exponential increase of the estimation error. These results show that our algorithm is robust to noisy measurements and thus applicable in real scenarios.

4.2 Experimental results

This section reports the results obtained by running our visual compass algorithm with real data taken from a paracatadioptric camera mounted on a mobile robot ActivMedia PIONEER 2X-DE. The robot motion is shown in Fig. 8(a). As the robot moves from location $\{1\}$ to $\{4\}$, our aim is to compute the relative camera orientation θ with respect to a Reference image. For the sake of clearness, Fig. 8(a) shows the silhouette of the robot both in the initial and Reference position.

The detected circles in the initial frame are shown in Fig. 7(a). Among these, note that $\mathcal{L}_3$ is orthogonal to the other lines. After the first run of the algorithm, only the lines $\mathcal{L}_1$ and $\mathcal{L}_2$ have been selected by the algorithm as the minimum two providing the



(a) (b)

Fig. 5. *Simulation results.* (a) Two parallel lines $\mathcal{L}_1$ and $\mathcal{L}_2$ are viewed by two paracatadioptric cameras in O and O'. The camera in O is translated by $\mathbf{t}$ and rotated about the z-axis of an angle $\theta = \pi/4$; (b) Association matrix: dark grey corresponds to a high association index while light grey to low association.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

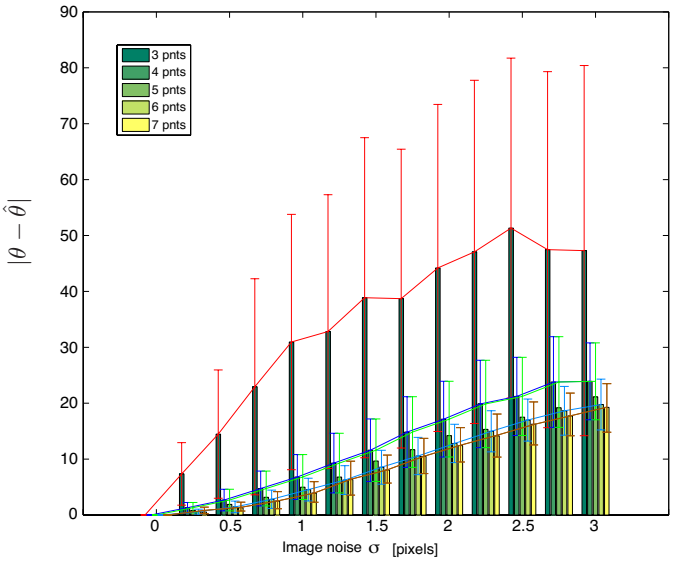


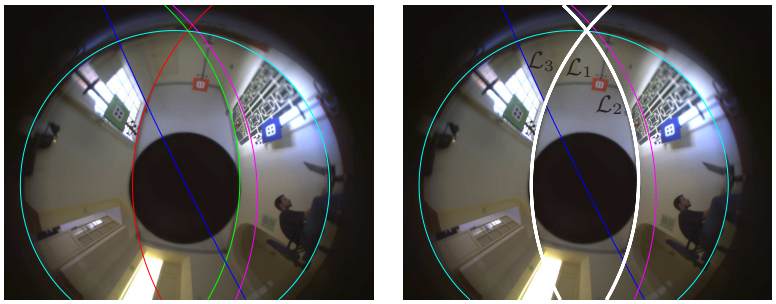
Fig. 6. *Simulation results*: Estimation error $|\theta - \widehat{\theta}|$ VS image noise power σ and fitting points number.

best estimate (i.e., as inliers, plotted in white in Fig. 7(b)). These lines are used during all the experiment to estimate the angle θ. Figure 8(b) shows the estimated angle $\hat{\theta}$ in correspondence with the robot waypoints $\{1\}$-$\{4\}$. As expected, in $\{4\}$ the estimated robot orientation is near zero because the robot is aligned with the Reference view.

5 Conclusion and future works

In this paper we have presented a new multi-view property called *disparity-circles constraint* valid for paracatadioptric cameras and used it to design a visual compass algorithm for the computation of the $z-$axis camera rotation angle. The algorithm does not require any knowledge of the internal camera calibration parameters, is suitable for real-time implementation and only uses the image projection of 3-D parallel lines. A RANSAC implementation of the algorithm makes it able to automatically detect those lines that are parallel in 3-D. Extensive simulation results have shown the effectiveness of the proposed paracatadioptric visual compass and its applicability in real scenarios. Experimental results with a mobile robotic platform are also presented. Future works will deal with the extension of this algorithm to full-axis orientation retrieval.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255



(a) (b)

Fig. 7. *Experimental results*: (a) Detected circles in the first frame; (b) Line $\mathcal{L}_1$ and $\mathcal{L}_2$ give the best estimate and line $\mathcal{L}_3$ discarded as outlier.

