Image-based servoing of non-holonomic vehicles using non-central catadioptric cameras

Hadi Aliakbarpour, Omar Tahri, Helder Araujo
Institute of Systems and Robotics
University of Coimbra
Email: {hadi,omartahri,helder}@isr.uc.pt

***Abstract*—Novel contributions on image-based control of a mobile robot using a general catadioptric camera model are presented in this paper. Visual servoing applications using catadioptric cameras have essentially been using central cameras and the corresponding unified projection model. So far only in a few cases more general models have been used. In this paper we address the problem of visual servoing using the so-called the radial model. The radial model can be applied to many camera configurations and in particular to non-central catadioptric systems with mirrors that are symmetric around an axis coinciding with the optical axis. In this case, we show that the radial model can be used with a non-central catadioptric camera to allow effective image-based visual servoing (IBVS) of a mobile robot. Two sets of experiments are carried. In one of the sets, IMU (Inertial Measurement Unit) is used to measure the relative rotation of the robot and in the other set visual features are solely used. The achieved results validate both the applicability and effectiveness of the proposed method for imaged-based control of a non-holonomic robot.**

I. Introduction

Image-based servoing approaches are versatile and effective methods to control robot motion by using camera observations. In practice, when a conventional camera is used there is no guarantee that the features remain in the camera's field of view (FOV). In order to overcome the problem of keeping the features in the camera's FOV, several methods have been developed namely: based on path planning [1], zoom adjustment [2], switching control [3]. Simpler and differing approaches consist on using omnidirectional vision sensors to increase the FOV using mirrors. Wide-angle cameras include catadioptric systems that combine mirrors and conventional cameras to create omnidirectional images providing 360^o panoramic views of a scene, or dioptric fish-eye lenses [4], [5]. Lately they have been subject of an increasing interest from robotics researchers [6], [7], [8], [9].

Having a single viewpoint in omnidirectional imaging systems is very practical [4], [10]. Such systems have a single center of projection, in such way that every image point measures the irradiance of the light passing through the same viewpoint. One can model a central imaging system as two consecutive projections: spherical and perspective. Geyer and Daniilidis in [11] derived the geometric model of these systems, and called it the *unified model*. In this model perspective projection corresponds to a special configuration. This formulation has been used by many research works in the area of visual servoing. Tahri et al. in [9] proposed an image-based visual servoing(IBVS) method to control the translational degrees of freedom (DOFs) which is invariant to rotational motion. In [7] an IBVS is proposed. This method is based on the auto-epipolar condition, which occurs when the current and desired catadioptric views undergo a pure translation. The method has been applied to control a holonomic mobile robot. Adaptation of the classical image-based visual servoing to a generalised imaging model was proposed in [12], by modeling the cameras as sets of 3D viewing rays. In [6], the projection of 3-D straight lines in the image plane on a central catadioptric system is used to control a $6DOFs$ holonomic robot and a non-holonomic mobile robot. As mentioned in [4], although the existing methods are effective for single-viewpoint catadioptric systems, in practice just a few realistic configurations lead to a single-viewpoint catadioptric system.

The problem of modeling the general case of a non-central catadioptric camera is a hard problem and still has only been tackled partially in computer vision. For this reason, iterative approaches are usually applied by some researchers to determine the reflection point on the mirror. Recently, a forward projection model has been proposed for the case of non-central catadioptric cameras consisting on a perspective camera and a rotationally symmetric conic reflector [13]. In the latter work, the optical path from a given 3D point to the given viewpoint is obtained by solving a $6th$ degree polynomial equation for general conic mirrors. For a spherical mirror, the forward projection equation reduces to a $4th$ degree polynomial, resulting in a closed form solution. In [14], an analytical forward projection equation for the projection of a 3D point reflected by a quadric mirror into the image plane of a perspective camera, with no restrictions on the camera placement is derived. They show that the equation is a $8th$ degree polynomial in a single unknown. In absence of an analytical and simple forward model, the determination of some elements like the interaction matrix required for image-based servoing becomes difficult.

For scene reconstruction or control purposes, a complete knowledge of the projection model is not always required. In [15], a technique to linearly estimate the radial distortion of a wide-angle lens given three views of a real-world plane has been proposed using the radial projection model. Based on [15], linear methods for the estimation of multi-view geometry of 1D radial cameras have been studied in [16] and [17].

In this paper, it will be shown that the simple radial projection model can be sufficient for mobile robot control using a large family of catadioptric cameras. More precisely, the contributions of this paper are:

- An image-based visual servoing method for mobile robots moving on a plane, valid for a large set of catadioptric cameras (including radially symmetric non-central cameras) is proposed;
- Using the radial model, new visual features with decoupling properties are derived;
- An efficient image-based visual servoing approach based on the desired value of the interaction matrix is proposed.
- The feasibility and effectiveness of the proposed method have been demonstrated with real experiments using a real robot.

