Featureless omnidirectional vision-based control of non-holonomic mobile robot

Youssef Alj, Guillaume Caron

The 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2015)
October 28 ∼ 30, 2015 / KINTEX, Goyang city, Korea

Featureless omnidirectional vision-based control of non-holonomic mobile robot

Youssef Alj, Guillaume Caron

Université de Picardie Jules Verne, "Modélisation, Information & Systèmes Laboratory", 33 rue Saint-Leu - 80039 Amiens Cedex 1 - France
(E-mail :youssef_alj@hotmail.com ; guillaume.caron@u-picardie.fr)

Abstract - This paper proposes featureless algorithms to address complex maneuvers for non-holonomic mobile robots. Using a single omnidirectional camera, the robot reaches the target position using an omnidirectional photometric visual servoing algorithm that simultaneously controls translation and rotation. Evaluation of the positioning task was performed on eight different positions on a circle and the error between the final position and the target one is measured.

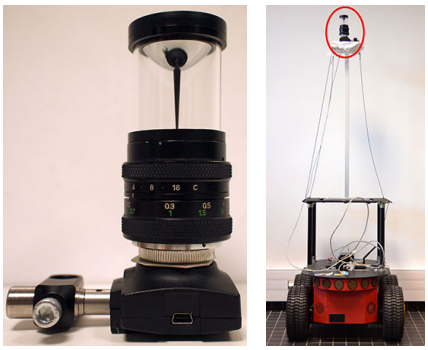
Keywords - visual servoing, omnidirectional camera, featureless, non-holonomic mobile robots.

1. Introduction

The goal of this work is to extend the omnidirectional photometric visual servoing [4] for complex maneuvers in mobile robot positioning and robust visual path following.

Controlling a robot with vision as input finds its interest in the acquisition of a lot of information in one image. This passive sensor is not range-limited, or only by the environment. While the field of view of a standard perspective camera is limited, an omnidirectional camera increases the field of view to 180 degrees, only changing the camera optics. Two families of such optics exist : the dioptric ones (fisheye lens) and the catadioptric ones (combination of lens and quadratic mirror). They allow acquiring a panoramic image in one shot, when their main optical axis is oriented vertically, a common use of these vision sensors on mobile robots (Fig. 1).

Vision-based robot control can be classified in two classes : feature-based or featureless. Contrary to visual feature-based control, featureless control has the interest of avoiding geometric features detection and matching, making the controller closer to data acquired by the sensor, that are pixel intensities of the image acquired by the camera : the photometry. Featureless or not, the vision-based control of a robot is described under the visual servoing framework [5]. It models the relationship between the motion of the camera and visual features in the image, leading to a kinematic proportional controller in basic formulations. Many visual features have been considered for visual servoing, as a point [8], a line [1] or sets of points [2], to only mention some of the ones introduced for omnidirectional vision in mobile robotics. Featureless, or photometric, visual servoing in omnidirectional vision for mobile robotics [4] considers the set of intensities of a desired image, *i.e.* acquired at the desired location of the mobile robot, as input of the mobile robot

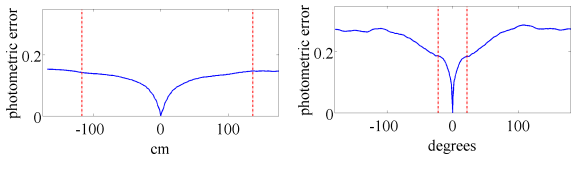


(a) (b)

Fig. 1 (a) The considered catadioptric camera, with a IDS uEye box ; (b) The camera vertically mounted and centered on a Pioneer 3-AT mobile robot.

control law [6]. Thus, visual servoing designs the control law to drive the robot minimizing the photometric error, *i.e.* the sum of squared differences (SSD) or other metrics (e.g. Hausdorff distance [9]), between the current image intensities and the desired ones.

However, every interest of featureless control has still not been shown because existing photometric visual servoing, even involving omnidirectional vision, has limitations, even for the most studied mobile robot kind, that is the unicycle robot. Indeed, such so called non-holonomic platform can not do instantaneously lateral motion leading to the need of maneuvers, if a lateral motion would be needed to achieve the mobile robot mission. Existing omnidirectional photometric visual servoing works well if such a motion is not needed but as soon as the mobile robot needs to rotate twenty degrees to reach its desired pose, the system may not converge on it. The reason will be clearly described in Section B. , but Figure 2(b) highlights it coarsely : the cost function, that is the photometric error, is much more sensitive to the rotational degree of freedom of the unicycle robot rather than its translational one (convergence area of 2.50 m width in



(a) (b)

Fig. 2 On an example : (a) Cost function for single axis pure translation ; (b) Cost function for pure rotation

Fig. 2(a)).

