September 25-30, 2011
San Francisco, California
IROS '11
IEEE/RSJ International Conference on
Intelligent Robots and Systems

ViCoMoR

IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)

Half Day Workshop

September 26th, 2011, San Francisco, California,
in conjunction with the IEEE/RSJ International
Conference on Intelligent Robots and Systems

http://vicomor.unizar.es

Organizers

Youcef Mezouar
LASMEA - Université Blaise Pascal, France

Gonzalo López-Nicolás
I3A - Universidad de Zaragoza, Spain

Contents

Contributions:

Web: http://vicomor.unizar.es
IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)
September 26th, 2011, San Francisco, California, in conjunction with the
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

The organization of this workshop was partly supported by Ministerio de Ciencia e Innovación/Uni6n Europea, DPI2009-08126, the ANR R-discover project and grant I09200 from Gyeonggi Technology Development Program funded by Gyeonggi Province.

Aims and Scope

Autonomous navigation of mobile robots is a complex problem that has attracted the attention of the research community during the last decades. Systems capable of performing efficient and robust autonomous navigation are of interest in many robotic applications: automation industry, material transportation, assistance to disabled people, surveillance, etc. In order to perform navigation, the mobile robots need to interact with the environment, and for this purpose different types of sensors are nowadays available. From all of them, vision systems stand out because they provide very rich information and because of their versatility and low cost. The purpose of this workshop is to discuss topics related to the challenging problems of visual control of mobile robots.

Visual control refers to the capability of a robot to visually perceive the environment and use this information for autonomous navigation. The visual control is a multidisciplinary field of research that requires the collaboration of computer vision and robotics control experts. Nowadays, there is still a gap between computer vision and robotics communities that prevents more profitable results from jointly research. Thus, one of the goals of this workshop is to provide a forum for the communication of new ideas between these communities. Moreover, different fields of research can promote new methods, for example from biological inspiration, or taking advantage of known methods applied in different areas, such as robotic manipulation or UAV. We propose a half-day workshop to enhance active collaboration of computer vision and robotic researchers, discuss formal methods for visual control and identify current trends of the field. Achievements, challenges and open questions related with the visual control of mobile robots are welcome.

Topics

Topics of interest include:

- Autonomous navigation and visual servoing techniques for mobile robots.
- Visual perception for visual control, visual sensors and integration of image information in the control loop.
- Visual control with constraints: nonholonomic constraints, motion in formation, coordination, distributed visual control, obstacle avoidance, motion planning, etc.
- New trends in visual control, innovative solutions or proposals in the framework of computer vision and control theory.

Program committee

Hector M. Becerra	(Universidad de Zaragoza, Spain)
Enric Cervera	(Universitat Jaume-I, Spain)
François Chaumette	(INRIA Rennes - IRISA, France)
Peter Corke	(Queensland University of Technology, Australia)
Warren Dixon	(University of Florida, USA)
Francisco Escolano	(Universidad de Alicante, Spain)
Nicholas R. Gans	(Univ. of Texas at Dallas, USA)
Jose J. Guerrero	(Universidad de Zaragoza, Spain)
Koichi Hashimoto	(Tohoku University, Sendai, Japan)
Seth Hutchinson	(University of Illinois at Urbana-Champaign, USA)
Nicolas Mansard	(LAAS/CNRS, France)
Patrick Rives	(INRIA Sophia Antipolis, France)
Carlos Sagues	(Universidad de Zaragoza, Spain)
Kai-Tai Song	(National Chiao Tung University, Taiwan)
Dimitris P. Tsakiris	(FORTH, Heraklion, Greece)
Andrew Vardy	(Memorial University of Newfoundland, Canada)
Xenophon Zabulis	(FORTH, Heraklion, Greece)

Organizers

Youcef Mezouar
LASMEA - Université Blaise Pascal
24 avenue des landais, 63 177 Aubière Cedex, France
Phone: +33 (0)4 73 40 75 89
Fax : +33 (0)4 73 40 72 62
Email: youcef.mezouar@lasmea.univ-bpclermont.fr
Web: http://wwwlasmea.univ-bpclermont.fr/Personnel/Youcef.Mezouar

Gonzalo López-Nicolás
Instituto de Investigación en Ingeniería de Aragón - Universidad de Zaragoza
María de Luna 1, E-50018 Zaragoza. Spain
Phone: (+34) 976 762344
Fax: (+34) 976 761914
Email: gonlopez@unizar.es
Web: http://webdiis.unizar.es/~glopez

Invited speakers

François Chaumette

INRIA Rennes - Bretagne Atlantique - IRISA, Campus de Beaulieu 35042, Rennes, France.

Web: http://www.irisa.fr/lagadic/team/Francois.Chaumette-eng.html

Title: Recent results in visual control of mobile robots

Abstract: The talk will present an overview of the results obtained recently in visual control of mobile robots in the Lagadic group at INRIA Rennes-Bretagne Atlantique. A first part will be devoted to the visual navigation of a mobile robot in an urban environment, including obstacle avoidance thanks to the use of a laser range sensor. A second part will be devoted to the visual localization of flying robots for landing applications.

Nicholas R. Gans

Department of Electrical Engineering. Erik Jonsson School of Engineering and Computer Science. The University of Texas at Dallas.

Web: http://www.utdallas.edu/~ngans

Title: Vision-based control beyond position and velocity regulation

Abstract: The methods developed for visual servoing thirty years ago are still successfully used today. Current advances in the field still often employ the original design methods. However, there are problems that are not well addressed by the classic approaches. Furthermore novel approaches to these problems can often alleviate other problems, such as computationally expensive or error prone, feature extraction and matching and nonholonomic constraints on mobile robots. This talk will present two such research problems, tracking multiple objects with mobile cameras, and maximizing the information content contained in an image.

Program

September 26th, 2011, San Francisco, California

09:20 – 09:30	Presentation of the workshop
09:30 – 10:00	**Vision-based control beyond position and velocity regulation** N. R. Gans
10:00 – 10:20	**Planning high-level paths for visual control tasks with sensor constraints** J. B. Hayet
10:20 – 10:40	**One homography to control multiple robots** M. Aranda, G. Lopez-Nicolas, Y. Mezouar and C. Sagues
10:40 – 11:00	**Visual road recognition using artificial neural networks and stereo vision** P. Y. Shinzato, F. S. Osorio and D. F. Wolf
11:00 – 11:30	Coffee break
11:30 – 12:00	**Recent results in visual control of mobile robots** F. Chaumette
12:00 – 12:20	**Vision based control for Humanoid robots** C. Dune, A. Herdt, E. Marchand, O. Stasse, P.-B. Wieber and E. Yoshida
12:20 – 12:40	**Weakly-calibrated visual control of mobile robots using the trifocal tensor and central cameras** H. M. Becerra, G. Lopez-Nicolas and C. Sagues
12:40 – 13:00	**Automatic regions-of-interest selection based on pearson's correlation coefficient** A. M. Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri

Planning high-level paths for visual control tasks with sensor constraints.

J.B. Hayet

Abstract— **Traditionally, vision-based control algorithms have been defined as local strategies that formulate the control of the robot actuators in terms of some image-related features. These features may take the form of landmark points (corners, interest points or lines) that have to be detected and recognized, e.g. to compute an homography between the current image and the target one. At the higher level, recent works have proposed to compute robot paths in such a way that one particular point is kept in sight all the time during the trajectory execution, which does not necessarily mean that all the interesting points will be effectively seen. Here, we propose to use a high-level planner that guarantees that all the points actually tracked will be kept in sight e.g. for computing homographies and basing the visual control on them.**

I. INTRODUCTION

Visual control techniques have played a key role in the development of autonomous mobile platforms. In image-based visual servoing, one specifies locomotion tasks in terms of the position of some features in the image. In the most typical setup, visual servoing is defined in terms of configurations of points in the image, by which are defined both the initial and goal configurations. As a consequence, an important pre-condition for this scheme to be successful is that, at a higher level, the robot path should be defined or guided in such a way that the features that will be used during the control will be kept in sight all along the trajectory. Our work aims at exhibiting such paths, that (1) could be used as reference paths in a visual servoing control and (2) could guarantee geometrically that the perspective entities used in the visual control – typically, a homography computed between two sets of four points, the starting and ending image configurations – can be effectively computed with a limited field-of-view camera, i.e. that none of the points get out of the image.

A. Existing approaches

Many visual servoing schemes have been described in the literature that rely on the estimation of the epipolar geometry between the current camera configuration and the goal camera configuration [5]. However, in indoor environments, it is quite common to have to rely only on planar scenes, with which the estimation of epipolar geometry is more difficult [6]. Hence, visual control schemes have been proposed based on the homographies induced by the visible planes.

In the work of [4], the authors have defined a set of homography-based control laws designed in such a way that, by applying these laws, the robot follows the shortest trajectories maintaining a given punctual target in sight, as illustrated by Fig. 1. The geometric characterization of these particular trajectories, and the complete synthesis of shortest paths have been described in two separate works. An initial description was provided in [1] and then corrected in a later work [7]. Locally, it is shown that these trajectories are either logarithmic spirals, straight lines or in-site rotations, and the most remarkable element in the synthesis is that any pair of configurations can be reached by trajectories ("words") involving no more than five elementary paths ("letters"). In [4], control laws are given for a sub-set of optimal trajectories, namely straight lines and concatenation of two spirals. In our opinion, these control laws could be generalized for the complete synthesis proposed in [7]. However, there remains an important limitation to this approach, as maintaining only one target point in sight does not necessarily imply that the planar features we use for computing the inter-image homographies necessary for the control will be visible in all frames. For example, some of the shortest paths in [7], such as the ones made of two spirals, may cause the robot to derive from the straight line joining the initial and final configurations and get closer to the main target. This may produce a break in the visibility of the features we are not keeping in sight explicitly.

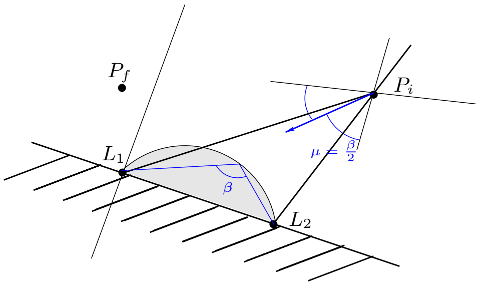


Fig. 1. Geometric setup for our problem: we aim at joining initial configuration $\mathbf{P}_i$ to final configuration $\mathbf{P}_f$, through visual controls based on homographies, where the homographies are computed with visual landmarks laying on the planar patch maintained in sight. β is the field of view. The shaded region is the locus of points such that the viewing angle on the planar path is superior to β (hence, not admissible configurations).

B. Contributions of this work

We propose a planner for a differential drive robot that guarantees that a given planar target is maintained in sight. In a sense, instead of [4], we do not *decouple* the problem of the visibility constraints (handled by the high-level planner) from the problem of vision-based control (handled by the control

J.B. Hayet is with Centro de Investigación en Matemáticas (CIMAT), 36240 Guanajuato, GTO., México jbhayet@cimat.mx

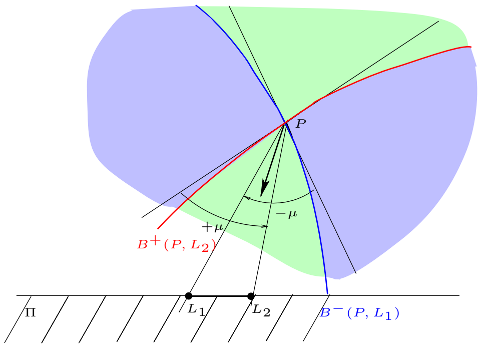


Fig. 2. Mobility around one configuration $\mathbf{P}$. The continuously derivable paths necessarily go through the green areas at $\mathbf{P}$. The $\mathcal{C}^1$ parts of shortest paths are necessarily among lines (in the interior of the green area) or logarithmic spirals (boundaries of the green area).

system), but instead, we consider the same features we use for the homography estimation as the geometric objects defining the visibility constraints. As it is shown below (in Section II), the shortest paths under these constraints are similar to the ones that keep one landmark in sight; in Section III, we will define a control scheme for realizing these paths, and in Section IV, we give the results of a few simulations based on this scheme. Finally, in Section V, we draw some conclusions about this approach.

II. HIGH LEVEL PLANS FOR VISUAL CONTROL

Consider Fig. 1 for a geometric setup of the problem. We want to plan paths in the plane, for a differential drive robot (hence, non-holonomic), controlled through its linear and angular velocities (v, ω). These paths should lead the robot from an initial configuration $\mathbf{P}_i$ to a final configuration $\mathbf{P}_f$, in such a way that a planar area, delimited by the two points $\mathbf{L}_1$ and $\mathbf{L}_2$, is kept in sight and is effectively used for the robot control, as explained in Section III. We suppose that the whole planar area is the surface of an obstacle (e.g., a wall), so that the robot can only be in the half-plane above the line $(\mathbf{L}_1\mathbf{L}_2)$. Let us call β the robot camera field-of-view and let us define $\mu = \frac{\beta}{2}$. The plane supporting the features will be referred to as Π.

A. Local properties of shortest paths

In [2], we have studied the main local properties of the shortest paths maintaining two landmarks in sight, and we recall here briefly the most important results. First, if one focuses on the neighborhood of a particular configuration $\mathbf{P}$, as in Fig. 2, one will immediately understand that the *local* nature of shortest paths is quite similar to the one studied in [1], [7] for the case of just one landmark. Indeed, any neighborhood will be partitioned into four zones, two of which being attainable from $\mathbf{P}$ by forward or backward motion, and the other two not being attainable directly. The curves delimiting these zones are, on the left (resp. right) side, the one on which the landmark $\mathbf{L}_1$ (resp. $\mathbf{L}_2$) is kept at a maximal viewing angle, i.e. with a value $+\mu$ (resp. $-\mu$), which means that the landmark is seen at the leftest (resp. rightest) position in the image. These curves are the same logarithmic spirals found in the 1-landmark case, the difference being that they are defined with respect to two separate centers. We will denote them by $\mathbf{B}^+(\mathbf{P}, \mathbf{L}_2)$ [1] (in red in Fig. 2) and $\mathbf{B}^-(\mathbf{P}, \mathbf{L}_1)$ (in blue). From this local structure, we have shown that, locally, and in this half-plane, the continuously derivable parts of the shortest paths can only be straight lines or parts of these spirals. Similarly to [1], we have also shown in [2] that the non-differentiable points on shortest paths could only correspond to in-site rotations from one saturated configuration to another one.

B. Globally optimal paths

From the previous remarks, we can define a new alphabet for shortest paths, based on straight line segments ($\mathbf{L}$ in the following), logarithmic spirals saturating the viewing angle on $\mathbf{L}_1$ on the left side of the image ($\mathbf{B}^-(\mathbf{L}_1)$), or saturating the viewing angle on $\mathbf{L}_2$ on the right side of the image ($\mathbf{B}^+(\mathbf{L}_2)$). A first important element of the synthesis is what is referred to in the quoted literature as the "S-sets" of a point $\mathbf{P}$, i.e. the set of points that are attainable from $\mathbf{P}$ through a straight line segment. This set is easy to build geometrically: it consists, *behind* $\mathbf{P}$, of an angular sector delimited by the lines corresponding to saturations of each of the two landmarks on their respective side, and *in front of* $\mathbf{P}$, of the intersection of two disks, each of them having $\mathbf{PL}_i$ $(i = 1, 2)$ as a chord, and denoting the maximal configurations (at saturation on $\mathbf{P}_i$) after starting a straight line from $\mathbf{P}$.

Moreover, as most of the properties already proven for one landmark are still valid for the case of two (see [2] for more details), we can reason in a similar fashion for establishing a synthesis of shortest paths, given a pair of initial and final configurations in the plane. However, an important difference with the 1-landmark case is the presence of an *obstacle* in the (x, y) space. Indeed, configurations for which the viewing angle on the segment $\mathbf{L}_1\mathbf{L}_2$ is superior to β are forbidden, so that a whole part of a disk, the locus of points $\mathbf{C}$ such that the angle $\widehat{\mathbf{L}_1\mathbf{C}\mathbf{L}_2} = \beta$, having $\mathbf{L}_1\mathbf{L}_2$ as a chord, has to be removed of the projection of the free space on the (x, y) plane. We will call this obstacle Γ_β. This new element complicates considerably the global characterization of shortest paths and implies to make a distinction among pairs of initial/final configurations: For some of them, the shortest path will be without contact with Γ_β, whereas for other pairs, the shortest path is necessarily in contact with Γ_β at some point.

a) **Paths in contact with Γ_β:** In this work, we will not try to exhibit the shortest paths in that case, which is illustrated by the example of Fig. 3. The paths are made of one or several cusps on Γ_β, at which point the tangent on both sides is the bisector of angle $\widehat{\mathbf{L}_1\mathbf{P}\mathbf{L}_2}$, and they appear as candidate optimal paths whenever the double-spiral curve that can be drawn from $\mathbf{P}_i$ to $\mathbf{P}_f$ first get in contact with Γ_β, before crossing the other spiral (this is the case in Fig. 3), or

[1] The plus indicates that the saturating viewing angle is positive.

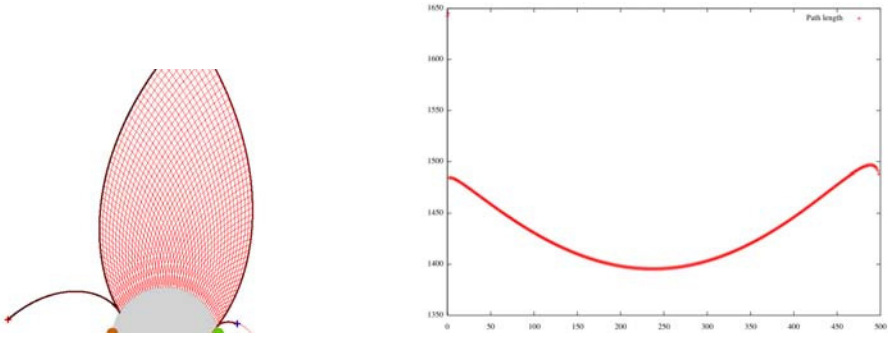


Fig. 4. Given a path with two cusps on the circle saturating the viewing angle (left), path length as a function of the position of the intermediary cusp on Γ_β (right).

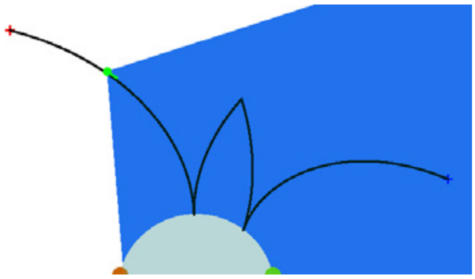


Fig. 3. An example of a path getting in contact with Γ_β. This path is made of four spirals, the landmarks are the two colored disks at the bottom, Γ_β is shaded.

if they do not cross at all. The main problem is in evaluating the number of cusps in the optimal path and their positions. Empirically, we have found that, given the double cusp of Fig. 3, it is *sometimes* possible to improve it by just dividing the inner curve of the cusps into two other curves, and that this happens when the two cusps are widely separated on Γ_β. It does not happen it they are close to each other. In Fig. 4, we give an example of this possible reduction: On the left hand side is one of the double cusp curve, for which we tried many alternative paths with one more intermediary cusp; On the right hand side, we plotted the path length function of the position of the intermediary cusp on Γ_β. Clearly, there is a reduction option for a cusp at some median value between the other two. Note in the same graph that the length function is not that simple: it seems to have two local maxima in addition of this minimum. For this paper, we will take the four spirals curve of Fig. 3 as the reference path, even if it is not always the shortest one.

b) **Paths not in contact with Γ_β:** For shortest paths not in contact with Γ_β, we can apply the same reasoning as with the 1-landmark case, especially as far as the number of possible "letters" in the "word" are possible. Shortest paths are done, in function of the relative position of $\mathbf{P}_i$ and $\mathbf{P}_f$, either through a $\mathbf{LB}^+(\mathbf{L}_2)\mathbf{B}^-(\mathbf{L}_1)\mathbf{L}$ word (or shorter), or a $\mathbf{LB}^-(\mathbf{L}_1)\mathbf{B}^+(\mathbf{L}_2)\mathbf{L}$ (or shorter). The aforementioned shorter variants of these are made of at maximum four non-rotation letters (in the case of four letters, we would add the initial, final and middle rotation) and are "sub-strings" of these two. For example $\mathbf{L}$, $\mathbf{LB}^+(\mathbf{L}_2)$, $\mathbf{B}^-(\mathbf{L}_1)\mathbf{B}^+(\mathbf{L}_2)\mathbf{L}$ are possible words. Now, the minimal length element inside the whole family of four non-rotation letter words joining two configurations is much more complicated to isolate than in the 1-landmark case. It involves transcendental equations to solve the intersection of the two spirals which are not defined in the same frames. However, it is a rather simple family to parameterize and simulate numerically, as can be seen in Fig. 5: each of this curve is uniquely determined by two angles, namely the angles defining the orientation of the starting and ending segments, and the minimum length curve in that case is in fact a $\mathbf{B}^-(\mathbf{L}_1)\mathbf{B}^+(\mathbf{L}_2)\mathbf{L}$ curve. For the remaining of this work, we will rely on this simple numerical optimization to compute the shortest path between configurations with candidate paths not in contact with Γ_β.

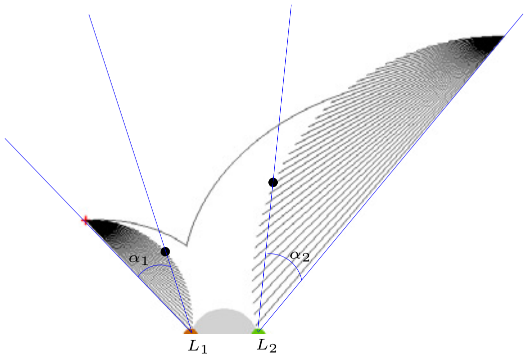


Fig. 5. Numerical evaluation of the shortest $\mathbf{LB}^+(\mathbf{L}_2)\mathbf{B}^-(\mathbf{L}_1)\mathbf{L}$ path between two configurations. The optimization is done in the (α_1, α_2) space, each possible curve of the family being uniquely characterized by a pair of angles (α_1, α_2). Each of the depicted line segment at the left (resp. right) correspond to a possible α_1 (resp. α_2).

III. APPLICATION TO VISUAL CONTROL STRATEGIES

In this section, we show how we can use a homography-based controller to follow the paths defined above, computed in function of the features used to estimate the homography. For simplicity we will suppose that the plane under observation is *vertical*.

A. Global control strategy

As a global control strategy, we will perform the following steps. Given a pair of start and goal images, in which a set of planar features $\{\mathbf{q}_k\}_k$ is present in both:

1) in the starting image, select a planar region of interest (not only a point), which horizontal bounds define the directions $\mathbf{l}_1$ and $\mathbf{l}_2$ to $\mathbf{L}_1$ and $\mathbf{L}_2$,
2) compute the homography between the two images,
3) decompose it [8] to get the (planar) rotation $\mathbf{R}$, the plane orientation $\mathbf{n}$, and, up to a scale factor, the translation between the robot position $\mathbf{t}$ and the distance to the plane, d,
4) choose an arbitrary scale factor,
5) with the scale factor, deduce $\mathbf{L}_1$ and $\mathbf{L}_2$ from $\mathbf{l}_1$ and $\mathbf{l}_2$, and $\mathbf{P}_f$ from t. We have now a map $(\mathbf{L}_1, \mathbf{L}_2, \mathbf{P}_i, \mathbf{P}_f)$ that we can use for planning.
6) Similarly, with this same scale factor, compute (1) the positions $\{\mathbf{Q}_k\}_{k\in[1,N}$ of the features $\{\mathbf{q}_k\}_{k\in[1,N}$ that will be used for homography estimation, (2) the shortest path sequence as in II and (3) the corresponding intermediary homographies,
7) apply the elementary control laws to reach each of the subgoal according to the corresponding path.

Given a shortest path computed by the numerical synthesis of Section II, we need to define image-defined sub-goals for each of the "letters" composing the shortest path. As an example, if the shortest path is given by a $\mathbf{LB}^-(\mathbf{L}_1)\mathbf{B}^+(\mathbf{L}_2)\mathbf{L}$ path (such as in Fig. 5), then we generate the five sets of feature projections, $\{\mathbf{q}_k^{(s)}\}_{k\in[1,N]}\}$ where s indexes the path primitive, and k the feature. These sets are the goal to reach in each sub-path, and the only information available during the execution are the current $\{\mathbf{q}_k\}_{k\in[1,N]}\}$, at one given instant (and their correspondences to the goal set).

B. Control strategy along elementary paths

To implement the homography-based control along each of the primitive paths, we have taken the same methodology as [4], where separate control laws are provided for straight line segments and concatenations of two spirals, for which stability results are given. Here, we process each primitive with an individual image-based goal and the corresponding control. We recall that in the planar motion case and with a vertical underlying plane, the inter-image euclidean homography $\hat{\mathbf{H}}$ corresponding to two robot configurations such that the first one is written (x, z, ϕ) or (ρ, α, ϕ) (polar coordinates) in the second one (z pointing in the direction of the robot)

$$\begin{array}{rcl} \hat{\mathbf{H}} & = & \mathbf{K}^{-1}\mathbf{H}\mathbf{K} \\ & \propto & \begin{pmatrix} \cos\phi + \rho\cos(\phi-\alpha)n_x & 0 & \sin\phi + \rho\cos(\phi-\alpha)n_z \\ 0 & 1 & 0 \\ -\sin\phi + \rho\sin(\phi-\alpha)n_x & 0 & \cos\phi + \rho\sin(\phi-\alpha)n_z \end{pmatrix}, \end{array}$$

where $\mathbf{K}$ are the camera intrinsic parameters, $\mathbf{H}$ the projective homography, that one computes from a set of corresponding pairs (generally by SVD). The controls developed in [4] act on these terms, more precisely on the equivalent terms of $\mathbf{H}$. By developing a bit further their expression, we can make appear clearly all the terms involved and how they need to be evaluated during the control. For example, to follow a spiral $\mathbf{B}$, we apply

$$\left\{ \begin{array}{rcl} v & = & k_v(\hat{\mathbf{H}}_{11} - \hat{\mathbf{H}}_{33}) \\ \omega & = & k_\omega(\hat{\mathbf{H}}_{13}(1 - \frac{\sin(\phi+\alpha+m)}{\sin(\phi+\alpha)}) + \frac{\sin m \sin\alpha}{\sin(\phi+\alpha)}) \\ m & = & \arctan\left(\frac{\mathbf{l}_{1,u} - W}{\alpha_u}\right) \end{array} \right. ,$$

where W is the image width, $\alpha_u = \mathbf{K}_{11}$ the focal in pixels and $\mathbf{l}_1$ is the projection on the image of $\mathbf{L}_1$. k_v and k_ω are control gains, whose sign may depend on the way a primitive is traversed, and on the relative orientation between the plane and the current translation vector, as explained in [4]. The terms α and ϕ correspond to a knowledge of the relative position of the robot w.r.t. the goal position, and they can be estimated by decomposition of the estimated homography $\mathbf{H}$. The controls for the spirals and the rotations are similar. As for the lines, since we already use position information for the other primitives, we choose a position-based controller

$$\left\{ \begin{array}{rcl} v & = & k_v(\hat{\mathbf{H}}_{11} - \hat{\mathbf{H}}_{33}) \\ \omega & = & k_\omega \phi \end{array} \right. .$$

IV. SIMULATION RESULTS

To validate the approach, we wrote a small JAVA program to simulate motion queries expressed as formulated above. To simplify the implementation, we supposed that all starting and ending rotations where automatically done, and we let the user choose the starting and final configurations in the Euclidean plane, from which we deduced the intermediate goal configurations. As stated previously, this is equivalent to define the initial and final configurations, the features for the homography, their planar support, and the field of interest directly from the image, after a "reconstruction" up to a scale.