A Appendix: Proof of Theorem 1

Proof. The interpretation planes through $\mathcal{L}_1$ and $\mathcal{L}_2$ (with respect to O'), have normal vectors $\mathbf{n}'_1 \triangleq [n'_{1x}\ n'_{1y}\ n'_{1z}]^T$ and $\mathbf{n}'_2 \triangleq [n'_{2x}\ n'_{2y}\ n'_{2z}]^T$ given by:

$$\mathbf{n}'_1 \sim \mathbf{X}'_{1(2)} \times \mathbf{X}'_{1(1)} \tag{7}$$

$$\mathbf{n}'_2 \sim \mathbf{X}'_{2(2)} \times \mathbf{X}'_{2(1)}. \tag{8}$$

Between any point $\mathbf{X}'_{2(i)}$ on $\mathcal{L}_2$ and the point $\mathbf{X}'_{1(j)}$ on $\mathcal{L}_1$, the followings hold true:

$$\mathbf{X}'_{2(2)} = \mathbf{X}'_{1(2)} + \mathbf{d}'_{21} \tag{9}$$

$$\mathbf{X}'_{2(1)} = \mathbf{X}'_{1(1)} + \mathbf{d}'_{21}, \tag{10}$$

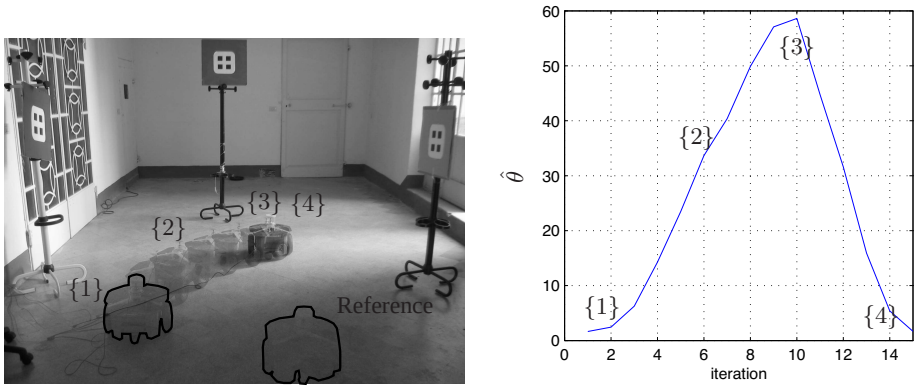


Fig. 8. *Experimental results*: (a) The mobile robot moves from the initial location $\{1\}$ to $\{4\}$ ($\{4\}$ is aligned with the Reference view); (b) Estimation of the angle θ with respect to the Reference view.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

being $\mathbf{d}'_{21} \triangleq \mathbf{X}'_{2(1)} - \mathbf{X}'_{1(1)}$ the vector in O', defined as the translation from $\mathcal{L}_1$ to $\mathcal{L}_2$. Substituting (9) and (10) in (8) we obtain:

$$\begin{aligned}\mathbf{n}'_2 &\sim (\mathbf{X}'_{1(2)} + \mathbf{d}'_{21}) \times (\mathbf{X}'_{1(1)} + \mathbf{d}'_{21}) \\ &\sim \mathbf{X}'_{1(2)} \times \mathbf{X}'_{1(1)} + (\mathbf{X}'_{1(2)} - \mathbf{X}'_{1(1)}) \times \mathbf{d}'_{21}\end{aligned}$$

and being $\mathbf{h}'_1 \triangleq \mathbf{X}'_{1(2)} - \mathbf{X}'_{1(1)}$ (Fig. 3(a)), it results that (8) can be also written as:

$$\mathbf{n}'_2 \sim (\mathbf{X}'_{1(2)} \times \mathbf{X}'_{1(1)}) + (\mathbf{h}'_1 \times \mathbf{d}'_{21}). \tag{11}$$

Let us now consider the expression of $\mathbf{n}_1$ and $\mathbf{n}_2$ w.r.t. O. Analogously to (9) and (10) we have $\mathbf{X}_{2(2)} = \mathbf{X}_{1(2)} + \mathbf{d}_{21}$ and $\mathbf{X}_{2(1)} = \mathbf{X}_{1(1)} + \mathbf{d}_{21}$. Note that $\mathbf{d}_{21} = \mathbf{R}^T_{z,\theta}\mathbf{d}'_{21}$ and $\mathbf{h}_i = \mathbf{R}^T_{z,\theta}\mathbf{h}'_i$, therefore

$$\mathbf{n}_2 \sim (\mathbf{X}_{1(2)} \times \mathbf{X}_{1(1)}) + \mathbf{R}^T_{z,\theta}(\mathbf{h}'_1 \times \mathbf{d}'_{21}). \tag{12}$$

Let us define:

$$\mathbf{e}_{21} \triangleq \mathbf{c}_2 - \mathbf{c}_1 = \gamma \begin{bmatrix} \frac{n_{2x}}{n_{2z}} - \frac{n_{1x}}{n_{1z}} \\ \frac{n_{2y}}{n_{2z}} - \frac{n_{1y}}{n_{1z}} \end{bmatrix}. \tag{13}$$