The contribution of the paper is to provide a solution to the latter issue, still directly using image intensities. In a few words, the core idea of the proposed method is to consider the re-orientation of the desired image with respect to the current one, thanks to visual compass, to overcome the sensitiveness of photometric visual servoing to rotation around the camera optical axis. Doing so has impacts on the control law that is to be described in Section 3. for the positioning issue, after reminding existing tools in Section 2A final experiment and a conclusion end the paper.

2. Existing tools

2.1 Notations

In the next sections we will use the following notations related to omnidirectional images :

- The desired image : $\mathbf{I}^*$; $\mathbf{I}^*_{\mathcal{S}}$ its spherical representation.
- The desired image rotated by θ : $\mathbf{I}^*_\theta$.
- The current image $\mathbf{I}$; $\mathbf{I}_{\mathcal{S}}$ its spherical representation.

2.2 Omnidirectional photometric visual servoing : the unicycle robot case

A. Method

Omnidirectional photometric visual servoing for a unicycle mobile robot defines its kinematic control inputs as :

$$\dot{\mathbf{q}} = (\upsilon_X, \omega_Z), \tag{1}$$

minimizing the photometric error (SSD) between the current and the desired omnidirectional image. Considering the camera and the robot axes are the same, that is the camera is at the robot center of rotation, its X-axis is pointing forward the robot and its Z-axis pointing up, the transformation matrix between the robot velocity $\dot{\mathbf{q}}$ and the camera velocity $\mathbf{v}$, can be neglected and the Levenberg-Marquardt-like control law is defined as [4] :

$$\dot{\mathbf{q}} = -\lambda \left(\mathbf{H} + \mu \operatorname{diag}(\mathbf{H})\right)^{-1} \mathbf{L}_{\mathbf{I}_{\mathcal{S}}}^{\top} \left(\mathbf{I}_{\mathcal{S}}(\mathbf{r}) - \mathbf{I}^*_{\mathcal{S}}(\mathbf{r}^*)\right) \tag{2}$$

with λ a gain, μ, the damping parameter,
$\mathbf{r} = [t_X, t_Y, t_Z, \phi * w_X, \phi * w_Y, \phi * w_Z]$, with $\mathbf{w} = [w_X, w_Y, w_Z]$ the 3D axis of rotation of unit norm and ϕ the angle, that are the six degrees of freedom (dof) camera pose (position and orientation) parameters. Furthermore, in Equation (2), $\mathbf{H} = \mathbf{L}_{\mathbf{I}_{\mathcal{S}}}^{\top} \mathbf{L}_{\mathbf{I}_{\mathcal{S}}}$, is the omnidirectional photometric Hessian, considering :

$$\mathbf{L}_{\mathbf{I}_{\mathcal{S}}} = \begin{bmatrix} \vdots \\ \mathbf{L}_{\mathbf{I}_{\mathcal{S}}} \\ \vdots \end{bmatrix}, \text{ with } \mathbf{L}_{\mathbf{I}_{\mathcal{S}}} = -\nabla \mathbf{I}_{\mathcal{S}}^{\top} \mathbf{L}_{\mathbf{X}_{\mathcal{S}}}, \tag{3}$$

is the interaction matrix related to luminance of the spherical image $\mathbf{I}_{\mathcal{S}}$, representing the omnidirectional image, at cartesian spherical coordinates $\mathbf{X}_{\mathcal{S}} = (X_{\mathcal{S}}, Y_{\mathcal{S}}, Z_{\mathcal{S}})$ corresponding to a pixel. The latter coordinates and spherical representation of the image are obtained considering the invert of the unified projection model [3] :

$$\mathbf{X}_{\mathcal{S}} = \begin{pmatrix} \frac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} x \\ \frac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} y \\ \frac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{pmatrix}, \tag{4}$$

where (x, y) are the coordinates of a normalized image point, obtained from pixel coordinates (u, v) and inverse intrinsic parameters, including the above mentioned ξ, a specific parameter of the unified projection model.