In these simulations, the homography decomposition is done through the SVD-based algorithm described in [8], proposed in the context of auto-calibration. We suppose that we know the intrinsic parameters matrix $\mathbf{K}$. The homographies between the current image and the goal image are computed by a SVD from the overdetermined set of equations given by the correspondences, we have used only 7 features, which explains that the computed homography is often quite noisy, in particular when the point positions are perturbed by noise.

The first experiment (Figs. 6, 7, 8) is in a nearly noiseless case (Gaussian $\sigma = 10^{-3}$ pixels additive noise on point coordinates). It corresponds to the execution of a $\mathbf{LB}^-(\mathbf{L}_1)\mathbf{B}^+(\mathbf{L}_2)$, that is made of four distinct primitives (counting the in-site rotation), for $\beta = 2$. In Fig. 6, we show the ground truth trajectory, and the sub-goals to reach. In Fig 7, we show that the error levels are kept quite low, both in the image and with respect to the ground truth position of the goal, except at the end of the second spiral, where one observe for some time stronger orientation errors and a non-null residual in the X error. Finally, Fig. 8 shows the linear and angular velocities. The different primitives are clearly distinguishable.

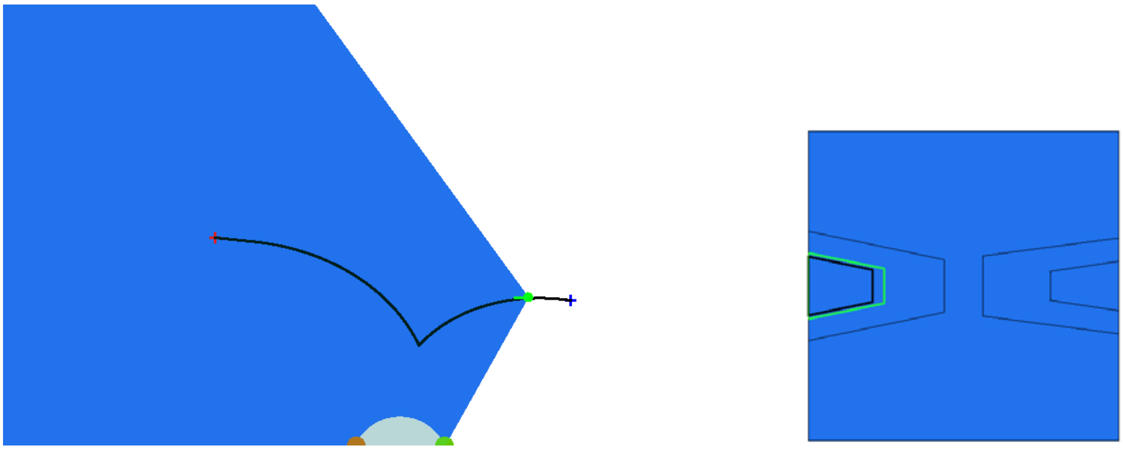


Fig. 6. Left, the planned path and its execution through homography-based control. The path is made of four primitives (a line, a forward spiral, an in-site rotation and a final backward spiral. Right, the different sub-objectives (for commodity, we only depicted the four surrounding features). Note the saturation on one side for all of them. The current position (left) and view (right) are depicted in green.

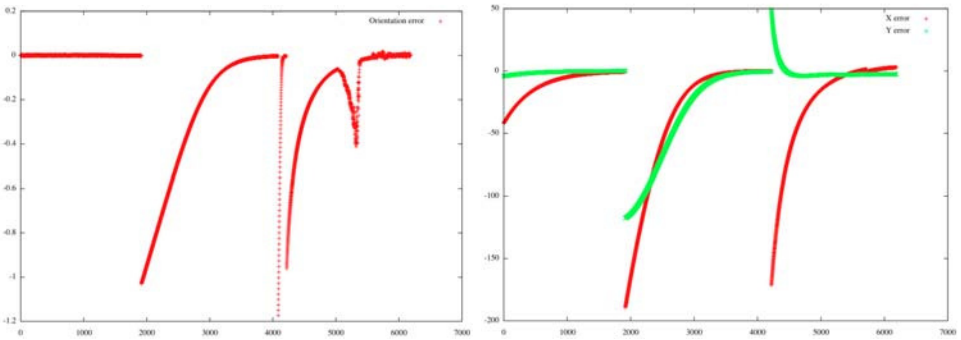


Fig. 7. Errors for the experiment of Fig. 6. Left, errors in orientation and, right, errors in X and Y (in both case, actual pose vs. pose corresponding to the goal). To give an idea of the scale of these errors, the planar area in which the experiment is done (i.e., the size of Fig. 6, left) is 900 × 500.

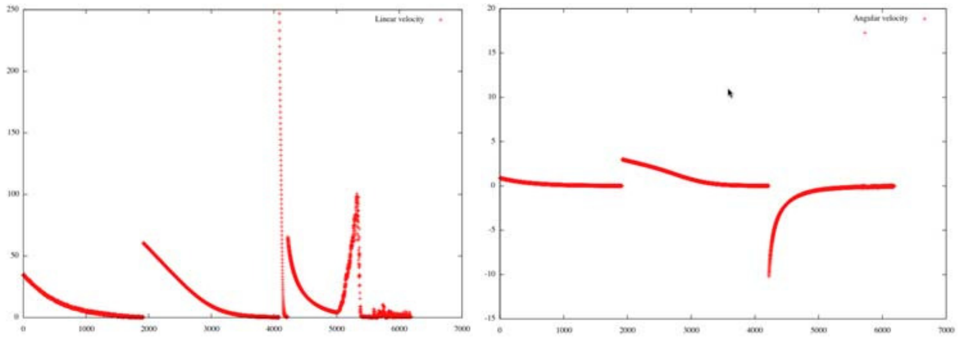


Fig. 8. Linear and angular velocities profiles.

With noisy data ($\sigma = 1$), we produced the second experiment, in which the path (see Fig. 9) is a $\mathbf{B}^{-}(\mathbf{L}_1)\mathbf{B}^{+}(\mathbf{L}_2)$ one. The error levels are higher, of course, but the execution of the path remains reasonable, the main problem occurring for the orientation, again, at the end of the second spiral, and with residual position errors at the end of both spirals. A more robust decomposition algorithm could perhaps improve the overall performance.

V. CONCLUSIONS AND FUTURE WORKS

We have presented a motion planner for differential drive robots that outputs paths maintaining two points into coverage, thus guaranteeing for example that an entire planar patch of the environment could stay visible, e.g. for visual control purposes. The planner is similar in many ways to the existing proposals for one landmark, in particular as it induces the same kind of local primitives for shortest paths, straight lines and logarithmic spirals. We have integrated it with an homography-based controller initially developed for

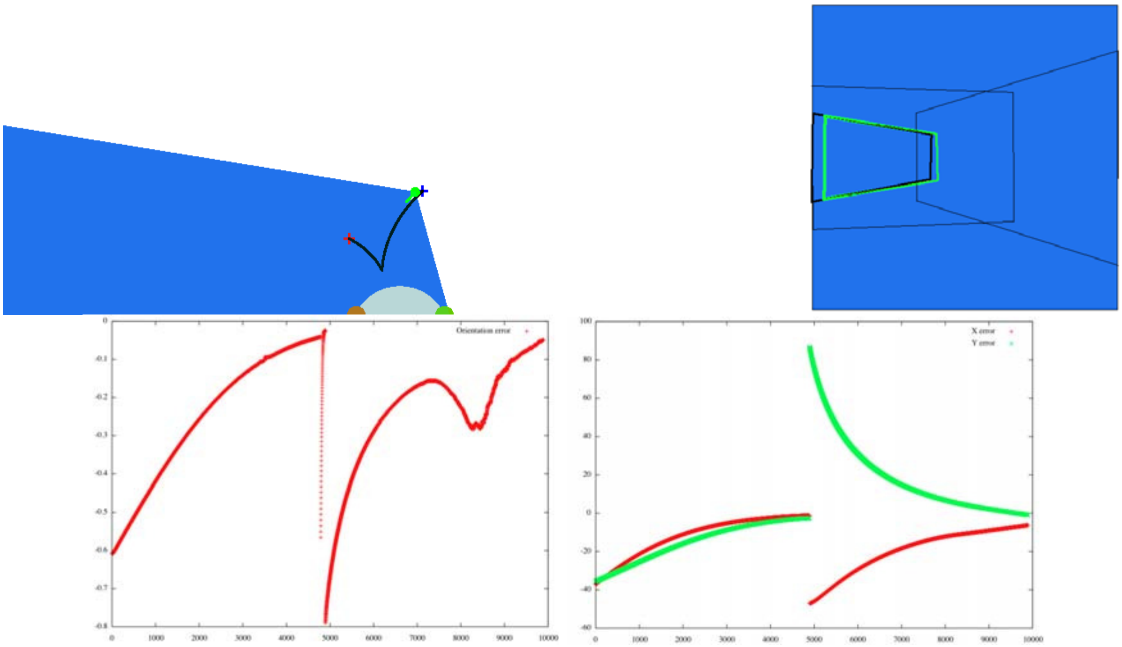


Fig. 9. Second experiment with noisy input data, $\sigma = 1$. The trajectory is a double spiral with an in-site rotation.

the primitives of the one-landmark case, and we have shown that it could be adapted in a straightforward way. In our ongoing work, we first aim to give a complete path synthesis for this specific problem, which is difficult in the case paths in contact with Γ_β are considered; we also want to use our works of [2] and [3] to extend this kind of strategy in two directions: (1) in surveillance contexts, it could be interesting to find optimal paths that keep a set of planar locations in sight (e.g. the vertices of a polygon), and (2) in the presence of obstacles, since in this scenario the free case primitives could be an element for building admissible paths among obstacles.

References

[1] S. Bhattacharya, R. Murrieta-Cid, and S. Hutchinson. Optimal paths for landmark-based navigation by differential-drive vehicles with field-of-view constraints. *IEEE Trans. on Robotics*, 23(1):47–59, 2007.

[2] J.B. Hayet. Local properties of the shortest length paths for a differential drive robot keeping a set of landmarks in sight. *Journal of Intelligent and Robotic Systems (In press)*, 2011.

[3] J.B. Hayet, Claudia Esteves, and Rafael Murrieta-Cid. A motion planner for maintaining landmark visibility with a differential drive robot. In Springer, editor, *Proc. of the Eight Int. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2008.

[4] G. López-Nicolás, S. Bhattacharya, J.J. Guerrero, C. Sagüés, and S. Hutchinson. Switched homography-based visual control of differential drive vehicles with field-of-view constraints. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 4238–4244, 2007.

[5] G. López-Nicolás, C. Sagüés, J.J. Guerrero, D. Kragic, and P. Jensfelt. Nonholonomic epipolar visual servoing. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'06)*, 2006.

[6] David Nister. An efficient solution to the five-point relative pose problem. In *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, volume 2, page 195, 2003.

[7] P. Salaris, D. Fontanelli, L. Pallottino, and A. Bicchi. Shortest paths for a robot with nonholonomic and field-of-view constraints. *IEEE Trans. on Robotics*, 26:269–281, 2010.

[8] Bill Triggs. Autocalibration from planar scenes. In *Proc. of the European Conference in Computer Vision (ECCV)*, pages 89–105, 1998.

IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)
September 26, 2011, San Francisco, California, USA

One Homography to Control Multiple Robots

M. Aranda[1], G. López-Nicolás[1], Y. Mezouar[2] and C. Sagüés[1]

Abstract— **This paper presents a visual control method to be used on a set of mobile robots. We consider a framework where the robots have nonholonomic constraints, move in a plane and are observed by a calibrated flying camera, which provides the only sensory information used for the control. The objective of the control task is to drive the group of robots to a desired configuration. This configuration is simply defined by an image, avoiding the need for additional information. The task is carried out through an image-based control scheme using the homography induced by the multi-robot system. Collision avoidance between the robots is also performed, using a simple image-based method. The performance of the proposal is illustrated through simulations.**

I. Introduction

Multi-robot systems are an important research area in current robotics, due to their suitability to perform certain tasks (such as exploration, surveillance, security or rescue operations) that are difficult for one single robot. In particular, a number of research works in this field focus on the problem of reaching and maintaining a robot team in a particular configuration [1] [2] [3].

Vision sensors have been extensively used for robot localization, navigation and control. Visual control is a wide field of research that has attracted the attention of many researchers [4]. In multi-robot systems, it is common to have a setup where each robot is equipped with a local perception system, and they share their information to accomplish the global task. This is the case, for example, in the localization method for multiple mobile robots presented in [5]. Another related work is [2], where groups of mobile robots are controlled to visually maintain formations, including the situation where communication between the robots is not available. The vision-based formation control with feedback-linearization proposed in [1] tackles the issue of switching between decentralized and centralized cooperative control.

Centralized multi-robot control approaches provide several advantages: they allow simple and cheap robots, and release their local resources by transferring expensive computations to an external computer. A centralized architecture is considered for the leader-follower control proposed in [3], where the perception system consists of a fixed camera on the ceiling. In general, vision-based tasks become more robust when multiple view geometry constraints are imposed [6]. Particularly, the homography is a well-known geometric model across two views induced by a plane of the scene, and it has been used extensively in visual control [7], [8], [9].

In this paper, we consider a framework where the multiple robots are assumed to have nonholonomic motion constraints and move in a planar surface. The goal of the proposed control scheme is to drive the multiple robots to a desired configuration defined by an image previously taken of that configuration. The visual information is acquired by a flying camera undergoing an arbitrary planar motion and looking downward at the robots. The camera moves in such a way that its translation is parallel to the motion plane of the robots and its rotation is parallel to the plane normal.

We propose a homography-based control approach that takes advantage of the planar motion constraint to parameterize the homography. With this particular parametrization, the approach can be used in a set of two or more robots. The image features we employ to compute the homography are the projections of the multiple robots on the image plane. Then, the computed homography gives information about the configuration of the set of robots. In particular, from the homography we can determine if the configuration of the robots is rigid, i.e. they maintain the desired configuration defined by the target image, or nonrigid, meaning that the robots are in a different configuration. We propose an image-based control law where a desired homography is defined as a reference for the control in order to drive the robots to the desired configuration. The use of the homography for multi-robot formation makes our approach different from other image-based techniques. In particular, the interaction is expressed and handled through this homography.

It is always necessary to perform collision avoidance when controlling multiple mobile robots. Being a key element of robotic navigation, obstacle avoidance has been extensively studied for many years. Classical solutions to the problem include the potential field methods [10]. These methods have some well-known shortcomings [11], which has prompted the appearance of many modifications and improvements on them [12], [13]. Potential-like methods continue to be widely used for obstacle avoidance, mainly due to their simplicity. Collision avoidance in multiple-robot scenarios has to deal with both inter-agent and external collisions. Simple, decentralized approaches are usually preferred. In our case, we use a potential-like collision avoidance method based on gyroscopic forces, similar to the approaches employed in [14], [15]. Our method is image-based, simple and fast to compute, well suited to nonholonomic vehicles, and can

[1] Instituto de Investigación en Ingeniería de Aragón. Universidad de Zaragoza, Spain. {marandac, gonlopez, csagues}@unizar.es
[2] LASMEA - Université Blaise Pascal, Aubiere 63177, France. youcef.mezouar@lasmea.univ-bpclermont.fr

This research was supported by grant I09200 from Gyeonggi Technology Development Program funded by Gyeonggi Province, by project DPI2009-08126, the ANR R-discover project, grant AP2009-3430 Ministerio de Educación and the I3A Fellowship Program.

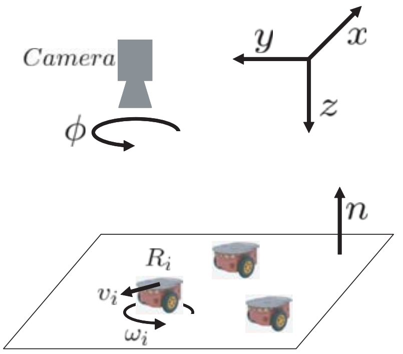


Fig. 1. Coordinate system: The motion of the camera occurs in the $x - y$ plane of the global reference and the robots undergo planar motion parallel to the $x - y$ plane. The rotation of the camera is also parallel to the plane normal $\mathbf{n}$.

be implemented as a decoupled term added directly to the control inputs.

This paper extends the work presented in [16]. Here, we propose a different control law which carries out the multi-robot configuration control task in one single step. With this scheme, only the positions of the robots in the configuration are controlled, while their final orientation is not regulated. This is suitable for cases where, for instance, the robots have omnidirectional capabilities, and the final orientation is irrelevant. The other main difference with respect to the aforementioned previous work is that in this paper we incorporate a collision avoidance mechanism, which is a necessary element in any multi-robot control implementation.

The paper is organized as follows. Section II presents the parametrization of the homography and the definition of the desired homography for reaching the multi-robot goal configuration. The control law for the multi-robot system and the collision avoidance method employed are presented in section III. In Section IV the performance of the proposal is illustrated through simulations. Finally, the conclusions of the paper are given in Section V.

II. Homography-Based Scheme

The setup of the multi-robot system and the flying camera is illustrated in Fig. 1, where the global fixed Left-handed coordinate system is depicted. In the following, we parameterize the homography in this framework and describe the method to compute linearly the homography. Then, we propose a procedure to define the desired homography that corresponds to the desired configuration of the multi-robot system.

A. Homography Parametrization

Two perspective images can be geometrically linked through a plane by a homography $\mathbf{H} \in \mathbb{R}^{3\times 3}$. This projective transformation $\mathbf{H}$ relates points of the plane projected in both images. Pairs of corresponding points $(\mathbf{p}, \mathbf{p}')$ are then related up to scale by $\mathbf{p}' = \mathbf{H}\,\mathbf{p}$. The calibrated homography can be related to camera motion and plane parameters as follows

$$\mathbf{H} = \mathbf{R} + \mathbf{T}\,\mathbf{n}^T/d\,, \tag{1}$$

where $\mathbf{R}$ and $\mathbf{T}$ are the relative rotation and translation of the camera, $\mathbf{n}$ is the unit normal of the plane with respect to the reference frame and d is the distance along $\mathbf{n}$ between the plane and the reference position. In the framework considered, the position of the camera $(x, y, z)^T$ is constrained to the plane $x - y$ (i.e. $z = 0$) and rotation ϕ about the z-axis. This constraint yields

$$\mathbf{R} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}, \tag{2}$$

with $\mathbf{T} = -\mathbf{R}(x, y, 0)^T$.

In our framework, the mobile robots move in a planar surface that generates the homography. Besides, the camera undergoes planar motion: the translation is parallel to the plane and the rotation is parallel to the plane normal, i.e. the z-axis, and $\mathbf{n} = (0, 0, -1)^T$. Notice that the distance d is the height of the camera with respect the motion plane of the robots. Therefore, the homography matrix is given by

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\phi & \sin\phi & -t_x/d \\ -\sin\phi & \cos\phi & -t_y/d \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{3}$$

B. Homography Computation

The homography across two views can be computed from a minimal set of four point correspondences solving a linear system [17]. In our framework, the points considered consist of the projection of the robots on the image plane, and they are denoted in homogeneous coordinates by $\mathbf{p} = (p_x, p_y, 1)$. A point correspondence $(\mathbf{p}, \mathbf{p}')$ is related up to scale by the homography as $\mathbf{p}' = \mathbf{H}\mathbf{p}$, which can be expressed in terms of the vector cross product as $\mathbf{p}' \times \mathbf{H}\mathbf{p} = \mathbf{0}$ [17]. From this expression two linearly independent equations in the entries of $\mathbf{H}$ (3) are obtained

$$\begin{bmatrix} p_x & p_y & 1 & 0 & -p'_x \\ p_y & -p_x & 0 & 1 & -p'_y \end{bmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{23} \\ h_{33} \end{pmatrix} = 0\,. \tag{4}$$

Each point correspondence gives two independent equations. Given that $\mathbf{H}$ is defined by seven unknown entries, and using the homography constraints $h_{11} = h_{22}$ and $h_{21} = -h_{12}$, a set of two point correspondences allows to determine the homography up to a scale factor by solving a linear system. Given that h_{33} is never zero because of the particular form (3), the scale of the homography can always be normalized and fixed by this entry.

C. The Target Homography

Each pair of robots induce a homography across two images, the current image and the image of the desired configuration. Given a set of N robots, the number of homographies defined by the different pair of robots is

$N(N-1)/2$. If all of these homographies are equal, the relative motion of the robots is rigid. Otherwise, if any of the homographies is different to the others, the relative motion of the set of robots is not rigid and they are not in the desired configuration. A desired homography computed using all robots needs to be defined in order to lead the robots to the desired configuration.

In the first case, the homography induced by the plane of the robots moving in the desired configuration is conjugate to a planar Euclidean transformation given by

$$\mathbf{H}_{rigid} = \begin{bmatrix} \cos\phi & \sin\phi & h_{13} \\ -\sin\phi & \cos\phi & h_{23} \\ 0 & 0 & 1 \end{bmatrix}. \qquad (5)$$

Notice that the upper left hand 2×2 matrix is orthogonal. The Euclidean transformation produces a translation and rotation of the image, and lengths and angles are invariants by this transformation.

The angle of rotation is encapsulated in the eigenvalues of (5) given by $\{1, e^{i\phi}, e^{-i\phi}\}$. Then, from the general expression of the homography, it can be deduced that $\mathbf{n} = (0,0,-1)^T$ and relative motion up to scale $(x,y,0)^T$ analogue as the assumptions defined for the homography parametrization. In this case, the robots are in formation with all the homographies induced by pairs of robots equal to the homography computed from all the robots (5) .

In the second case, the motion of the robots is not rigid, and they are not in the desired configuration. Then, the computation of the homography gives a matrix of the form

$$\mathbf{H}_{nonrigid} = \begin{bmatrix} s\cos\phi & s\sin\phi & h_{13} \\ -s\sin\phi & s\cos\phi & h_{23} \\ 0 & 0 & 1 \end{bmatrix}, \qquad (6)$$

where the upper left hand 2×2 matrix is no longer orthogonal. This previous matrix corresponds to a similarity transformation, i.e. translation, rotation and isotropic scaling represented by the scalar s. Angles and ratios of lengths are invariants by this transformation. The eigenvalues of this similarity are $\{1, s\,e^{i\phi}, s\,e^{-i\phi}\}$ and encapsulate the rotation angle. Comparison with the general expression of the homography leads again to $\mathbf{n} = (0,0,-1)^T$ but to a computed relative motion $(x, y, (s-1)d^2)^T$ up to scale, with $z \neq 0$. Therefore, the nonrigid motion of the robots induces a valid homography but not constrained to the assumed camera motion. We need to define a desired homography like $\mathbf{H}_{nonrigid}$, but being induced by a motion that keeps the camera motion constraints. This can be done normalizing (6) to make the upper left hand 2×2 matrix orthogonal and setting $h_{33} = 1$ to hold the planar motion constraint of the camera $(z = 0)$. Alternatively, we can simply normalize the upper left hand 2×2 matrix and obtain the desired homography with

$$\mathbf{H}^d = \mathbf{H}_{nonrigid} \begin{bmatrix} 1/s & 0 & 0 \\ 0 & 1/s & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad (7)$$

where s is computed as the norm of the upper left hand 2×2 matrix of $\mathbf{H}_{nonrigid}$. Then, the goal is to control the robots in such a way that all the homographies are led to $\mathbf{H}^d$ to reach the desired configuration.

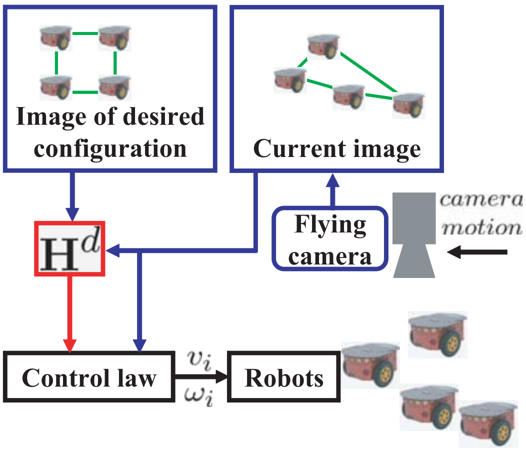


Fig. 2. Overview of the control loop. In each iteration of the control, the flying camera takes a current image of the robots, the desired homography $\mathbf{H}^d$ is obtained and used in the control law to compute the robot velocities necessary to reach the desired configuration of the multi-robot system.

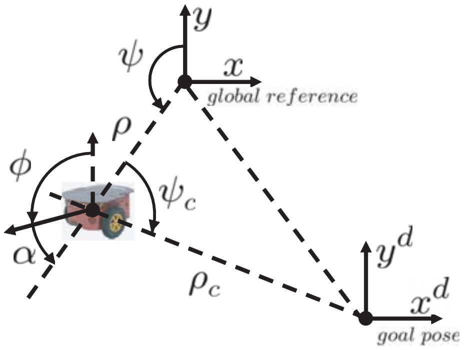


Fig. 3. Coordinate systems from a top view of the 3D scene. The robot position is given by $(x, y, \phi)^T$ or $(\rho, \alpha, \phi)^T$ in the global reference. The different parameters depicted are described along the text.