The rest of this paper is organized as follows: The proposed radial camera model and its usage for visual servoing are introduced in Section III. The issues related to the selection of adequate visual features from images are discussed in Section IV. A control law which uses the proposed features is introduced in the same section. In Section V we present our experimental setup and achieved results in a real scenario. Conclusions and future works are presented and discussed in Section VI.

II. Notation and Symbols

Throughout this article we use the following notations: Scalars are typeset in regular lower-case. Vectors are denoted by lower-case boldface. Matrices are typeset in capital boldface. Variables with the * (star) denote they are computed using the information corresponding to the robot's goal positions.

III. Radial Camera Model for Visual Servoing

A. Radial camera model

A catadioptric system made up by the combination of a conventional pinhole camera and a rotationally symmetric mirror, shown in Figure 1, is considered. The camera is positioned in such a way as to have its optical axis aligned with the mirror axis. Using the radial projection model, $3D$ point $\mathbf{p} = (X, Y, Z)$ is reflected first on a point on the mirror $\mathbf{p_r} = [X_r \, Y_r, Z_r]$ before being projected onto the image plane as $\widetilde{\mathbf{x}}_m$ (expressed in metric homogeneous coordinates):

$$\widetilde{\mathbf{x}}_m = (x_m, y_m, 1) = \frac{\mathbf{p_r}}{Z_r} \tag{1}$$

Point $\widetilde{\mathbf{x}}_m$ is projected into the catadioptric image at $\widetilde{\mathbf{x}}_d = (x_d, y_d, 1)$, expressed in pixels and can be obtained from $\widetilde{\mathbf{x}}_m$ using:

$$\widetilde{\mathbf{x}}_d = \mathbf{K}\widetilde{\mathbf{x}}_m \tag{2}$$

where $\mathbf{K}$ is the matrix of the camera intrinsic parameters, f_x and f_y being the focal lengths, μ_x and μ_y are the principle point coordinates (and zero skew) as following:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & \mu_x \\ 0 & f_y & \mu_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

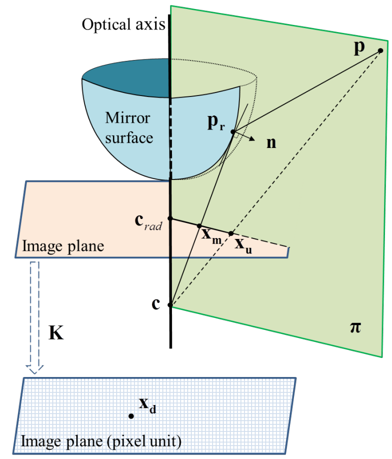

Fig. 1. Axial catadioptric system

From the laws of reflection, we have: (a) vector $\mathbf{n}$, the center of the projection $\mathbf{c}$, the $3D$ points $\mathbf{p}$ and $\mathbf{p_r}$ belong to the same plane π as shown in Fig. 1, (b) the angle between the incident ray and $\mathbf{n}$ is equal to the angle between the reflected ray and $\mathbf{n}$. In [17] and [16], the intersection of the planes π defined by the image points from multiple views has been used to recover linearly the structure of the scene.

The mirror is rotationally symmetric, and therefore the optical axis also belongs to π. Further, for symmetry reasons, the center of distortion (in our case the center of the image) ($\mathbf{c_{rad}}$) and the principal point coincide. In this paper, the normalized coordinates of $\mathbf{x_m}$ are used as follows:

$$\mathbf{x_n} = \frac{\mathbf{x_m}}{\| \mathbf{x_m} \|} \tag{4}$$

so that they belong to the unit circle. Later, $\mathbf{x_n}$ will be used in the derivation of the new features and image servoing algorithm. It can be proved that the computation of $\mathbf{x_n}$ from the image points expressed in pixel only requires the knowledge of the principal point coordinates (which coincides with the distortion center) and the ratio of the focal length parameters ($\rho = \frac{f_x}{f_y}$). Note that the center of the image (center of distortion) can be approximated by estimating the center of the mirror border (assumed to be a circle or an ellipse) [18].