In Equation (3), $\nabla \mathbf{I}_{\mathcal{S}}$ are the cartesian spherical gradients introduced in [4] and the interaction matrix $\mathbf{L}_{\mathbf{X}_{\mathcal{S}}}$ is similar to the Jacobian of $\mathbf{X}_{\mathcal{S}}$ with respect to the camera pose parameters $\mathbf{r}$. In the general case, $\mathbf{L}_{\mathbf{X}_{\mathcal{S}}}$ has six dof, so it involves 6 columns, but in the considered mobile robot case, with υ_X and ω_Z dof, we have :

$$\mathbf{L}_{\mathbf{X}_{\mathcal{S}}} = \begin{pmatrix} \frac{X_{\mathcal{S}}^2 - 1}{\rho} & Y_{\mathcal{S}} \\ \frac{X_{\mathcal{S}} Y_{\mathcal{S}}}{\rho} & -X_{\mathcal{S}} \\ \frac{X_{\mathcal{S}} Z_{\mathcal{S}}}{\rho} & 0 \end{pmatrix}. \tag{5}$$

The latter consideration of the unicycle robot two dof leads to some limitations of its maneuvering space that are described in Section B.

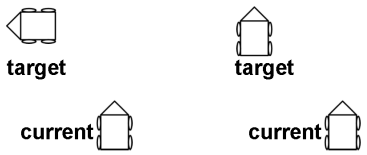
B. Limitations

Whereas the omnidirectional photometric visual servoing, recalled in Section A. , succeeds in situations where the robot motion only needs to be straightforward, or with a small change of orientation, other situations in which maneuvers are necessary may not meet success. Two common situations are identified as follows :

- First, if the needed robot motion, to reach the desired pose, would be too curvy (Fig. 3(a)), the limited convergence area highlighted in Figure 2(b) may lead to stop the robot rotation in a local minimum.
- Second, if the robot has to change its orientation during the motion for any reason, it will finally reach a pose in which its orientation is the same as the desired one but at a wrong position (Fig. 3(b)). Indeed, it woud be necessary to rotate but, by doing so, the orientation control would immediately counter that orientation change to minimize the photometric error.

Both situations involve limitations due to the control of orientation directly through the rotational degree of freedom as recalled at the end of Section A. , not the translation one that has a much larger convergence area (Fig. 2(a)). Thus, we propose to overcome these limitations by revisiting the orientation control to ensure the robot reaches the desired position and orientation, in the situations for which the existing omnidirectional photometric visual servoing succeeds and the two above mentioned situations (section 3.).

The proposed method involves a photometric visual compass that is described below.



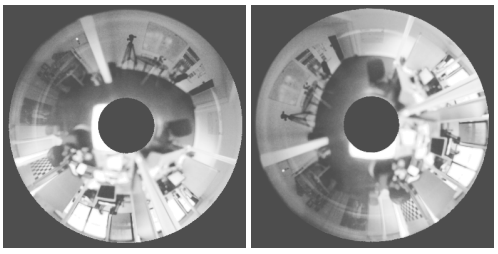
(a) (b)

Fig. 3 Two situations to address : (a) a very curvy trajectory is needed to reach the desired pose ; (b) a complex maneuver is needed to reach the desired pose

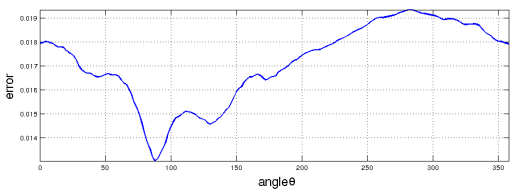
2.3 Visual compass

In the context of omnidirectional imaging, the visual compass refers to the algorithm that estimates the angle between an image $\mathbf{I}^*$ and a request image $\mathbf{I}$. In [7], the authors describe an algorithm that estimates such orientation angle. First, the two images are unwrapped into panoramic images. Assuming that the camera is perfectly vertical, a pure rotation around the camera axis will introduce a column-wise shift. The rotation angle is then computed by finding the best match between the reference image and the column-wise shift of the rotated image.