The $\mathbf{H}_{nonrigid}$ relates each point $\mathbf{p}$ of the current image with the corresponding point $\mathbf{p}'$ in the desired formation image with $\mathbf{p}' = \mathbf{H}_{nonrigid}\,\mathbf{p}$. The desired homography $\mathbf{H}^d$ is used now to define the goal location of the points in the image as $\mathbf{p}^d = (\mathbf{H}^d)^{-1}\,\mathbf{p}'$. Notice that the desired location of the robots in the image computed from the desired homography is not constant and varies along the time depending on the motion of the camera and the robots.

III. Visual Control Law

From the desired homography computed as explained in the previous section, we propose a control scheme to drive the robots to the desired configuration defined by an image of that configuration. An overview of the control loop is depicted in Fig. 2.

A. Robot Model and Coordinate Systems

Different coordinate systems defined in the 3D space are depicted in Fig. 3. The state of each robot is given by $(x, y, \phi)^T$, where ϕ is the orientation of the robot expressed

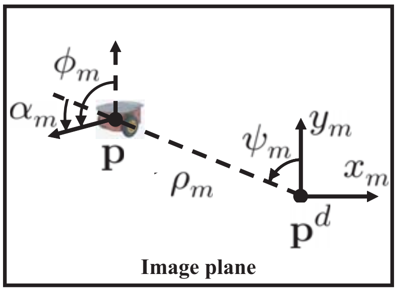


Fig. 4. Coordinate systems on the image plane for each robot. Subindex m denotes that the variable is defined on the image plane (the same variable without subindex m refers to the 3D space). Point $\mathbf{p}$ is the image projection of a robot and $\mathbf{p}^d$ its location to reach the desired configuration of the multi-robot system.

as the angle between the robot body y-axis and the world y-axis. Each robot has two velocity inputs, the linear velocity v and angular velocity ω, with v in the direction of the robot y-axis, and ω about the robot z-axis. The kinematics of each robot can be then expressed in general in polar or Cartesian coordinates in a fixed reference as

$$\left\{\begin{array}{l}\dot{\rho} = v\,\cos\alpha\\ \dot{\alpha} = \omega - \frac{v}{\rho}\sin\alpha\\ \dot{\phi} = \omega\end{array}\right., \text{and} \left\{\begin{array}{l}\dot{x} = -v\,\sin\phi\\ \dot{y} = v\cos\phi\\ \dot{\phi} = \omega\end{array}\right., \quad (8)$$

respectively, being

$$x = -\rho\sin\psi \quad \text{and} \quad y = \rho\cos\psi\,. \quad (9)$$

The alignment error α is defined as the angle between the robot body y-axis and the distance vector ρ,

$$\alpha = \phi - \psi\,. \quad (10)$$

We now introduce several variables, depicted in Fig. 4, to define the state of each robot on the image plane with (ρ_m, ψ_m, ϕ_m). The origin of the coordinate system for each robot $\mathbf{p}$ on the image plane is placed in the desired location $\mathbf{p}^d$, i.e. the robots are in the desired configuration when all of them are in the origin of their respective references ($\mathbf{p}^d$).

The variable ρ_m denotes the distance of the projection of a robot in the image $\mathbf{p}$ with respect to its desired position on the image $\mathbf{p}^d$, and so

$$\rho_m = \sqrt{(p_x - p_x^d)^2 + (p_y - p_y^d)^2}\,, \quad (11)$$

and also

$$\psi_m = atan2\left(-(p_x - p_x^d), (p_y - p_y^d)\right)\,, \quad (12)$$

where function $atan2$ returns the value of the arc tangent using the sign of the arguments to determine the quadrant. ϕ_m can be computed directly from the image of the robot with computer vision techniques or estimated with $\phi_m = atan2(-\Delta p_x, \Delta p_y)$, where Δp_x and Δp_y is the incremental motion of the robot in the image plane. The alignment error on the image α_m is also defined as $\alpha_m = \phi_m - \psi_m$.

B. Control Law

We define the control law in order to bring the robot team to the desired configuration. The control is carried out in one single step. It is assumed that the final orientation of the agents is not relevant. This would be the case, for example, in an application where a secondary task is performed by robots having omnidirectional capabilities. Having this assumption in mind, we define the controller as follows:

$$\left\{\begin{array}{l}v = -k_v\,\rho_m\\ \omega = -k_\omega\,(\alpha_m - \pi)\end{array}\right., \quad (13)$$

where $k_v > 0$ and $k_\omega > 0$ are control gains.

This control law drives the robots to their target positions so that the team reaches the desired configuration. The image projection of the distance to the desired position ρ_m and the alignment error α_m are measured directly in the image plane.

C. Collision avoidance

We choose to perform the collision avoidance task with a method based on gyroscopic forces. These types of approaches [14], [15] have their origin in navigation function methods (NFM) [18], and present some interesting properties. First, they are simple and fast. In addition, they are suitable to be used directly on a nonholonomic vehicle as steering commands. Gyroscopic forces can be implemented simply as an additive term in the control law, decoupling the collision avoidance from the control task. Also, their application does not change the energy of the system, which is an interesting characteristic from the stability viewpoint. Lastly, they are known to be able to avoid certain trap situations that come about with collision avoidance methods based on navigation functions.

In order to avoid inter-agent collisions, we use the following repulsive potential function [10] between any two given robots in the team:

$$f_{ij}(r_{ij}) = \left\{\begin{array}{ll}\frac{1}{2}\eta\left(\frac{1}{r_{ij}} - \frac{1}{r_0}\right)^2, & r_{ij} \leq r_0\\ 0, & r_{ij} > r_0.\end{array}\right. \quad (14)$$

This function has been widely used in obstacle avoidance contexts. It creates a repulsive force when the distance (r_{ij}) between two agents i and j is below a certain limit distance (r_0). In [14], [15], split-rejoin behaviors of the team of robots are desired, and therefore the potential functions employed are different from ours. Collision avoidance needs range information; however, due to the particular geometry of our framework, distances in the image plane are equivalent to distances between the robots, up to a constant scale factor. This allows us to implement the collision avoidance (i.e. compute (14)) using only visual information. The total repulsive potential acting on each robot is equal to the sum of the potentials due to all of the other robots in the group, i.e. $f_i = \sum_{j\neq i} f_{ij}(r_{ij})$.

We perform the collision avoidance as an additive angular velocity term. The linear velocity remains unchanged. The added steering term for every robot i is proportional to the negative gradient of the potential function projected on the direction orthogonal to the orientation of the robot $(v_i^{\perp})$:

$$\omega_i^{ca} = -k_{ca} < v_i^{\perp}, \nabla_{r_i} f_i >, \qquad k_{ca} > 0. \tag{15}$$

The total angular velocity of the control method is:

$$\omega^{total} = \omega + \omega^{ca}, \tag{16}$$

where ω is given by (13).

IV. Experiments

In this section, we present several simulations in order to illustrate the performance of the control scheme. It is assumed that the projection of each robot in the images can be detected and identified in order to match it with its correspondence in the other images.

The first experiment considers a configuration consisting of four robots in a square formation, with the flying camera undergoing a circular motion. The results from it are displayed in Fig. 5. We show the performance of the homography-based control with and without collision avoidance. In both cases the multi-robot system reaches the desired configuration (i.e. the relative positions between the robots in the formation). It can be seen that the collision avoidance mechanism performs well and generates more suitable paths from a practical point of view. As explained in section III-B, the control is performed in one single step and the final orientation of the robots is not corrected. The evolution of the homography entries is also displayed in the figure. As can be seen, all the individual homographies computed between each pair of robots converge to the desired homography. Notice that the desired homography is not constant, as it evolves depending on the motion of the camera.

In the second experiment, the desired configuration is a triangle formed by six robots, while the flying camera follows a spiral-like motion. Figure 6 displays the results from this experiment. Again, the target multi-robot configuration is correctly achieved, independently of the motion of the camera or the absolute position of the robot team. The minimum distance between robots is shown in order to illustrate the collision avoidance performance. Finally, we also carried out an experiment with a larger number of robots (twenty). It can be seen in Fig. 6 that in this case, the minimum distance between robots is also maintained above a certain threshold, avoiding drops in its value (possible collisions).

V. Conclusion

A new visual control scheme has been proposed to lead a group of robots to a desired configuration. The control law is based on a particular homography parametrization that allows to define the desired location of the robots in the image plane. The advantages of this approach are the simplicity of the definition of any arbitrary desired configuration for the set of robots, avoiding the need of metric information in the 3D space, and the fact that the planar motion of the flying camera can be unknown and arbitrary (thus allowing it to perform additional tasks simultaneously) without affecting the control performance. A vision-based collision avoidance method decoupled from the control task is also proposed. Simulations are presented in order to support the validity of the approach and illustrate its performance.

References

[1] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 813–825, 2002.

[2] R. Vidal, O. Shakernia, and S. Sastry, "Following the flock: Distributed formation control with omnidirectional vision-based motion segmentation and visual servoing," *Robotics and Automation Magazine*, vol. 11, no. 4, pp. 14–20, 2004.

[3] J. Chen, D. Sun, J. Yang, and H. Chen, "Leader-Follower Formation Control of Multiple Non-holonomic Mobile Robots Incorporating a Receding-horizon Scheme," *The International Journal of Robotics Research*, vol. 29, no. 6, pp. 727–747, 2010.

[4] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[5] H. Chen, D. Sun, and J. Yang, "Global localization of multirobot formations using ceiling vision SLAM strategy," *Mechatronics*, vol. 19, no. 5, pp. 617 – 628, 2009.

[6] G. López-Nicolás, J. J. Guerrero, and C. Sagüés, "Visual control of vehicles using two-view geometry," *Mechatronics*, vol. 20, no. 2, pp. 315 – 325, 2010.

[7] G. Blanc, Y. Mezouar, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," in *IEEE International Conference on Robotics and Automation*, April 2005, pp. 3365–3370.

[8] J. Courbon, Y. Mezouar, and P. Martinet, "Indoor navigation of a non-holonomic mobile robot using a visual memory," *Autonomous Robots*, vol. 25, no. 3, pp. 253–266, 2008.

[9] G. López-Nicolás, N. R. Gans, S. Bhattacharya, J. J. Guerrero, C. Sagüés, and S. Hutchinson, "Homography-based control scheme for mobile robots with nonholonomic and field-of-view constraints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 4, pp. 1115–1127, 2010.

[10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. Journal of Robotics Research*, vol. 5, pp. 90–98, 1986.

[11] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *IEEE International Conference on Robotics and Automation*, 1991, pp. 1398–1404.

[12] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, pp. 207–222, 2002.

[13] W. H. Huang, B. R. Fajen, J. R. Fink, and W. H. Warren, "Visual navigation and obstacle avoidance using a steering potential function," *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 288 – 299, 2006.

[14] D. E. Chang, S. C. Shadden, J. E. Marsden, and R. Olfati-Saber, "Collision avoidance for multiple agent systems," *IEEE International Conference on Decision and Control*, pp. 539–543, 2003.

[15] N. Moshtagh, N. Michael, A. Jadbabaie, and K. Daniilidis, "Vision-based, distributed control laws for motion coordination of nonholonomic robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 851–860, 2009.

[16] G. López-Nicolás, Y. Mezouar, and C. Sagüés, "Homography-based multi-robot control with a flying camera," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 4492–4497.

[17] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[18] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential fields," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

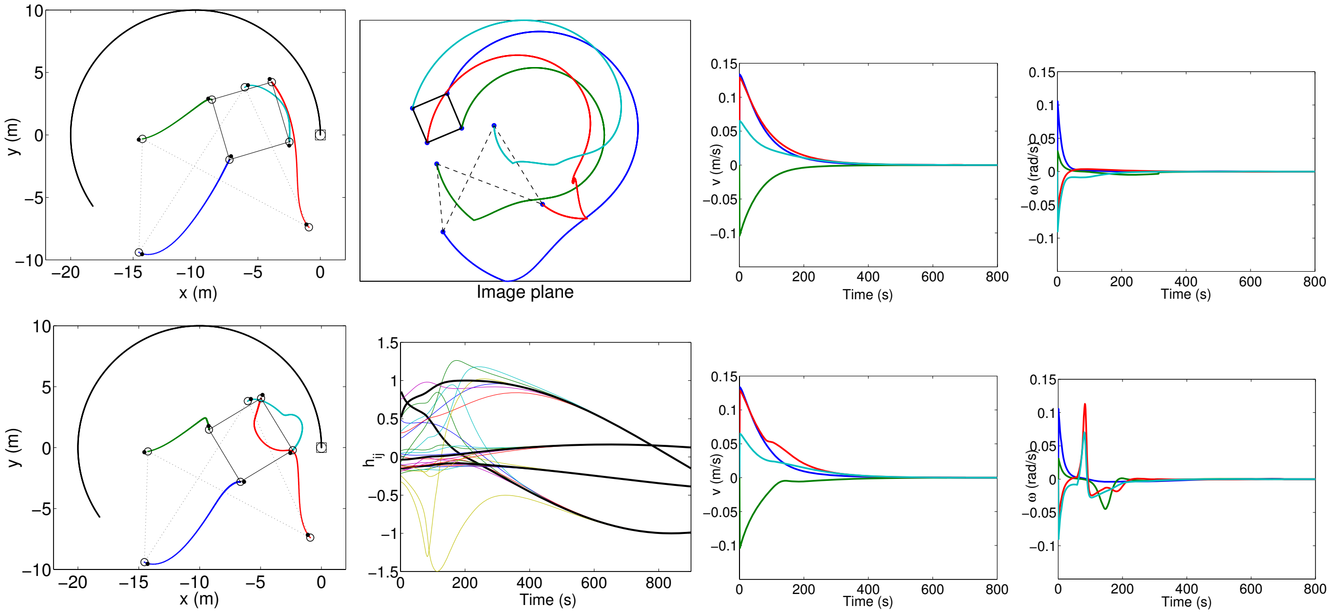


Fig. 5. Simulation with the flying camera undergoing a circular motion. The robots are initially in an arbitrary configuration and the goal is to reach the desired one, which is defined by the image of a square with vertices of $x-y$ coordinates: (-10,-8),(-10,-3),(-5,-8),(-5,-3). Top-left: top view of the camera (the initial position is depicted with a circle inside a square) and the robots, which are depicted as circles with a spot signalling their orientation. The initial configuration is drawn with a dashed line and the path followed by the robots to reach the desired configuration is shown (thick lines). This plot corresponds to the case where collision avoidance is not used. Top row-second column: trace of the robots in the image plane. Top row-third and fourth columns: linear and angular velocities of the robots, without collision avoidance. Bottom-left: robot paths with collision avoidance. Bottom row-second column: evolution of the homography entries (h_{11}, h_{12}, h_{13}, h_{23}) of the desired homography (thick lines) and the current homographies between the robots (thin lines). Bottom row-third and fourth columns: linear and angular velocities of the robots, with collision avoidance. The image plane and homography elements plots are from the case when collision avoidance is used.

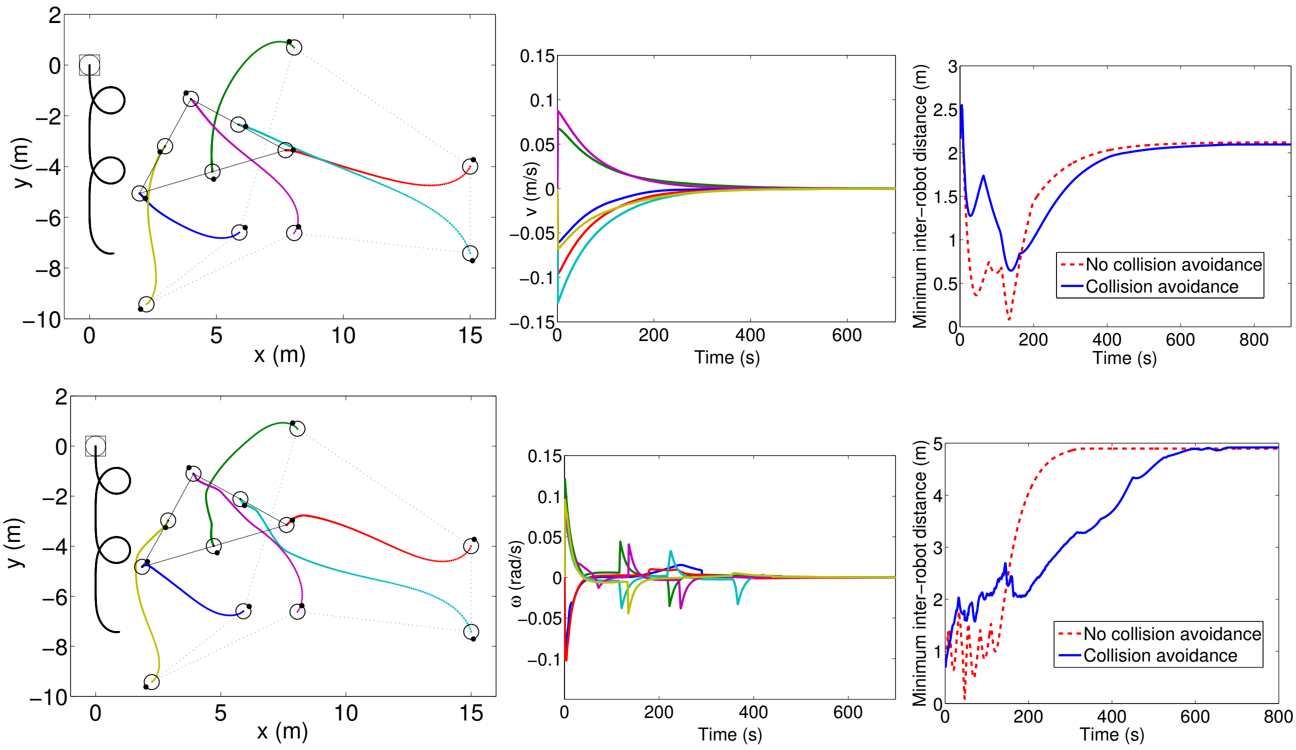


Fig. 6. Simulation results with the flying camera undergoing a spiral-like motion. The robots are initially in an arbitrary configuration and the goal is to reach the desired one, which is a triangle formed by six robots. Top-left: camera motion and robot paths, without collision avoidance. Bottom-left: camera motion and robot paths, with collision avoidance. Top row-center: linear velocities. Bottom row-center: angular velocities. The displayed velocities are for the case in which collision avoidance is used. Top-right: minimum inter-robot distance, with and without collision avoidance, for the six-robot triangular configuration. Bottom-right: minimum inter-robot distance, with and without collision avoidance, for a twenty-robot square configuration with circular camera motion.

IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)
September 26, 2011, San Francisco, California, USA

Visual Road Recognition Using Artificial Neural Networks and Stereo Vision

Patrick Y. Shinzato and Fernando S. Osorio and Denis F. Wolf
Mobile Robotic Laboratory,
Institute of Mathematics and Computer Science,
University of Sao Paulo - ICMC-USP,
Sao Carlos, SP Brazil
Email: shinzato@icmc.usp.br, fosorio@icmc.usp.br, denis@icmc.usp.br

Abstract—**Road recognition using visual information is an important capability for autonomous navigation in urban environments. Over the last three decades, a large number of visual road recognition approaches have been appeared in the literature. This paper propose a novel stereo vision based on artificial neural network that can identify the road using color, texture and disparity information from images. Several features are used as inputs of the ANN such as: average, entropy, energy and variance from different color channels (RGB, HSV, YCrCb). As a result, our system is able to estimate the road location and the confidence factor for each part of detected environment. Furthermore, our system presents a good generalization capacity. Experimental tests have been performed in several situations in order to validate the proposed approach.**

I. Introduction

Visual road recognition, also known as "lane detection", "road detection" and "road following", is one of the desirable skills to improve autonomous vehicles systems. As a result, visual road recognition systems have been developed by many research groups since the early 1980s, such as [1] [2] [3]. Details about these and others works can be found in several surveys [4] [5] [6] [7].

Most work developed before the last decade was based on certain assumptions about specific features of the road, such as lane markings [8] [9], geometric models [10] and road boundaries [11]. These systems have limitations and in most cases they showed satisfactory results only in autonomous driving on paved, structured and well-maintained roads. Furthermore they required favorable conditions of weather and traffic. Autonomous driving on unpaved or unstructured roads, and adverse conditions have also been well-studied in the last decade [12] [13] [14] [15]. We can highlight developed systems for the DARPA Grand Challenge [16] like [17] [18] [19] focusing on desert roads.

One of the most representative works in this area is the NAVLAB project [3]. Systems known as SCARF [20], UNSCARF [10], YARF [21], ALVINN [22], MANIAC [23] and RALPH [24] were also developed by the same research group. Among these systems, the most relevant reference for this paper are ALVINN and MANIAC because they are also based on artificial neural networks (ANN) for road recognition.

The idea of ALVINN consists of monitoring a human driver in order to learn the steering of wheels while driving on roads on varying conditions. This system, after several upgrades, was able to travel on single-lane paved and unpaved roads and multi-lane lined and unlined roads at speeds of up to 55 mph. However, it is important to emphasize that this system was designed and tested to drive on well-maintained roads like highways under favorable traffic conditions. Beyond those limitations, the learning step takes a few minutes [25] and the authors mention that when is necessary a retraining then this is a shortcoming [24]. According to [23], the major problem of ALVINN is the lack of ability to learn features which would allow the system to drive on road types other than that on which it was trained.

In order to improve the autonomous control, MANIAC (Multiple ALVINN Networks In Autonomous Control) [23] has been developed. In this system, several ALVINN networks must be trained separately on their respective roads types that are expected to be encountered during driving. Then the MANIAC system must be trained using stored exemplars from the ALVINN training runs. If a new ALVINN network is added to the system, MANIAC must be retrained. Both systems trained properly, ALVINN and MANIAC, can handle nonhomogeneous roads in various lighting conditions. However, this approach only works on straight or slightly curved roads [12].

Other group that developed visual road recognition based on ANN was the Intelligent Systems Division of the National Institute of Standards and Technology [26] [27]. They developed a system that make use of a dynamically trained ANN to distinguish between areas of road and nonroad. This approach is capable of dealing with nonhomogeneous road appearance if the nonhomogeneity is accurately represented in the training data. In order to generate training data, three regions from image were labeled as road and three others regions as nonroad, i.e., the authors made assumptions about the location of the road in the image, which causes problems in certain traffic situations. Additionally, this system works with the RGB color channel that suffers a lot of influence in the presence of shadows and lighting changes in the environment. A later work [28] proposed dynamic location of regions labeled as road in order to avoid these problems. However, under shadows situations, the new system becomes less accurate than the previous one because the dynamic location does not incorporate the road with shadow information in the training database.

In this work, we present a visual road detection system that use ANN with stereo images that contains depth information. Beyond the depth information, our ANN received several different image features as input. Features like averages, entropy, energy and variance from differents color channels (RGB, HSV, YCrCb) from sub-images. Other detail about this classification is that it provides confidence factor for each sub-image classification that can be used by control algorithm. Unlike [26], our system does not need to be retrained all the time because the generalization capacity of our system is more powerful than theirs. Therefore, our system does not require make assumption about location of road.

II. System Description

The system's goal is to identify the road region on a image obtained by a stereo camera attached to a vehicle. To accomplish this task, our system calculates the disparity of pixels using images left and right from camera. Immediately after, our system transforms the color image into set of sub-images and generates image features for each one. These features and disparity are used by ANN in order to classify if a sub-image belongs to a road class or not. A control algorithm uses the results in order to control the vehicle autonomously. After executing an action, the system captures another pair of images from environment and returns to first stage. The Fig. 1 show how the system works.

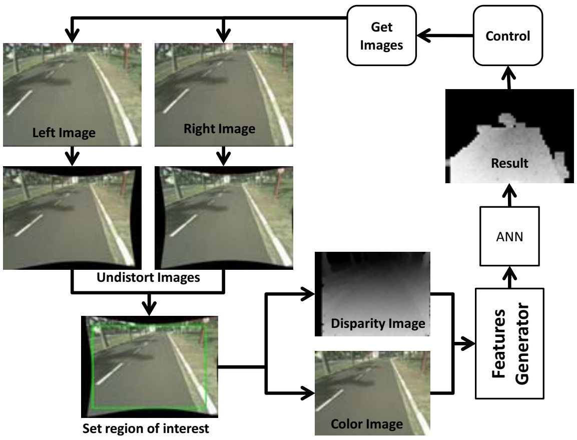


Fig. 1. (a) The System Architecture: Given a pair of images, the disparity is calculated. After that, disparity and color image are transformed into a set of sub-images that will be classified by ANN.

A. Stereo Vision

A stereo camera has two lenses to always capture a pair of images. These images have a shift between parts of the image proportional to the distance of the lens. Due to of this, it is possible to determine the depth of a point, estimating the difference of its position within the two images. This method is similar to the functioning of human vision.

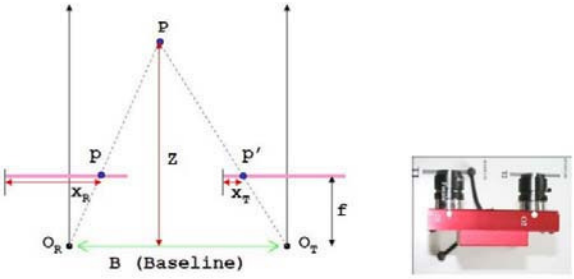


Fig. 2. Canonical system of a camera with two lenses. f is focal length, B is the distance between the lens.

Disparity of a point p is the distance on the X-axis with corresponding point p' in another image. Fig. 2 shows canonical system of a camera with two lenses. Match algorithms are used to calculate disparity and it has a high computational cost, therefore one should minimize the search space. In a canonical perfect system, such a search could be limited horizontally, ie, the search only happens in the neighbors of the same line. However, this does not happen in reality because of the camera lenses have distortions and are not perfectly aligned. Due to this, it is necessary calibrate camera in order to rectify the images and limit the search space. The method used to calibrate our camera and calculate the disparity is described in [29].