Let $\widetilde{\mathbf{x}}_u = (x_u, y_u, 1) = \frac{\mathbf{p}}{Z}$ be the point coordinates (homogeneous) in metric units of the projection of $\mathbf{p}$ using pinhole model as shown in Fig. 1. Let $\mathbf{x_u} = [x_n, y_n]$ be the non-homogeneous coordinates corresponding to $\widetilde{\mathbf{x}}_u$. Since the center of the pin-hole camera and $\mathbf{p}$ belong to plane π, the point $\mathbf{x_u}$ also belongs to the intersection of this plane with the image plane. Therefore, $\mathbf{c_{rad}}$, $\mathbf{x_u}$ and $\mathbf{x_m}$ belong to the same line. We have then $\mathbf{x_n} = \frac{\mathbf{x_u}}{\|\mathbf{x_u}\|}$, which leads to:

$$\mathbf{x_n} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \frac{1}{\sqrt{X^2+Y^2}} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{5}$$

Note that $\mathbf{x_n}$ is not defined only if the 3D point belongs to the camera optical axis (which does not happen in the case of the catadioptric camera). In the following $\mathbf{x_n}$ will be used to define new visual features to control the motion of a mobile robot moving on a plane.

B. Visual servoing

In visual servoing the time variation $\dot{\mathbf{s}}$ of the visual features $\mathbf{s}$ can be linearly expressed with respect to the relative camera-object kinematics screw as $\dot{\mathbf{s}} = \mathbf{L_s}\tau$, where $\mathbf{L_s}$ is the interaction matrix related to $\mathbf{s}$. Usually, the control scheme is designed to reach an exponential decoupled decrease of differences on the visual features to their goal value $\mathbf{s}^*$. If we consider an eye-in-hand system observing a static object, the corresponding control law is:

$$\tau_c = -\lambda \widehat{\mathbf{L_s}}^{+}(\mathbf{s}-\mathbf{s}^*), \tag{6}$$

where $\widehat{\mathbf{L_s}}$ is a model or an approximation of $\mathbf{L_s}$, $\widehat{\mathbf{L_s}}^{+}$ the pseudo-inverse of $\widehat{\mathbf{L_s}}$, λ a positive gain tuning the time to convergence, and $\tau_c = (\upsilon_c, \omega_c)$ the camera velocity sent to the low-level robot controller.

IV. Visual features selection and control law

In the next paragraph, new visual features are proposed and their corresponding interaction matrices derived. A control law using the goal values of the interaction matrices is also proposed and derived.

A. Visual features suitable to control camera translational velocities

In this paragraph, we introduce new visual features which are suitable for the control of the translation component of the camera movement. To do so, we use the inner product between the coordinates of two points $\mathbf{x_n}_i$ and $\mathbf{x_n}_j$ in the image:

$$c_{ij} = \mathbf{x_n}_i^{\top}\mathbf{x_n}_j \tag{7}$$

By taking the derivative of (7), we obtain:

$$\dot{c}_{ij} = \mathbf{x_{nj}}^{\top}\dot{\mathbf{x}}_{\mathbf{ni}} + \mathbf{x_{ni}}^{\top}\dot{\mathbf{x}}_{\mathbf{nj}} \tag{8}$$

The interaction matrix corresponding to $\mathbf{x_n}$ can be obtained by taking the derivative of (5):

$$\mathbf{L_{x_n}} = [\ \mathbf{L_{x_n}}_{\upsilon} \quad \mathbf{L_{x_n}}_{\omega}\ ] \tag{9}$$

with:

$$\mathbf{L_{x_n}}_{\upsilon} = \begin{bmatrix} -\frac{(1-x_n^2)}{d} & \frac{x_n y_n}{d} & 0 \\ \frac{x_n y_n}{d} & -\frac{(1-y_n^2)}{d} & 0 \end{bmatrix} \tag{10}$$

and

$$\mathbf{L_{x_n}}_{\omega} = \begin{bmatrix} -x_n y_n z_n & -(1-x_n^2)z_n & y_n \\ (1-y_n^2)z_n & x_n y_n z_n & -x_n \end{bmatrix} \tag{11}$$

where $d = \sqrt{X^2+Y^2}$ and $z_n = Z/d$. By combining (9) and (8), the interaction matrix $\mathbf{L}_{c_{ij}} = [\mathbf{L}_{c_{ij}\upsilon}\ \mathbf{L}_{c_{ij}\omega}]$ corresponding to c_{ij} can be then obtained by:

$$\mathbf{L}_{c_{ij}\upsilon} = \left[\ (\frac{-1}{d_j}+\frac{c_{ij}}{d_i})\mathbf{x_n}_i^{\top} + (\frac{-1}{d_i}+\frac{c_{ij}}{d_j})\mathbf{x_n}_j^{\top} \quad 0\ \right] \tag{12}$$

and

$$\mathbf{L}_{c_{ij}\omega} = [\ y_{nj}z_{ij} + y_{ni}z_{ji} \quad -x_{nj}z_{ij} - x_{ni}z_{ji} \quad 0\ ] \tag{13}$$

where $z_{ij} = z_{ni} - c_{ij}z_{nj}$ and $z_{ji} = z_{nj} - c_{ij}z_{ni}$. From (12) and (13), it can be seen that c_{ij} is invariant to the motion around the optical axis (which corresponds to the normal of the plane of motion). We assume that the camera is mounted on the mobile robot so that the translational motion takes place on the plane defined by the vectors $\mathbf{x}$ and $\mathbf{y}$ of the camera frame. Therefore, only the first two entries of the matrix $L_{c_{ij}\upsilon}$ are useful for the control of the translational motion with respect to the x-axis and the y-axis. In the next paragraph, we explain how to select an adequate feature to control the remaining DOF, namely the rotation around the optical axis.