Instead of unwrapping the omnidirectional images, we propose to find the best match by directly handling the omnidirectional images. Assuming the camera's intrinsic parameters are known, $\mathbf{I}^*$ is rotated around the camera's principal point (u_0, v_0) by an increment θ. The SSD is then evaluated between the reference image, rotated by the increment $\mathbf{I}^*_\theta$, and the request image $\mathbf{I}$. The final angle is computed as the angle θ that achieves the smallest SSD between $\mathbf{I}^*_\theta$ and $\mathbf{I}$. Fig. 4 shows an example of computing such orientation angle.



(a)reference image (b)rotated image



(c)photometric error between the request image and the rotated reference image, wrt angle θ.

Fig. 4 Computing the angle between a reference and a request image using visual compass algorithm

3. OPVS for positioning

3.1 Problem overview

In order to solve the positioning task issues depicted in Figure 3, we introduce the following notations :

- $\overrightarrow{X}$ the robot direction at the current pose.
- $\overrightarrow{X^*}$ the robot direction at the target pose.
- $\overrightarrow{X^*_\theta}$ the target direction obtained by rotating $\overrightarrow{X^*}$ by θ.
- $\overrightarrow{X_p}$ the direction joining the initial to the target positions.

We denote the coincidence between two directions $\overrightarrow{X}$ and $\overrightarrow{X'}$ with the symbol $\equiv$. We can write the following : $\overrightarrow{X} \equiv \overrightarrow{X'}$ if $\overrightarrow{X}$ and $\overrightarrow{X'}$ are coincident, and $\overrightarrow{X} \not\equiv \overrightarrow{X'}$ otherwise.

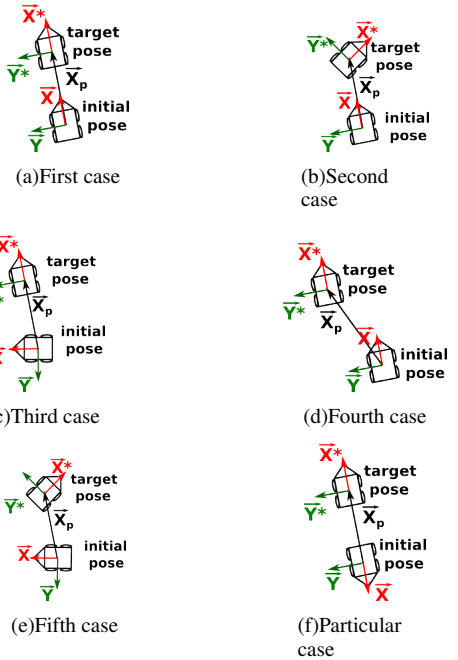
It turns out that five situations may be distinguished as well as a particular case (Fig. 5) :

1. First case (Fig. 5(a)) : a straightforward translation. This occurs when :

$$\overrightarrow{X_p} \equiv \overrightarrow{X} \equiv \overrightarrow{X^*} \tag{6}$$

2. Second case (Fig. 5(b)) : $\overrightarrow{X_p} \not\equiv \overrightarrow{X^*}$ but $\overrightarrow{X} \equiv \overrightarrow{X_p}$.
3. Third case (Fig. 5(c)) : $\overrightarrow{X_p} \not\equiv \overrightarrow{X}$ but $\overrightarrow{X^*} \equiv \overrightarrow{X_p}$.
4. Fourth case (Fig. 5(d)) : $\overrightarrow{X} \equiv \overrightarrow{X^*}$ and $\overrightarrow{X^*} \not\equiv \overrightarrow{X_p}$.
5. Fifth case (Fig. 5(e)) : which corresponds to the general case where $\overrightarrow{X_p} \not\equiv \overrightarrow{X}$, $\overrightarrow{X^*} \not\equiv \overrightarrow{X_p}$ and $\overrightarrow{X} \not\equiv \overrightarrow{X^*}$.

In the following, we discuss the five above-mentioned cases and propose a solution to each of them, gradually.



(a)First case (b)Second case

(c)Third case (d)Fourth case

(e)Fifth case (f)Particular case

Fig. 5 The possible cases to handle the proposed positioning task.

A. First case
In this case (Fig. 5(a)) the three axes $\overrightarrow{X}$, $\overrightarrow{X_p}$ and $\overrightarrow{X^*}$ are coincident. A simple translation using a visual servoing along $\overrightarrow{X}$ can be performed in order to reach the target pose.