B. Generating Of Sub-images With Features

This stage transforms an color image into set a of sub-images and generates image-features for each them. More specifically, a image resolution ($M \times N$) pixels was decomposed in many sub-images with ($K \times K$) pixels, as show Fig. 3(a) which is transformed in Fig. 3(b). Mathematically, it can be defined as follows: suppose an image represented by a matrix I of size ($M \times N$). The element $I(m,n)$ corresponds to the pixel in row m and collumn n of image, where ($0 \leq m < M$) and ($0 \leq n < N$). Therefore, sub-image(i,j) is represented by group $G(i,j)$ that contains all the pixels $I(m,n)$ such that ($(i*K) \leq m < ((i*K)+K)$) and ($(j*K) \leq n < ((j*K)+K)$). For each group, many image features are generated. These features will be used as input ANN that determine whether the sub-image belongs to a road class or not. If the sub-image is classified as belonging to road class, then all pixels from group are considered as belonging to this class. Fig.3(c) shows sub-images belonging to road class painted red. This strategy has been used to reduce the amount of data, allowing faster processing and obtaining information like texture from sub-images.

Several statistical measures like mean, entropy and variance were used as image features. For each image, all measures were calculated with each color channel - we used RGB, HSV, YCrCb and normalized RGB. Also, the average of disparity of all pixels inside each sub-image was calculated. Thus, we generated a group of 49 features to be used as inputs by ANN. However, how this is a large number of inputs, we decided

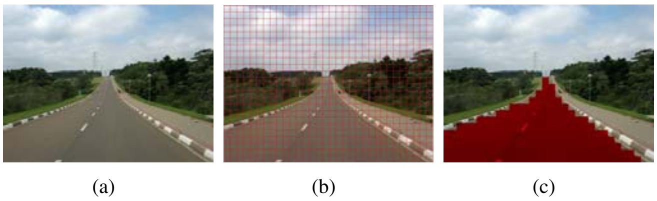


Fig. 3. In features generation stage, the image (a) is transformed into set of sub-images that represents each square from the image (b). After the classification, we can obtained results like (c), where all pixels from a square receive the same classification. Red squares were classified as belonging to road class.

to use a selection method “CFS”. This method selected the disparity-feature and all features shown in Table I. Finally, our system uses only 14 features to classify sub-images between road or non-road class.

TABLE I
FEATURES CALCULATED BY OUR SYSTEM. NOTE THAT RN, GN, BN ARE RGB CHANNELS NORMALIZED.

Measure	Channels from several color spaces											
	R	G	B	H	S	V	Y	Cr	Cb	RN	GN	BN
Mean	×	×	×	×	×	×	×	×	×	×		×
Entropy				×								
Variance												
Energy								×				

C. Artificial Neural Network

We used a multilayer perceptron (MLP) [30], which is a feedforward neural network model that maps sets of input data onto specific outputs. We use the back propagation technique [31], which estimates the weights based on the amount of error in the output compared to the expected results.

The ANN topology consists in, basically, two layers, where the hidden layer has seven neurons and the output layer has two neurons, as shows the Fig. 4, one neuron for road class and other for non-road class. The input layer has 14 inputs, and they are all normalized.

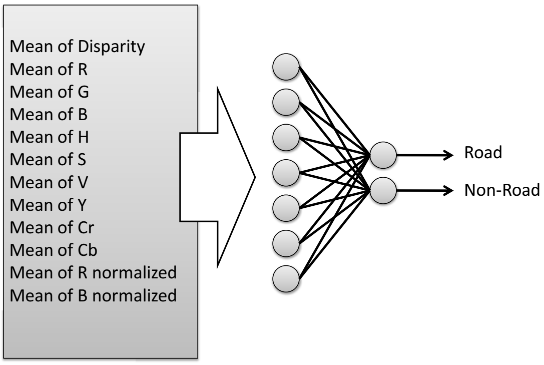


Fig. 4. ANN topology: The ANN uses some features, not all, to classify the sub-image between belonging to a road class or not.

Regarding ANN convergence evaluation, two metrics are frequently used: “MSE” and “Hit Rate”. The MSE is “Mean-Square Error” and usually the training step stops when the “MSE” converges to zero or some acceptable value. However, a small mean-square error does not necessarily imply good generalization [32]. Also this metric does not provide how many patterns are missclassified, i.e., if the error is higher in some patterns or if the error is evenly spread in all patterns. Other way of evaluating the convergence is checking how many patterns were classified correctly, or “Hit Rate”. In this case, the problem is to define a good precision to interpret the ANN output (i.e. given a ANN output, determine whether the output is equal to expected output or not), since the output may not be exactly the value expected. Seeking for a more adequate assessment to the proposed problem, we used a method known as AUC (area under an ROC curve).

According Fawcett [33], a receiver operating characteristics (ROC) graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC graps are two-dimensional graphs in which true positive rate is plotted on the Y-axis and false positive rate is plotted on the X-axis, as show Fig. 5. Each point in “ROC curve” is produced by different thresholds. To evaluate a classifier, the area under the ROC curve is calculated. This value will always be between 0 and 1.0. Is important to note that AUC values close to or below 0.5 indicate classifiers with poor performance. The closer to 1.0 the better the performance of the classifier.

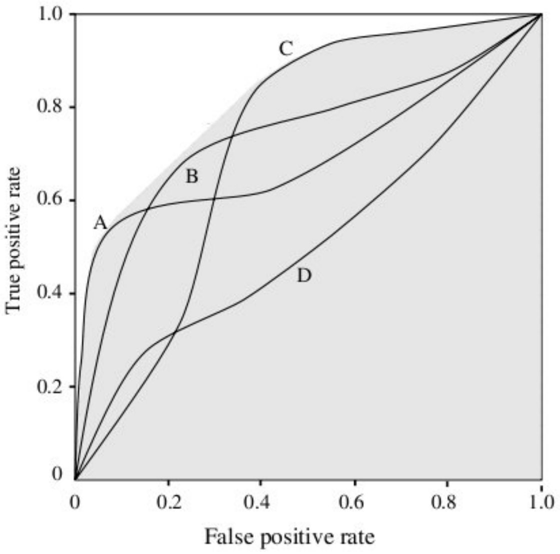
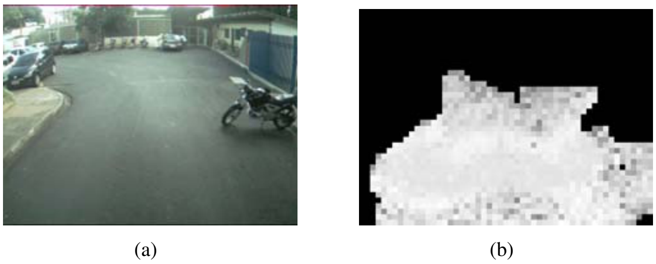


Fig. 5. ROC Curve sample for 4 classifiers. Image adapted from [33]

D. Visual Navigation Map (VNMap)

After classify all sub-images from an image with ANN, our system generate a VNMap filtering the resulting image with a growth algorithm. Thanks to it, the sub-images belonging to non-road class are painted pure black. Fig. 6 shows a sample of an image classified, where the Fig. 6(b) shows the VNMap in gray-scale - black represents non-road class, white represents road class and the gray represents the intermediate values. Is important to note that when the network does not achieve a good classification then the filter can consider the whole region as not navigable.


(a)

(b)

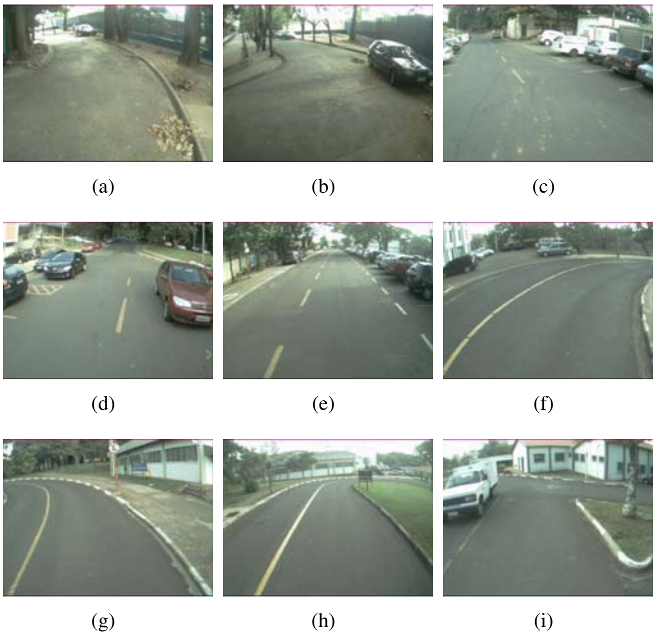
Fig. 6. (a) is a color image. (b) is results from a classification sample from our system. Black represents non-road class, white represents road class and the gray represents the intermediate values.

III. Experiments and Results

In order to validate the proposed system, several experiments have been perfomed. Our setup for the experiments was a eletric car equipped with (Videre STOC Color 15cm) camera. The image resolution was (640×480) pixels. The car and camera were used only for data collection. In order to execute the experiments with ANNs, we used the OpenCV library [29] that has also been used in the image acquisition and to visualize the processed results from system. The sub-image size used was $K = 10$, so each image has 3072 sub-images. We use the *semi-global block matching* method to calculate the disparity of pixels. The ANN has been trained until reaching 500 cycles.

Several paths traversed by the vehicle have been recorded using stereo camera. These paths are composed by road, sidewalks, parking, buildings, and vegetation. Also, some stretchs presents adverse conditions such as dirt (Fig. 7). We selected 128 pairs of images to compose our database. We randomly selected $\frac{3}{4}$ of images to use in training step and the last $\frac{1}{4}$ for evaluation step. The method 8-fold cross validation [32] was used in training step. The Table II shows results on evaluation set, ie, the AUC of each instance of ANN. Each line represents some instance of ANN trained with $\frac{7}{8}$ of training set and tested with last $\frac{1}{8}$ represented by each column. The last line represents the average of all instances evaluated. In general, our system achive AUC of 0.96 on evaluation set, which is a sactisfatory result.

The Fig. 8 shows some samples of classifications. It is possible to see that our system (ANN + filter) has achieved a high degree of certainty about the sub-images belonging to road class in all cases. In general, our system has been able to distinguish the road from the sidewalk and other items as cars and trees of the scenes evaluated. The small errors obtained are related to traffic lanes that have very different colors of asphalt and were manually interpreted as belonging to road class in this experiment. In addition, the system accumulates error with the loss of accuracy at the edges, since the method of generating attributes averaging a portions of the image.


(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

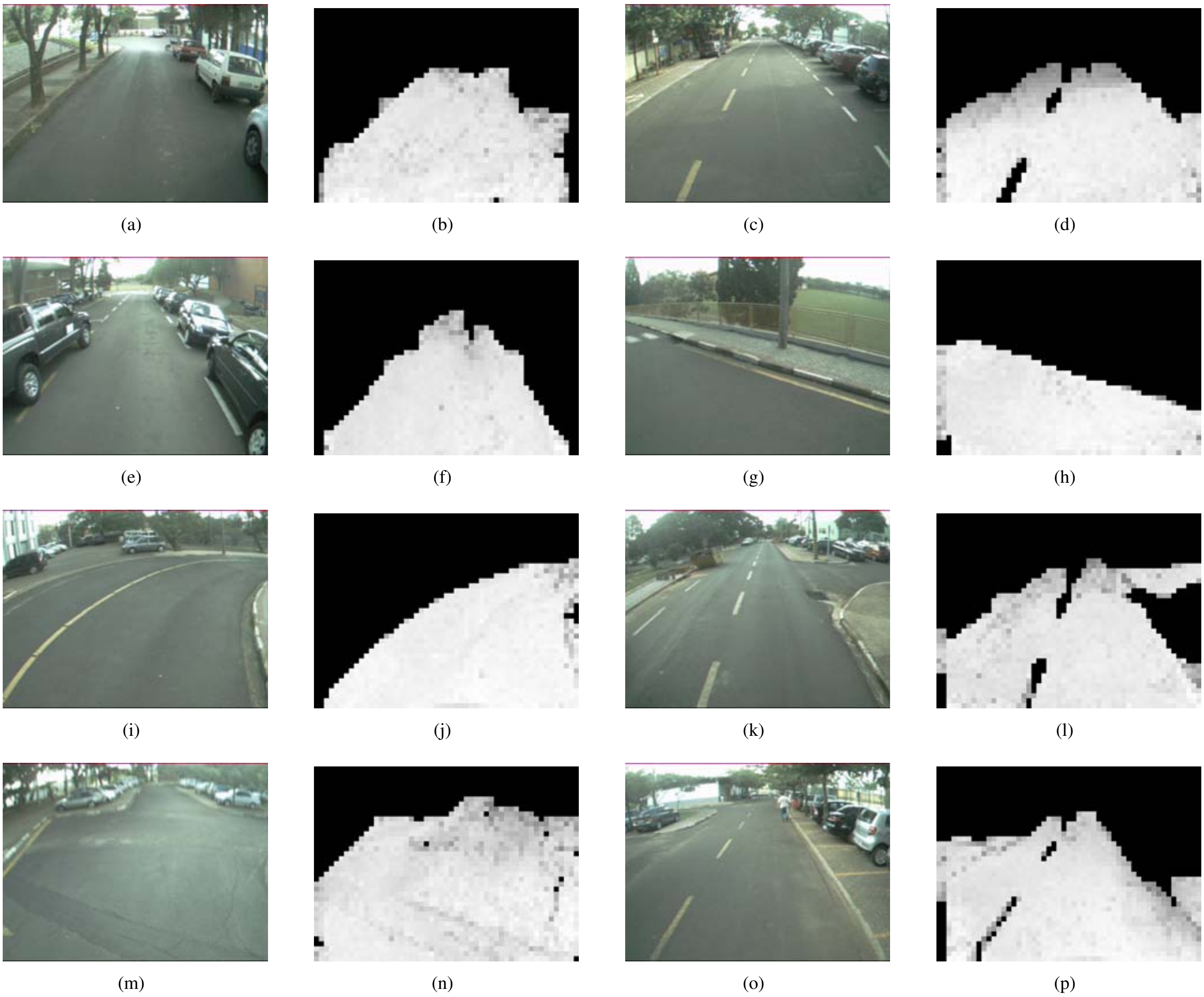
Fig. 7. Samples of scenarios used in this work.

TABLE II
Table of some AUC obtained on the set of evaluation

some instances	T1	T2	T3	T4	T5	T6	T7	T8
1	0.98	0.89	0.96	0.97	0.95	0.98	0.97	0.96
2	0.98	0.89	0.96	0.97	0.96	0.98	0.98	0.96
3	0.98	0.89	0.96	0.97	0.95	0.98	0.98	0.96
4	0.98	0.89	0.97	0.97	0.96	0.98	0.98	0.96
5	0.98	0.89	0.96	0.97	0.96	0.98	0.98	0.97
6	0.98	0.89	0.97	0.98	0.96	0.98	0.98	0.96
7	0.98	0.88	0.96	0.97	0.95	0.98	0.98	0.97
8	0.98	0.89	0.97	0.97	0.95	0.98	0.98	0.97
9	0.98	0.89	0.96	0.97	0.96	0.98	0.98	0.96
10	0.98	0.89	0.97	0.97	0.96	0.98	0.98	0.96
average	0.98	0.89	0.96	0.97	0.96	0.98	0.98	0.96

An important observation is that the disparity helped eliminate the misclassification of near obstacles with similar colors to the road - see Fig.8(e)-(f). Also, we may be concluded that the farther away is the observed point, less significant is the value of disparity. Because the sub-images near the vanishing point of the straight roads were not correctly classified - see Fig.8(c)-(d) and Fig.8(k)-(l).

Based on the experiments, we concluded that results are satisfatory and can be used by some control algorithm. Fig.9 shows in green the sector chosen by the algorithm based on direction of goal using a GPS and the classification from scene obtained by stereo camera. The other sectors colored with blue are other options avaiable in accordance with road identifier classification. When the sector is not colored so the sector not achieved the minimum threshold from polar histogram, which means that the vehicle should not go in those directions. This implementation only shows the direction that our vehicle must follow, we do not implement the control of the vehicle yet.


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Fig. 8. Samples of results from our system.

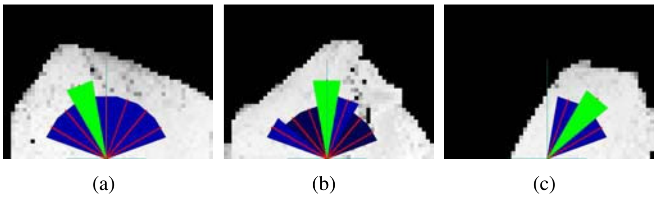

(a) (b) (c)

Fig. 9. Results from control algorithm, the image (a) is result when the goal is left of vehicle. (b) when goal is in the front of vehicle. And finally, (c) when the goal is the right of vehicle.

IV. Conclusion

Visual road recognition is one of the desirable skills to improve autonomous vehicles systems. We presented a visual road detection system that use ANN with depth information obtained by a stereo camera. Our ANN is capable to learn colors, textures and disparity of any sub-image instead of totally road appearance. Also, our training evaluation method is a more adequate assessment to the proposed problem, since many results with many low degrees of certainty lead to low scores. Finally, the system classification provides confidence factor for each pixel-group classification of image that can be used by a control algorithm.

In general, the results obtained in the experiments were relevant, since the system reached good classification results when the training step obtains good score. As future work, we plan to integrate it with other visual systems like *lane detection* in order to improve the system in urban scenarios. We intend integrate our road detection system with some control algorithm like a adaptation of VFH and control the vehicle. We also plan to integrate our approach with laser mapping in order to make conditions to retrain the ANN without human intervention and without making assumptions about the image. Finally, as the system classifies each block independently, we intend to improve the processing efficience using a GPU.

Acknowledgment

The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), processes 573963/2008-9 and 08/57870-9.

References

[1] A. Broggi, M. Bertozzi, and A. Fascioli, "Argo and the millemiglia in automatico tour," *Intelligent Systems and their Applications, IEEE*, vol. 14, no. 1, pp. 55 –64, January 1999.

[2] R. Gregor, M. Lutzeler, M. Pellkofer, K. Siedersberger, and E. Dickmanns, "Ems-vision: a perceptual system for autonomous vehicles," in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000, pp. 52 –57.

[3] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and navigation for the carnegie-mellon navlab," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 3, pp. 362 –373, May 1988.

[4] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent & Robotic Systems*, vol. 53, pp. 263–296, 2008, 10.1007/s10846-008-9235-4.

[5] G. Desouza and A. Kak, "Vision for mobile robot navigation: a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 237 –267, Feb. 2002.

[6] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 1–16, 2000.

[7] E. D. Dickmanns, "Vehicles capable of dynamic vision: a new breed of technical beings?" *Artif. Intell.*, vol. 103, pp. 49–76, August 1998.

[8] A. Takahashi, Y. Ninomiya, M. Ohta, and K. Tange, "A robust lane detection using real-time voting processor," in *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*, 1999, pp. 577 –580.

[9] M. Bertozzi and A. Broggi, "Gold: a parallel real-time stereo vision system for generic obstacle and lane detection," *Image Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 62 –81, Jan. 1998.

[10] J. Crisman and C. Thorpe, "Unscarf, a color vision system for the detection of unstructured roads," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, April 1991, pp. 2496 – 2501.

[11] A. Broggi and S. Bert, "Vision-based road detection in automotive systems: A real-time expectation-driven approach," *Journal of Artificial Intelligence Research*, vol. 3, pp. 325–348, 1995.

[12] C. Tan, T. Hong, T. Chang, and M. Shneier, "Color model-based real-time learning for road following," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, September 2006, pp. 939 –944.

[13] F. Diego, J. Alvarez, J. Serrat, and A. Lopez, "Vision-based road detection via on-line video registration," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, September 2010, pp. 1135 –1140.

[14] J. Alvarez and A. Lopez, "Novel index for objective evaluation of road detection algorithms," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, October 2008, pp. 815 –820.

[15] M. Manz, F. von Hundelshausen, and H.-J. Wuensche, "A hybrid estimation approach for autonomous dirt road following using multiple clothoid segments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 2410 –2415.

[16] DARPA05, http://www.darpa.mil/grandchallenge05/, 2005, visited in February/2011.

[17] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised monocular road detection in desert terrain," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[18] D. Lieb, A. Lookingbill, and S. Thrun, "Adaptive road following using self-supervised learning and reverse optical flow," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[19] J. Lee, C. Crane, S. Kim, and J. Kim, "Road following in an unstructured desert environment using monocular color vision as applied to the darpa grand challenge," in *International Conference on Control, Automation, and Systems (ICCAS)*, 2005.

[20] J. Crisman and C. Thorpe, "Scarf: A color vision system that tracks roads and intersections," *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp. 49 – 58, February 1993.

[21] K. Kluge and C. Thorpe, "The yarf system for vision-based road following," *Mathematical and Computer Modelling*, vol. 22, no. 4-7, pp. 213 – 233, 1995.

[22] D. A. Pomerleau, "Neural network vision for robot driving," in *The Handbook of Brain Theory and Neural Networks*. University Press, 1996, pp. 161–181.

[23] T. Jochem, D. Pomerleau, and C. Thorpe, "Maniac: A next generation neurally based autonomous road follower," in *Proceedings of the International Conference on Intelligent Autonomous Systems*, February 1993.

[24] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert*, vol. 11, no. 2, pp. 19 –27, Apr. 1996.

[25] D. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991.

[26] M. Foedisch and A. Takeuchi, "Adaptive real-time road detection using neural networks," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, October 2004, pp. 167 – 172.

[27] P. Conrad and M. Foedisch, "Performance evaluation of color based road detection using neural nets and support vector machines," in *Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. 32nd*, 2003, pp. 157 – 160.

[28] M. Foedisch and A. Takeuchi, "Adaptive road detection through continuous environment learning," in *Applied Imagery Pattern Recognition Workshop, 2004. Proceedings. 33rd*, October 2004, pp. 16 – 21.

[29] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O'Reilly, 2008.

[30] P. S. Churchland and T. J. Sejnowski, *The Computational Brain*. Cambridge, MA, USA: MIT Press, 1994.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," pp. 673–695, 1988.

[32] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*, 2nd ed. Prentice Hall, July 1998.

[33] T. Fawcett, "An introduction to roc analysis," *Pattern Recogn. Lett.*, vol. 27, pp. 861–874, June 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1159473.1159475

Vision based control for Humanoid robots

C. Dune, A. Herdt, E. Marchand, O. Stasse, P.-B. Wieber, E. Yoshida

***Abstract*— This paper presents a visual servoing scheme to control humanoid dynamic walk. Whereas most of the existing approaches follow a perception-decision-action scheme, we hereby introduce a method that uses the on-line information given by an on-board camera. This close looped approach allows the system to react to changes in its environment and adapt to modelling error. Our approach is based on a new reactive pattern generator which modifies footsteps, center of mass and center of pressure trajectories at the control level for the center of mass to track a reference velocity. In this workshop, we present three ways of servoing dynamical humanoïd walk : a naïve one that compute a reference velocity using a visual servoing control law, a second one that takes into account the sway motion induced by the walk and an on going work on vision predictive control that directly introduces the visual error in the cost function of the pattern generator. The two first approaches have been validated on the HRP-2 robot. These close loop approaches give a more accurate positioning than the one obtained when executing a planned trajectory especially when rotational motion are involved.**

I. INTRODUCTION

Humanoid robots are designed for human environments, defined as unstructured and dynamic environments [1] where objects move outside robots'control. In order to complete a specific task, humanoid robots must perceive and react to environmental changes. Vision based control may help them to perceive their surroundings in order to adapt their behaviour efficiently. Indeed, most of the humanoid robots are equipped with cameras that provide rich information without adding so much weight and size. The use of embedded camera is attractive because it avoids equipping the environment with additional sensors, and thus the system is more autonomous. Yet, extracting data from these cameras is a real challenge, especially while walking.

In this paper, we introduce a monocular visual servoing scheme to control the HRP-2 walk towards an object with taking into account the peculiar motion of the on-board camera induced by the stepping.

A. State of the art

Previous works on humanoid walking control assume that the robot path is defined before computing the actual joint control to realize it. They generally follow a perception-decision-action scheme: first, a sensor acquires data on the world and/or the robot state, then, suitable footsteps over a time horizon are decided, and the trajectories of the center of mass (CoM) and the center of pressure (CoP) are computed while respecting the stability constraints. Finally, the control of the legs is computed by inverse kinematics. This perception-decision-action loop has proven to be fast enough to realize impressive demonstrations for stair-climbing and obstacle avoidance [2], [3], [4], [5].

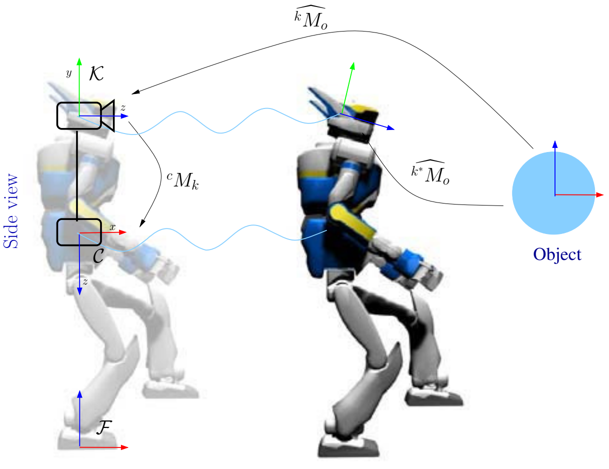


Fig. 1. The robot has to reach a desired position with regards to an object

We claim that a visual servoing control scheme is well suited for vision based walking motion generation because it compensates for model errors. Visual servoing proved to be successful for grasping tasks with standing [6], [7] or walking humanoids [8], [9]. In [8], visual servoing is used to control a humanoid avatar along landmarks. The upper body is approximated by the kinematic chain that links an on-board camera to the CoM. The lower body is controlled by adding two translational degrees of freedom to the CoM. The translational velocity of the CoM is sent to a kinematic locomotion module which control the legs motion. In [9] a whole body visual servoing scheme based on a hierarchical stack of task is introduced. However, the footsteps are predefined. The leg motion is thus set to be the task of higher priority. Therefore visual-servoing in this context is projected in the null-space of the pre-defined walking path. On the contrary in this work, the controller driving the walk is directly guided by vision.

Few work deal with footsteps, CoM and CoP trajectories modification inside the preview window. The work presented

C. Dune was with CNRS-AIST, JRL (Joint Robotics Laboratory), UMI 3218/CRT, Intelligent Systems Research Institute, AIST Central 2, Umezono 1-1-1, Tsukuba, Ibaraki 305-8568 Japan and is now with Handibio, EA4322 Université du Sud- Toulon Var.