B. Controlling camera rotation

Previously we introduced some features suitable for controlling the robot's transnational velocity. Beside this, the rotational velocity has to be controlled as well. In this paragraph, we introduce two approaches for this purpose. First one is to use an IMU (Inertial Measurement Unit), as a digital compass, to directly measure the relative rotational angle of the robot. The second approach is to define and extract a set of features which are sensitive to the camera's rotation. This last method was proposed in our previous work [19], nevertheless here we briefly remind it. Let $\mathbf{p_1}$ be a point defined by:

$$\mathbf{p_1} = \sum_{i=1}^{N} a_i \mathbf{x_{ni}} \tag{14}$$

From $\mathbf{p_1}$, we define a new point $\mathbf{v_1}$ belonging to the unit circle by:

$$\mathbf{v_1} = \frac{\mathbf{p_1}}{\parallel \mathbf{p_1} \parallel} \tag{15}$$

By taking the derivative of (15), the interaction matrix corresponding to $\mathbf{v_1}$ can be obtained by:

$$\mathbf{L_{v_1}} = \frac{\mathbf{I_2} - \mathbf{v_1}\,\mathbf{v_1}^{\top}}{\parallel \mathbf{p_1} \parallel}\sum_{i=1}^{N} a_i \mathbf{L_{x_{ni}}} \tag{16}$$

Let α_m be the angle defined by:

$$\alpha_m = atan2(v_{1y}, v_{1x}) \tag{17}$$

By taking the derivative of (17), it can be obtained:

$$\dot{\alpha}_m = v_{1x}\dot{v}_{1y} - v_{1y}\dot{v}_{1x} \tag{18}$$

By combining (18) with (16), $L_{\alpha_m\omega_z} = -1$ is obtained. As a result one can conclude that α_m varies linearly with respect to the velocity ω_z.

C. Control law

Let $\mathbf{s_c}$ be the feature vector obtained by stacking the features c_{ij} and $\mathbf{s_c^*}$ their goal values. Let $\mathbf{L_{s_c}}$ be the interaction matrix obtained by stacking the two first entries v_x and v_y of the interaction matrix corresponding to each feature c_{ij}. Only the two first entries are taken into account because we are only concerned with a planar motion and c_{ij} is invariant to the rotation around z-axis. We want to avoid computing the depths. Let us consider that the goal is to move the desired camera position towards the initial one. Therefore, the velocities that have to be applied to the goal position of the camera using its corresponding interaction matrix are obtained from:

$$\begin{cases} \begin{bmatrix} \upsilon_x^* \\ \upsilon_y^* \end{bmatrix} = -\lambda \mathbf{L}^+_{\mathbf{s_c}*}(\mathbf{s_c^*} - \mathbf{s_c}) \\ \omega_z^* = \lambda(\alpha_m^* - \alpha_m) - L_{\alpha_m \upsilon_x*}\, \upsilon_x^* - L_{\alpha_m \upsilon_y*}\, \upsilon_y^* \end{cases} \tag{19}$$

where $\mathbf{L}_{\alpha_\mathbf{m}\upsilon_{\mathbf{x}*}}$ and $\mathbf{L}_{\alpha_\mathbf{m}\upsilon_{\mathbf{y}*}}$ represent the variation of α_m with respect to the velocities υ_x and υ_y respectively. Let us consider the three frames shown in Figure 2-b. Let $\mathscr{F}_c$ and $\mathscr{F}_{c*}$ represent respectively the current and the goal camera frames and $\mathscr{F}_{ci}$ an intermediate frame that has the same position of the center as $\mathscr{F}_{c*}$ but the orientation of $\mathscr{F}_c$. As it can be seen from Figure 2-b, the translational velocity to be applied to the frame $\mathscr{F}_c$ to move it towards its desired position is equal to the negative of the velocities that move $\mathscr{F}_{ci}$ towards $\mathscr{F}_c$. Therefore, to control the translational motion of the current camera position, it is more adequate to use the interaction matrix corresponding to $\mathbf{s_c}$ computed for the position corresponding to $\mathscr{F}_{ci}$:

$$\begin{bmatrix} \upsilon_x \\ \upsilon_y \end{bmatrix} = -\lambda \mathbf{L}^+_{\mathbf{s_{ci}}}(\mathbf{s_c} - \mathbf{s_{ci}}) \tag{20}$$