B. Second case
In this case (Fig. 5(b)) $\overrightarrow{X} \not\equiv \overrightarrow{X^*}$ but $\overrightarrow{X} \equiv \overrightarrow{X_p}$. In order to satisfy condition of equation (6) the robot needs to perform the following tasks :

- Simulate a target rotation : the visual compass is used once for the computation of the angle θ between the current and the desired images. The desired image is then rotated with this angle as $\mathbf{I}^*_\theta$ to fit the orientation of the current image $\mathbf{I}$. A new target direction $\overrightarrow{X^*_\theta}$ is then obtained.
- Perform a translational visual servoing along $\overrightarrow{X_p}$ using the following control law :

$$\upsilon_X = -\lambda \left(\mathbf{H} + \mu \operatorname{diag}(\mathbf{H})\right)^{-1} \mathbf{L}^{\mathsf{T}}_{\mathbf{I}_\mathcal{S}} \left(\mathbf{I}_\mathcal{S}(\mathbf{r}) - (\mathbf{I}^*_{\mathcal{S}_\theta})(\mathbf{r}^*)\right), \tag{7}$$

  considering $\mathbf{I}^*_{\mathcal{S}_\theta}$ as the spherical representation of $\mathbf{I}^*_\theta$ as described in section 2.1 and $\mathbf{L}_{\mathbf{I}_\mathcal{S}}$ has only one column.
- Perform a robot rotation : regulation of the angle θ to zero (i.e. θ is re-computed for each current image).

$$\omega_Z = \gamma * \theta \tag{8}$$

Where γ is a tunable gain that controls the robot speed to reach the desired orientation.

C. Third case
In this case $\overrightarrow{X}$ and $\overrightarrow{X_p}$ are different but $\overrightarrow{X^*}$ and $\overrightarrow{X_p}$ are coincident. In order to have $\overrightarrow{X_p} \equiv \overrightarrow{X}$ the robot must rotate. A naive approach would be to perform a visual servoing on ω_Z. However, as stated in the introduction (Fig.2(b)), the convergence area of the cost function is limited to 22 degrees. To solve this issue, as in the second case, the desired image is oriented, as $\mathbf{I}^*_\theta$, to fit the current image orientation. The difference wrt the second case is that the axes $\overrightarrow{X}$ and $\overrightarrow{X^*_\theta}$ are only parallel but not coincident. Thus, we consider υ_Y to control the robot orientation. Indeed, υ_Y, computed with visual servoing, is null if and only if $\overrightarrow{X^*_\theta}$ and $\overrightarrow{X}$ are coincident. If not, the sign of υ_Y gives the rotation velocity sign, so :

$$\omega_Z = \alpha * \upsilon_Y, \tag{9}$$

where α is another tunable gain that controls the robot rotation speed to reach the desired orientation.
To sum up, the third case can be solved following these steps :

- simulate a target rotation to obtain $\overrightarrow{X^*_\theta}$ that is parallel to $\overrightarrow{X}$.
- rotate the robot using a visual servoing on υ_Y as described by Equation (9) to make $\overrightarrow{X}$ and $\overrightarrow{X_\theta}$ coincident, for the correct angle, hence coincident with $\overrightarrow{X_p}$ too.
- perform a translational visual servoing along $\overrightarrow{X}$.

D. Fourth case
In this case $\overrightarrow{X} \equiv \overrightarrow{X^*}$ but $\overrightarrow{X} \not\equiv \overrightarrow{X_p}$. Since $\overrightarrow{X^*}$ is already parallel to $\overrightarrow{X}$, we first need to rotate the robot using a visual servoing on υ_Y as in the third case, before performing the pure translation.
Finally, a second pure rotation is performed as the last step of the second case.

E. Fifth case
The axes $\overrightarrow{X}$, $\overrightarrow{X_p}$ and $\overrightarrow{X^*}$ have different orientations. To make visual servoing successful in this general case, we proceed first as the third case and finally considering the last step of the second case. The control algorithm is presented in the next section.

3.2 The proposed solution

Three main stages describe the positioning control algorithm :

1. Step 1 : First pure rotation, based on the initial and the desired images, the robot rotates to point to the desired position (Algorithm 1).