O. Stasse and E. Yoshida are with CNRS-AIST, JRL (Joint Robotics Laboratory), UMI 3218/CRT, Intelligent Systems Research Institute, AIST Central 2, Umezono 1-1-1, Tsukuba, Ibaraki 305-8568 Japan

A. Herdt and P.-B. Wieber are with Bipop team Inria Rhône Alpes, Grenoble, France

E. Marchand is with Lagadic team Inria Rennes Bretagne Atlantique, Rennes, France

in [10] shades some light on this problem. It shows that modifying the next landing position of the flying foot might impose a new CoP trajectory going out of the support polygon. This can jeopardize the equilibrium of the robot. To solve the problem, the stepping period may be modified to reduce this instability [10], at the cost of slowing down the robot. A recent method proposes to modify the footsteps according to a perturbation applied to the CoP [11]. In the current paper the desired velocity computed by a visual servoing based controller is directly used to change footsteps, while ensuring walking stability constraints and with time intervals of constant length. Another difference lays on the fact that, on the one hand, the CoP is constrained at the center of the footprints, and on the other hand the CoP can move freely inside the support polygon.

B. *Contribution*

Our approach is based on a new pattern generator (PG) that has been proposed by Herdt et al [12], [13]. It computes a reactive stable walking motion for the CoM to track an instant reference velocity without predefined footsteps. This paves the way to reactive walking motion based on current environmental perception.

In this paper, we introduce a real time vision based control of HRP-2 walking motion. It is based on the visual servoing scheme we introduced in [14] applied to a positioning task. This scheme has the advantage of handling both fixed and mobile object. The only requirement is to know at least partially the 3d model of the object : some 3d edges for angular objects or the diameter of a sphere for a ball.

C. *Paper overview*

Section II is dedicated to the new PG description. Section III presents the model based tracker and the visual servoing control and Section IV presents the results obtained using our approach with regards to the execution of a planned trajectory. Section V draws the conclusion and perspectives.

II. Prediction Control Scheme for Reactive Walking Motion

This section presents the on-line walking motion generator introduced in [12], [13]. The robot is modelled as a linear inverse pendulum which fits fairly well with the HRP-2 distribution of mass. The control is based on a Linear Model Predictive Control scheme that computes the footsteps and the optimal jerk of the point mass model to minimise the difference between a reference CoM velocity and the previewed one.

A. *Systems Dynamics*

The humanoid robot is modelled as an oriented mass point centred on the robot CoM. This paragraph describes the dynamics of a stable walking motion.

1) Motion of the Center of Mass: Let us consider a frame $\mathcal{C}$ attached to the position of the CoM of the robot and to the orientation of its trunk. The position and orientation of this frame will be noted $c = \begin{bmatrix} c^x & c^y & c^z & c^\varphi & c^\psi & c^\theta \end{bmatrix}$, with Cardan angles c^φ, c^ψ and c^θ.

The acceleration $\ddot{c}$ of this frame has to be continuous for being realized properly by usual actuators. We will consider here that it is in fact piecewise linear on time intervals of constant length τ, with a piecewise constant jerk $\dddot{c}$ (third derivative of the position) on these intervals. The trajectory of this frame over longer time intervals of length $n\tau$ can be simply obtained by integrating over time the piecewise constant jerk together with the initial speed $\dot{c}$ and acceleration $\ddot{c}$. For any coordinate $\alpha \in \{x, y, z, \varphi, \psi, \theta\}$, this leads to simple linear relationships

$$C^\alpha_{i+1} = S_p c^{\hat{\alpha}_i} + U_p \dddot{C}^\alpha_i, \tag{1}$$

$$\dot{C}^\alpha_{i+1} = S_v \hat{c}^\alpha_i + U_v \dddot{C}^\alpha_i, \tag{2}$$

$$\ddot{C}^\alpha_{i+1} = S_a \hat{c}^\alpha_i + U_a \dddot{C}^\alpha_i, \tag{3}$$

where the initial state is $\hat{c}^\alpha_i = \begin{bmatrix} c^\alpha(t_i) & \dot{c}^\alpha(t_i) & \ddot{c}^\alpha(t_i) \end{bmatrix}^T$, and C^α_{i+i} is the vector of the state on the prediction horizon that can is defined by

$$C^\alpha_{i+1} = \begin{bmatrix} c^\alpha(t_{i+1}) \\ \vdots \\ c^\alpha(t_{i+n}) \end{bmatrix}, \dots \; \dddot{C}^\alpha_{i+1} = \begin{bmatrix} \dddot{c}^\alpha(t_{i+1}) \\ \vdots \\ \dddot{c}^\alpha(t_{i+n}) \end{bmatrix}$$

The matrices $U_\bullet, S_\bullet, Z_\bullet$ introduced here follow directly from recursive application of the dynamics (details on matrices can be found in [12]), let T be the sampling period, and N the length of the time horizon. The matrix related to the position prediction are :

$$S_p = \begin{pmatrix} 1 & T & \frac{T^2}{2} \\ \vdots & \vdots & \vdots \\ 1 & NT & N^2T^2 \end{pmatrix} U_p = \begin{pmatrix} \frac{T^3}{6} & 0 & 0 \\ \vdots & \ddots & 0 \\ (1{+}3N{+}3N^2)\frac{T^3}{6} & \dots & \frac{T^3}{6} \end{pmatrix}$$

the one related to the velocity prediction on time horizon are :

$$S_v = \begin{pmatrix} 0 & 1 & T \\ \vdots & \vdots & \vdots \\ 0 & 1 & NT \end{pmatrix} U_v = \begin{pmatrix} \frac{T^2}{2} & 0 & 0 \\ \vdots & \ddots & 0 \\ (1{+}2N)\frac{T^2}{2} & \dots & \frac{T^2}{2} \end{pmatrix}$$

and the one related to the acceleration prediction on time horizon are :

$$S_a = \begin{pmatrix} 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{pmatrix} U_a = \begin{pmatrix} T & 0 & 0 \\ \vdots & \ddots & 0 \\ (1+N)T & \dots & T \end{pmatrix}$$

2) Motion of the Center of Pressure: The position z of the Center of Pressure (CoP) on the ground can be approximated by considering only the inertial effects that are due to the translation of the CoM, neglecting the other effects due to the rotations of the different parts of the robot. This proves to be a very effective approximation, which leads to the simple relationships:

$$z_i^x = c_i^x - (c_i^z - z_i^z)\ddot{c}_i^x/g, \text{ and } z_i^y = c_i^y - (c_i^z - z_i^z)\ddot{c}_i^y/g,$$

where the difference $c_i^z - z_i^z$ corresponds to the height of the CoM above the ground, and g is the norm of the gravity force. We will consider here only the simple case where the height of the CoM above the ground is constant. In that case, we can obtain a relationship similar to (1)-(3):

$$\begin{aligned} Z_{i+1}^x = S_z\hat{c}_i^x + U_z\dddot{C}_i^x \text{ and } Z_{i+1}^y = S_z\hat{c}_i^y + U_z\dddot{C}_i^y, \\ \text{with } S_z = S_p - (c_i^z - z_i^z)S_a/g, \\ U_z = U_p - (c_i^z - z_i^z)U_a/g. \end{aligned}$$

3) Foot step generation: Basically, humanoid nominal walking cycle can be divided into two stages: a double support phase, where the two feet are on the ground and a single support phase, where only one foot is firmly on the ground on the other one is flying from its previous position to the next one. In this paper the stepping period is set to be $800ms$ with a double support phase of $100ms$ and single support phase of $700ms$.

The new pattern generator selects on-line the feasible footsteps on the preview window with regards to the robot mechanical properties [15]. Let note F_{i+1} the vector of the footstep position on the time horizon. The position of the footsteps is then used twice: first to ensure the stability constraints on the CoP trajectory and secondly to be included in the cost function to attract the CoP trajectory towards the center of the polygon of support.

B. Constraints definition

To be stable, the dynamics control of the walking motion must comply with the following stability constraints.

1) Constraints on the CoP : since the feet of the robot can only push on the ground, the CoP can lie only within the support polygon, that is the convex hull of the contact points between the feet and the ground [16]. Any trajectory not satisfying this constraint cannot be realized properly. This needs to be taken into account when computing a walking motion with the MPC scheme (4). The foot on the ground is assumed to have a polygonal shape, so that this constraint can be expressed as a set of constraints on the position of the CoP which are linear with respect to the position of the foot on the ground but nonlinear with respect to its orientation.

2) Constraints on the foot placement: we need to assure that the footsteps decided by the above mentioned algorithm are feasible with respect to maximum leg length, joint limits, self-collision avoidance, maximum joint velocity and similar geometric and kinematic limitations. In order to keep the Linear MPC structure of the algorithm, simple approximations of all these limitations are expressed in the form of linear constraints defined in [15].

C. Following a reference velocity

This section sets the optimisation problem to solve to ensure that the CoM velocity tracks a reference velocity. In order to keep the constraints linear, the optimisation is split in two steps: first, translations are treated, then rotations along the vertical axis are considered. This control is used as the highest priority task in a general inverse kinematics framework to compute whole body motion.

1) Translational velocity: It has been proposed in [12] to generate walking motions by directly following a reference velocity $\dot{C}^*$. Only horizontal translations were considered. Secondary objectives were also introduced to help obtaining a more satisfying behaviour: centring the position of the feet with respect to the position of the CoP, and minimizing the jerk $\dddot{c}(t)$ to slightly smoothen the resulting trajectory.

$$\begin{aligned} \min \ & \frac{\alpha}{2}\left\|\dot{C}_{i+1}^x - \dot{C}_{i+1}^{x,*}\right\|^2 + \frac{\alpha}{2}\left\|\dot{C}_{i+1}^y - \dot{C}_{i+1}^{y,*}\right\|^2 \\ & + \frac{\beta}{2}\left\|\bar{C}_{i+1}^x - \dot{C}_{i+1}^{x,*}\right\|^2 + \frac{\beta}{2}\left\|\bar{C}_{i+1}^y - \dot{C}_{i+1}^{y,*}\right\|^2 \\ & + \frac{\gamma}{2}\left\|F_{i+1}^x - Z_{i+1}^x\right\|^2 + \frac{\gamma}{2}\left\|F_{i+1}^y - Z_{i+1}^y\right\|^2 \\ & + \frac{\varepsilon}{2}\left\|\dddot{C}_i^x\right\|^2 + \frac{\varepsilon}{2}\left\|\dddot{C}_i^y\right\|^2 \end{aligned} \tag{4}$$

where $\bar{C}$ is the mean speed of the CoM over two steps. Introducing the vector $u_i = \begin{bmatrix}\dddot{C}_i^x & F_{i+1}^x & \dddot{C}_i^y & F_{i+1}^y\end{bmatrix}$ of motion parameters which automatically computed, this optimization problem can be expressed as a canonical Quadratic Program with the aforementioned constraints [12].

2) Following a reference rotational velocity: If the robot trunk has to rotate, then the orientations of the feet have to be adapted properly. Yet, introducing θ as a variable in II-B.2 would result in non-linear constraints. In order to keep the linear form, Herdt et al [12], [13] chose to predetermine the orientation of the feet before solving the translational Quadratic Program.

To increase the robustness of trunk rotational motion, the feet orientations have to be aligned with the trunk orientation. Furthermore, feet and trunk acceleration and velocity have to be limited to avoid infeasible trajectories. This leads to the formulation of a decoupled Quadratic Program:

$$\min_{u_i^\theta} \quad \tfrac{\delta}{2}||C_{i+1}^\theta - F_{i+1}^\theta||^2 + \tfrac{\epsilon}{2}||\dot{C}_{i+1}^\theta - \dot{C}_{i+1}^{\theta,*}||^2 \tag{5}$$

$$s.t. \quad \dot{F}_{i+1}^{\theta,s} = 0 \tag{6}$$

$$||F_{i+1}^{\theta,r} - F_{i+1}^{\theta,l}|| < \theta_{max}^{rl} \tag{7}$$

$$||F_{i+1}^{\theta} - C_{i+1}^{\theta}|| < \theta_{max}^{FT} \tag{8}$$

$$||\dot{F}_{i+1}^{\theta} - \dot{C}_{i+1}^{\theta}|| < \dot{\theta}_{max}^{FT} \tag{9}$$

$$||\ddot{F}_{i+1}^{\theta} - \ddot{C}_{i+1}^{\theta}|| < \ddot{\theta}_{max}^{FT}, \tag{10}$$

The two terms of the above objective ensure that the trunk follows the desired rotational velocity and that at the same time the feet are aligned as much as possible with the trunk. The constraints assure the feasibility of the desired motions.

D. Over-all behaviour of pattern generator

In order to compute a proper control law for the walk, we have to understand the over-all behaviour of the pattern generator. The PG ensures that the CoM tracks a reference velocity yet on average and in the limit of the dimension of the robot (length of legs, actuator torque limit, etc.). We describe here these two aspects of the PG (see Fig. 2).

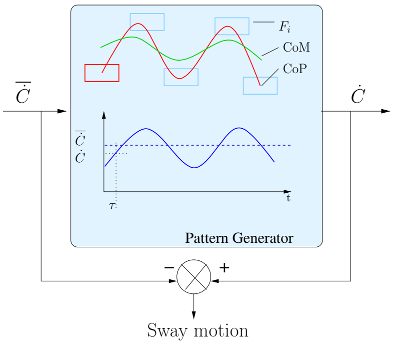


Fig. 2. The pattern generator ensures that the input velocity is tracked on average on the preview horizon. The output of the Model Predictive Control is the first control computed on the preview horizon. The difference between the reference velocity and the real velocity is mostly a sway motion due to the stepping.

1) Limiting the velocity: In order to ensure the tracking of the reference velocity, the three velocity components have to be limited to feasible ones, i.e. velocities that respect the walking constraints which mainly depends on the robot geometry and actuators capabilities. It can be shown that the maximum speed for the HRP-2 robot is : $\dot{c}_{limit} = \begin{pmatrix}0.2 & 0.2 & 0.2\end{pmatrix}$ for the considered PG [13].

2) Sway Motion: In most of the existing PG, the stepping motion induces a lateral sway motion that prevents the CoM velocity from following instantaneously the expected one. The sway motion is mandatory for a proper walk and the control law should not compensate for it but cancels its effects on the visual error computation.

Let us define $\dot{b}$ the additional sway of period $T = \tau_{step}/\tau$. Let assume that $\dot{b}$ is such that $\int_{l=i}^{i+T} \dot{b}(t)dt = 0$. Then the behaviour of the PG can be approximated by

$$\dot{c} = \dot{\bar{c}} + \dot{b} \tag{11}$$

where $\dot{\bar{c}}$ is the velocity of a virtual average CoM that corresponds to a displacement without the stepping. The camera velocity can then be written :

$$\dot{k} = \dot{\bar{k}} + {}^kV_c\dot{b} \tag{12}$$

where ${}^cV_k$ is the twist matrix related to the camera-center of mass transform ${}^cM_k$ (see Fig 1).

III. Vision Based Control

In this section, a position based visual servoing scheme is introduced to compute the velocity that is given as a reference to the reactive PG.

When the robot walks, its stepping makes its head shake and oscillate. Each time a foot hits the ground, the impact propagates to the robot's head and the camera jolts which causes blur and shift in the image. Moreover, the inherent sway motion disturbs the control law. It makes the use of on-board images challenging. We will first describe a model-based tracking [17] that is robust enough to track a known object in such a difficult image sequence. Then, we present our visual control law that modify online the current measurement to cancel the sway motion.

A. Model Based Tracking

The model-based tracking introduced in [17] provides a robust solution to the challenging issue of tracking an object while walking. It can be used to track geometrical shapes (lines, cylinders, ellipsoids, ...) as soon their perspective projection can be computed. It estimates on-line the position of a known object in the camera frame $\widehat{{}^kM_o}$. This tracking algorithm can be divided in two steps: i) 2D tracking where contour points are locally tracked and ii) pose estimation that is based on a non linear iterative algorithm.

Fig. 3 depicts the tracking principle. 1) Starting from an initial pose, the lines of the 3D object model are projected on the image and sampled (light blue lines and black points). Then the normal to the line are computed for every sampled points (yellow lines). Pixels are tested in the neighbourhood of the sampled points and along the normal to find the maximum gradient response (red points). 2) In a second step, a virtual visual servoing is used to find the object position by controlling a virtual camera so that the projection of the 3D model fits best with the tracked points (black lines). The current visual features are the projection of the 3D lines l_i according to the pose $\widehat{{}^kM_o}$ and the desired visual features are the tracked points p_i. The error is the distance between a point and a line (see bottom right frame Fig. 3).

Finally the optimisation problem can be written as:

$$\widehat{{}^kM_o} = \arg\min_{{}^kM_o} \sum_i \mathcal{C}(d_\perp(p_i, l_i({}^kM_o))) \tag{13}$$

where $\mathcal{C}$ is a robust function that allows to handle outliers. The distance $d_\perp$ is represented in Fig. 3.

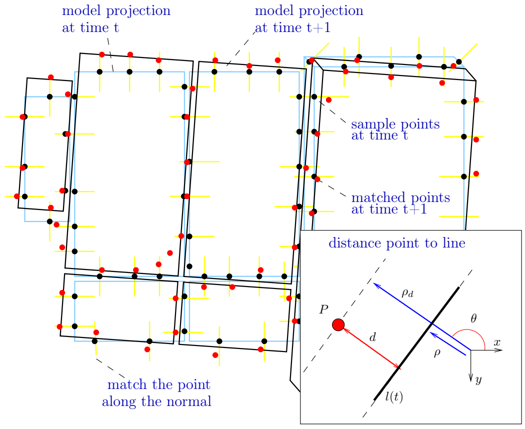


Fig. 3. Model-based tracking principle.

The robust part of the tracker can not be found in the features extraction itself but in the weighting of their contributions relatively to the confidence given in each measurement. Classically, the outliers are rejected using Hough or RANSAC methods. The considered tracker is based on a statistical methods, the M-Estimator [18]. Further details on the algorithm for robust tracking may be found in [17].

B. Visual Servoing

Classically, visual servoing aims to regulate an error vector $e = s - s^*$ between some current features observed in an image s and some desired visual features s^* [19].

Here the current and the desired features are the current pose of the object in the camera frame ${}^kM_o$ and the pose of the object in the desired camera frame ${}^{k^*}M_o$. The positioning task is regulated when ${}^{k^*}M_k = I$. The task error can be expressed as 6 dimensional pose vector $s = (t, \theta u)$: the first three coordinates are the three translations t and the last three coordinates are a rotation vector in a (θu) representation where θu defines the angle and axis of the rotation of the current camera with regards to the desired one.

The key feature in this control scheme is the *interaction matrix* L that links the time variation of the visual features $\dot{s}$ to the relative camera velocity $\dot{k}$. It is defined by:

$$\dot{s} = L\dot{k} \tag{14}$$

Then, the control law that regulates e with an exponential decrease $\dot{e} = -\lambda e$ is [19]:

$$\dot{k} = -\lambda \widehat{L}^+ e \tag{15}$$

where $\widehat{L}^+$ denotes the Moore-Penrose pseudo inverse of an approximation or a model of L.

For the chosen type of features, the interaction matrix is [19]

$$L = \begin{pmatrix} {}^{k^*}R_k & 0_{3\times3} \\ 0_{3\times3} & J_\omega \end{pmatrix} \tag{16}$$

where $J_\omega = L_\omega$ and L_ω is such that $L_\omega^{-1}\theta u = \theta u$.

Then we can write the CoM reference speed $\dot{\bar{c}}$:

$$\dot{\bar{c}} = -\lambda {}^cV_k \widehat{L}^+ e \tag{17}$$

C. Cancelling the sway motion

Due to the sway, the features oscillate in the image. Using (12) and (15) the feature variations can be written:

$$\dot{s} = \widehat{L}\dot{\bar{k}} + \widehat{L}^k V_c \dot{b} \tag{18}$$

Let us define a virtual camera (Fig. 4) $\overline{\mathcal{K}}$ that corresponds to the position of the on-board camera if there was no sway motion. The velocity of this virtual camera is $\dot{\bar{k}}$, it is actually the velocity that is input into the reactive PG. Its value is given in(23). In order to compute a control law that does not include the sway motion, we will servo this virtual camera $s(\bar{k})$ to $s(\bar{k}^*)$.

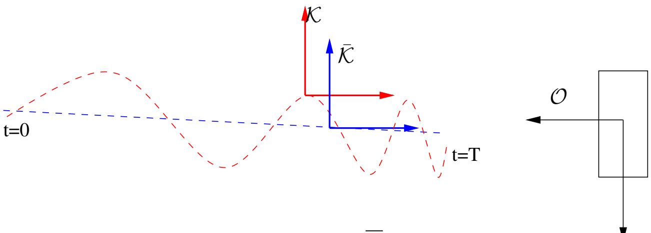


Fig. 4. $\mathcal{K}$ is the current camera frame and $\overline{\mathcal{K}}$ is the camera position obtained if the visual servoing velocity is applied without the walking constraints.

We have now to express $\bar{s} = s(\bar{k})$ with regards to the current measurement $s = s(k)$. With (14) we can write:

$$s(t) - s(0) = \int_0^t \widehat{L}\dot{k}dt = \int_0^t \widehat{L}(\dot{\bar{k}} + \dot{b}_k)dt \tag{19}$$

$$\text{and } \bar{s}(t) - \bar{s}(0) = \int_0^t \widehat{L}\dot{\bar{k}}dt \tag{20}$$

Then assuming that $s(0) - \bar{s}(0) = E$ and using (19) and (20) we obtain

$$s(t) = \bar{s}(t) + \int_0^t \widehat{L}^k V_c \dot{b}dt - E \tag{21}$$

from which we can deduce the corrected visual error

$$\bar{e}(t) = \bar{s}(t) - s^* = e(t) - (\int_0^t \widehat{L}^k V_c \dot{b}dt - E) \tag{22}$$

Notice that when $\bar{e} \longrightarrow 0$ then $e \longrightarrow \int_0^t L\dot{b}_k$. In this study, we do not expect e to converge to zero but to oscillate around zero with a period T. The convergence of the control law is then reached when $\int_{t-T}^t edt = 0$, which is obtained if $\int_{t-T}^t \int_0^t \widehat{L}^k V_c \dot{b}dt = 0$. Let us define $E = \int_{t-T}^t \int_0^t \widehat{L}^k V_c \dot{b}dt$ and note that in general $E \neq 0$. It can be estimated over one period of time T. In order to avoid drift accumulation in the comutation of E we can use a sliding windows to define the current virtual error e and deduce the control law

$$\dot{\bar{k}} = -\lambda \widehat{L}^+ (e - (\int_0^t \widehat{L}^k V_c \dot{b}dt - \int_{t-T}^t \int_0^t \widehat{L}^k V_c \dot{b}dt)) \tag{23}$$

And finally, the CoM reference velocity can be computed as :

$$\dot{\bar{c}} = -\lambda^c V_k \widehat{L}^+ (e - (\int_0^t \widehat{L}^k V_c \dot{b}dt - \int_{t-T}^t \int_0^t \widehat{L}^k V_c \dot{b}dt)) \tag{24}$$

D. Vision based control

Visual Model Predictive Control Scheme has been studied to deal with constraints, eg to ensure the visibility of the target or avoid joint limits [20]. In order to improve the results presented in this paper, we propose to write a general non linear model predictive control scheme to select the optimal jerk of the CoM $\dddot{C}$ regarding some visual criteria. Then the function to minimise is now

$$\min_{\dddot{C}, F} \frac{1}{2} \sum_{i=1}^{N} \| s(k_i) - s_i^* \|^2 \tag{25}$$

IV. A Preliminary Result

Fig. 5, Fig. 6 depicts an experiment of visual servoing for dynamic walking that shows the feasibility of the approach on the HRP2 robot.

The experimental scheme is the following : the robot has to reach a desired position with regards to a partially known object. First, the system is given a desired pose of the object in the camera frame. It can be arbitrarily set or it can be estimated by placing the robot at the desired position. The object is then tracked in the image and its pose is estimated

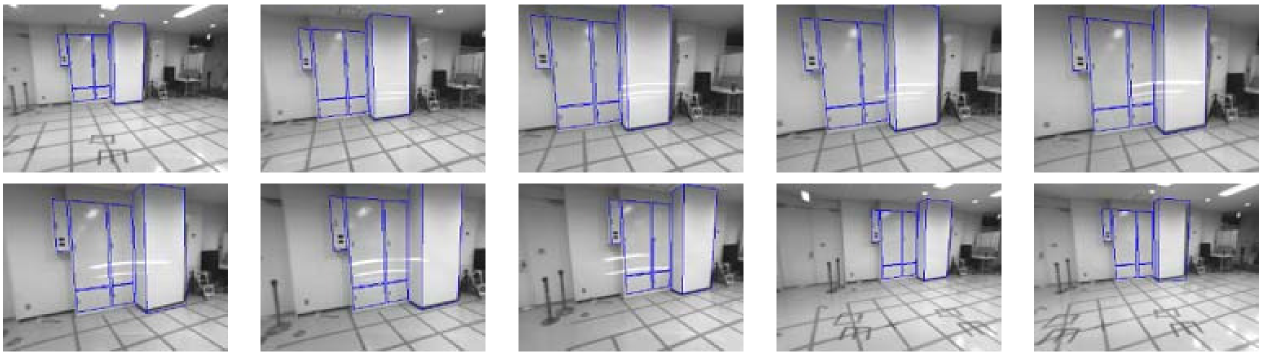


Fig. 5. Model tracking while walking on a square on the ground. The robot firstly walks forward, then sideways, then backwards, and sideways again to reach its initial position. On images 2-8, we can notice white horizontal curved line. They are induced by the reflexion of the light on the dark plastic shield. The model-tracker gives good positioning results when the tracked object has 3 dimensional edges.

in the camera frame. It allows to learn $k^{*}M_o$. The advantage of the learning solution is that the position is estimated using the same camera as the one used on-line for visual servoing, which compensate for calibration errors. Secondly the robot is placed to an initial arbitrary position ${}^{k}M_o$ from where the reference object can be seen. In this paper, the robot has to perform both translational motions and rotational motion to reach the desired position.

Fig. 5 presents some tracking results while the robot is walking. On images 2-8, we can notice white horizontal curved line. They are induced by the reflexivity of the light on the dark plastic shield that protects the HRP-2 camera. They cause some partial occlusion and worse, they can masquerade object lines and make the tracking fail. The model-based tracker we use is designed for convex objects. If the current projection of the model only allows to track 2D planes, it can happen that the optimisation problem falls in a local minimum. The order of magnitude of the model-tracking accuracy, while walking 2m away from the object, is about $0.1m$ and $0.1rad$.