In the case of the projection onto the sphere, it was shown in [9] that two interaction matrices $\mathbf{L}^\mathbf{2}_{\mathbf{i_n}}$ and $\mathbf{L}^\mathbf{1}_{\mathbf{i_n}}$ related to an invariant to the $3D$ rotation i_n and computed respectively for two camera poses 1 and 2 separated by a rotational motion are related by equation:

$$\mathbf{L}^\mathbf{2}_{\mathbf{i_n}} = \mathbf{L}^\mathbf{1}_{\mathbf{i_n}}\, {}^\mathbf{1}\mathbf{R_2} \tag{21}$$

where ${}^\mathbf{1}\mathbf{R_2}$ is the rotation matrix. Similarly, it can be shown for feature $\mathbf{s_{ci}}$ that if only a rotation is considered between $\mathscr{F}_{ci}$ and $\mathscr{F}_{c*}$, $\mathbf{L_{s_{ci}}}$ can be obtained from $\mathbf{L_{s_c*}}$ by:

$$\mathbf{L_{s_{ci}}} = \mathbf{L_{s_c}}\, {}^{\mathbf{c}*}\mathbf{R_i} \tag{22}$$

where ${}^{\mathbf{c}*}\mathbf{R_i}$ is the 2-dimensional rotation matrix corresponding to the rotation angle γ between $\mathscr{F}_{c*}$ and $\mathscr{F}_{ci}$. We continue to prove that c_{ij} will be invariant to any rotation around z axis. Based on (7), c_{ij} is a function of $\mathbf{x_n}_i$ and $\mathbf{x_n}_j$:

$$c_{ij} = g(\mathbf{x_n}_i, \mathbf{x_n}_j) = \mathbf{x_n}_i^\top \mathbf{x_n}_j \tag{23}$$

Now we apply an arbitrary rotation $\mathbf{R}$ to the inputs of the function g:

$$g(\mathbf{R}\mathbf{x_{ni}}, \mathbf{R}\mathbf{x_{nj}}) = (\mathbf{R}\mathbf{x_{ni}})^\top \mathbf{R}\mathbf{x_{nj}} = \mathbf{x_{ni}}^\top \mathbf{R}^\top \mathbf{R}\mathbf{x_{nj}} = \mathbf{x_{ni}}^\top\, \mathbf{x_{nj}} \tag{24}$$

Equation (24) shows that $g(\mathbf{R}\mathbf{x_{ni}}, \mathbf{R}\mathbf{x_{nj}})$ is equal to $g(\mathbf{x_n}_i, \mathbf{x_n}_j)$ and as a result the feature c_{ij} is invariant to any rotation around z axis. Since c_{ij} is invariant to a z-axis rotation one obtains $s_{ci} = \mathbf{s_{c*}}$. By combining this result and (22) in (20), we obtain:

$$\begin{bmatrix} \upsilon_x \\ \upsilon_y \end{bmatrix} = -\,{}^\mathbf{i}\mathbf{R_{c*}}\lambda \mathbf{L}^+_{\mathbf{s_{c*}}}(\mathbf{s_c} - \mathbf{s_{c*}}) \tag{25}$$

By combining (25) and (19), we finally obtain:

$$\begin{bmatrix} \upsilon_x \\ \upsilon_y \end{bmatrix} = -\,{}^\mathbf{i}\mathbf{R}_* \begin{bmatrix} \upsilon_{x*} \\ \upsilon_{y*} \end{bmatrix} \tag{26}$$

On the other hand, since and the z-axis has the same orientation in the current and the goal camera poses, we choose $\omega_z = -\omega_{z*}$. In the next section, we explain how to effectively approximate ${}^\mathbf{i}\mathbf{R_{c*}}$.

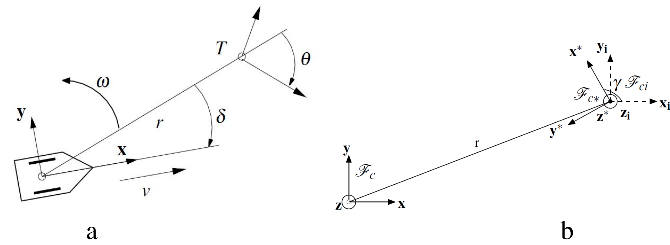

Fig. 2. (a) Egocentric polar coordinate system with respect to the observer (b) Camera frame position.