Since $\mathbf{n}_1 = \frac{\mathbf{X}_{1(2)} \times \mathbf{X}_{1(1)}}{\|\mathbf{X}_{1(2)} \times \mathbf{X}_{1(1)}\|}$, then we can substitute it into (12) and obtain

$$\mathbf{n}_2 \sim \mathbf{n}_1 \lambda + \mathbf{R}^T_{z,\theta}\mathbf{p}' \tag{14}$$

with $\lambda \triangleq \|\mathbf{X}_{1(2)} \times \mathbf{X}_{1(1)}\|$ and $\mathbf{p}' \triangleq \mathbf{h}'_1 \times \mathbf{d}'_{21}$. Owing to (14) we simply obtain:

$$\begin{aligned} n_{2x} &\sim \lambda n_{1x} + p'_x \cos\theta + p'_y \sin\theta \\ n_{2y} &\sim \lambda n_{1y} - p'_x \sin\theta + p'_y \cos\theta \\ n_{2z} &\sim \lambda n_{1z} + p'_z. \end{aligned} \tag{15}$$

Substituting (15) in (13), after some simple manipulations, one obtains

$$\mathbf{e}_{21} \sim \begin{bmatrix} p'_x \cos\theta + p'_y \sin\theta - p'_z c_x \\ -p'_x \sin\theta + p'_y \cos\theta - p'_z c_y \end{bmatrix}. \tag{16}$$

The computation of $\mathbf{n}'_2$ can be derived analogously, and hence it results that:

$$\mathbf{e}'_{21} \sim \begin{bmatrix} p'_x - p'_z c'_x \\ p'_y - p'_z c'_y \end{bmatrix}. \tag{17}$$

Now, it is immediate to see that $c_x \sim c'_x \cos\theta + c'_y \sin\theta$ and $c_y \sim c'_y \cos\theta - c'_x \sin\theta$. Using these into (16) and comparing the obtained expression of $\mathbf{e}_{21}$ with $\mathbf{e}'_{21}$, we finally obtain:

$$\begin{aligned} e_{21}(1) &\sim e'_{21}(1)\cos\theta + e'_{21}(2)\sin\theta \\ e_{21}(2) &\sim e'_{21}(2)\cos\theta - e'_{21}(1)\sin\theta \end{aligned}$$

from which (5) follows directly.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255

References

1. A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(25):578–589, 2003.
2. J.P. Barreto and H. Araujo. Geometric properties of central catadioptric line images. In *7th European Conference on Computer Vision*, pages 237–251, 2002.
3. J.P. Barreto, F. Martin, and R. Horaud. Visual servoing/tracking using central catadioptric images. In *8th International Symposium on Eperimental Robotics, Advanced Robotics Series*, pages 863–869, 2002.
4. R. Benosman and S.B. Kang. *Panoramic Vision: Sensors, Theory and Applications*. Springer Verlag, New York, 2001.
5. M. W. M. Gamini Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.
6. Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
7. P. Krishnamurthy G. Schindler and F. Dellaert. Line-based structure from motion for urban environments. In *3D Data Processing Visualization and Transmission (3DPVT)*, pages 846–853, 2006.
8. C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems. In *Sixth European Conference on Computer Vision*, pages 445–462, 2000.
9. C. Geyer and K. Daniilidis. Paracatadioptric camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:687–695, 2002.
10. C. Geyer and K. Daniilidis. Properties of the catadioptric fundamental matrix. In *7th European Conference on Computer Vision*, pages 140–154, 2002.
11. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
12. J. Košecká and W. Zhang. Video compass. In *7th European Conference on Computer Vision*, pages 476 – 491, 2002.
13. A. Makadia, C. Geyer, S.S. Sastry, and K. Daniilidis. Radon-based structure from motion without correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 796–803, 2005.
14. E. Marchand, F. Spindler, and F. Chaumette. ViSP for visual servoing. *IEEE Robotics and Automation Magazine*, 12(4):40–52, 2005.
15. G. L. Mariottini and D. Prattichizzo. EGT for multiple view geometry and visual servoing: robotics vision with pinhole and panoramic cameras. *IEEE Robotics and Automation Magazine*, 12(4):26–39, 2005.
16. G.L. Mariottini, F. Morbidi, D. Prattichizzo, G.J. Pappas, and K. Daniilidis. Leader-Follower Formations: Uncalibrated Vision-Based Localization and Control. In *Proc. IEEE Int. Conf. Robot. Automat*, pages 2403–2408, 2007.
17. J.M.M. Montiel and A.J. Davison. A visual compass based on SLAM. In *IEEE International Conference on Robotics and Automation*, pages 497 – 503, 2006.
18. T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49(1):23–37, 2002.
19. R. Vidal, O. Shakernia, and S. Sastry. Omnidirectional vision-based formation control. In *Fortieth Annual Allerton Conference on Communication, Control and Computing*, pages 1625–1634, 2002.
20. Z. Zhang. Determining the epipolar geometry and its uncertainty: a review. *International Journal of Computer Vision*, 27(2):161–198, 1998.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 242-255