In the experiment illustrated Fig. 6 a position based visual servoing is given the Pattern Generator as an input. In this experiment, the robot has approximatly to move $1m$ forward, $1.5m$ sideways and $0.7rad$ in rotation. A convergence threshold is arbitrarily set to $0.1m$ in translation and $0.1rad$ in rotation. Then the accuracy of the positioning reaches these values at best. Besides, the reference velocity is limited to $0.2m/s$ in translation and $0.2rad/s$ in rotation. This limits have be chosen to secure the robot mechanical parts. It may be increase in the future.

The top left figure depicts the estimated position of the object frame with regards to the camera frame. Notice that the on board camera axis are not parallel to the ground plane. The robot head is oriented slightly towards the ground. Also remark that the position estimation could be replaced by any localisation technique (such as SLAM) except that the model based scheme has the advantage to handling mobile object tracking [1]. In the top left graph, we can see that the lateral motion oscillates. This is directly due to the stepping motion. This lateral motion can be also found in the bottom right figure that is the output velocity of the pattern generator. The walking control guaranties this sway motion to be minimal. Furthermore, the upper body of the HRP2 robot can not compensate for this sway motion due to a lack of degrees of freedom. Anyway, the model tracker proved to be robust enough to track an object even when the camera oscillates under the sway motion. The top right figure first shows an increase of the error and then a visual servoing classical exponential decrease. The increase of the error is directly related to the variation of the pose estimation that can be observed in the top left figure. Both changes are due to the motion induced by the robot first steps. Usually, the robot needs two steps to reach the desired velocity and make the error decrease. The bottom left figure presents the visual control law that is the reference pattern generator velocity.

The resulting motion has been compared to a planned trajectory executed with a Kajita's PG. As shown in [21], when the robot is walking forward or backward, the open loop execution of the planned trajectory results in a good positioning (less than $1cm$ in translation and less than $0.1rad$ in rotation). However, lateral motion induces a large drift and an error in rotation, such that the difference between the initial position and the final position is more than 60cm in translation and more that $0.5rad$ in rotation. Since we have set the convergence threshold to 0.1, the translational error is less than $2cm$. Yet, the error in rotation is small and less than $0.1rad$ even after lateral motion. The difference between the first position and the final position was less that $15cm$ and the error in rotation less than $0.1rad$. As excepted, the greater error are found along the sagital plane. Indeed, this is the direction where the pose estimation is the more uncertain for a monocular camera.

V. Conclusion

We think that vision based control is well suited to control the walk of the humanoid robot HPR-2. The method proved to be robust to model errors and gives better result than executing a planned trajectory without closing the control loop. Our going work on vision based pattern generator

[1] some examples of rolling ball tracking and other objects tracking are available on the lagadic team website : http://www.irisa.fr/lagadic/demo.html

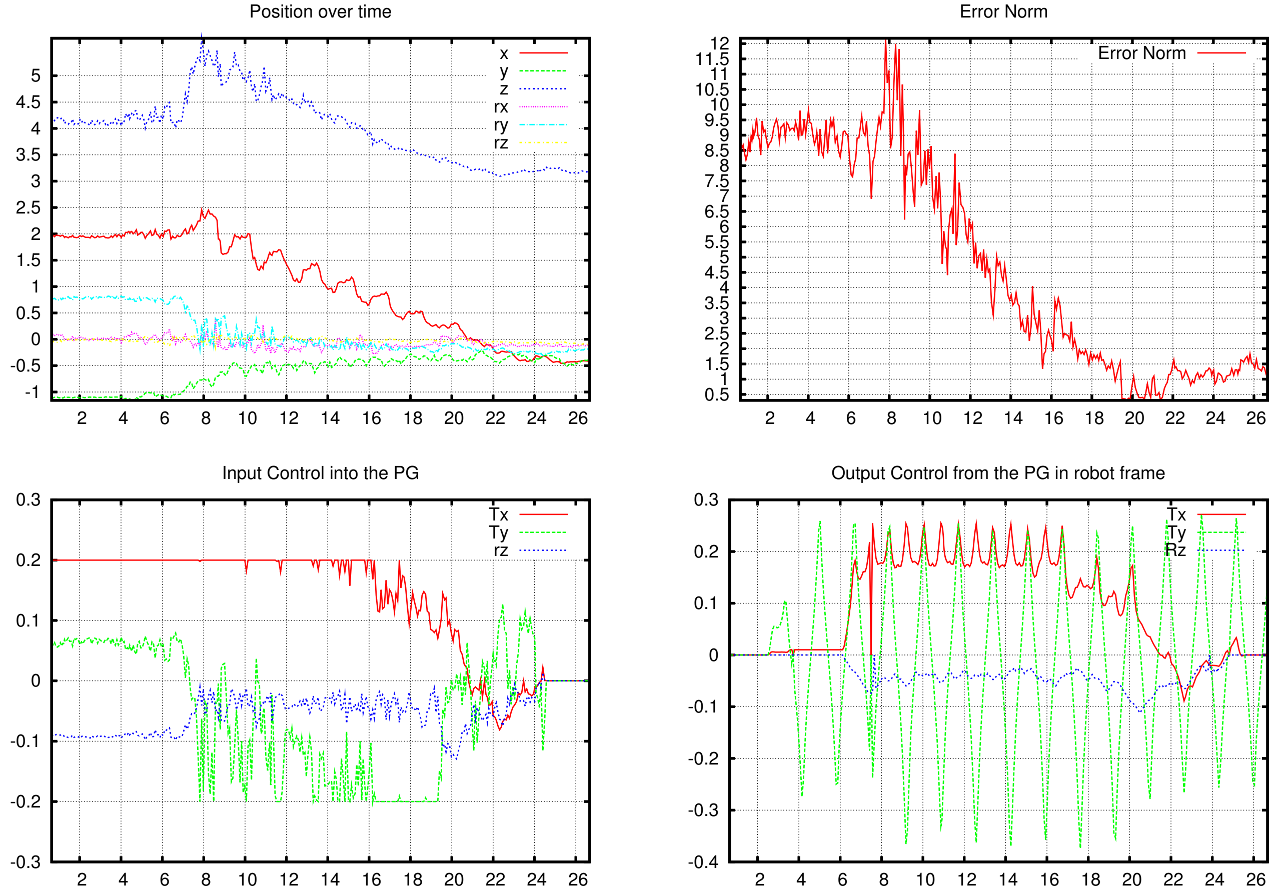


Fig. 6. Visual servoing of dynamic walking experiment. Top left : trajectory of the object's pose in the camera frame. Top right : evolution of the error norm. Bottom left : control Input of the pattern generator (reference CoM velocity $\bar{\dot{c}}$). Bottom right : control output of the pattern generator (real CoM velocity $\dot{c}$), we can remark two picks to zero which are only due to client reading error from the middle ware and these values are not the one sent to the system

is expected to improve these results in several ways : the predictive framework allows to include both balance and visibility constraints, we expect the system to be more reactive and we expect more natural trajectories that do not necessary follows an exponential decrease of the error.

VI. Acknowledgments

The first author is supported by Grant-in-Aid from the Japanese Society for the Promotion of Science (JSPS) Fellows P-09721. The second, third and fourth authors are supported by a grant from the RBLINK Project, Contrat ANR-08-JCJC-0075-01. Special thanks to F. Chaumette for his precious advices and discussions.

References

[1] C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments," *IEEE Robotics and Automation Magazine*, pp. 20–29, march 2007.

[2] O. Lorch, A. Albert, J. Denk, M. Gerecke, R. Cupec, J. Seara, W. Gerth, and G. Schmidt, "Experiments in vision-guided biped walking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 2484–2490.

[3] J. Chestnutt, P. Michel, J. Kuffner, and T. Kanade, "Locomotion among dynamic obstacles for the honda asimo," in *IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 2572–2573.

[4] P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, "Gpu-accelerated real-time 3d tracking for humanoid locomotion and stair climbing," in *IEEE Int. Conf. on Robotics and Automation*, October 2007, pp. 463–469.

[5] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3d perception and environment map generation for humanoid robot navigation," *International Journal of Robotics Research*, vol. 27, no. 10, pp. 1117–1134, 2008.

[6] J. Coelho, J. Piater, and R. Grupen, "Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot," *Robotics and Autonomous Systems*, vol. 37, no. 2-3, pp. 195–217, November 2001.

[7] G. Taylor and L. Kleeman, "Flexible self-calibrated visual servoing for a humanoid robot," in *Australian Conference on Robotics and Automation (ACRA2001)*, 2001, pp. 79–84.

[8] N. Courty, E. Marchand, and B. Arnaldi, "Through-the-eyes control of a virtual humanoïd," in *IEEE Computer Animation 2001*, H.-S. Ko, Ed., Seoul, Korea, November 2001, pp. 74–83.

[9] N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi, "Visually-guided grasping while walking on a humanoid robot," in *IEEE Int. Conf. on Robotics and Automation, ICRA'07*, Roma, Italy, April 2007, pp. 3041–3047.

[10] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," in *IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 3989–3994.

[11] K. Nishiwaki and S. Kagami, "Online walking control system for humanoids with short cycle pattern generation," *Int. Journal of Robotics Research*, vol. 28, pp. 729–742, 2009.

[12] A. Herdt, D. Holger, P. Wieber, D. Dimitrov, K. Mombaur, and D. Moritz, “Online walking motion generation with automatic foot step placement,” *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, April 2010.

[13] A. Herdt, N. Perrin, and P.-B. Wieber, “Walking without thinking about it,” in *IEEE Int. Conf. on Robotics and Automation*.

[14] C. Dune, A. Herdt, O. Stasse, P.-B. Wieber, E. Yoshida, and K. Yokoi, “Cancelling the sway motion of dynamic walking in visual servoing,” in *IEEE Int. Conf. on Robotics and Automation*, 2010.

[15] N. Perrin, S. Stasse, F. Lamiraux, and E. Yoshida, “Approximation of feasibility tests for reactive walk on hrp-2,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4243–4248.

[16] P.-B. Wieber, “On the stability of walking systems,” in *Int. Workshop on Humanoid and Human Friendly Robotics*, 2002.

[17] A. Comport, E. Marchand, M. Pressigout, and F. Chaumette, “Real-time markerless tracking for augmented reality: the virtual visual servoing framework,” *IEEE Trans. on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 615–628, July 2006.

[18] Huber, *Robust Statistics*. Wiler New York, 1981.

[19] F. Chaumette and S. Hutchinson, “Visual servo control, part i: Basic approaches,” *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, December 2006.

[20] G. Allibert, E. Courtial, and F. Chaumette, *Visual Servoing via Advanced Numerical Methods*. Springer, Avril 2010, ch. 20- Visual Servoing via Nonlinear Predictive Control, pp. 393–412.

[21] O. Stasse, A. Davison, R. Sellouati, and K. Yokoi, “Real-time 3d slam for humanoid robot considering pattern generator information,” in *IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 348–355.

IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)
September 26, 2011, San Francisco, California, USA

Weakly-Calibrated Visual Control of Mobile Robots using the Trifocal Tensor and Central Cameras

H. M. Becerra, G. López-Nicolás and C. Sagüés

Abstract— **In this paper, we present the synthesis of two control schemes that exploit the properties of the trifocal tensor computed from bearing measurements (1D TT), for the pose-regulation problem of mobile robots. Both control schemes are valid for vision systems obeying a central projection model, in such a way that visibility constraint problems can be overcome. The use of the 1D TT avoids the need of a complete camera calibration for any type of central camera, so that, weakly-calibrated control schemes are obtained. This benefit of the 1D TT as measurement is exploited in an image-based (IB) approach as well as in a position-based (PB) approach. The IB scheme employs direct feedback of the elements of the tensor without commuting to any other approach during the control task. The PB approach relies on the feedback of the pose estimated dynamically from the 1D TT. Both visual control schemes are evaluated through real-world experiments using a hypercatadioptric imaging system.**

I. Introduction

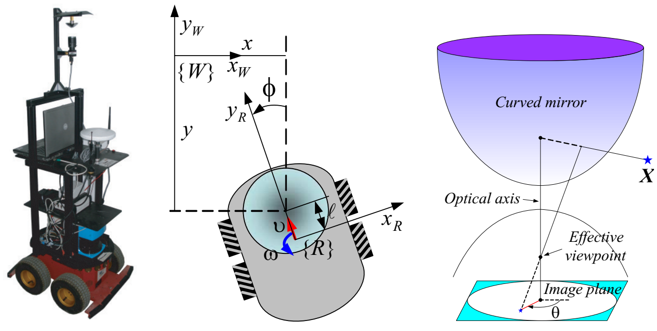
Visual control of mobile robots is an interesting research field, motivated by the introduction of this type of robots as *service robots*. Particularly, wheeled mobile robots (WMR) are well appreciated in service tasks, where the positioning at a desired location is an important aspect. This paper describes an approach to drive a WMR equipped with a central generic camera onboard to a desired location, which is specified by a target image previously acquired, i.e., using a teach-by-showing strategy. Along the years, the research on visual control has dedicated important efforts to find suitable error functions in order to obtain a desired behavior of the robotic system in terms of stability and robustness of the closed loop control. The basic approaches are typically separated in image-based (IB) schemes, in which the error function consists of a set of features that are directly available in the image data, and position-based (PB), in which a set of 3D parameters must be estimated from image measurements [1].

The goal of steering a robot to a desired location by visual servoing (VS) is carried out by minimizing an error function that relates visual data, typically from two images: the current and the target one. We propose to take advantage of more information by using three views and the geometric constraint that describes the complete geometry between them, the trifocal tensor (TT). This geometric constraint is more robust and more stable than those based on two views an it is also independent of the observed scene [2]. Its simplified version constrained to planar motion, the 1D TT, has proved its effectiveness for localization in [3] and [4], but has been less studied for control applications. In these works, conventional perspective and omnidirectional cameras are converted to 1D virtual cameras through a transformation of bearing measurements. The authors of [5] assert that the radial 1D camera model is sufficiently general to represent the great majority of omnidirectional cameras under the assumption of knowing the center of radial distortion. The 2D TT has been introduced for visual control of mobile robots in [6]. This approach shows good results reaching the target location, but it uses a non-exact system inversion that suffers of potential stability problems. An application of the TT related to camera-motion estimation is presented in [7]. It introduces a filtering algorithm with the TT as measurement model to tackle the vision-based pose-tracking problem for augmented reality applications. The use of more than two views in VS provides robustness and enough information to correct also depth from visual feedback, which is not possible from two views.

In this paper, we present two visual control schemes that exploit the property of the 1D TT of being estimated from bearing information. This provides the advantage that parameters related to focal length do not appear in the control laws, so that, weakly-calibrated schemes are obtained. Additionally, the simplified representation of the imaging systems as 1D virtual cameras provides the versatility of the schemes to be applied using any central camera [8]. First, an IB scheme that uses direct feedback of the elements of the 1D TT is presented, as a summary of our previous work [9]. The proposed switching control law turns out to be a square control system that consists of two controllers, which correct position and orientation in two steps. Secondly, we present a PB scheme that feeds back the robot pose estimated dynamically from the 1D TT, which has been introduced in [10]. We show the property of observability of the system with the 1D TT as measurement using linear theory. The proposed PB scheme corrects the robot position and orientation using smooth robot velocities from a single control law. Real-world experiments using a hypercatadioptric imaging system as sensor show the validity of the proposed approaches.

The paper is organized as follows. Section II specifies the mathematical modeling of the mobile robot and the 1D TT geometric constraint. Section III presents the development of the image-based approach and Section IV describes the position-based approach. Section V shows the performance of the proposed approaches via real-world experiments. Finally, Section VI provides the conclusions.

This work was supported by project DIP 2009-08126. The authors are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza C/ María de Luna 1, E-50018 Zaragoza, Spain {hector.becerra, gonlopez, csagues}@unizar.es



(a) (b) (c)

Fig. 1. Description of the robot-camera configuration. (a) Our robotic platform with an hypercatadioptric camera on top. (b) Robot frame definition. (c) A generic central catadioptric system.

II. Mathematical Modeling

A. Robot Model

This work focuses on controlling a wheeled mobile robot through the information given by a generic central camera mounted onboard, as shown in Fig. 1(a), and under the framework that is depicted in Fig. 1(b). The camera can be eventually translated a distance ℓ along the longitudinal axis of the robot. The kinematic motion model of the camera-robot system as expressed in state space is

$$\begin{aligned} \dot{x} &= -\omega\ell\cos\phi - \upsilon\sin\phi, \\ \dot{y} &= -\omega\ell\sin\phi + \upsilon\cos\phi, \\ \dot{\phi} &= \omega. \end{aligned} \tag{1}$$

By applying an Euler approximation (forward difference) on the continuous derivatives, the discrete version of the camera-robot model is obtained:

$$\begin{aligned} x_{k+1} &= x_k - T_s\left(\omega_k\ell\cos\phi_k + \upsilon_k\sin\phi_k\right), \\ y_{k+1} &= y_k - T_s\left(\omega_k\ell\sin\phi_k - \upsilon_k\cos\phi_k\right), \\ \phi_{k+1} &= \phi_k + T_s\omega_k, \end{aligned} \tag{2}$$

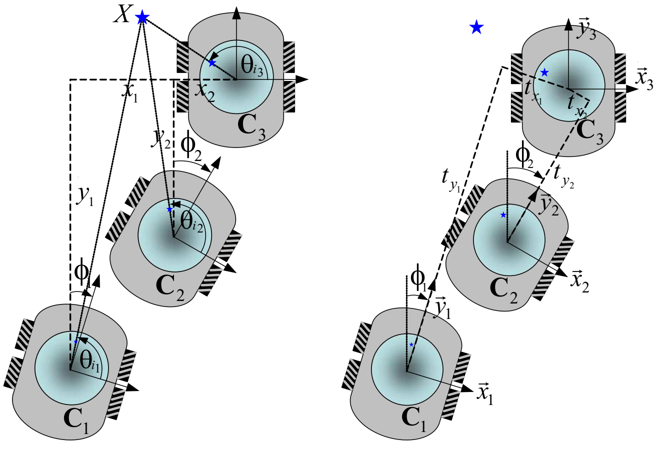
where T_s is the sampling period. In the sequel, we use the notation $s\phi = \sin\phi$, $c\phi = \cos\phi$.

B. The Trifocal Tensor for Central Cameras

The procedure to estimate the trifocal tensor (TT) is basically the same for conventional and central catadioptric cameras if it is formulated in terms of rays that emanate from the effective viewpoint [8]. In the case of planar motion, the simplified version of the tensor, the 1D TT, particularly adapts to the property of omnidirectional images to preserve bearing information regardless of the high radial distortion induced by lenses and mirrors. Fig. 1(c) shows the bearing angle of an observed feature in a hypercatadioptric system looking upwards. Therefore, the bearing measurement θ can be converted to its 1D projection as $\mathbf{p} = (\sin\theta, \cos\theta)^T$. For conventional cameras looking forward, the projective formulation can be obtained using the normalized x-coordinate of the point features with respect to the principal point, i.e., $\mathbf{p} = (\mathrm{u}_n, 1)^T$. By relating this representation for three different views of a feature, it results in the trifocal constraint

$$\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} T_{ijk}\mathrm{u}_i\mathrm{v}_j\mathrm{w}_k = 0, \tag{3}$$

where $\mathbf{u} = (\mathrm{u}_1,\mathrm{u}_2)^T$, $\mathbf{v} = (\mathrm{v}_1,\mathrm{v}_2)^T$ and $\mathbf{w} = (\mathrm{w}_1,\mathrm{w}_2)^T$ are given by bearing measurements projected as described before into the 1D virtual retina of each view, and T_{ijk} are the eight elements of the 1D TT. In general, the trifocal tensor elements are estimated linearly by stacking seven constraints (3).



(a) (b)

Fig. 2. Framework of the three-view geometry. (a) Global reference definition with origin in the third view and bearing measurements θ of a feature. (b) Relative location between cameras with a fixed reference frame on each view.

Let us define a global reference frame as depicted in Fig. 2(a) with the origin in the third camera. Then, the camera locations with respect to that global reference are $\mathbf{C}_1 = (x_1, y_1, \phi_1)$, $\mathbf{C}_2 = (x_2, y_2, \phi_2)$ and $\mathbf{C}_3 = (0, 0, 0)$. The relative locations between cameras are defined by a local reference frame in each camera as shown in Fig. 2(b). The geometry of the three views is encoded in the tensor elements as follows:

$$\mathbf{T}^m_{ijk} = \begin{bmatrix} T^m_{111} \\ T^m_{112} \\ T^m_{121} \\ T^m_{122} \\ T^m_{211} \\ T^m_{212} \\ T^m_{221} \\ T^m_{222} \end{bmatrix} = \begin{bmatrix} t_{y_1}s\phi_2 - t_{y_2}s\phi_1 \\ -t_{y_1}c\phi_2 + t_{y_2}c\phi_1 \\ t_{y_1}c\phi_2 + t_{x_2}s\phi_1 \\ t_{y_1}s\phi_2 - t_{x_2}c\phi_1 \\ -t_{x_1}s\phi_2 - t_{y_2}c\phi_1 \\ t_{x_1}c\phi_2 - t_{y_2}s\phi_1 \\ -t_{x_1}c\phi_2 + t_{x_2}c\phi_1 \\ -t_{x_1}s\phi_2 + t_{x_2}s\phi_1 \end{bmatrix} \tag{4}$$

where $t_{x_i} = -x_i c\phi_i - y_i s\phi_i$, $t_{y_i} = x_i s\phi_i - y_i c\phi_i$ for $i = 1, 2$. Some details on deducing the trifocal constraint (3) and the expressions in (4) can be found in [4]. Additional constraints $-T_{111} + T_{122} + T_{212} + T_{221} = 0$, and $T_{112} + T_{121} + T_{211} + T_{222} = 0$ are accomplished when the 1D TT is computed from a calibrated retina. These calibration constraints allow us to estimate the 1D TT from only five triplets of point correspondences, which improves the estimation [4]. These additional constraints can be always used for central cameras, because the bearing measurements are independent on focal length. Only the center of projection for omnidirectional images or the principal point for conventional cameras is required to estimate the 1D TT. Thus, the use of this tensor as measurement results in weakly-calibrated control schemes, in contrast to previous approaches [11], [12], [13]. It is worth mentioning that it is needed to normalize the tensor elements in order to fix a scale of the measurements, where normalize means to divide each element by one of them that can be assumed as constant ($T_N^m = T_{121}$).

III. Image-based Control from the 1D TT

The problem of taking three variables to desired values $(x_2, y_2, \phi_2) = (0, 0, 0)$ may be solved with at least three outputs being controlled, but defining more than two outputs generate a non-square control system, in which its non-invertibility makes difficult to prove stability. The trifocal tensor is an overconstrained measurement; however, it is possible to find two outputs to drive them to desired values and then a third variable remains as a DOF to be corrected a posteriori. By taking into account the values of the tensor elements at the final location, the solution of the homogeneous linear system generated when the outputs are equal to zero and the invertibility of the matrix relating the output dynamics with the robot velocities, we find that it is feasible to design a square control system that corrects both longitudinal and lateral error, leaving the orientation as a DOF. The orientation error can be corrected in a second step considering that the robot uses a differential drive.

A. First-Step - Position Correction

Let us define the following sum of normalized tensor elements as outputs to be controlled:

$$\begin{aligned} \xi_1 &= T_{112} + T_{121}, \\ \xi_2 &= T_{212} + T_{221}. \end{aligned} \tag{5}$$

A robust tracking controller is proposed to take the value of both outputs to zero in a smooth way. Let $e_1 = \xi_1 - \xi_1^d$ and $e_2 = \xi_2 - \xi_2^d$ be the corresponding tracking errors, where ξ_1^d and ξ_2^d are suitable sinusoidal references. Using the time derivatives of these errors and considering that the camera location coincides with the vertical axis of rotation of the robot ($\ell = 0$), we obtain the error system

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} -\frac{c\phi_1}{T_N^m} & T_{122} - T_{111} \\ -\frac{s\phi_1}{T_N^m} & T_{222} - T_{211} \end{bmatrix} \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} - \begin{bmatrix} \dot{\xi}_1^d \\ \dot{\xi}_2^d \end{bmatrix}. \tag{6}$$

This system has the form $\dot{\mathbf{e}} = \mathbf{D}(\mathbf{T}, \phi_1)\,\mathbf{u} - \dot{\xi}^d$, where $\mathbf{D}(\mathbf{T}, \phi_1)$ corresponds to the decoupling matrix, $\dot{\xi}^d$ represents a feedforward term depending on the known references [9] and T_N^m is defined at the end of Section II.B. We treat the tracking problem as the robust stabilization of the error system (6). A control law based on sliding mode control [14], which has been already applied in the context of visual control [15], is proposed as follows:

$$\mathbf{u}_{db} = \begin{bmatrix} \upsilon_{db} \\ \omega_{db} \end{bmatrix} = \mathbf{D}^{-1} \begin{bmatrix} \dot{\xi}_1^d - \kappa_1 \text{sign}(s_1) - \lambda_1 s_1 \\ \dot{\xi}_2^d - \kappa_2 \text{sign}(s_2) - \lambda_2 s_2 \end{bmatrix} \tag{7}$$

where $\kappa_1 > 0$, $\kappa_2 > 0$, $\lambda_1 > 0$, $\lambda_2 > 0$ are control gains and $s_1 = e_1$, $s_2 = e_2$ are the so-called sliding surfaces. Note that the control law depends on the orientation of the fixed auxiliary camera ϕ_1. This parameter can be fixed to zero and any error with respect to the real value is tackled by the robust control. The control law uses the inverse of the decoupling matrix $\mathbf{D}$ to compute the robot velocities, which presents a singularity when the robot reaches the target position. This entails the problem that the rotational velocity ω_{db} increases to infinity as the robot reaches the target. However, we propose the commutation to a direct sliding mode controller when $\det(\mathbf{D})$ is near to zero in order to keep ω_{db} bounded. This kind of controller has been studied for output tracking through singularities [16]. For our case, a bounded sliding mode controller is

$$\mathbf{u}_b = \begin{bmatrix} \upsilon_b \\ \omega_b \end{bmatrix} = \begin{bmatrix} k_\upsilon \text{sign}(s_1) \\ -k_\omega \text{sign}(s_2 \left(T_{222} - T_{211}\right)) \end{bmatrix} \tag{8}$$

where k_υ and k_ω are suitable control gains.