V. Experiments

In this section we present experimental results that were carried out. In our previous work [19], simulated results were presented for different scenarios. In this paper we present real results obtained in a real scenario. A Pioneer 3-DX [20] has been used as differential drive non-holonomic vehicle. The experiments were performed using the open-source meta-operating system, ROS [21]. The coordinate system related to this robot is shown in Fig. 2-a. The applied control law in this work is as follows:

$$\begin{bmatrix} \upsilon_l \\ \omega \end{bmatrix} = \begin{bmatrix} k_r\, r \\ k_\delta \delta + k_\theta \theta \end{bmatrix} \tag{27}$$

where r is the distance from the initial to the desired camera pose which can be approximated by $r = \frac{\sqrt{\upsilon_x^2 + \upsilon_y^2}}{\lambda}$ after removing the time unit. The angle δ can also be estimated as the direction of the velocity to be applied to the current camera pose from $\delta = atan2(\upsilon_y, \upsilon_x)$ (since the camera is rigidly attached to the robot). The angle θ is $\theta = \frac{\omega_z}{\lambda} - \delta$. Note that in (27), the following conditions have to be satisfied for the sake of stability:

$$k_r > 0\,, k_\theta < 0 \tag{28}$$

and

$$k_\delta + \frac{5}{3}k_\theta - \frac{2}{\pi}k_r > 0 \tag{29}$$

Once the robot is placed on the goal position (Fig. 3-a) in the scene an image is grabbed as the *goal image* (Fig. 3-c). As seen in the figures, some black circles (here nine circles) were fixed on the walls and used as visual beacons. The centers of these beacons are automatically extracted (using openCV library) and used as our *goal points* for the goal image. In our experiments, ρ has been given the value 1 and the center of the image/distortion was computed as the center of the mirror border image. It has as coordinates $\mu_x = 636$, $\mu_y = 536$. As already mentioned, one of the advantages of the method proposed in this paper is that the camera does not need to be fully calibrated.

The robot is placed on an arbitrary position, the *initial position* (Fig. 3-b). Then the robot moves, so that it can reach the goal position with the goal orientation. After the

Fig. 3. Images of the robot and of images acquired by the robot on its position. (a) and (c): the robot in the goal position and the corresponding image acquired by the robot, respectively. (b) and (d) the robot on an arbitrary initial position and its corresponding image, respectively.

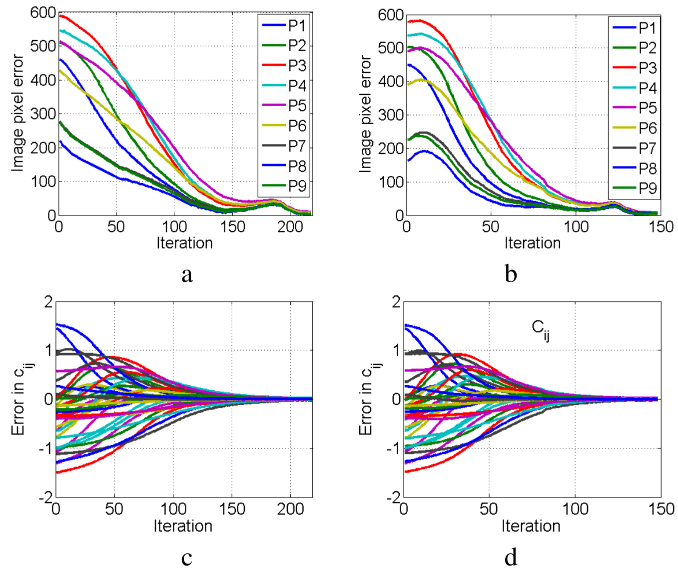

Fig. 4. Diagram of errors in terms of points and features during the convergence: (a) and (b) show the reductions of errors between the nine goal image points and their correspondences in the current images. Using these nine image points, 36 features were defined. The errors between the goal features and current features are plotted in (c) and (d). Images at the left correspond for the case of using just visual features and images at the right for the case of using IMU for rotational controlling)

extraction of features from both goal and initial images, the proposed control law is applied to the robot. In our experiments, $k_\theta = -0.1$, $k_r = \lambda = 0.15$ and $k_\delta = 0.31$ are used as constant parameters in the control law (27).