Algorithm 1: First Pure Rotation

Data: Desired image $\mathbf{I}^*$
Result: Velocity component ω_Z.
begin
for *each acquired Image* $\mathbf{I}$ **do**
Find θ the angle between $\mathbf{I}$ and $\mathbf{I}^*$ using a visual compass algorithm.
Rotate the desired image by θ, the result is $\mathbf{I}^*_\theta$.
Perform a visual servoing iteration using $\mathbf{I}^*_\theta$ on υ_Y.
Apply the velocity $\omega_Z = \alpha * \upsilon_Y$ to the robot.
end
end

At the end of the first pure rotation, the robot is not accurately pointing to the desired position. Thus using a pure translation to let the robot reaching the desired position is not sufficient. Hence, robot orientation should simultaneously be controlled with the translation instead of a pure translation control.

2. Step 2 : Combined rotation and translation, the robot is controlled by a simultaneous rotation/translation commands in order to reach the desired position (Algorithm 2).

Algorithm 2: Combined rotation and translation

Data: Desired image $\mathbf{I}^*$
Result: Velocity components υ_X and ω_Z.
begin
for *each acquired image* $\mathbf{I}$ **do**
Find θ the angle between $\mathbf{I}^*$ and $\mathbf{I}$.
Rotate $\mathbf{I}^*$ by θ, the result is $\mathbf{I}^*_\theta$.
Perform a visual servoing iteration using $\mathbf{I}^*_\theta$ on υ_X and υ_Y.
Apply the velocity $(\upsilon_X, \omega_Z = \beta * \upsilon_Y)$ to the robot.
end
end

3. Step 3 : Second pure rotation, finally the robot performs a pure rotation until the angle θ, between the current and the desired image, is null (Algorithm 3).

Algorithm 3: Second pure rotation

Data: Desired Image $\mathbf{I}^*$
Result: Velocity component ω_Z.
begin
for *each acquired image* **I do**
Find angle θ between current image **I** and desired image $\mathbf{I}^*$.
Apply velocity $\omega_Z = \gamma * \theta$ to the robot.
end
end

4. Particular case (Fig. 5(f)) : in this case the axes $\overrightarrow{X} \equiv \overrightarrow{X_p} \equiv \overrightarrow{X^*}$ but $\overrightarrow{X} \equiv -\overrightarrow{X^*}$. In order to accomplish the robot manoeuvre, the robot needs to simulate a target rotation as in the third case so that $\overrightarrow{X_\theta^*}$ has the same orientation as $\overrightarrow{X}$. The robot needs then to perform a backward translation in order to reach the target pose (unlike the previous cases where a forward translation is performed). Finally, the robot accomplishes a pure rotation as the last step of the second case.

3.3 Validation

Experiments were conducted using the setup presented in Fig. 1 : omnidirectional camera and the mobile robot pioneer 3-AT. For these experiments, eight scenarios corresponding to different initial positions on a circle of radius 60 cm (Fig. 6) are carried out. Starting from an initial position on the circle, the robot has to reach the target pose which corresponds to the center of this circle. The arrow corresponds to the robot orientation. We distinguish between two algorithms used in these experiments :

A. Using the proposed classical Omnidirectional Photometric Visual Servoing (classical OPVS)

In these experiments, we use the classical version of omnidirectional photometric visual servoing (*cf.* Section A.). First a desired image is taken and saved in the hard drive of the computer. Then for each of the eight poses, the robot moves using the classical OPVS. The videos showing these experiments can be found in this link : `http://home.mis.u-picardie.fr/~youssef/Videos/OPVS/classical_OPVS/`.

B. Using the enhanced omnidirectional photometric visual servoing

In these experiments, we applied our enhanced version of the omnidirectional photometric visual servoing (enhanced OPVS). It is noteworthy that the enhanced OPVS is also applied for poses 3 and 7 so that we can compare the performance of the enhanced version with the classical one. Regarding the pose 3, the robot is expected to not perform any rotation and then moving backward to the desired pose. Instead, as the video corresponding to the pose 3 shows, the robot performs a rotation of about 180 degrees and moves towards the desired target. This can be explained by the fact the visual compass algorithm outputs a non-zero angle at pose 3 which drives the robot to the next minimum corresponding to $v_Y = 0$ (i.e. particular case of Fig.5-f is almost theoretical). Figure 7 shows the generated paths for poses 1 and 6. The videos corresponding to these experiments can be found in this link `http://home.mis.u-picardie.fr/~youssef/Videos/OPVS/enhanced_OPVS/`.