B. Second-Step - Orientation Correction

Once position correction has been reached, we can use any single tensor element whose dynamics depends on ω and with desired final value zero to correct orientation. We select the dynamics $\dot{T}_{122} = -T_{112}\omega$. A suitable input ω is

$$\omega = \lambda_\omega \tfrac{T_{122}}{T_{112}}, \tag{9}$$

where $\lambda_\omega > 0$ is a control gain. This rotational velocity assigns the following dynamics to T_{122}, which is exponentially stable:

$$\dot{T}_{122} = -T_{112}\left(\lambda_\omega \tfrac{T_{122}}{T_{112}}\right) = -\lambda_\omega T_{122}.$$

Note that (9) never becomes singular because at the beginning of this step $T_{112} = -t_{y_1}\cos\phi_2$, and it tends to $-t_{y_1} \neq 0$ as final value. Although only a rotation is carried out in this second step, we keep the translational velocity υ_b given in (8) in order to keep closed loop control along the whole motion.

IV. Position-based Control from the 1D TT

The elements of the 1D TT are very useful providing information of position and orientation of a camera [4]. We propose to make use of the information provided by the 1D TT to estimate the camera motion dynamically, according to the nonholonomic motion model (2). Once the robot pose is estimated, it can be used to control the robot in the Cartesian space.

Consider the problem of estimating the state $\mathbf{x}_k = (x_k, y_k, \phi_k)^T$ of the discrete model of the robot (2) by using measurements $\mathbf{y}_k$, which depend on the robot state through a nonlinear function h. It is assumed that the robot state and the measurements are affected by Gaussian noises $\mathbf{m}_k$ and $\mathbf{n}_k$, respectively. The noisy system and measurement model can be expressed in compact form as follows:

$$\begin{aligned} \mathbf{x}_{k+1} &= f\left(\mathbf{x}_k, \mathbf{u}_k\right) + \mathbf{m}_k, \\ \mathbf{y}_k &= h\left(\mathbf{x}_k\right) + \mathbf{n}_k \end{aligned} \tag{10}$$

where it is accomplished $\mathbf{m}_k \sim N(0, \mathbf{M}_k)$, $\mathbf{n}_k \sim N(0, \mathbf{N}_k)$ and $E\left[\mathrm{m}_{k,i}\mathrm{n}_{k,j}^T\right] = 0$, with $\mathbf{M}_k$ the state noise covariance and $\mathbf{N}_k$ the measurement noise covariance. This estimation problem can be solved by a filtering approach using an Extended Kalman Filter (EKF), however, the property of observability must be ensured in order to achieve a consistent estimation.

A. Linear Observability from Measurements of the 1D TT

There are few works concerned about observability when an estimation based on Kalman filtering is applied. Some of them are [17] and [18]. To analyze our case, let us consider the linear approximation ($\mathbf{F}_k$, $\mathbf{G}_k$, $\mathbf{H}_k$) of the system (2) in the time k, where

$$\mathbf{F}_k = \left|\frac{\partial f}{\partial \mathbf{x}_k}\right|_{\substack{\mathbf{x}_k=\hat{\mathbf{x}}_k^+,\\ \mathbf{m}_k=0}} \quad \mathbf{G}_k = \left|\frac{\partial f}{\partial \mathbf{u}_k}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_k^+,} \quad \mathbf{H}_k = \left|\frac{\partial h}{\partial \mathbf{x}_k}\right|_{\substack{\mathbf{x}_k=\hat{\mathbf{x}}_k^-\\ \mathbf{n}_k=0}}.$$

Due to the matrices $\mathbf{F}_k$ and $\mathbf{H}_k$ are changing at each instant time, observability may not be ensured, which affects the convergence properties of the estimation algorithm. As mention in [17], a system that is locally observable over every time segment $[t_k, t_{k+1}]$ in the interval $[t_0, t_{k+1}]$ will also be completely observable over the interval $[t_0, t_{k+1}]$. Then, the condition to accomplish for every k to ensure the system to be completely observable is

$$rank\left(\left[\begin{array}{cccc} \mathbf{H}_k^T & (\mathbf{H}_k\mathbf{F}_k)^T & \cdots & \left(\mathbf{H}_k\mathbf{F}_k^{n-1}\right)^T \end{array}\right]^T\right) = n.$$

Because of the triangular form of the matrix $\mathbf{F}_k$, the rows of the observability matrix become linearly dependent. The only possibility of achieving the full rank condition is by building $\mathbf{H}_k$ of full space. It can be done by taking three elements of the TT as outputs. By analyzing the Jacobian of each element of the tensor, we find that a suitable selection of measurements is T_{122}, T_{211}, T_{111}, in such a way that

$$\mathbf{H}_k = \left[\begin{array}{ccc} c\phi_1 c\hat{\phi} & c\phi_1 s\hat{\phi} & t_{y_1} c\hat{\phi} - \hat{t}_y c\phi_1 \\ -c\phi_1 s\hat{\phi} & c\phi_1 c\hat{\phi} & -t_{x_1} c\hat{\phi} + \hat{t}_x c\phi_1 \\ -s\phi_1 s\hat{\phi} & s\phi_1 c\hat{\phi} & t_{y_1} c\hat{\phi} + \hat{t}_x s\phi_1 \end{array}\right], \quad (11)$$

where $\hat{\phi} = \hat{\phi}_{k|k-1}$, $\hat{t}_x = -\hat{x}_{k|k-1} c\hat{\phi}_{k|k-1} - \hat{y}_{k|k-1} s\hat{\phi}_{k|k-1}$, $\hat{t}_y = \hat{x}_{k|k-1} s\hat{\phi}_{k|k-1} - \hat{y}_{k|k-1} c\hat{\phi}_{k|k-1}$, and t_{x_1}, t_{y_1} and ϕ_1 are constant values. The measurement matrix in (11) ensures local observability for every k even for some particular initial conditions, for instance $\phi_1 = 0$, in which case this matrix remains full rank due to the cosines in the main diagonal. It is worth emphasizing that the result is valid for normalized tensor elements, although we show the previous expressions for non-normalized elements for simplicity. Actually, observability from only one element of the tensor as measurement is ensured according to a nonlinear analysis [19]. In the same paper can be found a method for the EKF initialization using the information provided by the tensor.

B. Control Law using the Estimated Robot Pose

In this section, we assume that the robot pose is available, given by the EKF using three elements of the TT as measurements. The outputs to be controlled are the camera position coordinates x_k and y_k. Consequently, the orientation (ϕ_k) is left as a DOF which is automatically corrected by tracking suitable desired trajectories. To take the value of both outputs to zero in a smooth way, we design a tracking controller. Let us define the tracking errors as $\xi_k^1 = x_k - x_k^d$, $\xi_k^2 = y_k - y_k^d$, with x_k^d and y_k^d suitable references [19]. Thus, the error dynamics $\xi_k = (\xi_k^1, \xi_k^2)^T$ obey the following difference equation:

$$\xi_{k+1} = \xi_k + T_s \left[\begin{array}{cc} -s\phi_k & -\ell c\phi_k \\ c\phi_k & -\ell s\phi_k \end{array}\right] \mathbf{u}_k - T_s \left[\begin{array}{c} \dot{x}_k^d \\ \dot{y}_k^d \end{array}\right]. \quad (12)$$

We can see that the control inputs appear in the first order difference equation of each output. Then, the system (2) with outputs (x_k, y_k) has a vector relative degree {1,1}. Then, we have a first order zero dynamics, which corresponds to the DOF of the control system, the orientation ϕ_k. A static state feedback control law that achieves global stabilization of the system (12) is

$$\left[\begin{array}{c} \hat{\upsilon}_k \\ \hat{\omega}_k \end{array}\right] = \frac{1}{\ell}\left[\begin{array}{cc} -\ell s\hat{\phi}_k & \ell c\hat{\phi}_k \\ -c\hat{\phi}_k & -s\hat{\phi}_k \end{array}\right]\left[\begin{array}{c} \hat{\nu}_k^1 \\ \hat{\nu}_k^2 \end{array}\right], \quad (13)$$

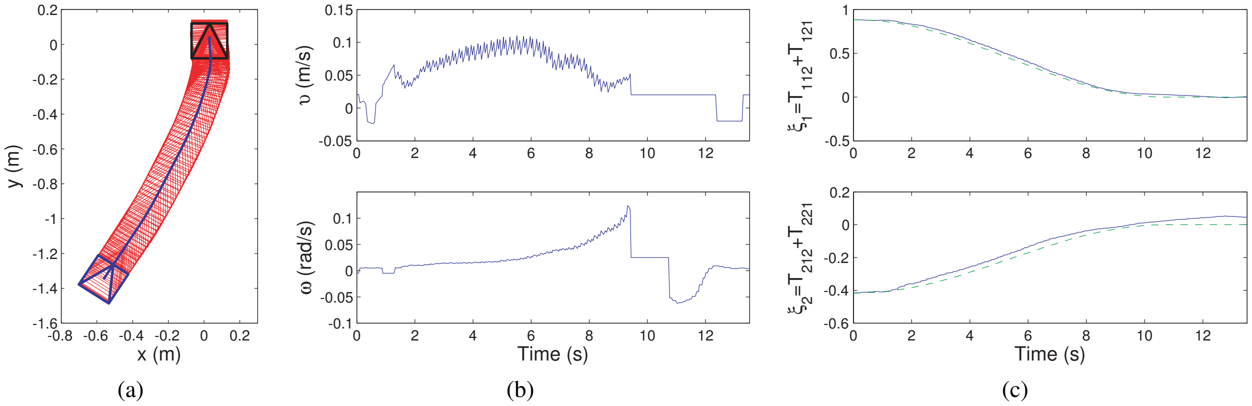
where $\hat{\nu}_k^1 = -k_1\hat{\xi}_k^1 + \dot{x}_k^d$ and $\hat{\nu}_k^2 = -k_2\hat{\xi}_k^2 + \dot{y}_k^d$. The error behavior will be exponentially stable iff $k_1 > 0$, $k_2 > 0$. Note that this input-output linearization via static feedback is valid for $\ell \neq 0$. Otherwise, a singular decoupling matrix is obtained. However, the case of having the camera shifted from the robot rotation axis over the longitudinal axis is a common situation. Orientation correction is simultaneously achieved by tracking a parabolic path in the Cartesian space, which is demonstrated in [19].

V. Experimental Evaluation

Both proposed approaches have been tested in real-world experiments using the robot presented in Fig. 1(a). The camera acquires images of size 800×600 pixels. The 1D TT is estimated using the five-point method as described in Section II.B with estimated projection center ($x_0 = 404$, $y_0 = 316$) as the only required calibration parameter. These experiments have been carried out using a tracking of features as implemented in the OpenCV library.

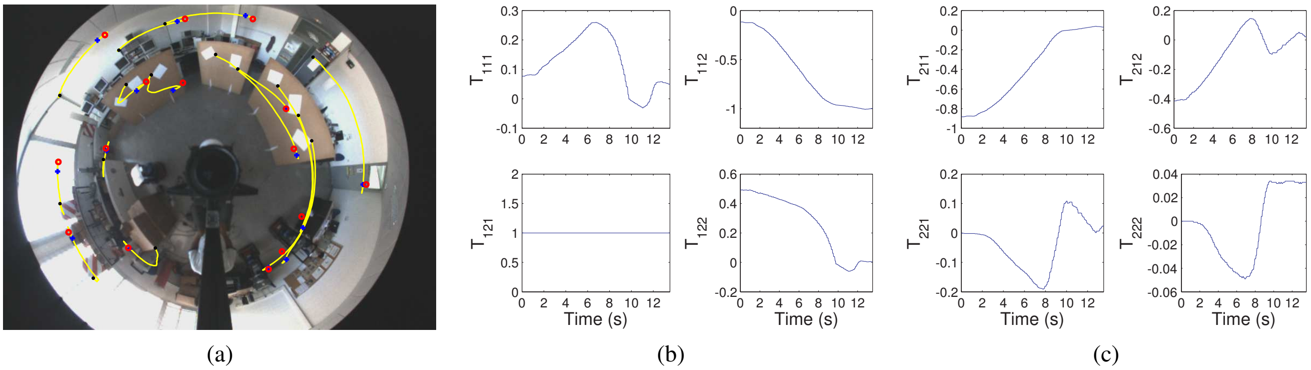
A. Image-based Approach

The experiment shown in this section corresponds to a trial of the image-based approach. Fig. 3(a) presents the resultant path, given by odometry, from the ground truth initial location (-0.55 m,-1.35 m,-35 deg). The time to accomplished the pose-regulation task is almost 14 s. The execution time of the first step is set to 9.4 s through fixing a number of iterations in which the tracked references reach zero. Before that, we can see in Fig. 3(b) that the bounded sliding mode control law is applied when the singularity appears. Fig. 3(c) shows that the behavior of the outputs is always close to the desired one but with a small error. The reason of the remaining error is that our robotic platform is not able to execute commands at a frequency higher than 10 Hz, and consequently the performance of the sliding mode control is not the optimum. According to Fig. 4(a) the motion



(a) (b) (c)

Fig. 3. Experimental results with the image-based approach. (a) Resultant path from the data given by the robot odometry. (b) Computed velocities. (c) Controlled outputs.



(a) (b) (c)

Fig. 4. Behavior of the visual measurements for the experiment using the image-based approach. (a) Motion of the image points. The marker "·" corresponds to the initial points, the marker "O" to the target points and the marker "+" are the points in the image at the end of the motion. (b) Evolution of the first four tensor elements. (c) Evolution of the second four tensor elements.

of the image points along the sequence does not exhibit a damaging noise, in such a way that the tensor elements evolve smoothly during the task, as presented in Fig. 4(b)-(c). The robot finally reaches the target with good precision.

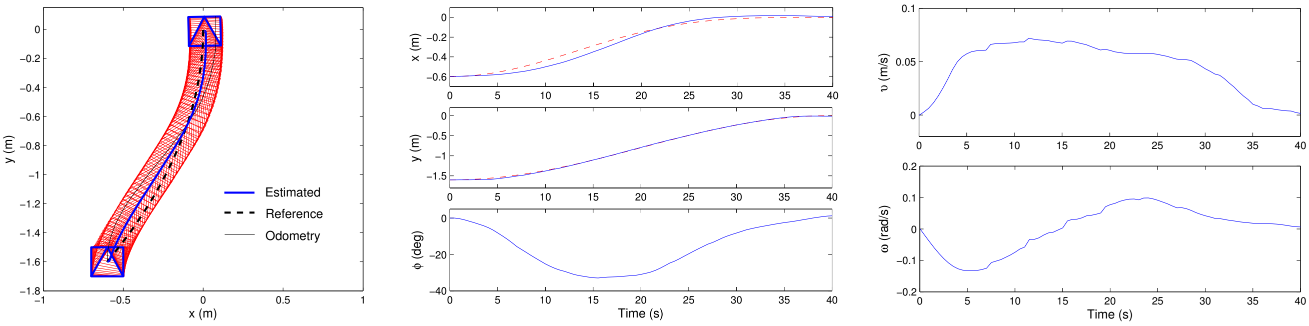
B. Position-based Approach

For the evaluation of the position-based approach the sampling time T_s is set to 0.5 s. The distance from the camera to the rotation axis of the robot has been roughly set to $\ell = 10$ cm. The ground truth initial location is (-0.6 m,-1.8 m,0 deg). Fig. 5(a) presents the resultant path, given by the estimated state of the robot, for one of the experimental runs. The figure also shows the reference path and the one given by odometry. It can be seen that the estimated path is closer to the reference than the path obtained from odometry. The duration of the pose-regulation task is fixed to 40 s, when the tracked references reach zero. Fig. 5(b) shows the behavior of the estimated state together with the tracked references for the position coordinates. According to Fig. 5(c) the robot velocities behave smoothly along the task, which represent an improved performance with respect to the image-based approach. Fig. 6(a) presents the well behaved motion of the image points along the sequence. The evolution of tensor elements is shown in Fig. 6(b). It is worth noting that the tensor estimation is not affected when the robot is reaching the target, i.e., there is no problem with the short baseline.

VI. CONCLUSIONS

In this paper, we have presented and evaluated experimentally two control schemes that rely on monocular vision to solve the pose-regulation problem of mobile robots. The proposed schemes are valid for vision systems obeying a central projection model, so that visibility constraint problems are avoided with the adequate sensor. In both proposed schemes, an adequate set of visual measurements are taken from the 1D trifocal tensor (TT). This tensor is estimated from bearing information of the visual features, which avoids the need of a complete camera calibration for any type of central camera and therefore, weakly-calibrated control schemes are obtained.

The properties of the 1D TT have been exploited in image-based (IB) and position-based (PB) schemes. The proposed IB scheme employs the direct feedback of elements of the 1D TT without commuting to any other approach during the whole task. This scheme is a two-step control law that is based on the sliding mode control technique. The proposed PB approach relies on the feedback of the estimated pose for control in the Cartesian space, with the benefits of reducing the dependence of the servoing on the visual data and facilitating the planning of complex tasks. We have shown that the 1D TT provides enough information to estimate the robot pose dynamically through a linear observability

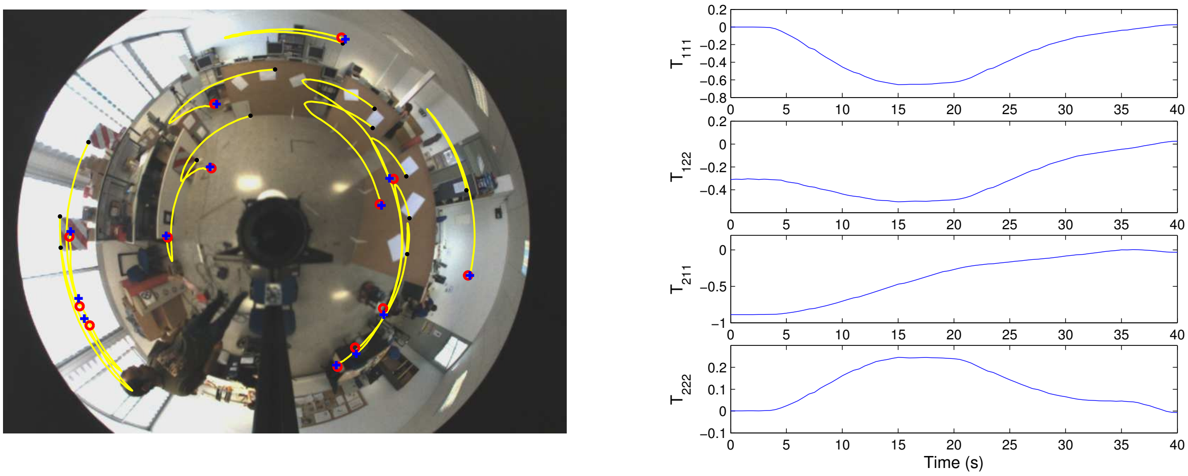


(a)

(b)

(c)

Fig. 5. Experimental results with the position-based approach. (a) Resultant path plotted using the estimated camera-robot state, although the reference path and the odometry are also shown. (b) Estimated camera-robot state. (c) Computed velocities.



(a)

(b)

Fig. 6. Behavior of the extracted information from the images for the experiment with the position-based approach. (a) Motion of the point features. Initial points - "·", target points - "O" and final points - "+". (b) Four normalized tensor elements, three of them are taken as measurements.

analysis. The PB scheme is a single-step control law that corrects the robot pose using smooth robot velocities.

References

[1] F. Chaumette and S. Hutchinson. Visual servo control. I. Basic approaches. *IEEE Robotics and Autom. Mag.*, 13(4):82–90, 2006.

[2] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[3] F. Dellaert and A. W. Stroupe. Linear 2D localization and mapping for single and multiple robot scenarios. In *IEEE International Conference on Robotics and Automation*, pages 688–694, 2002.

[4] J.J. Guerrero, A.C. Murillo, and C. Sagüés. Localization and matching using the planar trifocal tensor with bearing-only data. *IEEE Transactions on Robotics*, 24(2):494–501, 2008.

[5] S. Thirthala and M. Pollefeys. Trifocal tensor for heterogeneous cameras. In *Proc. of 6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS)*, 2005.

[6] G. López-Nicolás, J.J. Guerrero, and C. Sagüés. Visual control through the trifocal tensor for nonholonomic robots. *Robotics and Autonomous Systems*, 58(2):216–226, 2010.

[7] Y.K. Yu, K.H. Wong, M.M.Y Chang, and S.H. Or. Recursive camera-motion estimation with the trifocal tensor. *IEEE Trans. on Systems, Man, and Cyber. - Part B: Cyber.*, 36(5):1081–1090, 2006.

[8] C. Geyer and K. Daniilidis. An unifying theory for central panoramic systems and practical implications. In *European Conference on Computer Vision*, pages 445–461, 2000.

[9] H. M. Becerra, G. López-Nicolás, and C. Sagüés. Omnidirectional visual control of mobile robots based on the 1D trifocal tensor. *Robotics and Autonomous Systems*, 58(6):796–808, 2010.

[10] H. M. Becerra and C. Sagüés. Pose-estimation-based visual servoing for differential-drive robots using the 1D trifocal tensor. In *IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, pages 5942–5947, 2009.

[11] G. L. Mariottini and D. Prattichizzo. Image-based visual servoing with central catadioptric cameras. *The International Journal of Robotics Research*, 27(1):41–56, 2008.

[12] S. Benhimane and E. Malis. A new approach to vision-based robot control with omni-directional cameras. In *IEEE International Conference on Robotics and Automation*, pages 526–531, 2006.

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

[14] J. Guldner V. Utkin and J. Shi. *Sliding Mode Control in Electromechanical Systems*. CRC Press, Boca Raton, 1999.

[15] H. M. Becerra, G. López-Nicolás, and C. Sagüés. A sliding mode control law for mobile robots based on epipolar visual servoing from three views. *IEEE Transactions on Robotics*, 27(1):175–183, 2011.

[16] R. M Hirschorn. Output tracking through singularities. In *IEEE Conference on Decision and Control*, pages 3843–3848, 2002.

[17] M. Bryson and S. Sukkarieh. Observability analysis and active control for airbone SLAM. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1):261–280, 2008.

[18] T. Vidal-Calleja, M. Bryson, S. Sukkarieh, A. Sanfeliu, and J. Andrade-Cetto. On the observability of bearing-only SLAM. In *IEEE Int. Conf. on Robotics and Autom.*, pages 4114–4119, 2007.

[19] H. M. Becerra and C. Sagüés. Exploiting the trifocal tensor in dynamic pose-estimation for visual control. *Submitted for a Journal Paper*, 2011.

Automatic Regions-of-Interest Selection based on Pearson's Correlation Coefficient

A. Miranda Neto, A. Corrêa Victorino, I. Fantoni and D. E. Zampieri

Abstract—**Navigation of a Mobile Robot is based on its interaction with the environment, through information acquired by sensors. Particularly for Mobile Robot navigation in unknown environment, the type and number of sensors determines the data volume necessary to process and compose the image from the environment. Nevertheless, the excess of information imposes a great computational cost in data processing. Taking into account the temporal coherence between consecutive frames, a Discarding Criteria methodology based on Pearson's Correlation Coefficient (PCC) was proposed and applied as a Dynamic Power Management solution to a robotic visual-machine perception. In this context, this work proposes an environment observer method based on PCC that instead of processing all image pixels, it selects automatically only the regions-of-interest (ROI) and processes it in real time in order to perform a task: road detection and obstacle avoidance. This real-time visual perception system has been evaluated from real data obtained by two experimental platforms.**

I. Introduction

Lately, several applications for control of autonomous vehicles are being developed, and, in most cases, the machine vision is an important part of the set of sensors used for navigation. Some of these applications include: the aerial robots that offer great perspectives in many applications as search and rescue, real-time monitoring, high risk aerial missions, mapping, etc [1], [2]. Similarly, the development of Unmanned Aerial Vehicles (UAVs) has been of interest for military applications, however, one limitation is their maximum flight time; therefore they cannot carry large fuel payloads [3]. Future exploration of Mars requires long-endurance UAVs that use resources that are plentiful on Mars [4], [5], [6], [7]. Finally, for military or civil purposes, vehicular applications [8], [9], [10] have as objective the development of autonomous and semi-autonomous systems capable of traversing unrehearsed and off-road terrain, driving a car autonomously in an urban environment and also to help the driver in its driver task.

The perception is a common task to all cases previously presented, and an important factor is the variety and complexity of environments and situations. These real-time intelligent platform developments have a common issue: providing to the platform the capability of perceiving and interacting with its neighbour environment, managing power consumption, CPU usage, etc. Then, taking into account the temporal coherence between consecutive frames, this work proposes an environment observer method based on Pearson's Correlation Coefficient that observes if there are no significant changes in the environment, permitting that regions-of-interest (ROI) are automatically selected in order to perform a task: road detection and obstacle avoidance.

The Section II presents a review of previous works. The Section III introduces the Pearson's Correlation Coefficient, followed by the Discarding Criteria method in Section IV. The Automatic Regions-of-Interest (ROI) Selection is presented in the Section V. The results and conclusions are given in Section VI and VII.

II. Related Works

A. Sensor Perception

Environment perception is a major issue in autonomous vehicles. It uses many types of sensors [8], [9], including ultrasonic sensors, laser rangefinders, radar, cameras, etc. However, when incorporating several types of sensors, there is an increase of autonomy and "intelligence" degrees, especially in relation to navigation in unknown environments. In contrast, the type and quantity of sensors determine the volume of data for processing that requires, in most cases, a high computational cost. For unstructured environments, the scenario for study is dynamic, with several elements in motion. Thus, running a semi- or autonomous system involves carrying out complex, and non-deterministic operations in real time.

Moreover, a real-time system must satisfy explicit response-time constraints, including failure. This system is one whose logical correctness is based on both the correctness of the outputs and their timeliness [11]. Furthermore, there is a considerable complexity in the sense that correctness not only depends on the logical ordering of events of the systems, but also on the relative timing between them [12].

Aware that in the majority of the semi- and autonomous navigation systems, the machine-vision system is working

Manuscript received July 31, 2011.

Arthur de Miranda Neto is Ph.D. Student in Mechanical Engineering at the State University of Campinas – Brazil and in Information Technology and Systems at the University of Technology of Compiègne, France. Actually, he is researcher engineer at the Heudiasyc Lab CNRS UMR 6599.

Alessandro Corrêa Victorino is associate professor at the University of Technology of Compiègne – Heudiasyc Lab CNRS UMR 6599, France.

Isabelle Fantoni is researcher at the University of Technology of Compiègne – Heudiasyc Lab CNRS UMR 6599, France.

Douglas Eduardo Zampieri is associate professor of the Computational Mechanics Department, at the State University of Campinas, Brazil.

together with other sensors, added to its low cost, this paper proposed a monocular vision-based sensor. Because it uses simple techniques and fast algorithms, the system is capable to achieve a good performance, where the compromise between processing time and images acquisition is fundamental.