Two sets of experiments have been carried out depend on how the camera rotational velocity is obtained. In one set, just visual features are used for obtaining the rotational components and for the other set IMU is solely used. For the first case the rotation matrix ${}^{i}\mathbf{R}_{c*}$ (expressed in (26)) is estimated using as a rotation angle $(\alpha_m - \alpha_m^*)$ and for the second case it is obtained as the difference between the angles measured by the IMU. As expected the robot performs smooth trajectories, converges and stops nearly on the goal position. Figures 4-a and -b depict the errors (distances in pixels) between the goal image points and their corresponding points in the current as a function of servoing iterations (a: just visual features are used for obtaining rotation, b: just IMU to measure rotation). As it can be seen, convergences are quite smooth and with negligible errors. The same behaviors are shown in Figures 4-c and -d, where the errors between the features $(c_{ij}^* - c_{ij})$ are plotted (c: rotation is estimated from vision, d: rotation is measured using IMU). The velocities of the camera are shown in Figures 5-a and -b. Figures 5-c and -d depict the robot's velocities in the two mentioned scenarios. Figure 6-a presents the errors in the angle α_m values during servoing. The convergence of the camera's current rotation to the camera's goal rotation is plotted Figure 6-b where the angles are measured by IMU. It should be mentioned that two movies, each one corresponding to one of the scenarios, have been filmed and can be accessed at *https://sites.google.com/site/otahri*.

VI. Conclusion

In this paper, we have used the radial camera model to propose a novel IBVS. New features were extracted based on this model and their corresponding interaction matrices were derived. The method does not require a fully calibrated camera. The only calibration parameters that are require are the ratio of the two focal lengths ($\rho = \frac{f_x}{f_y}$) and the coordinates of the principal point (μ_x and μ_y). In general, both these parameters can be estimated automatically by using the image of the mirror border (circle or ellipse). Furthermore, only the goal value of the interaction matrix is used to compute the velocities, which allows avoiding the estimation of the depths of the points, as well as the inversion of the interaction matrix, during servoing.

As a result of using a simple radial model, the proposed IBVS method can be applied for a large class of catadioptric cameras, both central and non-central. The proposed method has been implemented using a ROS-based robotic platform. The results obtained show the validity and effectiveness of the proposed approach. Our future work includes extending the method for a 6 $DOFs$ robot and the use of global visual features as well as multiple-view geometry in the control law.

Acknowledgements

Hadi AliAkbarpour, Omar Tahri and Helder Araujo are with Institute for Systems and Robotics, Polo II 3030-290 Coimbra, Portugal. The authors would like to thank the support of project Morfeu–PTDC/EEA-CRO/108348/2008 funded by the Portuguese Science Foundation (FCT) and H. Araujo would like to thank the support of project FCT/PTDC/EIA-EIA/122454/2010, funded also by the Portuguese Science Foundation (FCT) by means of national

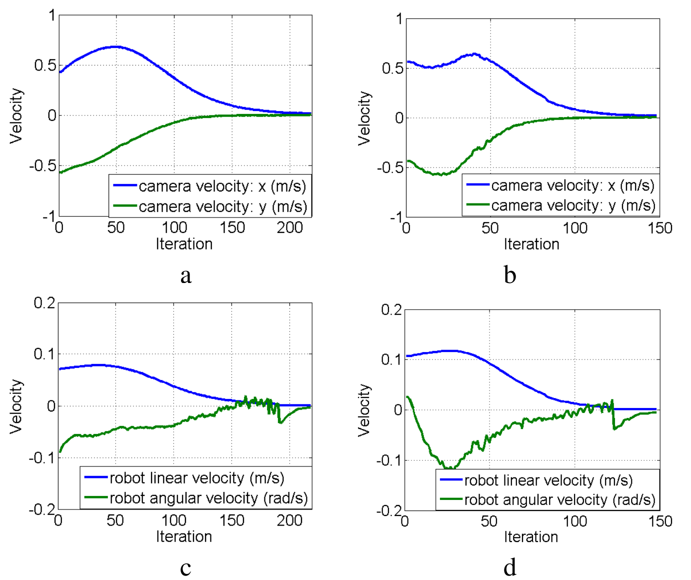

Fig. 5. Plots for the velocities during the iterations: (a) Depicts the camera linear velocity when just visual features are used. (b) Depicts the camera linear velocity for the case of using solely IMU for the rotation. (c) shows the linear and angular components of the velocity of the robot during the convergence when just visual features are used. (d) Shows the linear and angular components of the velocity of the robot during the convergence for the case of using IMU for the rotation.

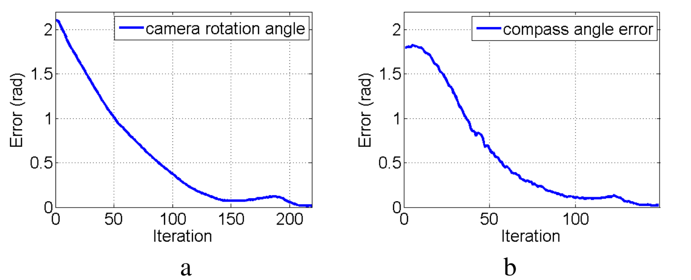

Fig. 6. Convergence of the camera's current angle to its goal angle. a: the rotation is obtained using just visual features (corresponding to α_m in (17)) and b: the rotation is measured by using IMU.

funds (PIDDAC) and co-funded by the European Fund for Regional Development (FEDER) through COMPETE Operational Programme Competitive Factors (POFC).