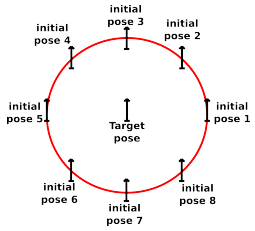
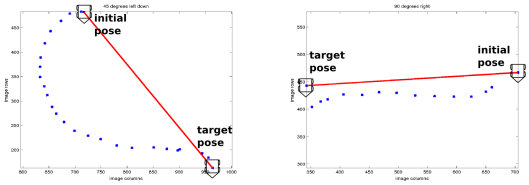


Fig. 6 The experiments with the mobile robot for the positioning task problem.



(a)Pose 6 (b)Pose 1

Fig. 7 Trajectories performed by the robot for the poses 1 and 6. The red line corresponds to the shortest path joining the initial to the target pose and the blue points correspond to the actual path.

3.4 Results analysis

The results in terms of metric error between the final position of the robot and the desired position are measured using a ruler and are presented in Table 1. The error is evaluated for the two schemes corresponding to the classical and enhanced OPVS and according to the X-axis (resp. Y-axis) wich corresponds to ΔX (resp. ΔY).

The parameters α, β, γ are experimentally set as the following $(\alpha, \beta, \gamma) = (5.0, 4.5, 0.5)$. The camera frame rate is 15 fps.

These results show that the enhanced OPVS outperforms the classical OPVS in terms of metric error. Actually, the classical OPVS works only for poses 3 and 7, while the enhanced OPVS works for all the poses. Regarding poses 3 and 7, the classical OPVS provides better results than the enhanced OPVS.

4. Conclusion

In this paper, we presented a new featureless visual servoing algorithm for non-holonomic robot navigation. Considering only image intensities, the proposed visual servoing algorithm allows to control the robot translation and orientation in order to reach the desired pose by performing pure rotation phases and computing the rotation velocity component during the translation step. In future works, we plan to improve the control scheme, still only

OPVS algorithm	Classical OPVS		Enhanced OPVS	
Error(cm) / Poses (Fig. 6)	ΔX	ΔY	ΔX	ΔY
Pose 1	0	60	0	2
Pose 2	2	41	0	5
Pose 3	0	1	4	1
Pose 4	1	47	1	5
Pose 5	5	58	3	3
Pose 6	0	46	0	2
Pose 7	0	1	3	3
Pose 8	4	38	3	0
Mean	1.5	36.5	1.75	2.62
Standard deviation	2	23.17	1.67	1.76

Table 1 Error in centimetres of the enhanced OPVS for positioning task versus the classical OPVS scheme.

considering the photometric feature, but allowing smooth motion instead of a decoupled one.

Acknowledgments

This work was supported by the project COALAS. The project COALAS was selected under the European cross-border cooperation program INTER-REG IV A France (Channel) England, co-funded by the ERDF.

References

[1] H.H. Abdelkader, Y. Mezouar, N. Andreff, and P. Martinet. Image-based control of mobile robot with central catadioptric cameras. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 3522 – 3527, april 2005.

[2] H. Aliakbarpour, O. Tahri, and H. Araujo. Visual servoing of mobile robots using non-central catadioptric cameras. *Robotics and Autonomous Systems*, 62(11) :1613 – 1622, 2014.

[3] J. P. Barreto and H. Araujo. Issues on the geometry of central catadioptric imaging. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 422–427, Hawaii, USA, December 2001.

[4] G. Caron, E. Marchand, and E. Mouaddib. Photometric visual servoing for omnidirectional cameras. *Autonomous Robots, AURO*, 35(2) :177–193, October 2013.

[5] F. Chaumette and S. Hutchinson. Visual servo control, Part I : Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4) :82–90, December 2006.

[6] Yoshihiko Mochizuki and Atsushi Imiya. Featureless visual navigation using optical flow of omnidirectional image sequence. In *Workshop of SIMPAR*, volume 8, pages 307–318, 2008.

[7] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Trans. on Robotics*, 24(5) :1015–1026, Oct 2008.

[8] R. Tatsambon Fomena, H. Yoon, A. Cherubini, F. Chaumette, and S. Hutchinson. Coarsely calibrated visual servoing of a mobile robot using a catadioptric vision system. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 5432–5437, St Louis, USA, October 2009.

[9] Jianguo Zhao, Yunyi Jia, Ning Xi, Weixian Li, Bo Song, and Liang Sun. Visual servoing using non-vector space control theory. In *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, pages 87–92. IEEE, 2012.