Additionally, the vision-based sensors are defined as passive sensors and the image scanning is performed fast enough for Intelligent Transportation Systems [13]. Furthermore, on the safety front, the progressive safety systems will be developed through the manufacturing of an "intelligent bumper" peripheral to the vehicle in answering new features as: blind spot detection, frontal and lateral pre-crash, etc. The objective in terms of cost to fill ADAS functions has to be very lower than the current Adaptive Cruise Control (500 Euros) [14].

B. Machine Vision

Autonomous robots can perform desired tasks in unstructured environments without continuous human guidance. An important variable is the state conditions in combination with environment events, because they may determine the system behavior.

In this way, Dynamic Power Management (DPM) is a design methodology for dynamically reconfiguring systems to provide the requested services and performance levels with a minimum number of active components or a minimum load on such components. For example, DPM and Real-Time Scheduling (RTS) techniques were presented in [15]. They were applied to reduce the power consumption of mobile robots. At the same time that scheduling is a key concept in computer multitasking and real-time operating system, the DPM dynamically adjusts power states of components adaptive to the task's need, reducing the power consumption without compromising system performance.

The machine vision, part of the embedded computers, is an important component of the set of sensors. Although extremely complex and highly demanding, thanks to the great deal of information it can deliver, it is a powerful means for sensing the environment and it has been widely employed to deal with a large number of tasks in the automotive field [13]. However, complex machine vision systems can lead to some losses due to the processing time.

A great amount of information would not necessarily lead to better decisions and could also harm the performance of the system, overloading it. Then, taking into account that it has been estimated that humans perceive visually about 90% of the environment information required for driving [13], it is not a bad idea to reduce information acquired by a vision system, in order to reduce processing time.

The method proposed here is based in an automatic image discarding criteria [16], [17], a low complexity and easy implemented solution. It improves the performance of a real-time system by choosing, in an automatic way, which images should be discarded and which ones should be treated at the visual perception system.

Furthermore, the fundamental premise for the applicability of DPM is that systems experience non-uniform workloads during operation time. A second assumption is that it is possible to predict, with a certain degree of confidence, the fluctuations of workload [18]. In this case, a simple DPM method shuts down a component when it is idle and it is essentially a prediction problem [15]. Thus, according to [18], the rationale in all predictive techniques is that of exploiting the correlation between the past history of the workload and its near future in order to make reliable predictions about future events.

III. Pearson's Correlation Coefficient (PCC)

According to [19], an empirical and theoretical development that defined regression and correlation as statistical topics were presented by Sir Francis Galton in 1885. In 1895, Karl Pearson published the Pearson's Correlation Coefficient (PCC) [20]. The Pearson's method is widely used in statistical analysis, pattern recognition and image processing. Applications on the latter include comparing two images for image registration purposes, disparity measurement, etc [21]. It is described in (1):

$$r_1 = \frac{\sum_i (x_i - x_m)(y_i - y_m)}{\sqrt{\sum_i (x_i - x_m)^2}\sqrt{\sum_i (y_i - y_m)^2}} \tag{1}$$

where x_i is the intensity of the i^{th} pixel in image 1, y_i is the intensity of the i^{th} pixel in image 2, x_m is the mean intensity of image 1, and y_m is the mean intensity of image 2. The PCC threshold, r_1, has value 1 if the two images are identical, 0 if they are completely uncorrelated, and –1 if they are completely anti-correlated, for example, if one image is the negative of the other.

IV. Discarding criteria

The discarding criteria was presented in [16] as a simple solution to improve the performance of a real-time navigation system by choosing, in an automatic way, which images should be discarded and which ones should be treated at the visual perception system. It was a new approach to the Pearson's Correlation Coefficient (PCC).

In Fig. 1, basically, if the PCC indicates that there is a high correlation between a reference frame and another new frame acquired, the new frame is discarded without being processed, Fig. 1 (c). In this case, some logical components may be shut down to save processor energy consumption, and/or to make the CPU available for running concurrent processes (the system can repeat a last valid command). Otherwise, the frame is processed and it is set as the new reference frame for the subsequent frame. For all cases presented in this paper a threshold is chosen in an empirical way, as explained next.

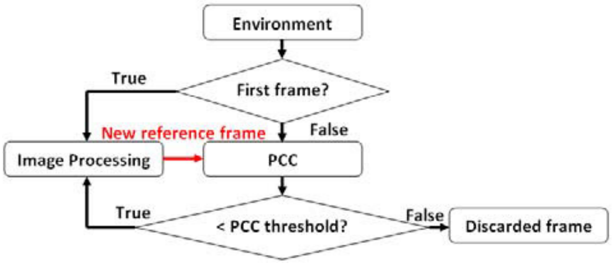


Fig. 1 – Discarding criteria [16].

Whereas the main problem that has to be faced when real-time imaging is concerned and which is intrinsic to the processing of images is the large amount of data [13], a logical dynamic optimization methodology based on Pearson's Correlation Coefficient (PCC) was introduced in [22]. To better understand this proposal, the Fig. 2 presents the accumulated time of a hypothetical image processing time (15ms) versus the gain obtained by using the discarding criteria, which could allow significant savings in CPU power consumption. In desert context were discarded 470 of 530 frames, whilst in off-road context were discarded 5595 of 6740 frames. For these two cases, the discarding rate remains over 80%.

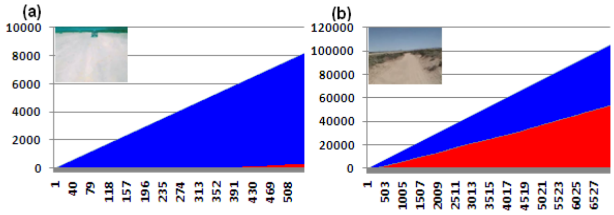


Fig. 2 – (a) Desert video [23]; (b) Off-road video [23]; In blue: the cumulative impact computations (ms) without the discarding criteria; In red: the cumulative computations (ms) by using the discarding criteria.

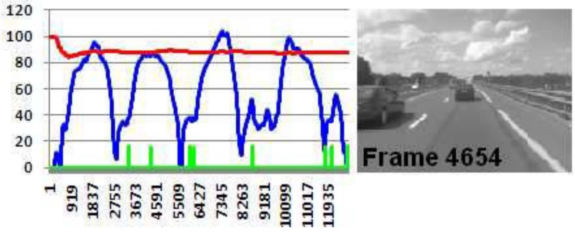


Fig. 3 – Real environment: Red line: discarding rate; Blue line: the vehicle speed; Green line: computational time (ms).

However, this discarding rate is not dependent on the video sequence or image size, but on the obstacles / objects influence. As shown in Fig. 3, it is also important to notice that there is no diffeomorphism between the robot speed and the PCC variation, because if there are no changes between consecutive frames, the PCC threshold remains static. In this case, the isomorphism cannot be guaranteed and it ensures more efficiency for our proposal. In real, dynamic and unknown environments, this rate remains over 65%. For these examples, the PCC threshold was fixed at 0.85.

V. Automatic Regions-of-Interest (ROI) Selection

According to the Pearson's correlation, in a certain analysis window (pair of frames), if the obstacle/object occupies a big portion of the scene, the PCC threshold tends to be low. Conversely, if obstacle/object occupies a small portion of the frame, it means that it is away from the vehicle and the system will have time enough to react. Nevertheless, where are these interest points/pixels? Or, which pixels (ROI) of the pair of images contributed most to the Pearson's coefficient computed? Which of them really need to be reprocessed (or resent to a server)?

Right after the Pearson's correlation in (1), it has x_m and y_m, respectively: the mean intensities of images 1 and 2, i.e. r_{1Xm} and r_{1Ym}.

From these two values, it begins again the process's correlation in (2), where for each pair of pixels analyzed, the only possible result is: [-1 or +1]. That is, all pixels with intensities below these means will be candidates for interest points. The Fig. 4 (c) and (f) present this process, where the red pixels (ROI) represent $r_2 = -1$.

$$r_2 = \frac{\sum_i (x_i - r_{1Xm})(y_i - r_{1Ym})}{\sqrt{\sum_i (x_i - r_{1Xm})^2}\sqrt{\sum_i (y_i - r_{1Ym})^2}} = \begin{cases} -1 \\ or \\ +1 \end{cases} \qquad (2)$$

where x_i is the intensity of the i^{th} pixel in image 1, y_i is the intensity of the i^{th} pixel in image 2, r_{1Xm} and r_{1Ym} were obtained in (1).

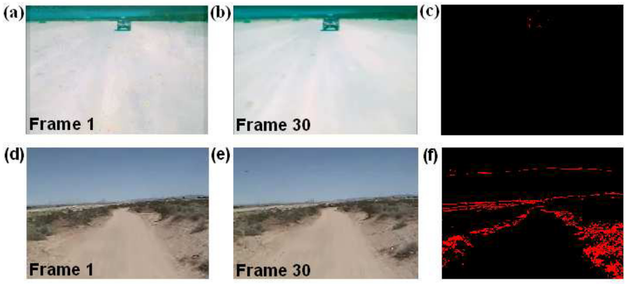


Fig. 4 – (a) and (b) are frames of the desert video [23]; (d) and (e) are frames of the off-road video [23]; (c) and (f) present the process's correlation in (2), where the red pixels (ROI) represent $r_2 = -1$.

A. *Road detection*

Different techniques on automatic and semi-automatic road extraction methods are proposed in the literature. With respect to these specific tasks, a road detection method based on Otsu thresholding algorithm was proposed in [24], [25].

As shown in Fig. 1, the road detection process (imaging processing) is performed only when the PCC indicates that there is a low correlation between the reference frame and the current frame. Otherwise, the current frame is discarded without being processed. When an image is discarded, the system keeps the previous segmentation result, which is linked to an Otsu threshold. This stored threshold represents the navigable area. Consequently, in order to further improve the navigable area detection, for each discarded image, in Fig. 1, it classifies only the interest pixels (ROI) obtained in (2) from the last stored Otsu threshold. As an example, Fig. 5 (d) presents the interest pixels from the correlation between the Fig. 5 (a) and (b), where the white pixels (ROI) represent $r_2 = -1$. In Fig. 5 (e), these interest pixels are classified as navigable area in blue.

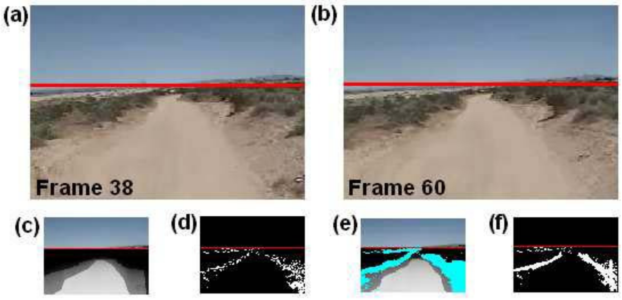


Fig. 5 – (a) and (b) are frames of the off-road video [23]; (c) the road detection process result; (d) the interest pixels (white pixels); (e) the interest pixels are classified as navigable area in blue; (f) line detection using Hough transform from the interest pixels (ROI) in Fig. 5 (d). The horizontal red lines represent the horizon detection process [25].

B. *Identifying the limits (boundaries) of the road*

In order to identify the limits of the road (which includes the obstacles), many research works proposing methods to detect road boundary. As an example, the Canny edge detector [26] can be employed as input of Hough transform [27] due to its robust performance and accurate edge localization. Nevertheless, to demonstrate another application for the proposed method here, Fig. 5 (f) presents the line detection using Hough transform from the Fig. 5 (d), where the white pixels represent $r_2 = -1$.

C. *Obstacle avoidance*

According to [28], the robots enter in the military context especially when it is necessary to reduce human exposure to hazardous situations. Many of these robotic missions can be observed, among them, as shown in Fig. 6 (d), an Improvised Explosive Device (IED) detonation. In this context, obstacle avoidance is a robotic discipline that includes reactive control in real time, i.e. reactive obstacle avoidance.

In this way, as discussed earlier, in order to make reliable predictions about future events, a predictive technique explores the correlation between the past history of the workload and its near future [18]. On the other hand, the Pearson's method explores regression and correlation aspects. Then, in order to reduce the risk of collision, for the obstacle avoidance task, it uses a PCC threshold equal to 0.65, which put in evidence the past history properties. In a computational process running in parallel to what was discussed in section V-A, for each frame processed, evidencing the past history properties, another old Otsu threshold is also stored. Besides the results presented in the Fig. 6, this procedure also allows a greater level of security, especially when the camera does not “see” the navigation area (i.e. in front of a wall), as will be shown in next section.

Therefore, for each discarded image, in Fig. 1, it classifies only the interest pixels (ROI) obtained in (2) from the last Otsu threshold obtained by a road detection algorithm (reference frame). Fig. 6 (c) and (f) represent the interest pixels classified as obstacle, where the white pixels (ROI) represent $r_2 = -1$. As shown in Fig. 6 (b) and (e), the interest pixels are classified as obstacle in yellow.

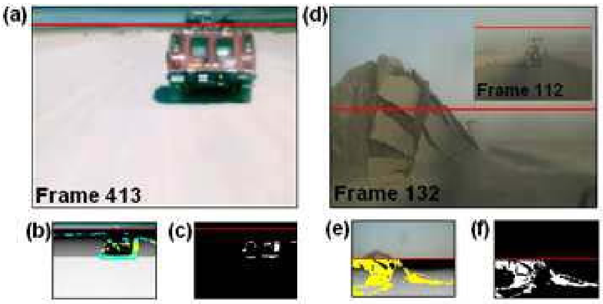


Fig. 6 – (a) is a frame of the off-road video [23]; (d) Improvised Explosive Device example; (b) and (e): the interest pixels are classified as navigable area in blue; (b) and (e): the interest pixels classified as obstacle in yellow. The horizontal red lines represent the horizon detection process [25].

VI. Experimental Results

Besides the experimental DARPA test-banks [23], this section presents results on real, dynamic and unknown environments, and they were obtained using two experimental vehicles, from a moving vehicle with a Sony DFW-VL500 camera. In order to reduce the number of data, it includes a resolution reduction of image to 160x120 pixels.

From displacements on the outskirts of the Heudiasyc Laboratory in France, the test-bank presented in the Fig. 7, 9 and 10 contains images recorded using the Carmen vehicle shown in Fig. 11 (a). The data were collected while a driver was driving the vehicle. Results for two different types of image texture (road surfaces) were selected and its results are presented in the Fig. 7 (b) and (e).

The next cases present autonomous displacements at Renato Archer IT Center (CTI) in Brazil. This stage of testing evaluates the proposed algorithm at low speed on real-time conditions using the vehicle VERO shown in Fig. 11 (b). The VERO platform is equipped with SICK LMS and Hokuyo UTM30 outdoor laser scanners, GPS receiver, a monocular camera, and a CAN-bus interface which grants access to encoder data for the four wheels and steering, and allows commands to be sent to two independent motors driving the rear wheels and to control the steering angle. In the Fig. 8 (b) and (e), the trajectory control system had a single goal: to keep the robot in the center of the navigable path based on the mass of center. Fig. 8 (d) shows the task execution to go through a gate in off-road context. Fig. 8 (c) and (f) present the line detection using Hough transform from navigable area pixels. Two types of image texture (road surfaces) also were selected and its results are presented in Fig. 8 (b) and (e).

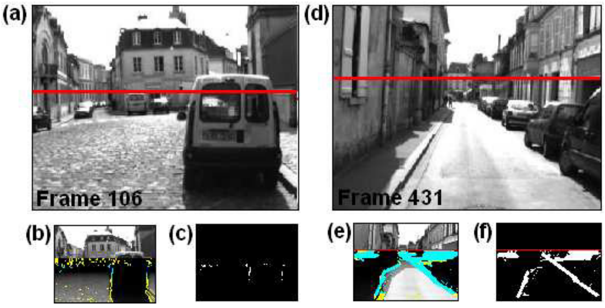


Fig. 7 – (a) and (d) are frames in real environments; (b) and (e): the interest pixels are classified as navigable area in blue; (b) and (e): the interest pixels are classified as obstacle in yellow; (c) and (f): line detection using Hough transform from navigable area pixels. The horizontal red lines represent the horizon detection process [25].

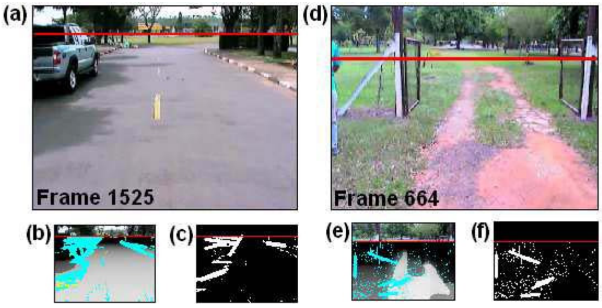


Fig. 8 – (a) and (d) are frames in real environments; (b) and (e): the interest pixels are classified as navigable area in blue; (b) and (e): the interest pixels are classified as obstacle in yellow; (c) and (f): line detection using Hough transform from navigable area pixels. The horizontal red lines represent the horizon detection process [25].

As expected, in grass areas, on the parallelepiped streets or where an excessive noise is observed, the efficiency of the method decreases considerably, what can be improved with the application of a smoothing filter, and/or from region-merging algorithm that mainly aims to represent homogeneous regions.

As shown in Fig. 9, it is not expected that a single camera provides all needed information to the safe navigation system to take decisions on routes. However, following what was presented in section VI-C: Obstacle avoidance, the new results in front of walls are presented in Fig. 10.

A video showing the application of this method is available in [29].

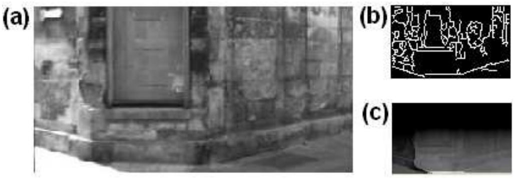


Fig. 9 – (a) Original image (in front of a wall) and its road detection result in (c); (b) Canny edge detection result.

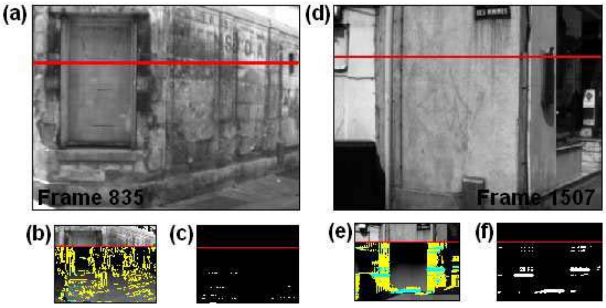


Fig. 10 – (a) and (d) are frames in real environments; (b) and (e): the interest pixels are classified as navigable area in blue; (b) and (e): the interest pixels are classified as obstacle in yellow; (f) and (i): line detection using Hough transform from navigable area pixels. The horizontal red lines represent the horizon detection process [25].

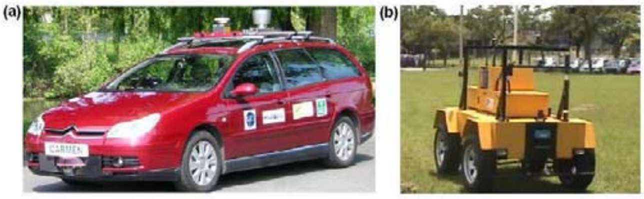


Fig. 11 – The experimental vehicles: (a) Carmen vehicle at Heudiasyc Laboratory in Compiegne, France; (b) Autonomous vehicle (VERO) at Renato Archer IT Center (CTI) in Campinas, Brazil.

VII. Conclusion

This work proposes a real-time machine vision algorithm based on monocular vision. It is important to notice that this algorithm is not based on previous knowledge of the environment (lane shape, geometric inference, etc) neither camera calibration. A remarkable characteristic of methodology presented in this work is its independence of the image acquiring system and of the robot itself. The same implementation works in different mobile robots, with different embedded vision systems, without the need of adjusting parameters. Moreover, this visual-observer methodology may be extended to other sensors and components. Future work would be also focused to provide ground truth measurements from a front mounted radar and/or LIDAR system.

ACKNOWLEDGMENT

The authors wish to thank Dr. Luiz Gustavo Bizarro Mirisola, Dr. Samuel S. Bueno, Dr. Josué Jr. G. Ramos, Mr. Hélio Azevedo and Mr. Gerald Dherbomez for their kind attention to our work, for the useful discussions and support in data acquisition. This work was supported in part by French National Research Agency / ANR Percoive Project.

REFERENCES

[1] F. Bonin-Font, A. Ortiz, G. Oliver, (2008), “Visual Navigation for Mobile Robots: A Survey”, Journal of Intelligent and Robotic Systems, 53, 3, 263-296, doi: 10.1007/s10846-008-9235-4

[2] B. Kim, P. Hubbard, D. Necsulescu, (2003), “Swarming Unmanned Aerial Vehicles: Concept Development and Experimentation, A State of the Art Review on Flight and Mission Control”, DRDC-OTTAWA-TM-2003-176; Technical Memorandum.

[3] S. R. Anton, D.J. Inman, (2008), “Energy Harvesting for Unmanned Aerial Vehicles”, In: Proceeding of SPIE.

[4] A. T. Klesh and P. T. Kabamba, (2009), “Solar-Powered Aircraft: Energy-Optimal Path Planning and Perpetual Endurance”, Journal of Guidance, Control, and Dynamics.

[5] The Vision for Space Exploration, NASA, Feb. 2004, http://www.nasa.gov/pdf/55583main_vision_space_exploration2.pdf [retrieved 29 Nov. 2010].

[6] Energetically Autonomous Tactical Robot, (2009), DARPA Contract W31P4Q-08-C-0292.

[7] R. Finkelstein, Robotic Technology Inc, (2009), “Energetically Autonomous Tactical Robot and Associated Methodology of Operation”, Patent Application number: 12/612,489, Publication number: US 2010/0155156 A1.

[8] S. Thrun, et al. (2006), “Stanley, the robot that won the DARPA Grand Challenge”, Journal of Robotic Systems, Volume 23 , Issue 9, DARPA Grand Challenge, 661-692, ISSN:0741-2223, doi=10.1002/rob.v23:9

[9] Team Berlin (2007), Spirit of Berlin: An Autonomous Car for the DARPA Urban Challenge Hardware and Software Architecture, http://www.darpa.mil/grandchallenge/ TechPapers/Team_Berlin.pdf [retrieved 02 Dec. 2010].

[10] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, (2006) “Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations,” Vehicle System Dynamics, vol. 44, no. 7, pp. 569–590, doi=10.1080/00423110600563338

[11] P. Laplant, (2004), “Real-Time System Design and Analysis”, IEEE Press, 3nd ed.

[12] P. Bouyer et al., (2010), “Quantitative analysis of real-time systems”. Journal Communications of the ACM.

[13] M. Bertozzi. A. Broggi and A. Fascioli, (2000), “Vision-based intelligent vehicles: state of the art and perspectives”, Robotics and Autonomous systems 32, 1-16, doi=10.1.1.36.3682

[14] Radio Spectrum Committee, European Commission, Public Document, Brussels, 5 July 2010, RSCOM10-35 http://ec.europa.eu/information_society/policy/ecomm/radio_spectrum/_document_storage/rsc/rsc32_public_docs/rscom10_35.pdf [retrieved 02 Dec. 2010].

[15] H. Y. L. C. Yongguo Mei, Yung-Hsiang Lu, (2005), “A case study of mobile robot's energy consumption and conservation techniques”, Proceedings of the the IEEE International Conference on Advanced Robotics, ICAR 2005, pp. 492-497, doi=10.1109/ICAR.2005.1507454

[16] A. Miranda Neto, L. Rittner, N. Leite, D. E. Zampieri, R. Lotufo and A. Mendeleck, (2007), “Pearson's Correlation Coefficient for Discarding Redundant Information in Real Time Autonomous Navigation Systems”, Proceedings of the 2007 IEEE Multi-conference on Systems and Control, doi: 10.1109/CCA.2007.4389268

[17] A. Miranda Neto, L. Rittner, N, Leite, D. E. Zampieri and A. C. Victorino, (2008), “Nondeterministic Criteria to Discard Redundant Information in Real Time Autonomous Navigation Systems based on Monocular Vision”, ISIC Invited Paper, 2008 IEEE Multi-conference on Systems and Control, San Antonio, Texas, EUA, September 3-5, doi: 10.1109/ISIC.2008.4635955

[18] L. Benini, A. Bogliolo, and G. D. Micheli. (2000), “A Survey of Design Techniques for System-Level Dynamic Power Management”, IEEE Transactions on Very Large Scale Integration Systems, 8(3):299-316, doi=10.1109/92.845896

[19] J. L. Rodgers and W. A. Nicewander, (1988), “Thirteen Ways to Look at the Correlation Coefficient”, The American Statistician, 42, 59-66.

[20] K. Pearson, (1895), Royal Society Proceedings, 58, 241.

[21] Y. K. Eugene and R.G. Johnston, “The Ineffectiveness of the Correlation Coefficient for Image Comparisons”, Technical Report LA-UR-96-2474, Los Alamos, 1996.

[22] A. Miranda Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri, (2011), “Real-Time Dynamic Power Management based on Pearson’s Correlation Coefficient”, Proceedings of the IEEE International Conference on Advanced Robotics, ICAR 2011, to appear.

[23] DARPA 2005. “DARPA Grand Challenge”, http://www.darpa.mil/grandchallenge05/

[24] A. Miranda Neto and L. Rittner, (2006), “A Simple and Efficient Road Detection Algorithm for Real Time Autonomous Navigation based on Monocular Vision”, Proceedings of the 2006 IEEE 3rd Latin American Robotics Symposium, doi=10.1109/LARS.2006.334325

[25] A. Miranda Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri, (2011), “Robust Horizon Finding Algorithm for Real-Time Autonomous Navigation based on Monocular Vision”, submitted to Proceedings of the IEEE Int. Conf. on Intelligent Transportation Systems, ITSC 2011, to appear.

[26] J. F. Canny, (1986), “A computational approach to edge detection”. IEEE Trans. Pattern Anal. Machine Intell. 8 (6), 679–698, doi=10.1109/TPAMI.1986.4767851

[27] D. Ballard, (1981), “Generalized Hough transform to detect arbitrary shapes”, IEEE Trans. Pattern Anal. Machine Intell. 13 (2), 111–122.

[28] R. Chatila, (2010), “Dual research: Issues in robotics”, Oral presentation, Mission for Research and Innovation Scientific 2010. Directorate General of Armaments (DGA), Paris, France.

[29] http://www.youtube.com/watch?v=VcUQVC1F8Xw