REFERENCES

[1] Y. Mezouar, F. Chaumette, Path planning for robust image-based control, IEEE Trans. on Robotics and Automation 18 (4) (2002) 534–549.

[2] S. Benhimane, E. Malis, Vision-based control with respect to planar and non-planar objects using a zooming camera, in: The 11th International Conference on Advanced Robotics Coimbra, Portugal, Coimbra, Portugal, 2003, pp. 863–866.

[3] G. Chesi, K. Hashimoto, D. Prattichizzo, A. Vicino, Keeping features in the field of view in eye-in-hand visual servoing: a switching approach, IEEE Transactions on Robotics 20 (5) (2004) 908–914.

[4] S. Baker, S. Nayar, A theory of catadioptric image formation, Int. Journal of Computer Vision 35 (2) (1999) 175–196.

[5] J. Courbon, Y. Mezouar, L. Eck, M. Martinet, A generic fisheye camera model for robotic applications, in: IROS, 2007, pp. 1683–1688.

[6] H. Hadj-Abdelkader, Y. Mezouar, P. Martinet, F. Chaumette, Catadioptric visual servoing from 3d straight lines, IEEE Trans. on Robotics 24 (3) (2008) 652–665.

[7] G. L. Mariottini, D. Prattichizzo, Image-based visual servoing with central catadioptric camera, International Journal of Robotics Research 27 (2008) 41–57.

[8] P. Corke, D. Strelow, S. Singh, Omnidirectional visual odometry for a planetary rover, in: In IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 4, Sendai, Japan, 2004, pp. 4007–4012.

[9] O. Tahri, Y. Mezouar, F. Chaumette, P. Corke, Decoupled image-based visual servoing for cameras obeying the unified projection model, IEEE Trans. on Robotics 26 (4) (2010) 684 – 697.

[10] T. Svoboda, T. Pajdla, Epipolar geometry for central catadioptric cameras, Int. Journal on Computer Vision 49 (1) (2002) 23–37.

[11] C. Geyer, K. Daniilidis, Mirrors in motion: Epipolar geometry and motion estimation, Int. Journal on Computer Vision 45 (3) (2003) 766–773.

[12] A. Comport, R. Mahony, F. Spindler, A visual servoing model for generalised cameras: Case study of non-overlapping cameras, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 5683 –5688. doi:10.1109/ICRA.2011.5979678.

[13] A. Agrawal, Y. Taguchi, S. Ramalingam, Analytical forward projection for axial non-central dioptric and catadioptric cameras, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), ECCV 2010, Vol. 6313/2010 of Lecture Notes in Computer Science, 2010, pp. 129–143.

[14] A. Agrawal, Y. Taguchi, S. Ramalingam, Beyond alhazen's problem: Analytical projection model for non-central catadioptric cameras with quadric mirrors, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 2993–3000.

[15] S. Thirthala, M. Pollefeys, The radial trifocal tensor: a tool for calibrating the radial distortion of wide-angle cameras, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005, pp. 321 – 328.

[16] C. Sagues, A. Murillo, J. Guerrero, T. Goedeme, T. Tuytelaars, L. Van Gool, Localization with omnidirectional images using the radial trifocal tensor, in: IEEE Int. Conf. on Robotics and Automation, Orlando, FL, 2006, pp. 551 – 556.

[17] S. Thirthala, M. Pollefeys, Multi-view geometry of 1d radial cameras and its application to omnidirectional camera calibration, in: Tenth IEEE International Conference on Computer Vision, Vol. 2, Beijing, China, 2005, pp. 1539–1546.

[18] C. Mei, P. Rives, Single view point omnidirectional camera calibration from planar grids, in: IEEE Int. Conf. on Robotics and Automation, 2007, pp. 3945–3950. doi:10.1109/ROBOT.2007.364084.

[19] O. Tahri, O. Araujo, Non-central catadioptric cameras visual servoing for mobile robots using a radial camera model, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, 2012, pp. 1683–1688.

[20] S. Zaman, W. Slany, , G. Steinbauer, Ros-based mapping, localization and automatic navigation using pioneer 3-dx robot and their relevant issues, in: Saudi International Electronics, Communications and Photonics Conference, Riad, Saudi-Arabia, IEEE, 2011, pp. 1–5.

[21] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, , A. Y. Ng, Ros: an open-source robot operating system, in: Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA), 2009.